

International Journal of
Info & Com **S**ciences for
Decision **M**aking

ISSN:1265-499X

N°13 - Janvier 2004

CONTENTS

- Numéro spécial Colloque MAJECSTIC'03** **Page 4**
Yann Bertacchini., Université de Toulon-Var, Laboratoire LePont.
- Modélisation par réseaux de Petri pour une programmation VHDL. Exemple d'application en robotique mobile d'assistance au handicap** **Page 5**
Alexandre Abellard, Mohamed Ben Khelifa, Moez Bouchouicha, Patrick Abellard, *Université de Toulon-Var (La Garde, France)*
- Détection de véhicules dans un environnement autoroutier à l'aide de données télémétriques et visuelles** **Page 12**
Sonia Izri, Eric Brassart, Laurent Delaboche, Arnaud Clémentin, *IUT d'Amiens (Amiens, France)*
- Reconstruction de la cornée par analyse ascendante-descendante** **Page 23**
Sébastien Richard, Rémy Bulot, Jean Sequeira, *Laboratoire LSIS (Marseille, France)*
- Speech recognition : new techniques for speaker adaptation** **Page 30**
Olivier Bellot, Driss Matrouf, *LIA (Avignon, France)*
- GXcast : une généralisation du protocole Xcast** **Page 39**
ALi Boudani, Alexandre Guitton, Bernard Cousin, *Université Rennes I (Rennes, France)*
- Une approche pour le contrôle de la qualité des systèmes de filtrage collaboratif** **Page 46**
Lizbeth Gallardo-Lopez, Catherine Berrut, Nathalie Denos, *CLIPS-IMAG (Grenoble, France)*
- J-RADEX : un simulateur de SGBDTR convivial** **Page 56**
Jérôme Haubert, Bruno Sadeg, Laurent Amanton, Roland Coma, *Université du Havre (Le Havre, France)*
- Extension d'un protocole de contrôle de concurrence des transactions temps-réel** **Page 64**
Jérôme Haubert, Bruno Sadeg, Laurent Amanton, *Université du Havre (Le Havre, France)*

-
- Approche connexionniste de la perception du mouvement** **Page 72**
Claudio Castellanos Sanchez, Bernard Girau, Frédéric Alexandre, *LORIA INRIA-Lorraine (Vandoeuvre-lès-Nancy, France)*
- Classification automatique de comportements d'élèves apprenant l'algèbre** **Page 82**
David Renaudie, *IMAG (Grenoble, France)*
- NSCSP : définition et résolution par transformation** **Page 91**
Alexis Anglada, *Dassault Aviation (Paris, France)*
- Analyse hors-ligne d'applications temps-réel fortement couplées comportant des tâches à durées variables** **Page 98**
Stéphane Pailler, Annie Choquet-Geniet, *LISI-ENSMA (Poitiers, France)*
- Comparaison d'approches de simulations distribuées à évènements discrets d'entités spatialisées** **Page 105**
Gauthier Quesnel, Raphaël Duboz, Eric Ramat, *Université du Littoral (Calais, France)*
- Construction de spécifications multi-vues UML et B** **Page 114**
Dieudonné Okalas Ossami, Jeannine Souquières, Jean-Pierre Jacquot, *LORIAS (Vandoeuvre-lès-Nancy, France)*
- The auditive consistence property in distributed music performance : a conductor based synchronisation** **Page 129**
Nicolas Bouillot, *CNAM- CEDRIC (Paris, France)*
- Versions on-line du problème du stable maximum** **Page 138**
Bruno Escoffier, Vangelis Th. Paschos, *Université Paris IX Dauphine (Paris, France)*
- Détection automatique de frontières des phrases : Un système adaptatif multi-langues** **Page 145**
Anna Pappa, Gilles Bernard, Hind Oukerradi, *Université Paris VIII (Paris, France)*
- Un point sur la relation entre Technologies de l'Information et de la Communication et productivité** **Page 154**
Marc Bourreau, Chiraz Karamti, *ENST (Paris, France)*
- Une approche pour la génération automatique de tests de robustesse** **Page 167**
Cyril Pachon, *Laboratoire Vérimag (Gières, France)*
- Une solution optimale pour un problème de routage dans un canal dense avec contraintes de capacité** **Page 176**
Maria Zrikem, *CNAM (Paris, France)*
- Vers une architecture multi-agents pour du clustering dynamique d'une population d'agents** **Page 184**
Roland Coma, *LIH (Le Havre, France)*

Adaptation des applications réparties à base de composants aux terminaux mobiles en environnement sans fil **Page 191**

Nabil Kouici, Denis Conan, Guy Bernard, *CNRS Samovar (Evry, France)*

Segmentation d'image par minimisation de la complexité stochastique **Page 201**

Frédéric Galland, Nicolas Bertaux, Philippe Réfrégier, *Faculté St Jérôme (Marseille, France)*

Description et conception d'une plte-forme robuste combinant des analyseurs d'énoncé **Page 209**

Francis Brunet-Manquat, *CLIPS-IMAG (Grenoble, France)*

Réseaux cellulaires de Voronoï **Page 221**

Anne-Elisabeth Baert, *Université de Picardie Jules Verne (Amiens, France)*

Suivi de l'activité de pilotage par prédiction et recalage **Page 229**

Charles Lesire, *Onera-Cert, DCSD (Toulouse, France)*

Editors in chief : Pr. H.Dou, Pr. Ph.Dumas, Dr Y.Bertacchini

All correspondences about I.S.D.M or submission should be sent to: Dr Y.Bertacchini
Université de Toulon, LePont, C205, BP 132, 83957 La Garde Cedex, France
e-mail : bertacchini@univ-tln.fr www server : <http://www.isdm.org>

NUMERO SPECIAL MAJECSTIC'03

En janvier 2003 est née l'idée de créer une conférence francophone pour jeunes chercheurs en informatique. Cela a donné lieu à la mise en place de MAJECSTIC'03, MANifestation des JEunes Chercheurs du domaine des STIC. Pour cette première édition, un nombre important de thèmes a été proposé afin de couvrir un vaste panorama des activités de recherche en Informatique et Génie Informatique. Cette approche permettra de réunir un grand nombre de jeunes chercheurs, en lieu favorable aux échanges. De fait, MAJECSTIC s'est donné l'objectif d'être le point de départ de collaborations et d'échanges entre doctorants, avec l'espoir que de ces échanges émergeront de nouvelles collaborations qui constitueront la base de futurs projets inter-laboratoires.

La communauté francophone a répondu présente à l'appel avec 116 soumissions émanant de plus de 50 laboratoires ou organismes de recherche. Les 68 articles retenus devraient donner une image assez précise de la recherche en informatique dans la communauté francophone, de son dynamisme et de son positionnement vis-à-vis de la communauté internationale.

Nous souhaitons remercier les présidents d'honneurs, le comité d'organisation et le comité de programme pour leurs suggestions et relectures qui ont conduit à améliorer la qualité de cette conférence. Nous voudrions également remercier les conférenciers qui ont accepté notre invitation pour les sessions plénières et tous les chercheurs qui, par leur proposition de publications, feront de cette première manifestation un réel succès. Dans nos remerciements, nous n'oublions pas tous ceux qui ont travaillé dans l'ombre et sans qui cette conférence n'aurait pu voir le jour. Nous adressons aussi un remerciement tout particulier au directeur du département GII de l'école Polytechnique Universitaire de Marseille qui accueille cette première édition de MAJECSTIC.

MAJECSTIC n'aurait pu rassembler autant de jeunes chercheurs sans le soutien appuyé : du Ministère délégué à la Recherche et aux Nouvelles Technologies, du département STIC du CNRS, de la région PACA, du Conseil Général des Bouches-du-Rhône, de la Ville de Marseille, de l'Université d'Aix-Marseille II, de l'Université d'Aix-Marseille III, de l'ENSAM, de l'ESIL, du GDR I3, de l'association SPECIF, d'Eurocopter, du groupe Arcelor, du Laboratoire des Sciences de l'Information et des Systèmes ([LSIS](#) UMR CNRS 6168) et de l'Association des Doctorants et Docteurs du LSIS ([A2DL](#)). Qu'ils trouvent ici l'expression de notre gratitude.

Gageons que MAJECSTIC'03 sera le début d'une longue série qui permettra, années après années, de conforter les liens et les échanges entre jeunes chercheurs de la communauté informatique francophone.

NDLR : ce Numéro Spécial de ISDM a pu voir le jour par la coopération déjà ancienne entre M.Ben Khelifa, Enseignant-Chercheur en 61^e section, laboratoire SIS, et Yann Bertacchini, Enseignant-Chercheur, HDR, en 71^e section du laboratoire LePont. Ces deux laboratoires font partie de l'Université du Sud (Toulon-Var)

Yann Bertacchini., Université de Toulon-Var, Laboratoire LePont

***MODELISATION PAR RESEAUX DE PETRI POUR UNE PROGRAMMATION VHDL.
EXEMPLE D'APPLICATION EN ROBOTIQUE MOBILE D'ASSISTANCE AU HANDICAP***

Alexandre Abellard,

Doctorant en Traitement de l'Image et du Signal

alexandre.abellard@univ-tln.fr , + 33 4 94 14 22 48

Mohamed Ben Khelifa,

Maître de conférences en Génie Automatique et Traitement du Signal

khelifa@univ-tln.fr +33 4 94 14 20 77

Moez Bouchouicha,

Attaché Temporaire d'Enseignement et de Recherche en Génie Automatique et Traitement du Signal

moez@univ-tln.fr +33 4 94 14 24 93

Patrick Abellard,

Professeur des Universités en Génie Automatique et Traitement du Signal

abellard@univ-tln.fr +33 4 94 14 22 48

Adresse professionnelle

Laboratoire SIS/AI ★ Université de Toulon-Var ★ BP 20132 ★ F-83957 La Garde Cedex ★ France

Résumé : La recherche des solutions au problème de la conception conjointe matériel/logiciel est une tâche majeure dans la définition d'une méthodologie unique, structurée et automatique, permettant l'accélération du processus de conception et l'évaluation dynamique de différents compromis . Les Réseaux de Petri à Flux de Données sont une solution efficace pour y parvenir, ainsi que leur implantation sur circuits programmables. Nous décrivons une application relative à un fauteuil roulant contrôlé par une manette gérée par réseau neuronal.

Summary : The research of solutions to the problem of hardware/software codesign is a major task in the definition of a unique, structured and automatic methodology. Data Flow Petri Nets can be an efficient way to do this. Thus, it is possible to write a program and to compile it in order to implement it on programmable chips. The following paper describes their implementation in a wheelchair, whose lever is handled by a neural network.

Mots clés : Réseaux de Petri, réseaux de neurones, VHDL, fauteuil roulant.

Modélisation par réseaux de Petri pour une programmation VHDL. Exemple d'application en robotique mobile d'assistance au handicap

La recherche des solutions au problème de la conception conjointe matériel/logiciel est une tâche majeure dans la définition d'une méthodologie unique, structurée et automatique, permettant l'accélération du processus de conception et l'évaluation dynamique de différents compromis. Les Réseaux de Petri à Flux de Données (RdPFD) sont une solution efficace pour y parvenir, et l'utilisation d'un langage comme VHDL permet ensuite leur implantation sur circuits programmables.

L'exemple qui est décrit concerne un fauteuil roulant ergonomique et à commande adaptable au handicap. Pour certains handicapés, l'utilisation d'un fauteuil roulant peut être difficile du fait de capacités physiques trop faibles ou de troubles cognitifs. L'interface homme - machine doit donc être modulaire et configurable, facile à mettre en place, apporter une grande fiabilité, et utiliser le plus souvent possible du matériel non spécifique. Le projet FRACAH (Fauteuil Roulant A Commande Adaptée au Handicap) a été ainsi développé. Sa manette est gérée par un réseau de neurones artificiels qui enregistre les limitations fonctionnelles de la main puis qui les compense. Les expériences pratiquées avec des personnes présentant un handicap plus léger ont montré que la rentabilité dans l'exécution des tâches à accomplir était accrue, avec une sécurité maximale.

1 - INTRODUCTION

La conception conjointe de systèmes matériel/logiciel (hardware/software codesign) désigne la conception de systèmes comportant une partie matérielle avec un ou plusieurs circuits spécifiques (ASIC, composants reconfigurables...) et une partie logicielle qui est exécutée sur une architecture à base de processeurs standards ou spécifiques (microcontrôleurs, DSP...). Elle englobe la spécification, la validation et l'exploration des différentes possibilités de conception d'un système mixte dans le but d'optimiser des critères de coût et/ou de performances. Dias (2000) a montré que la recherche de

l'ensemble des solutions est une tâche majeure dans la définition d'une méthodologie unique, structurée et automatique car elle permet de définir des caractéristiques fondamentales telles que le coût et les performances temporelles.

Les avantages de cette intégration de conception matérielle et logicielle sont l'accélération du processus de conception et la possibilité d'évaluer dynamiquement les différents compromis possibles comme l'a fait Rompaey (1996). Par exemple, dans le cas de DSP et de circuits spécifiques, le processus de conception doit supprimer l'écart existant entre la spécification fonctionnelle hétérogène et son implantation. Au niveau système, Rubini et Lavenier (1997) ont démontré que les DSP ont besoin d'une combinaison entre les modèles flots de données et les modèles flots de contrôle pour leur spécification complète. L'utilisation des Réseaux de Petri à Flux de Données (RdPFD) est une solution efficace pour y parvenir. En effet, l'algorithme à implanter est facilement décrit par RdPFD et simulé avec n'importe quel outil standard. C'est à partir des résultats des simulations, qu'il est alors possible d'écrire un programme VHDL et de le compiler pour son implantation sur circuits programmables comme le propose Belhadj (1994).

2 - LES RESEAUX DE PETRI A FLUX DE DONNEES

Ils ont été définis par J. Alhmana (1983) et permettent de modéliser efficacement les machines multiprocesseurs de type flux de données / flux de contrôle, dans lesquelles la recherche des performances nécessite que soient supprimées des difficultés aussi diverses que : partage de bus, conflits d'accès mémoire, échanges inter-processeurs.... Ce type d'architecture est structurellement différent des multiprocesseurs classiques car il ne comporte ni processeur central, ni mémoire centrale. Ce sont les données qui dirigent les opérations. Abellard et Grall (2000) ont recensé un certain nombre de réalisations qui existent avec des modèles de fonctionnement variés.

Les Réseaux de Petri à Flux de Données (RdPFD) réalisent la combinaison en un modèle, de flots de données et de flots de contrôle. Abellard (2001) les utilise comme outil de spécification, de simulation et validation d'architectures à flux de données. Ce modèle, basé sur la notion de places biparties (opérateurs P_o et variables P_v) permet de modéliser de nombreuses structures : synchrones, asynchrones, systoliques, neuronales... (figure 1).

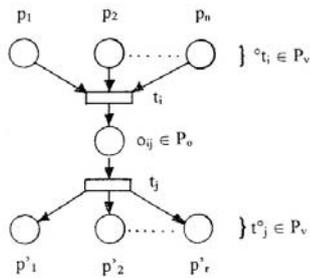


Figure 1 : Réseau de Petri à Flux de Données

Un RdPFD est un septuplet $(R, \varphi, \xi, \psi, X, O, C)$ défini de la façon suivante :

➤ R est un Réseau de Petri à Places bi-Parties (RdP PbP) conforme.

➤ φ est une application

$$\varphi \mid P_v : P \rightarrow X$$

$$\varphi \mid P_o : P \rightarrow O$$

telle que

$$\forall p_i \in P_o, p_j \in P_o \text{ et } \varphi(p_i) = \varphi(p_j) \text{ pour } i \neq j$$

alors

$$\forall t_l \in {}^o P_i \text{ et } t_k \in {}^o P_j, \{\varphi({}^o t_l) \neq \varphi({}^o t_k)\}$$

Ainsi, deux opérateurs identiques ne peuvent pas travailler sur le même ensemble de données.

➤ ξ est une application injective

$$\xi : X \rightarrow M = \{ME_1, ME_2, \dots, ME\}$$

telle que :

$$\forall p \in P_v, \exists ME \in M \text{ avec } ME = \xi(\varphi(p))$$

M est appelé ensemble de zones mémoires.

➤ ψ est une application qui permet de poser des conditions sur le franchissement d'une transition $\psi : T \rightarrow C$.

➤ $X = \{x_1, x_2, \dots, x_s\}$ est un ensemble de variables (réelles, entières, logiques...) prenant leurs valeurs respectivement dans les domaines D_1, D_2, \dots, D_u .

➤ $O = \{o_1, o_2, \dots, o_v\}$ est un ensemble fini d'opérateurs définis comme des applications internes de $D_1 \times D_2 \times \dots \times D_u$.

➤ $C = \{c_1, c_2, \dots, c_r\}$ est un ensemble de prédicats sur les variables de X .

3 - EXEMPLE D'APPLICATION

3.1 - Présentation

L'exemple qui est décrit concerne une réflexion sur la mise au point d'un fauteuil roulant ergonomique et à commande adaptable au handicap. Le fauteuil roulant électrique est un élément essentiel pour l'autonomie des personnes handicapées. Mais pour certaines, l'utilisation peut parfois être difficile, voire impossible, du fait de capacités physiques résiduelles trop faibles, d'une trop grande spasticité, d'une fatigabilité importante ou de troubles cognitifs. Par exemple, certaines personnes ne peuvent actionner qu'un simple capteur tout ou rien, ce qui rend difficile la commande d'un fauteuil roulant comme l'a expliqué Bourhis (2000). On peut alors apporter des améliorations à l'interface homme - machine standard de type joystick en utilisant des asservissements locaux ou des fonctionnalités intelligentes empruntées à la robotique : détection et évitement d'obstacles, accostage et passage de portes, suivi de parcours, planification de trajectoires... comme ceux réalisés par Ben Khelifa (2001) dans son travail de thèse.

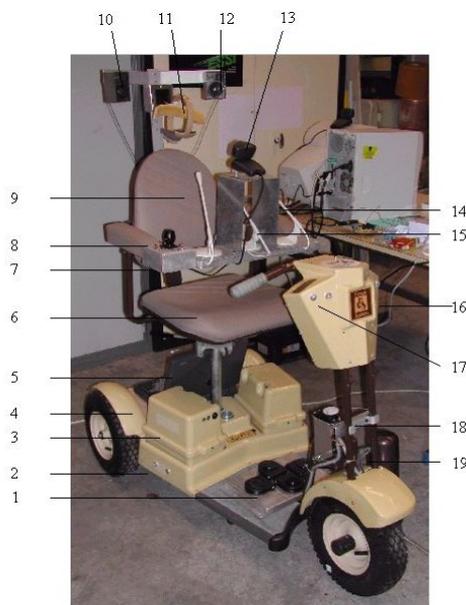
La robotique mobile de réadaptation et l'étude de fauteuils roulants intelligents s'apparentent au domaine de la robotique de service qui présente deux caractéristiques importantes : d'une part le robot partage son espace d'évolution avec les personnes et d'autre part, l'utilisateur n'est pas a priori compétent dans le domaine technologique, ce qui montre l'importance de l'interface homme - machine.

La grande diversité des utilisateurs et des structures est une caractéristique importante de l'application. Par exemple, l'environnement

peut être le domicile, un centre de réadaptation, un hôpital... et l'encombrement potentiel des chemins ainsi que les distances à parcourir sont de ce fait, très variables. Par ailleurs, les patients en fauteuil ont des possibilités physiques et cognitives très différentes.

L'interface homme - machine doit donc être modulaire et configurable, tant du point de vue logiciel que des capteurs de commande. Elle doit être facile à mettre en place et apporter une sécurité maximale avec une grande fiabilité. Par ailleurs, pour optimiser le rapport coût / performances, il est important d'utiliser chaque fois que possible, du matériel non spécifique à l'application, comme par exemple, une base de fauteuil roulant du commerce.

Le projet FRACAH (Fauteuil Roulant A Commande Adaptée au Handicap) a été développé en respectant ces exigences. La base est un ancien fauteuil roulant 3 roues qui a été restauré mais n'importe quel fauteuil du commerce peut convenir. Elle fournit un support expérimental de tests de différentes solutions techniques d'aide au pilotage : bouton - poussoir, joystick, commande par la tête, par le pied, le souffle, la voix... (figure 2).



1. Commande par pédales
2. Détecteurs de proximité droite
3. Compartiment batteries
4. Compartiment cartes électroniques
5. Micro-ordinateur
6. Siège motorisé en rotation et élévation
7. Commande par manette neuronale
8. Voyants - guides directionnels
9. Commande par le souffle (Avant et Droite)

10. Haut-parleur droit (synthèse vocale)
11. Commande par la tête
12. Haut-parleur gauche (synthèse vocale)
13. Caméra (états des yeux, mouvements de tête, suivi du regard, expressions du visage), guidage automatique
14. Commande par le souffle (Arrière et Gauche)
15. Microphone pour commande vocale
16. Détecteurs de proximité Avant Gauche
17. Détecteurs de proximité Avant Droit
18. Servo-moteur de direction
19. Moteur de propulsion

Figure 2 : Fauteuil roulant appareillé

La manette est gérée par un réseau de neurones artificiels qui enregistre les limitations fonctionnelles de la main en phase d'apprentissage, puis qui compense tous les gestes incomplets et élimine les tremblements et les mouvements involontaires désordonnés. En plus des capteurs actifs de sécurité, une caméra embarquée surveille l'état des yeux de l'enfant qui ne doit pas s'attarder trop longtemps à regarder sa manette, et les expressions de son visage (inquiétude, angoisse, peur...).

3.2 - Commande par manette neuronale

Pour prendre en compte les limitations fonctionnelles d'un enfant IMC (Infirme Moteur Cérébral), un réseau de neurones réalise la compensation automatique du geste après une phase d'apprentissage de la manette de contrôle pour que le matériel s'adapte à l'enfant handicapé et non pas le contraire.

Après plusieurs tests en simulation, le réseau de neurones retenu est constitué de 3 couches :

- une couche d'entrée qui reçoit la position (x,y) de la manette,
- une couche cachée,
- une couche de sortie qui fournit la direction appropriée à la commande du fauteuil et sa propulsion

Parmi les différentes architectures neuronales possibles et compte - tenu de la nature du problème, l'algorithme de fonctionnement est basé sur la rétro-propagation du gradient. La figure 3 en donne la structure avec $v_i^{[0]}$ les données d'entrée, $w_{ij}^{[k]}$ les poids synaptiques, $v_i^{[1]}$ les potentiels de la couche cachée, $v_i^{[2]}$ les sorties obtenues, et k l'indice de couche.

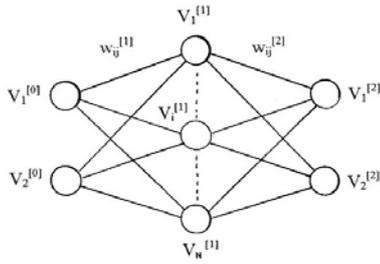


Figure 3 : Réseau de neurones

Les calculs à réaliser sont classiques :

1 - présentation de la donnée d'entrée $v_i^{[0]}$

2 - calcul des potentiels

$$v_i^{[k]} = F \left[\sum_{j=1}^{n_{k-1}} w_{ij}^{[k]} \cdot v_j^{[k-1]} \right]$$

3 - calcul des gradients :

- pour les cellules de sortie :

$$\delta_i^{[k]} = 2F' \left[\sum_{j=1}^{n_{k-1}} w_{ij}^{[k]} \cdot v_j^{[k-1]} \right] \cdot (s_{di}^{[k]} - s_i^{[k]})$$

- pour les autres cellules :

$$\delta_i^{[k-1]} = F' \left[\sum_{j=1}^{n_{k-1}} w_{ij}^{[k]} \cdot v_j^{[k-1]} \right] \cdot \sum_{m=1}^{n_k} w_{mi}^{[k]} \cdot \delta_m^{[k]}$$

avec λ le pas du gradient, F la fonction d'activation et F' sa dérivée.

4 - modification des poids synaptiques :

$$w_{ij}^{[k]} \leftarrow w_{ij}^{[k]} + \lambda \cdot \delta_i^{[k]} \cdot v_j^{[k-1]}$$

Pour illustrer ces calculs par un exemple, nous nous limitons dans le cadre de cet article à la modélisation du calcul du potentiel du premier neurone (figure 4).

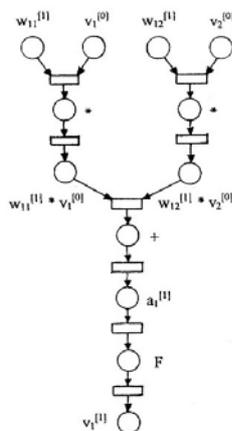


Figure 4 : Réseau de Petri à Flux de Données du calcul du potentiel d'un neurone

Chaque opérateur élémentaire est décrit suivant le principe de la figure 5. Cette description est répétée pour couvrir l'intégralité du RdPFD, comme indiqué figure 6 sur laquelle 3 opérateurs sont pris en compte pour le calcul de $a_1^{[1]} = w_{11}^{[1]} \cdot v_1^{[0]} + w_{12}^{[1]} \cdot v_2^{[0]}$.

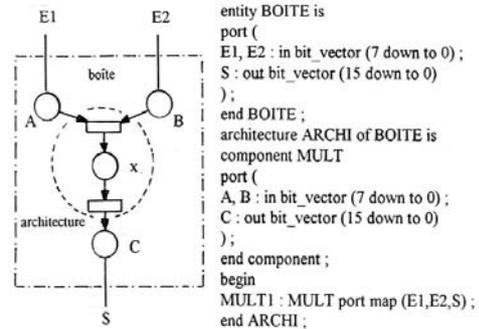


Figure 5 : Description VHDL d'un opérateur

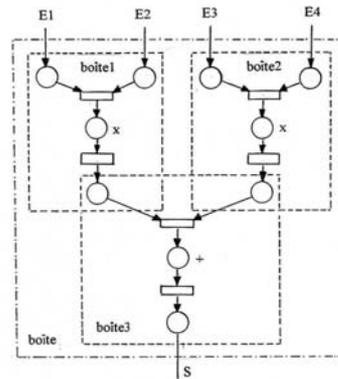


Figure 6 : Description de plusieurs opérateurs

Le programme peut alors être testé et simulé pour une implantation sur diverses cartes commerciales ou personnelles utilisant des composants programmables. La figure 7 donne un exemple de résultats obtenus par Abellard (2002) avec un ensemble logiciel et matériel Altera, après apprentissage dans différentes directions de la manette du fauteuil roulant.

1	1	1	1	1	1	1	1	1	1	1	1	1	1	0,98	0,95	0,45	0,15	0,05
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0,99	0,92	0,56	0,17	0,04
0,6	1	1	1	1	1	1	1	1	1	1	1	1	1	0,97	0,67	0,19	0,04	0,01
0,12	0,59	1	1	1	1	1	1	1	1	1	1	1	1	0,99	0,78	0,23	0,04	0,01
0,03	0,08	0,92	1	1	1	1	1	1	1	1	1	1	1	0,94	0,34	0,04	0,01	0,01
0,01	0,01	0,99	1	1	1	1	1	1	1	1	1	1	1	0,99	0,74	0,08	0,01	0,01
0,01	0,01	0,05	0,98	0,99	1	1	1	1	1	1	1	1	1	1,05	0,37	0,02	0,01	0,01
0	0	0	0	0,59	0,88	0,88	0,98	1	1	1	1	1	1	0,99	0,62	0,11	0,01	0,01
0	0	0	0	0,59	0,5	0,94	0,91	1	1	1	1	1	1	0,89	0,24	0,02	0,01	0,01
0	0	0	0	0,05	0,28	0,17	0,23	0,73	0,98	0,52	0,05	0,01	0,01	0,01	0,01	0,01	0,01	0,01

Figure 7 : Exemple de résultats obtenus après apprentissage

3.3 - Résultats

Sur le plan de l'application, pour amener l'enfant lourdement handicapé et souvent

limité intellectuellement à apprendre comment utiliser fonctionnellement une manette dans la perspective d'une commande de fauteuil roulant électrique, l'ergothérapeute doit impérativement procéder par étapes : étapes progressives, voire parfois régressives dans lesquelles un certain nombre de paramètres doivent être considérés. Une méthodologie de quantification de l'évolution de l'apprentissage a donc été mise au point par J.P. Belheur (2002). Elle repose sur 5 phases :

- Evaluation fonctionnelle des possibilités de la main de l'enfant (figure 8),
- Chronologie rééducative pour la préparation au maniement de la manette (figure 9),
- Exploitation des feedbacks multisensoriels pour l'orientation cardinale optimale du levier de la manette (figure 10),
- Réglages progressifs des paramètres de fonctionnalités de la manette (figure 9),
- Quantification et analyse des progrès / régressions (figure 11).

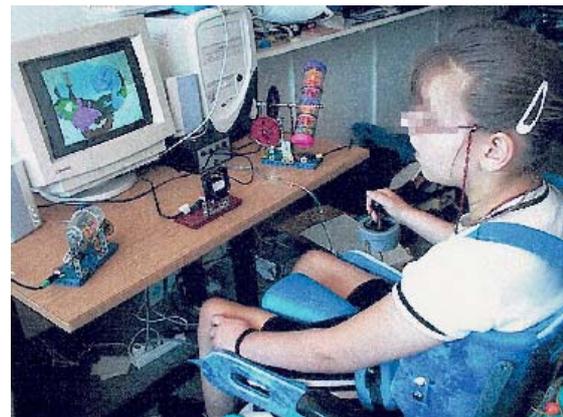


Figure 10 : Jeux sensoriels

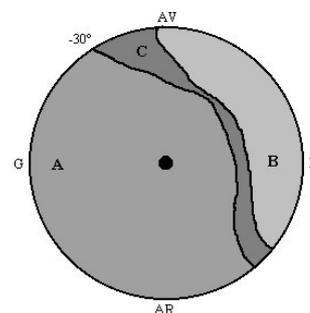


Figure 11 : Analyse de résultats par superposition de zones (A : zone gestuelle fonctionnelle, B : zone difficile à atteindre, C : évolution gestuelle)

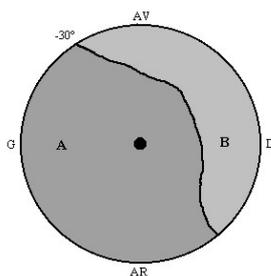


Figure 8 : Evaluation fonctionnelle initiale avant rééducation (A : zone gestuelle fonctionnelle, B : zone difficile à atteindre)

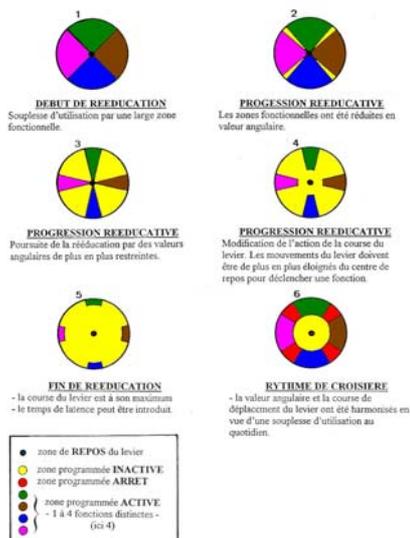


Figure 9 : Chronologie de rééducation gestuelle

4 - CONCLUSION

Les RdPFD constituent un outil efficace de description du parallélisme d'un algorithme de calcul. L'implantation sur une architecture parallèle utilisant des composants programmables est obtenue facilement à partir d'une description VHDL.

Pour l'application décrite, l'architecture neuronale est une bonne solution pour la prise en compte des progressions/régressions. Le rôle des ergothérapeutes est d'aller encore plus loin dans les interventions et les investigations afin de faciliter la vie des enfants IMC. L'ensemble de ces actions devrait permettre d'améliorer l'existant de façon significative car pour certains jeunes IMC, ce n'est pas toujours le temps qui aide le mieux à accéder à l'autonomie, mais plutôt le choix des stratégies rééducatives et des outils mis en oeuvre. Les expériences pratiquées avec des personnes présentant un handicap plus léger ont montré que l'approche innovante de ces travaux accroît la rentabilité dans l'exécution des tâches à accomplir (réduction du temps de

réponse, limitation des erreurs, diminution du facteur émotionnel...) avec une sécurité maximale.

L'électronique et l'informatique sont des outils qui nous conduisent à réfléchir et voir que d'autres personnes adultes lourdement handicapées, même avec des pathologies différentes peuvent elles aussi, accéder à l'autonomie.

REMERCIEMENTS

Les auteurs remercient le Conseil Régional PACA pour la bourse d'étude attribuée pour ce travail, la Chambre de Commerce et d'Industrie du Var et le Centre d'Etudes et de Recherches de Toulon et du Var pour leur contribution financière, et M. J.P. Belheur, ergothérapeute à l'Institut Pomponiana-Olbia (Hyères) et Président de l'Association TROPHEE d'aide à la personne handicapée, pour ses conseils et la fourniture du matériel.

BIBLIOGRAPHIE

- Abellard, P., Grall, P. (2000), "A low cost data flow architecture", *ACIDCA*, Monastir.
- Abellard, P., Grall, P., Abellard, A. (2001), "Adéquation algorithme architectures", *CNR'IUT 2001*, Roanne.
- Abellard, A., Enéa, G., Abellard, P., Belheur, J.P., Nolibé, G. (2002) : "Aides techniques en appareillage pour handicapés : de l'éducation gestuelle au projet FRACAH", *Colloque Recherche et Handicap*, Lyon.
- Alhmana, J. (1983) : "Modélisation par Réseaux de Petri à Flux de Données. Application à la synthèse de l'opérateur de Riccati rapide", Thèse de Doctorat, Université de Marseille III.
- Belhadj, M. (1994) : "Conception d'architectures en utilisant SIGNAL et VHDL", Thèse de Doctorat, Rennes 1.
- Belheur J.P, Pailloux M., Petitprez B., Abellard A., Abellard P. (2002), "Aides techniques d'éducation gestuelle pour enfants IMC", *Colloque HANDICAP 2002*, Paris.
- Ben Khelifa, M. (2001), "Vision par ordinateur et robotique mobile d'assistance. Application au projet MARH, Mobile Autonome Robotisé pour Handicapés", Thèse de doctorat, Université de Toulon.
- Bourhis, G., Gelin, R., Pruski, A. (2000), *Robotique d'aide aux personnes handicapées. Applications non manufacturières de la robotique mobile*, pp. 193-238, Ed. Hermès.
- Dias, A.F., "Contribution à l'implantation optimisée d'algorithmes bas niveau de traitement du signal et des images sur des architectures mono-FPGA à l'aide d'une méthodologie d'Adéquation Algorithmes Architectures". Thèse de Doctorat, Paris XI Orsay.
- Rompaey, K., Verkest, D., Bolsens, I., Deman, H. (1996) : "Coware : a design environment for heterogenous hardware/software systems" in *Design Automation for Embedded Systems*, vol 1, n° 4, pp 357-362.
- Rubini, S, Lavenier, D. (1997) : *Les architectures reconfigurables. Les calculateurs parallèles*, Vol 9, n°1.

DETECTION DE VEHICULES DANS UN ENVIRONNEMENT AUTOROUTIER A L'AIDE DE DONNEES TELEMETRIQUES ET VISUELLES

Sonia Izri,
Doctorante en Robotique
Izri@iut-amiens.fr + 33 3 22 53 40 88

Eric Brassart,
Maître de conférence en Robotique
Eric.brassart@iut.u-picardie.fr + 33 3 22 53 40 80

Laurent Delahoche
Maître de conférence en Robotique
Laurent.delahoche@iut.u-picardie.fr + 33 3 22 53 40 83

Arnaud Clémentin
Docteur en Robotique
Arnaud.clémentin@iut.u-picardie.fr + 33 3 22 53 40 85

Adresse professionnelle
IUT d'Amiens : Département informatique Université de Picardie Jules–Verne,
Avenue des facultés, 80025 Amiens CEDEX 1

Résumé : Nous présentons dans cet article les principes de traitements de données mis en œuvre dans le cadre d'un projet de recherche sur l'amélioration de la sécurité automobile, intitulé **SAACAM** (**S**ystèmes **A**ctifs d'**A**ide à la **C**onduite pour **A**méliorer la sécurité automobile), qui est contractualisé dans le cadre du pôle régional **DIVA** (**D**iagnostic et **V**éhicules **A**vancés) de la région Picardie.

Ce projet est basé sur l'utilisation conjointe de deux systèmes de perception extéroceptive :

- Un télémètre laser (le SICK LMS 200)
- Un capteur de vision omnidirectionnel.

Nous montrons dans cet article les résultats obtenus dans un environnement structuré de type autoroutier, et nous indiquons plus spécifiquement les temps de calcul des différents processus mis en œuvre.

Nous terminons en présentant quelques perspectives qui envisagent de définir un système de perception bi-sensoriel avec les capteurs extéroceptifs et tenter de fusionner deux types de primitives issues de ces derniers dans le but de localiser de façon relative les véhicules avoisinant.

Summary : We will present in this article the principles of data processing put into operation within the framework of a research project on the improvement of automobile safety, entitled **SAACAM** (**S**ystems **A**ctifs d'**A**ide à la **C**onduite pour **A**méliorer la sécurité automobile : *Active Systems of Driving Assistance for Improvement of Motorcar Safety*) which is contracted within the framework of the **DIVA** (**D**iagnosis and **V**ehicles **A**vancés : *Diagnosis and Advanced Vehicles*) regional pole of the Picardy region.

This project is based on the simultaneous use of two systems of exteroceptive perception :

- A telemetric laser (the SICK LMS 200)
- An omnidirectional vision sensor.

In this article we will show the results obtained in a structured motorway-type environment and indicate more specifically the calculation times employed.

We will end by presenting some perspectives, which attempt to define a bi-sensory system of perception with the exteroceptive sensors and try to merge two types of primitive results of these latter with the aim of localizing the neighbouring vehicles in a relative way

Mots clés : Perception, Vision omnidirectionnelle, contours actifs, segmentation.

Détection de véhicules dans un environnement autoroutier a l'aide de données télémétriques et visuelles

1. Introduction

Les projets concernant l'assistance à la sécurité routière par intégration de capteurs dits 'intelligents' sont extrêmement nombreux [CARSENSE, RADARNET, DENSETRAFFIC, EAST-EEA, etc.] et correspondent à une nécessité de plus en plus perceptible dans notre vie quotidienne. Cette problématique est, de surcroît, fortement encouragée par les constructeurs automobiles, les conducteurs de véhicules, la vie politique, le corps médical et les associations.

Quelque soit le niveau d'intégration des systèmes sécuritaires, quelque soit l'architecture fonctionnelle déployée, le problème récurrent reste la quantité d'information à traiter par rapport à la dynamique mise en jeu. Ainsi la synchronisation des traitements, les temps de traitement, l'acquisition temps réel sont autant de contraintes qui, rendent l'objectif final difficile à atteindre. Dans ce contexte on cherchera évidemment à privilégier des 'solutions capteurs' qui permettront soit d'obtenir un maximum d'informations en une seule acquisition, soit une information pré-traitée, qui de plus pourra être facilement embarquée sur un véhicule. Les systèmes de vision omnidirectionnelle [10], [11] sont à ce titre très intéressants à utiliser puisque lorsqu'ils sont embarqués sur un véhicule, ils permettent de détecter des dangers proches sur 360 degrés et cela en une seule acquisition. L'intégration d'un tel capteur sur un véhicule pour la détection des dangers proches est originale puisqu'elle n'a pas encore été faite à ce jour. En outre des données visuelles, nous utilisons dans le projet un autre système de perception extéroceptive qui est le télémètre laser. Ce dernier permet d'obtenir des nuages de points issus de mesures télémétriques, sur lesquels, nous cherchons à identifier les objets présents dans l'image.

Cette solution que nous présentons fait parti du projet intitulé SAACAM (Système Actifs d'Aide à la Conduite pour AMéliorer la sécurité automobile) et est contractualisée dans le cadre du pôle régional DIVA (DIagnostic et Véhicules Avancés) de la région Picardie.

Dans cet article nous détaillerons d'une part, les mécanismes des différents traitements effectués sur les données télémétriques. D'autre part, nous présenterons l'étude d'extraction d'indices visuels dans les images panoramiques, basée sur l'utilisation des modèles déformables de type 'snakes'. Nous terminerons cet article par la présentation des premiers résultats obtenus dans un environnement structuré de type autoroutier. Finalement nous donnerons quelques perspectives à la suite de nos travaux.

2. Traitement des données télémétriques

2.1. Présentation générale du SICK LMS 200

Le SICK LMS 200 est un scanner de mesure à balayage 2 dimensions : il scrute son environnement et récupère les coordonnées polaires de celui-ci. C'est une combinaison d'un télémètre à temps de vol avec un système de rotation du faisceau de mesure, grâce à cette technique on obtient une vision sous forme de radar.



Figure 1. SICK LMS_200

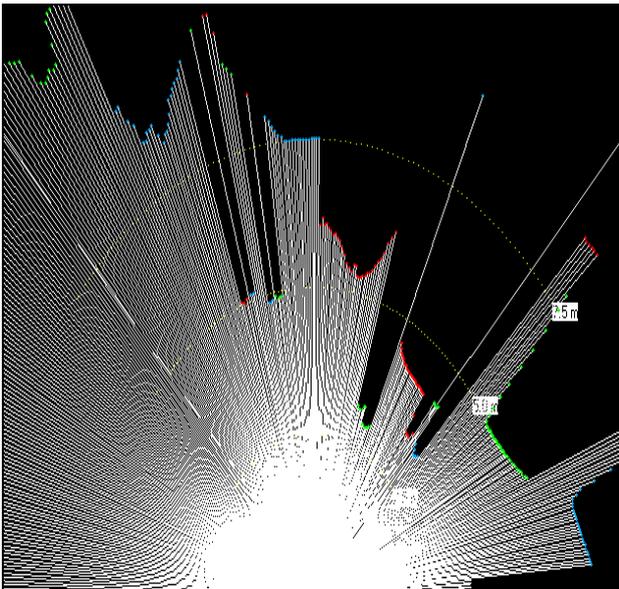


Figure 2 Image laser acquise avec le télémètre

2.2. Fonctionnement du SICK LMS 200

Une impulsion (un tir) est émise par une diode laser et simultanément, une horloge est démarrée. Cette impulsion lumineuse sera renvoyée par le premier obstacle rencontré sur son chemin. L'impulsion lumineuse, renvoyée par un obstacle, arrive sur un récepteur qui déclenche l'arrêt de l'horloge. A partir de ces informations, nous déduisons la distance séparant le télémètre et l'obstacle. Et connaissant la position angulaire du laser, nous arrivons à déterminer la position de l'objet dans le plan.

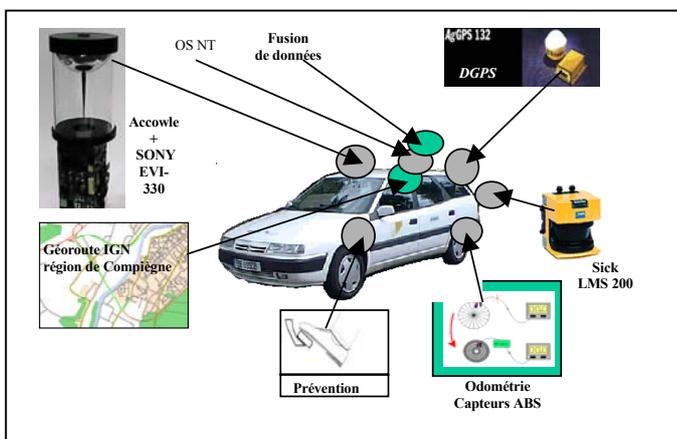


Figure 3 Schéma de fonctionnement

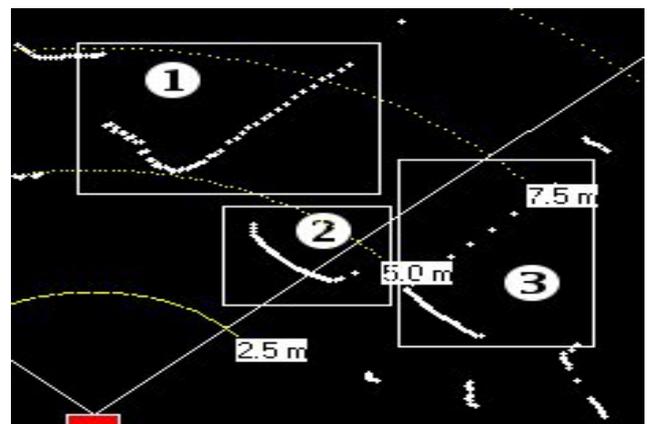
2.3. Intégration du SICK LMS 200 dans le projet

Le télémètre a été placé à l'arrière du véhicule pour des raisons sécuritaires, ce qui permet de récupérer une image des véhicules suiveurs, comme le montre la figure ci-dessous (**figure 3**).

2.4. Traitement des données du télémètre laser

2.4.1. Le "clustering"

Le "clustering" permet de regrouper des points consécutifs afin de répertorier l'ensemble des objets présents. Pour cela, nous avons calculé la distance qui sépare un point N de son voisin immédiat N+1. Dans l'exemple suivant, nous montrons, par les encadrements, ce que nous devrions obtenir à l'issue de cette étape.



Dans cet extrait de carte nous pouvons distinguer :

- ❶ La face avant et le côté gauche d'un véhicule.
- ❷ La face avant d'un véhicule.
- ❸ La face avant et le côté droit d'un véhicule.

Figure 4. Extrait d'une image avec identification des objets caractéristiques

Les alignements de points sont caractéristiques de la détection d'arêtes appartenant à des objets. Quand la continuité n'est pas respectée, il faut donc créer une séparation de cluster.

La détermination des seuils de rupture dans l'identification des différents objets se fait actuellement de manière heuristique. Ce seuil est pour l'instant fixé à 50 cm.

Pour pouvoir mieux identifier les résultats, nous les avons affichés en 3 couleurs différentes (RVB) qui se répèteront successivement.

Le schéma ci-dessous illustre le traitement effectué sur les données télémétriques en l'occurrence la lecture des données et l'affichage des clusters coloriés.

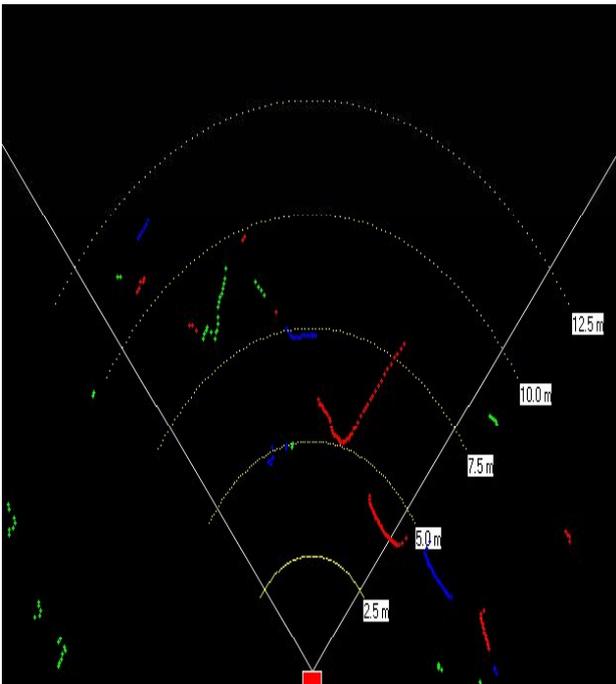


Figure 5. Lecture des données et affichage des clusters coloriés

2.4.2. La segmentation

Après avoir "clusteriser" les données, vient l'étape de segmentation, celle-ci permet de déterminer les différentes parties d'un cluster. Ce dernier est composé d'un ou plusieurs points représentant une partie d'un objet ou sa totalité.

Les points n'étant pas tous alignés, la segmentation va permettre de tracer les segments de droites correspondant à un sous-ensemble de points.

L'algorithme de recherche utilisé est celui de Duda-Hart [13]. Il consiste à regrouper

récurivement des ensembles de points alignés par rapport à un critère de distance point / droite support. L'algorithme est stoppé lorsqu'il n'y a plus de points qui vérifient la condition de distance. En appliquant cette méthode sur les différents clusters, on pourra déterminer les différentes droites qui composent chaque cluster.

2.4.3. Filtrage

2.4.3.1. Suppression des « petits » segments

Les 'petits' clusters ne permettent pas de conclure quant à la nature de l'objet qui est représenté. Et les clusters composés d'un seul point ne sont pas utilisables non plus.

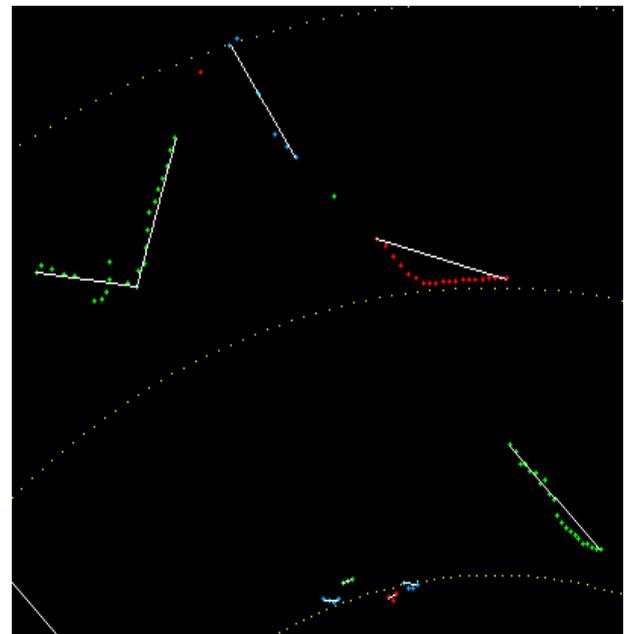


Figure 6 . Suppression des clusters non significatifs

Pour supprimer ces éléments, un seuil minimum de points est fixé.

En parcourant toute la liste des clusters, nous recherchons le nombre de points qui compose chaque cluster et si ce nombre est inférieur au minimum imposé, alors il est supprimé de la liste.

2.4.3.2. Fusion de droites alignées

La fusion de droites alignées permet de fusionner deux droites qui sont :

- Alignées avec un angle θ de $\pm 10^\circ$,
- Une distance.

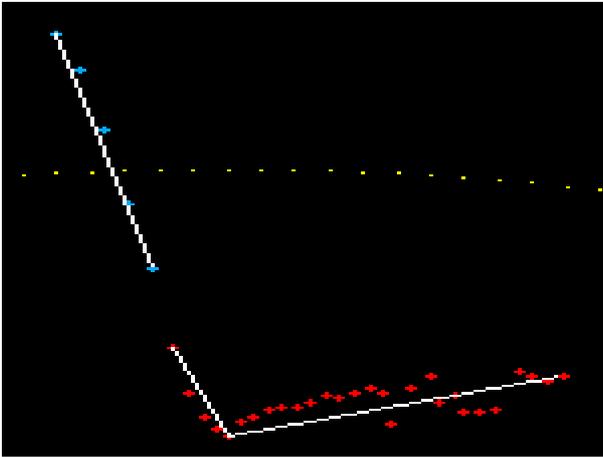


Figure 7. Exemple de fusion de droites alignées

2.4.3.3. Fusion de droites orthogonales

Pour la fusion de droites orthogonales, on va supposer que ces deux éléments n'en forment qu'un seul suivant des critères définis et modifiables par l'utilisateur.

Dans le schéma ci-contre, nous pouvons remarquer que si on prolonge les deux droites, elles se coupent en un point I. Nous pourrions ainsi dire que cet élément forme un objet et non deux objets.

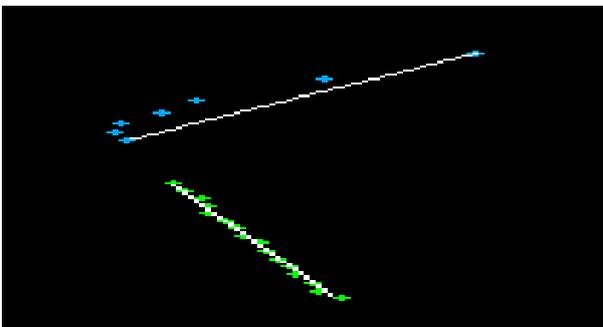


Figure 8. Exemple de fusion de droites orthogonales

3. Traitement des données omnidirectionnelles

3.1. Les systèmes de vision omnidirectionnelle

Ces systèmes de perception associent une caméra CCD et un miroir convexe (sphérique) permettant d'obtenir une vue sur 360 degrés de l'environnement, comme le montrent les figures ci-dessous (**figure 9, 10**). Nous comprendrions aisément son intérêt dans le cas d'une utilisation sur un véhicule dans un milieu autoroutier, pour une application de type 'dangers proches'.

Voici un exemple de ce l'on peut obtenir :



Figure9 . Capteur sphérique Accowle



Figure 10. Image obtenue par le capteur sphérique

3.2. Modélisation par contours actifs

La première utilisation des contours actifs date de 1987, lorsque Kass, Wiltkin et Terzopoulos [2] présentent leurs *snakes* comme un outil permettant de résoudre plusieurs sujets majeurs de la vision par ordinateur, à savoir la détection de contours et de segments [8], la mise en correspondance en stéréovision, et surtout le suivi d'objets. La compacité et l'efficacité du concept [9] lui ont assuré un succès rapide et on trouve actuellement une multitude de publications sur la variation de sa forme originelle.

Le principe consiste à placer dans l'image un *snake* qui 'se colle' au contour de la cible. Une énergie lui est associée, fonction de sa déformation et de sa position dans l'image. Le *snake* cherche en permanence à minimiser cette énergie. Celle-ci est modélisée de telle manière que le comportement résultat est une suite de déplacements et de déformations jusqu'à ce que le *snake* se stabilise sur un minimum local de son énergie qui correspond alors à la forme de l'objet suivi. Un *snake* se comporte comme une corde élastique qui serpente vers les contours de l'image grâce à un ensemble de forces locales gérées par un processus itératif. Les forces internes conservent la forme et assurent une continuité spatio-temporelle. Les forces externes tirent et guident le *snake*. L'initialisation et la normalisation des paramètres du *snake* sont difficiles. Le *snake* doit être placé assez proche du contour pour atteindre le minimum global de la fonction d'énergie. Ce système est utilisé dans de nombreux cas où le traitement de l'image est fait en temps réel comme en imagerie médicale, mais également pour détecter des contours déjà connus comme en botanique ou en paléontologie pour l'identification des espèces.

Le choix de l'utilisation de cette méthode se justifie par diverses raisons :

- méthode extensible
- interaction de l'utilisateur
- intégration de caractéristiques très variantes (informations géométriques et physiques)

Cependant, ce modèle présente certains défauts :

- l'influence de l'initialisation du *snake*
- le choix des paramètres élastiques est dépendant de l'image de travail
- il conserve sa topologie initiale au cours de ses déformations.

3.2.2 Les différentes forces du snake

La méthode de détection par *snake* ou contour actif, consiste à placer aux alentours de la forme à détecter une liste de points voisins deux à deux, représentée par une courbe.

Le *snake* peut être représenté (figure 11) de la manière suivante :

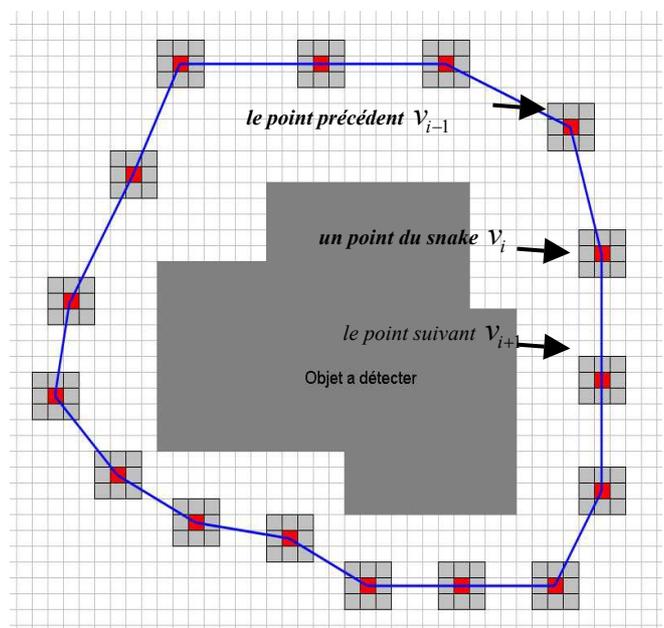


Figure 11. Schématisation d'un snake autour de l'objet à détecter

Comme le montre la figure 12, on ne s'intéresse pas qu'à chaque point du *snake* mais également à chaque voisinage de ces points, ce voisinage a une dimension de 3 x 3 pixels.

- Le *snake* est constitué d'un ensemble de n points, $v_i = (x_i, y_i)$.
- Chaque point v_i , est caractérisé par ses voisins proches : Le point précédent v_{i-1} et

le point suivant v_{i+1} . Un voisinage \mathcal{N}_i de taille m (dans notre cas $m=8$), est défini pour chaque point v_i .

•

La courbe ainsi obtenue que l'on peut formuler dans le plan par une forme paramétrique : $v(s) = (x(s), y(s))$, $s \in [0, 1]$, va se déformer de manière itérative selon l'action de plusieurs forces qui vont la tirer vers la forme.

Les forces impliquées dérivent de trois énergies associées au *snake* :

- Une énergie propre, due uniquement à la forme du contour, dite énergie interne E_{int} ,
- Une énergie potentielle imposée par l'image E_{image} . C'est elle qui va attirer le *snake* vers les contours réels présents,
- Une énergie de contraintes E_{cont} qui exprime certaines contraintes supplémentaires qui peuvent être imposées par l'utilisateur vu le *snake* qu'il veut obtenir.

L'énergie totale du *snake* est donc donnée par :

$$E_{snake} = \int_0^1 E_{snake}(v(s)) ds = \int_0^1 E_{int}(v(s)) + E_{image}(v(s)) + E_{cont}(v(s)) ds. \quad (1)$$

• **L'énergie interne :**

Dépend uniquement de la forme du *snake*. C'est une contrainte de régularité sur la courbe. Elle s'écrit :

$$E_{int} = (\alpha(s) \|v_s(s)\|^2 + \beta(s) \|v_{ss}(s)\|^2) / 2 \quad (2)$$

α et β sont les poids respectifs accordés à la dérivée première v_s et à la dérivée seconde v_{ss} .

Le premier coefficient contrôle le terme de premier ordre, qui agit sur la continuité (longueur) de la courbe, et le deuxième contrôle le terme de deuxième ordre qui agit sur sa courbure.

• **L'énergie potentielle liée à l'image :**

$$E_{image} = \gamma(s) E_{image} \quad (3)$$

Elle caractérise les éléments vers lesquels on veut attirer le *snake* sur l'image. Pour la détection de contours, on va s'intéresser aux lignes de fort gradient, on choisit, donc, le gradient de l'image au point considéré.

$$E_{image} = - \int_0^1 |\nabla I(v(s))|^2 ds \quad (4)$$

• **L'énergie de contraintes :**

Elle est définie par l'utilisateur selon les spécificités du problème. Par exemple imposer une distance minimale ou maximale entre deux points consécutifs du contour actif.

Une fois les énergies de chaque point du voisinage calculées, on est à même de décider vers quel pixel de l'image le point du *snake* va migrer (vers le point du voisinage dont la somme des 3 énergies est la plus faible) tel qu'on peut le voir sur la (figure 12).

Voici la manière dont sont codés les emplacements des points dans le voisinage.

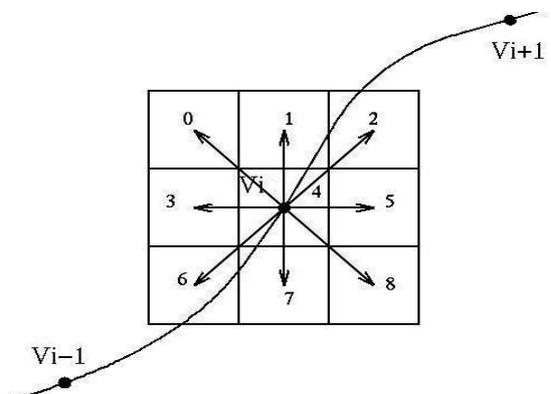


Figure 12. Déplacements possibles d'un point dans son voisinage

4. Résultats obtenus

4.1. Transformation omnidirectionnelle en panoramique

Les résultats obtenus initialement avec les images omnidirectionnelles ne permettent pas d'avoir des détections de contours correctes sur les objets. De ce fait, nous avons développé une méthode permettant la

conversion des images omnidirectionnelles en images panoramiques ; plus facilement interprétables d'un point de vue humain, et permettant de rejeter les zones de l'image où l'information visuelle ne subit pas le bruit du capteur. Aussi, nous avons décidé de ne travailler que sur certaines portions d'images en définissant des « imagettes » qui présentent l'avantage de ne pas trop pénaliser en temps de calcul le processus de traitement. Le procédé mis en œuvre pour réaliser cette transformation est basé sur la méthode de l'interpolation bilinéaire [10], qui permet d'obtenir des images redressées et non pixélisées.

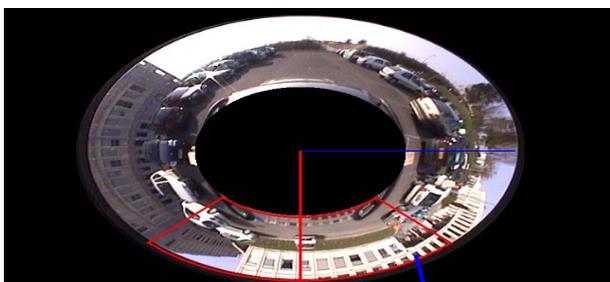


Figure 13. Application d'un masque sur l'image omnidirectionnelle et tracé d'une imagette



Figure 14. Imagette panoramique résultante

4.2. Décomposition de l'imagette en trois composantes RVB

La caméra utilisée dans le projet nous a permis d'obtenir des images couleurs. Ce type d'images nous offre une information supplémentaire par rapport aux images en niveaux de gris et qui est apportée par la chrominance. On peut imaginer, par exemple que le changement de couleur entre deux

objets de luminance équivalente définira un contour que seule la donnée de luminance n'aurait pas permis de détecter. Nous avons d'abord, procédé à la décomposition de chaque imagette couleur en trois composantes RVB, ensuite calculer le gradient de chaque composante et avons choisi parmi les méthodes permettant de déterminer les contours d'une image couleur celle basée sur le calcul du Max des trois gradients sur les trois composantes couleurs. C'est ce dernier, qui va attirer le *snake* vers les contours présents.

Cela permettra de constituer une image finale moins volumineuse et plus adaptée à un traitement temps réel.

La figure ci-dessous illustre la décomposition d'une imagette panoramique en trois composantes RVB.

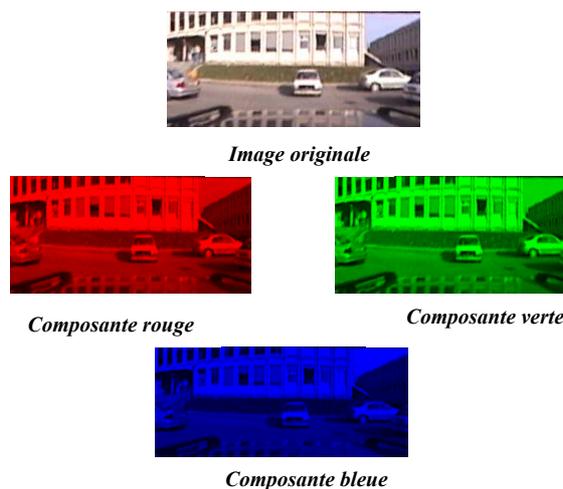


Figure 15. Décomposition de l'imagette couleur en composantes RVB

4.3. Résultats des contours actifs

La mise en place du processus dans son ensemble se fait par un ajustement de plusieurs paramètres liés à l'utilisation des *snakes*. Pour notre application nous avons, en fonction de différents tests effectués au préalable, positionné les valeurs des paramètres, α , β caractérisant l'énergie interne respectivement à 0.2, 0.8, les

traitements images de l'énergie externe sont basés sur un calcul de gradient localisé de type Sobel (gradient qui fournit le meilleur résultat dans ces types d'images), le paramètre γ varie de 1 à 4 selon l'image et le gradient choisi. L'initialisation du *snake* s'effectue pour l'instant, à la main, et nous nous sommes limités ponctuellement au repérage d'un seul objet.

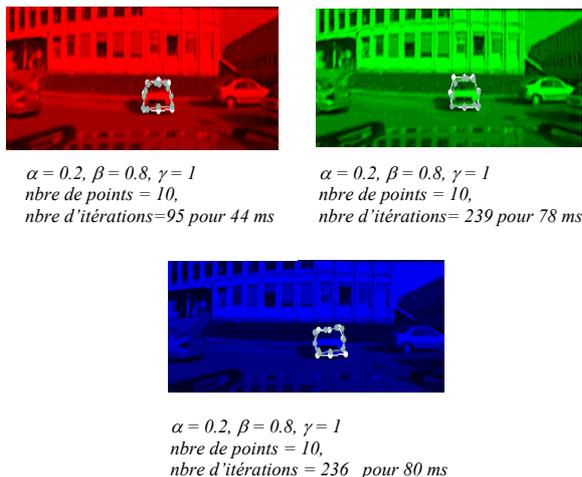


Figure 16. Résultats obtenus par application des snakes sur les trois composantes RVB

Nous montrons dans la figure 16, des exemples de résultats que nous avons obtenus dans un environnement d'extérieur avec un capteur omnidirectionnel sphérique. Ces derniers sont très encourageants puisque, après initialisation *du snake* [9], nous arrivons à suivre, sur une séquence d'images, un objet spécifique avec des temps de traitement variant de 40 ms à 100 ms.

5. Conclusion

Dans cet article, nous avons étudié et mis au point le mécanisme de détection des objets à partir de données télémétriques et visuelles. Après avoir détaillé le principe des différents traitements effectués sur les données télémétriques, ainsi que le formalisme des contours actifs, appliqué sur les images omnidirectionnelles, différentes

expérimentations ont été menées sur des environnements réels, hautement structurés de type autoroutier. Une analyse des résultats obtenus montre la compacité et l'efficacité de la méthode. Elle intègre des caractéristiques très différentes, telles que des informations géométriques, de contraste et d'intensité. Par ailleurs, le modèle travaille indépendamment de l'orientation de l'image, ce qui permet de réaliser le suivi d'objets dans l'image.

6. Perspectives

Nous nous sommes attachés à définir un système de perception et des méthodes d'extraction de primitives par rapport à un contexte autoroutier.

Notre stratégie est basée sur le suivi des véhicules proches, pour se faire, nous nous proposons d'utiliser une architecture de fusion de données multi-niveaux et multi-critères (**figure 17**).

Rappelons que l'objectif est de localiser de façon relative les véhicules avoisinant. Notre démarche consistera donc à propager une mise en correspondance d'une acquisition n à une acquisition $n + 1$ tant que le ou les véhicules proches apparaissent comme des situations de danger potentiel.

Nous souhaitons nous appuyer sur la théorie de l'évidence de Dempster-Shafer [14] pour réaliser une fusion de plusieurs critères qui devront caractériser de façon robuste la croyance accordée à chaque primitive sensorielle détectée, issue du modèle bi-sensoriel. Cette théorie permet de bien manipuler des données incertaines.

En effet, contrairement à la théorie de Bayes, on peut porter de la masse sur les hypothèses singleton, mais aussi sur les unions d'hypothèses.

A chaque niveau, on manipule un cadre de discernement composé de deux hypothèses :

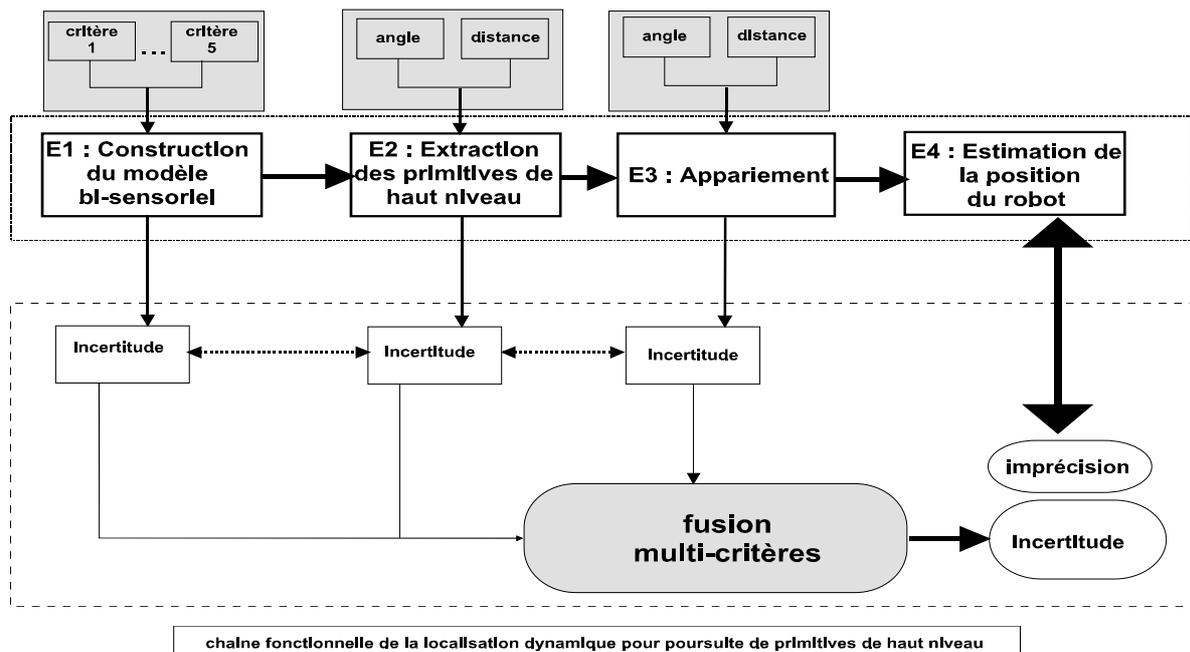


Figure 17. Schéma fonctionnel de la fusion de données multi-critères

OUI et NON. Par exemple, à l'étape E1, l'hypothèse OUI correspond à l'affirmation « OUI, le segment existe », et NON à « NON, le segment n'existe pas ». Ainsi, à l'issue de la fusion des différents critères, l'incertitude du segment sera dénotée par trois valeurs : $m(\text{OUI})$, $m(\text{NON})$, et $M (\text{OUI} \cup \text{NON})$, ces trois valeurs nous permettent d'obtenir une quantification de l'incertitude sur le segment.

La redondance et la complémentarité des données sensorielles seront exploitées pour ne garder que des primitives qui correspondront aux véhicules proches.

L'étape E2 de classification de ces primitives sera de déterminer l'incertitude liée à l'extraction des primitives de 'haut niveau' (en l'occurrence des véhicules) par propagation des incertitudes associées à chaque segment.

Une fois les cibles étiquetées au sein du modèle sensoriel obtenu à un instant t , la phase de suivi multi-cibles E3 consistera à mettre en correspondance ces amers de type voiture avec ceux obtenus à l'instant $t-1$: nous souhaitons développer une solution basée sur la fusion multi-critères, les critères seront liés à des stratégies d'association de type 'prédiction-observation'.

La démarche permettant de réaliser le suivi multi-cibles sera donc de nouveau basée sur l'utilisation des règles de Dempster-Shafer. Nous souhaitons évidemment traiter le problème de la « non observabilité momentanée » dans le cadre du suivi et de la mise à jour des incertitudes. Ainsi, l'incertitude gérée dans les niveaux précédents sera prise en compte dans cette dernière étape. Nous avons montré dans [11][12] que l'utilité de gérer ces incertitudes est multiple. En effet, cette approche nous permet d'écartier les pistes non fiables et ainsi travailler avec des données robustes ce qui permet notamment de ne pas supprimer

brutalement une piste dès qu'un appariement n'est pas réalisé. Enfin, avec cette stratégie il nous sera possible de déduire l'incertitude globale sur la localisation par propagation des incertitudes calculées sur chacune des pistes. Cette gestion orientée sur la cohérence des incertitudes aura pour conséquence de ne déclencher des alarmes que dans des configurations à fort niveau de dangerosité.

7. Bibliographie

- [1] Laurent Cohen. "Modèles déformables", CEREMADE, URA CNRS 749, Université Paris-9 Dauphine.
- [2] M. Kass, A. Witkin and D. Terzopoulos. "Snakes: Active contour models". Proc. 1st Int. Conference on Computer Vision, London, 1987, pp. 259-268.
- [3] F. Leymarie and M. Levine. "Tracking deformable objects in the plane using an active contour model". IEEE Trans. on PAMI, Vol. 16 (1993), pp. 617-634.
- [4] S. de Pablo, S. Lorenzo, S. Cáceres and J.A. Domínguez. "Dynamic Image Segmentation based on DSP". Proc. of Int. Conference on Signal and Image Processing (SIP'95), Las Vegas (Nevada-USA), 1995, pp. 109-112.
- [5] Donna J. Williams and Mubarak Shah - "A Fast Algorithm For Active Contours and Curvature Estimation" – Image Understanding, Vol55, N°1, January 1992, pp14-26.
- [6] Lavanya Viswanathan – « Equations For Active Contours », Novembre 1998
- [7] J.F. Canny - "A Computational Approach to Edge Detection" , IEEE, Trans. Pattern Anal. Mach. Intelligence, PAMI-8(1986), pp679-698
- [8] A.Y. Yuille, D.S. Cohen and P.W. Hallinan - "Feature Extraction From Faces Using Deformables templates", in CVPR, 1989, pp104-109
- [9] V. Caselles and B. Coll. "Snakes in Movement". Preprints of AERFAI (Córdoba-Spain) 1995, pp. 683-684.
- [10] Cyril Cauchois. "Modélisation et Calibration du Capteur Omnidirectionnel SYCLOP : Application à la localisation Absolue en Milieu Structuré". Université de Picardie Jules Verne. Décembre 2001.
- [11] Arnaud Clémentin."Localisation d'un robot mobile par coopération multi-capteurs et suivi multi-cibles", Université de Picardie Jules Verne. Décembre 2001.
- [12] Arnaud CLERENTIN, Laurent DELAHOUCHE, Eric BRASSART, Cyril CAUCHOIS – "Mobile robot localization based on multi target tracking" - proc. of the IEEE International Conference on Robotics and Automation (ICRA 2002), Washington, USA, Mai 2002.
- [13] J. Crowley, "World modelling and position estimation for a mobile robot using ultrasonic ranging", Proc. Of IEEE Conference on Robotics and Automation, Scottsdale, May 1989, p. 674-680.
- [14] SHAFER G.A. "A mathematical theory of evidence", Princeton university press, 1976.

RECONSTRUCTION DE LA CORNEE
PAR ANALYSE ASCENDANTE-DESCENDANTE

Sébastien Richard,

Doctorant en Informatique
sebastien.richard@lsis.org , + 33 4 91 82 85 35

Rémy Bulot,

Professeur
remy.bulot@lsis.org + 33 4 91 82 85 36

Jean Sequeira,

Professeur
jean.sequeira@lsis.org +33 4 91 82 85 21

Adresse professionnelle

Laboratoire LSIS, UMR CNRS 6168 Equipe LXAO – Campus de Luminy (ESIL)
Marseille, FRANCE

Résumé : La vidéokératographie est un procédé qui permet d'étudier la géométrie de la surface antérieure de la cornée à partir de l'image 2D de la mire réfléchiée par cette surface. Bien que la reconstruction ne soit pas fiable dans de nombreux cas, ce type de mesure est utilisé dans le cadre d'interventions chirurgicales en ophtalmologie. Nous proposons une nouvelle méthodologie de reconstruction basée sur un processus d'analyse ascendant-descendant. La phase ascendante permet de fiabiliser les différentes étapes du traitement pour aboutir à un modèle fiable et robuste de la cornée. La phase descendante permet d'ajuster plus finement le premier modèle obtenu en confrontant ce dernier directement sur les données initiales. On s'affranchit ainsi du « bruit » introduit dans la mesure par les différentes étapes intermédiaires de la phase ascendante. Ce nouveau procédé, qui met en œuvre des techniques de modèle déformable et de lancer de rayons, fiabilise la mesure en limitant les hypothèses de départ et en prenant en compte les caractéristiques tridimensionnelles de la cornée pendant la reconstruction 3D.

Summary : Videokeratography can provide detailed information about the subtle shape of the corneal surface. The three-dimensional reconstruction is not reliable in many cases however this technique is used by ophthalmologic surgeons. We propose a new reconstruction methodology based on ascending-downward analysis. The ascending analysis provides reliable model of the cornea and the downward analysis refines the precedent model using directly data acquired by videokeratograph. The downward analysis deletes noise created by intermediate steps of ascending analysis. This new method makes reliable the measurement not only by limiting the strating assumptions but also by taking into account corneal three-dimensional properties during 3D reconstruction.

Mots clés : Imagerie médicale, reconstruction cornéenne tridimensionnelle, modèle déformable, stratégie d'analyse ascendante-descendante.

Reconstruction 3D de la cornée par analyse ascendante-descendante

La vidéokératographie est un procédé qui permet d'étudier la géométrie de la surface antérieure de la cornée à partir de l'image 2D de la mire réfléchi par cette surface. Bien que la reconstruction ne soit pas fiable dans de nombreux cas, ce type de mesure est utilisé dans le cadre d'interventions chirurgicales en ophtalmologie. Nous proposons une nouvelle méthodologie de reconstruction basée sur un processus d'analyse ascendant-descendant. La phase ascendante permet de fiabiliser les différentes étapes du traitement pour aboutir à un modèle fiable et robuste de la cornée. La phase descendante permet d'ajuster plus finement le premier modèle obtenu en confrontant ce dernier directement sur les données initiales. On s'affranchit ainsi du « bruit » introduit dans la mesure par les différentes étapes intermédiaires de la phase ascendante. Ce nouveau procédé, qui met en œuvre des techniques de modèle déformable et de lancer de rayons, fiabilise la mesure en limitant les hypothèses de départ et en prenant en compte les caractéristiques tridimensionnelles de la cornée pendant la reconstruction 3D.

1 - INTRODUCTION

La forme de la cornée joue un rôle essentiel dans le processus de la vision humaine (près des trois quarts de la puissance réfractive de l'œil). Lorsque cette forme ne permet pas une focalisation correcte des images sur la rétine, les pathologies associées (myopie, hypermétropie, astigmatisme) peuvent être corrigées par des techniques chirurgicales telles que la kératotomie radiaire. Ces techniques occupent en ophtalmologie une place de plus en plus importante avec l'utilisation, en particulier, de « lasers excimer » dans le but d'effectuer des entailles dans la cornée ou procéder à l'ablation d'une partie de celle-ci. Pour ce type d'opération, il est indispensable de pouvoir évaluer au préalable et de manière très fine la surface antérieure de la cornée.

La vidéokératographie [1] (figures 1 et 2) est un des procédés les plus employés pour cette étude et utilise pour cela le principe de réflexion de rayons lumineux sur une surface convexe. Le processus de mesure peut se résumer en trois étapes distinctes :

- **l'acquisition** : une mire (généralement constituée d'anneaux lumineux concentriques portés par un cône) est projetée sur la cornée qui est utilisée comme un miroir convexe et l'image réfléchi est acquise au moyen d'une caméra vidéo ;
- **l'analyse 2D** : les contours de la mire sont recherchés dans cette image en niveau de gris ;

- **la reconstruction 3D** : les déformations observées sur ces contours sont analysées pour effectuer une reconstruction tridimensionnelle de la surface étudiée.

Toutefois, les algorithmes employés actuellement dans les vidéokératographes font un certain nombre d'hypothèses parfois bien trop restrictives (comme la propriété de symétrie cylindrique de la cornée) qui ne permettent pas une modélisation correcte pour de nombreuses pathologies (comme dans le cas de kératocônes). En particulier, nous avons constaté que, pour une même cornée, deux prises de vue faites sous des angles légèrement différents peuvent aboutir à des résultats éloignés. Les erreurs observées relèvent aussi bien de la phase d'analyse que de la phase de reconstruction et s'accumulent dans le résultat final. Il nous a donc paru intéressant de revoir tout le processus de mesure pour le fiabiliser.

Nous avons déjà proposé une approche originale [2] pour sélectionner dans l'image acquise par un vidéokératographe les informations robustes et significatives de la mire (analyse 2D qui est une étape déterminante avant la reconstruction effective de la surface).

Nous nous intéresserons ici à la dernière étape dans le processus de mesure : la phase de reconstruction 3D. La principale originalité de ce travail porte sur la méthodologie d'analyse plus que sur les algorithmes mis en œuvre.

2 – LES DONNEES

2.1 – Remarques préliminaires

Nous désignerons par :

- **plan méridien**, un plan contenant l'axe optique du vidéokératographe ;
- **sommet** de la cornée, le point d'intersection entre la cornée et l'axe optique de l'appareil ;
- **projection** d'un point A sur une surface (plan image, mire, cornée), le point d'intersection de cette surface et du rayon lumineux issu de A et passant par le diaphragme (après une éventuelle réflexion sur la cornée).

2.2 – Les données d'acquisition

Le vidéokératographe que nous employons fournit une image plane de 512 pixels de large par 480 de haut avec 256 niveaux de gris. La mire projetée est constituée de 8 anneaux lumineux concentriques, soit au total 16 contours à détecter (voir figures 1 et 2).

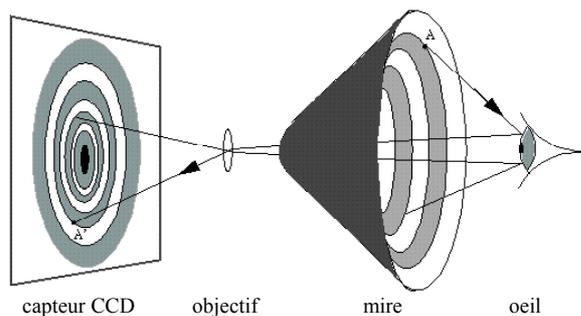


Figure 1 : Principe de vidéokératographie

Toutefois, on observe des effets parasites dans cette image qui vont compliquer notablement la reconnaissance de ces contours. Par exemple, l'ombre des paupières ou du nez (figure 2) occulte parfois largement la mire et, par la même, les informations caractérisant la cornée du patient dans ces zones.

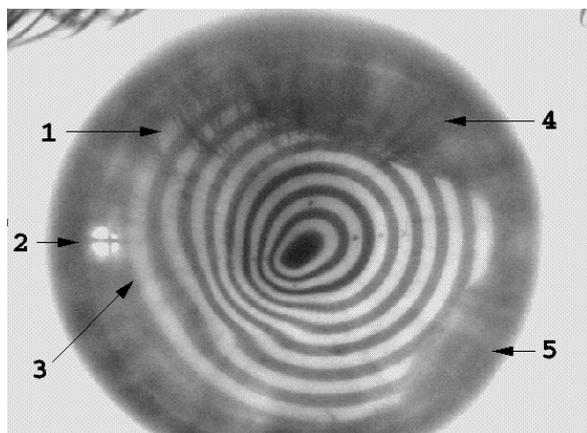


Figure 2 : L'image fournie par un vidéokératographe présente un certain nombre de phénomènes parasites (1 : ombre des cils; 2 : mire de télémétrie; 3 : contraste peu marqué; 4 : ombre des paupières; 5 : ombre du nez).

2.3 – Les contours de la mire dans l'image

Nous avons développé une technique d'analyse robuste permettant d'extraire les contours de la mire [2][3][4] (figure 3). Toutefois, la modélisation de ces contours reste empirique car elle est basée uniquement sur des contraintes géométriques 2D (suite de segments de B-splines cubiques) et ne tient compte ni des caractéristiques tridimensionnelles de la cornée, ni des propriétés mécaniques comme l'élasticité des tissus ou la pression intraoculaire.

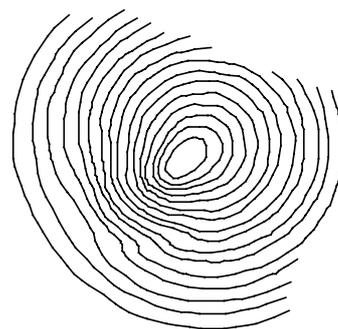


Figure 3 : Contours extraits de l'image 2D

3 – PRINCIPE DE RECONSTRUCTION 3D

Actuellement, les vidéokératographes sont calibrés au moyen de quatre billes de différents diamètres. La position des anneaux dans l'image permettra, par extrapolation linéaire, de déduire la courbure méridienne des cornées qui seront étudiées par la suite. La technique de reconstruction que nous employons est basée sur la simulation d'une vidéokératographie par lancer de rayons [5] (figure 4). Contrairement au système de calibrage par billes, cette approche permet de mieux prendre en compte les caractéristiques physiques et géométriques de la cornée. L'hypothèse de symétrie cylindrique (peu réaliste) n'est plus nécessaire à la reconstruction.

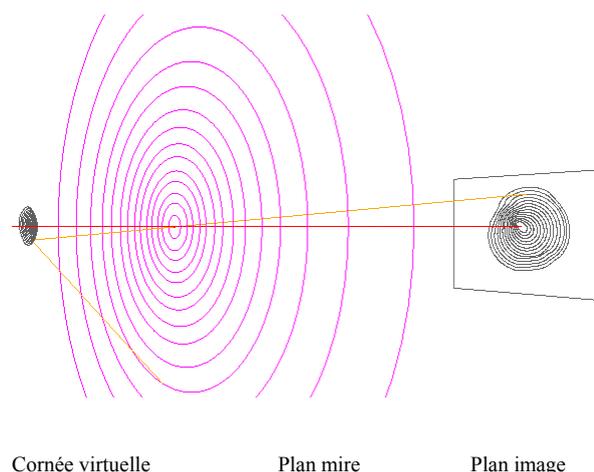


Figure 4 : Simulation d'une vidéokératographie par lancer de rayons (en jaune).

Le principe est le suivant :

- on définit une cornée virtuelle ;
- on calcule la projection de la mire sur le plan image après réflexion sur la cornée (loi de la normale) et passage au travers du diaphragme ;

- les contours obtenus par simulation sont comparés à ceux détectés dans l'image acquise et la cornée virtuelle est progressivement ajustée pour que l'image simulée converge vers l'image acquise.

On postule alors moyennant quelques précautions, que la cornée virtuelle obtenue correspond à la cornée du patient.

4 – MODELE DE CORNEE

Pour reconstruire la surface de la cornée à partir d'une image 2D, il faut disposer d'un **modèle a priori** qui permettra de compenser la perte d'information liée au principe d'acquisition (passage du 3D au 2D).

4.1 – Les hypothèses sur le modèle

Les hypothèses sont d'abord de natures géométriques (elles vont contraindre les propriétés de réflexion de la lumière) :

- la surface est supposée G^2 continue ;
- elle est convexe le long d'un méridien pour garantir que les images de deux contours distincts de la mire ne peuvent s'intersecter.

Elles pourront également être de nature mécanique pour compenser localement des manques d'information comme les ruptures de contours dues aux ombres du nez ou des paupières :

- élasticité et rigidité du tissu cornéen ;
- pression intraoculaire ;
- etc.

4.2 – Un modèle de cornée

Plusieurs modèles géométriques de surface peuvent être utilisés, mais nous en retiendrons un à partir des deux remarques suivantes :

- la forme de la cornée est proche d'une calotte sphérique;
- la mire effectue une discrétisation non-isotrope de la cornée : on ne dispose que de 32 points de mesure le long d'un plan méridien (intersections de ce plan avec les 8 anneaux), alors qu'un contour d'anneau représente de 100 à 1000 pixels suivant son diamètre.

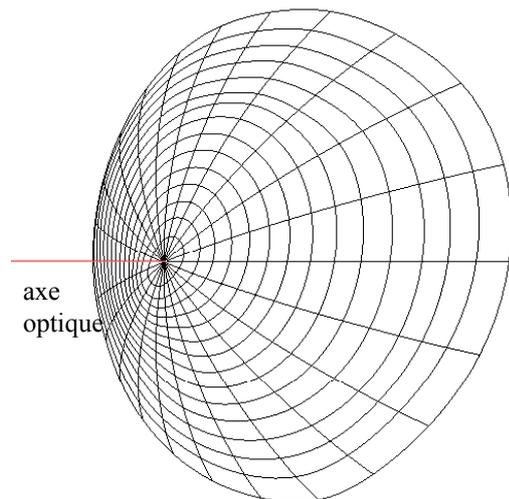


Figure 5 : Modélisation de la cornée avec des carreaux de B-splines bi-cubiques cylindriques.

Notre choix s'est porté sur un modèle de surface à base de carreaux de B-splines cylindriques bi-cubiques [6] [7] [8] (figures 3 et 4). L'axe optique représente la profondeur et porte le sommet de la surface, tandis que les points de contrôle sont positionnés dans 12 plans méridiens également répartis. Un point de la surface est donc localisé par des coordonnées (ρ, θ, z) où :

- ρ représente la distance de ce point à l'axe optique;
- θ représente l'angle du plan méridien contenant le point;
- z est la projection orthogonale de ce point sur l'axe optique.

L'axe ρ étant faiblement échantillonné comparé à l'axe θ , les points de contrôle sont répartis de sorte que les nœuds des carreaux se projettent sur les contours de l'image 2D. On compense ainsi le caractère anisotropique de la mire en faisant une interpolation de la surface de la cornée le long d'un même contour et une extrapolation de cette surface entre deux contours consécutifs.

Les nœuds de la surface sont initialement positionnés sur une sphère correspondant à une cornée « moyenne ».

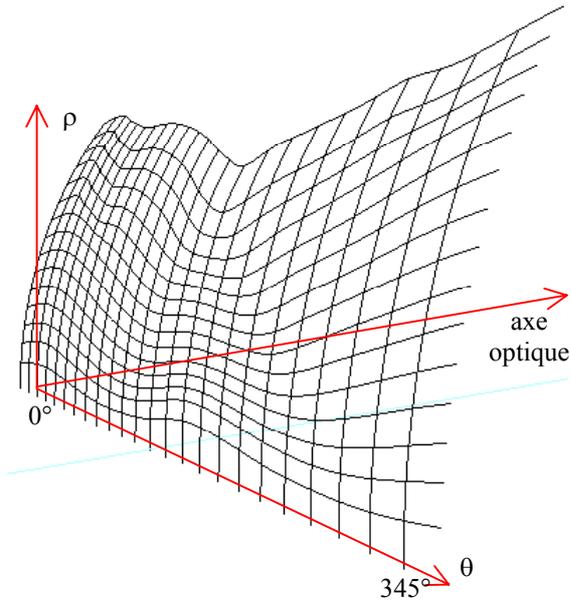


Figure 6 : Représentation en coordonnées cylindriques de la cornée et répartition des nœuds sur l'axe ρ .

4.3 – Quelques propriétés particulières du modèle

Le modèle de surface est défini par une matrice de 18x24 points de contrôle.

Les 24 derniers points de contrôle ($P_{17,j}$ pour j variant de 0 à 23) définissent le plan tangent sur la périphérie de la cornée. Ce sont eux qui vont contrôler l'angle de réflexion des rayons lumineux issus du dernier contour de la mire.

Les 24 premiers carreaux de surface sont de forme triangulaire dans leur représentation cartésienne :

- ils se rejoignent en un seul et même point O sur l'axe optique ;
- le plan tangent en O est identique quel que soit le carreau considéré.

Ces deux propriétés sont assurées par une organisation particulière des deux premières lignes de points de contrôles :

- $P_{-1,j} = P_{1, opp(j)}$ avec $opp(j) = (j+12)$ modulo 24
- $P_{0,j} = P_{0, j+12}$ pour $0 \leq j < 12$

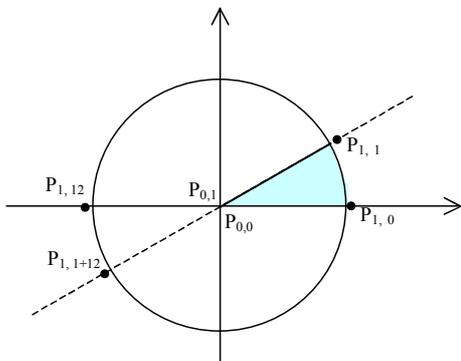


Figure 7 : Carreau de surface au centre de la cornée.

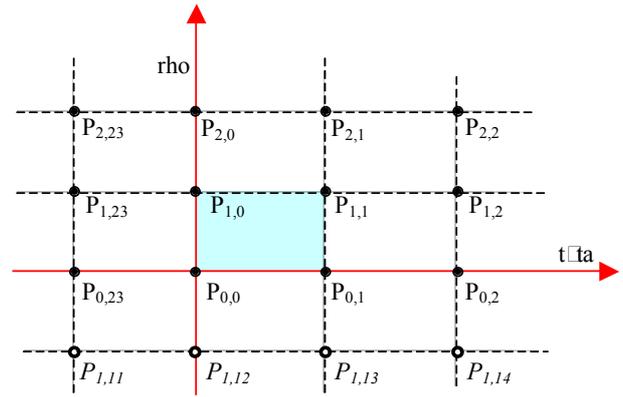


Figure 8 : Topologie des points de contrôle au centre de la cornée.

5 – PREMIERE PHASE D'AJUSTEMENT DU MODELE

5.1 – Lancer de rayon inverse

Soit un contour n° i dans l'image 2D et un point $C_{i,j}$ de ce contour. Il existe un unique rayon lumineux arrivant sur ce point $C_{i,j}$ après être passé par le diaphragme de l'appareil. En suivant la trajectoire inverse de ce rayon, il est aisé de calculer le point de réflexion $S_{i,j}$ sur notre modèle de cornée. L'évaluation de la normale en $S_{i,j}$ permet de déduire la direction du rayon incident, et donc, son point d'origine $N_{i,j}$ sur la surface de la mire.

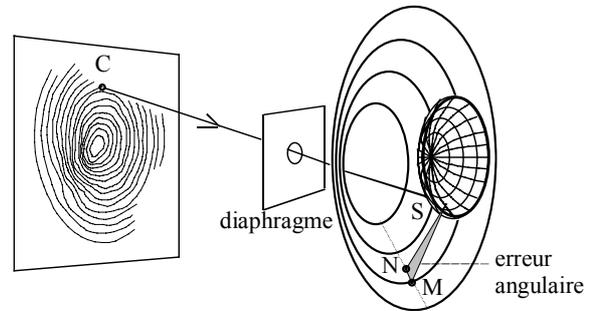


Figure 9 : Principe du lancer de rayon inverse.

Ce point $N_{i,j}$ se trouve dans un plan méridien qui intersecte le contour n° i de la mire en $M_{i,j}$. L'écart angulaire entre les segments $[S_{i,j}, N_{i,j}]$ et $[S_{i,j}, M_{i,j}]$ permet de caractériser l'erreur entre la cornée virtuelle et la cornée du patient.

L'intérêt de cette technique réside dans la faible complexité du processus résultant de la connaissance immédiate de la trajectoire du rayon réfléchi (ce qui n'est pas le cas du rayon incident).

5.2 – Ajustement du modèle

Pour un modèle de cornée donné, le lancer de rayons inverse permet de connaître pour chaque

point de contour $C_{i,j}$ dans l'image, l'erreur angulaire $a_{i,j}$ pour atteindre la mire sur le contour n° i . Il faudrait donc corriger l'incidence du plan tangent (plan de réflexion sur la cornée) de $a_{i,j}/2$ pour que le contour cible de la mire soit atteint.

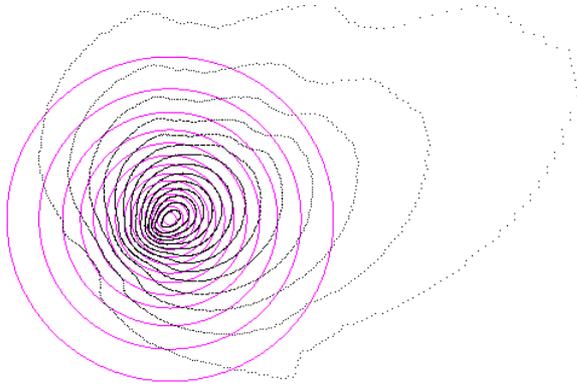


Figure 10 : Projection des points de contours (en noir) sur le plan de la mire (violet) avec le modèle initial de cornée.

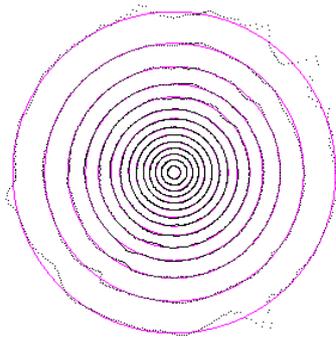


Figure 11 : Projection des points de contours (en noir) sur le plan de la mire (violet) après ajustement du modèle de cornée.

Le système étant surdéfini pour chaque point de contrôle de la surface, on minimise l'erreur quadratique (algorithme de Levenberg-Marquardt) pour connaître le plan tangent en chaque noeud et contraindre ainsi la forme de la cornée.

6 – AFFINER LA MESURE

Cette première phase d'ajustement assez rapide a permis d'obtenir un **modèle qualitatif** de la cornée du patient. Elle est qualifiée **d'analyse ascendante** car elle enchaîne, avec un degré d'interprétation croissant, une succession de traitements en partant des données pour aboutir à un premier modèle représentatif de la cornée du patient.

En particulier, la reconstruction s'est appuyée sur une étape intermédiaire d'extraction de contours dans l'image initiale 2D.

Le but était de :

- limiter la complexité des calculs en sélectionnant les informations principales ;
- obtenir un résultat qualitativement fiable (i.e. proche de la solution recherchée).

Toutefois, cette étape de segmentation a nécessairement introduit un « bruit » dans la mesure, notamment parce qu'elle est uniquement basée sur des techniques d'analyse d'image 2D qui sont complètement déconnectées des caractéristiques 3D de la cornée.

La deuxième étape d'ajustement est qualifiée d'**analyse descendante** : elle consiste à ajuster directement le modèle sur les données initiales, et ceci sans passer par les étapes intermédiaires. On disposera alors d'un modèle plus précis que l'on pourra qualifier de **modèle quantitatif**.

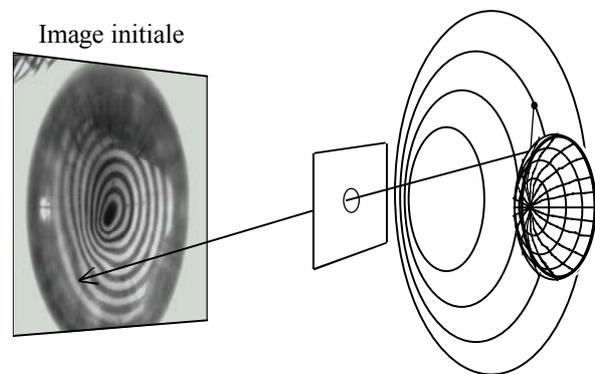


Figure 12 : Lancer de rayon direct

Le principe de ce deuxième ajustement s'appuie sur le lancer de rayon direct (figure 12). Pour un point $M_{i,j}$ de la mire, on cherche par dichotomie le point de réflexion $S_{i,j}$ sur la cornée de sorte que le rayon réfléchi passe par le diaphragme. On connaît alors le point d'impact $P_{i,j}$ du rayon dans l'image. La valeur du laplacien en $P_{i,j}$ (elle doit être nulle sur le contour d'un anneau) renseigne si le rayon est intérieur ou extérieur au contour cible ; la recherche du zéro du laplacien dans la direction normale au contour permet de connaître l'erreur angulaire du rayon réfléchi et d'agir sur le modèle pour diminuer cette erreur.

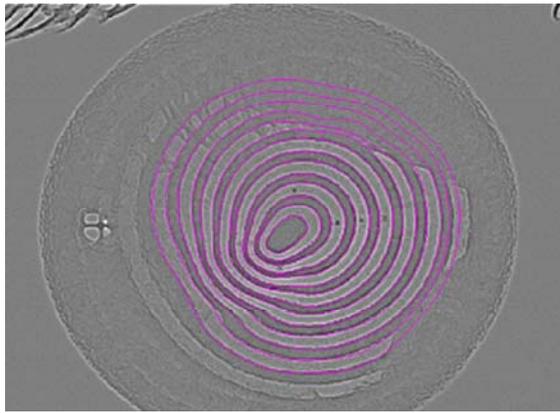


Figure 13 : *Projection de la mire dans le plan image et superposition sur le laplacien de l'image.*

Il est à noter que cette phase d'analyse descendante est réalisable uniquement parce que l'on dispose d'un premier modèle « proche » de la solution recherchée. En effet, les « attracteurs » dans l'image n'étant plus étiquetés, il est capital que la mire soit projetée à proximité de ces derniers pour que le modèle de cornée converge correctement.

7 – CONCLUSION

La méthodologie employée pour la reconstruction de la cornée peut s'appliquer à de nombreux problèmes d'analyse. La phase ascendante permet de proposer un nombre restreint de solutions approchées concurrentes tout en contrôlant la complexité de l'analyse. La phase descendante permet alors de faire un choix en recherchant dans les données initiales des informations plus fines qui distinguent le plus ces solutions. C'est généralement l'occasion d'affiner la solution retenue.

Dans le cas de la vidéokératographie, la phase ascendante a pour but de proposer une seule solution approchée. La phase descendante permet alors de mieux prendre en compte les caractéristiques tridimensionnelles de la cornée et d'éliminer les erreurs introduites par les traitements intermédiaires de la phase ascendante.

Nous étudions actuellement l'insertion d'un modèle physique réaliste de la cornée sur notre modèle géométrique. Les paramètres de ce modèle physique pourraient être adaptés en fonction du patient pour affiner la modélisation.

La qualité du modèle ne se mesure pas directement sur sa géométrie, mais sur ses propriétés optiques de réflexion (distance entre la projection de la mire virtuelle et l'image acquise). Toutefois la validation de notre procédé en situation réelle reste un problème car nous ne disposons pas de « fantômes » sur lesquels valider la mesure. Nous devrons probablement nous limiter à une validation à partir de modèles virtuels en simulant une vidéokératographie et en vérifiant si le processus de

reconstruction permet de retrouver la surface initialement choisie.

BIBLIOGRAPHIE

- [1] Schanzlin D., Robin B. (1992), *Corneal topography*, Springer Verlag.
- [2] Caprioglio M. (1998), *Reconnaissance de contours: application à la vidéokératographie*, thèse de l'université de la Méditerranée, Marseille.
- [3] Bulot R., Boi J-M., Sequeira J., Caprioglio M. (1996), *Contour segmentation using Hough transform*, ICIP 96, Lausanne, Suisse.
- [4] Boi J-M., Bulot R., Sequeira J. (1995), *A new methodology to analyze images from videokeratography*, IEEE-EMBS 95, Montréal, Canada.
- [5] Glassner A. (1989), *An introduction to ray tracing*, Academic Press.
- [6] Bartels R., Beatty J., Barsky B. (1988), *Mathématiques et CAO, B-splines*, volume 7, Edition Hermès.
- [7] Rockwood, Chambers (1996), *Interactive curves and surfaces*, Series in Computer Graphic and Geometric Modeling, Morgan Kauffmann.
- [8] Chu L., Barsky B., Klein S. (1999), *Cylindrical coordinate representation for modeling surfaces of the cornea, and contact lens*, International Conference on shape modeling and applications, Aizu-Wakamatsu, Japon.

SPEECH RECOGNITION : NEW TECHNIQUES FOR SPEAKER ADAPTATION

Olivier Bellot,

Doctorant en informatique

olivier.bellot@lia.univ-avignon.fr + 33 4 90 84 35 58

Driss Matrouf,

Maître de conférences en informatique

driss.matrouf@lia.univ-avignon.fr + 33 4 90 84 35 26

Adresse professionnelle

Laboratoire Informatique d'Avignon (LIA), 339, chemin des Meinajaries Agroparc BP 1228 84911
AVIGNON Cedex 9 FRANCE

Résumé : Les systèmes de reconnaissance de la parole utilisant des modèles acoustiques dépendants du locuteur sont plus performants que ceux basés sur des modèles indépendants du locuteur. Le but des techniques d'adaptation est d'améliorer ces derniers modèles pour s'approcher des performances obtenues avec un modèle dépendant du locuteur. Dans cet article, nous proposons deux nouvelles méthodes d'adaptation. La première utilisant les données de test et d'apprentissage pour adapter les modèles indépendants du locuteur, la seconde étant une technique d'adaptation basée sur une classification hiérarchique des gaussiennes composant le modèle acoustique. Ces stratégies d'adaptation ont été évaluées sur le corpus de test de l'AUPELF, ARC B1. Ces deux techniques permettent respectivement un gain relatif par rapport au système initial de 15% pour la première technique et de 16% pour la seconde.

Summary : The speaker-dependent HMM-based recognizers gives lower word error rates in comparison with the corresponding speaker-independent recognizers. The aim of speaker adaptation techniques is to enhance the speaker-independent acoustic models to bring their recognition accuracy as close as possible to the one obtained with speaker-dependent models. In this paper, we propose two new method: the first method operates in two steps using test and training data and the second is a hierarchical. These adaptations strategies were evaluated in a large vocabulary speech recognition task. The first method leads to a relative gain of 15 % with respect to the baseline system and 10 % with respect to the conventional MLLR adaptation, whereas the second leads to a relative gain of 16% with respect to the baseline system.

Mots clés : Communication Homme-Machine, Reconnaissance automatique de la parole, modèles acoustiques, adaptation au locuteur.

Key words : Human-computer communication, Automatic speech recognition, acoustic models, speaker adaptation.

Speech Recognition: New techniques for speaker adaptation

The speaker-dependent HMM-based recognizers have lower Word Error Rates (WER) than speaker-independent ones. In fact, modeling inter-speaker variability is usually performed by training acoustic models with as large as possible population of speakers. This training manner leads to a relative high variance in acoustic models and hence reduces discriminatory capabilities between different phonemes, especially in the context of larger perplexity tasks. Nevertheless, in the speaker-dependent case, the requirement of large amount of training data for each test speaker reduces the utility and portability of such systems.

The aim of speaker adaptation techniques is to enhance the speaker-independent acoustic models to bring their recognition accuracy as close as possible to the one obtained with speaker-dependent models.

In this paper, we will present two different approaches to increase the robustness of speech recognizer with respect to the speaker acoustic variabilities. The first one is a method using test and training data for acoustic model adaptation. This method operates in two steps: the first one performs an *a priori* adaptation using the transcribed training data of the closest training speakers to the test speaker. This adaptation is done with MAP procedure allowing reduced variances in the acoustic models. The second one performs an *a posteriori* adaptation using the MLLR procedure on the test data, allowing mapping of Gaussians means to match the test speaker's acoustic space. This adaptation strategy was evaluated in a large vocabulary speech recognition task. Our method leads to a relative gain of 15% with respect to the baseline system and 10% with respect to the conventional MLLR adaptation.

The second method presented in this paper is based on tree structure. Within the framework of speaker-adaptation, a technique based on tree structure and the maximum a posteriori criterion was proposed (SMAP)[15]. In SMAP, the parameters estimation, at each node in the tree is based on the assumption that the mismatch between the training and adaptation data is a Gaussian PDF which parameters

are estimated by using the Maximum Likelihood criterion. To avoid poor transformation parameters estimation accuracy due to an insufficiency of adaptation data in a node, we propose a new technique based on the maximum a posteriori approach and PDF Gaussians Merging. The basic idea behind this new technique is to estimate an affine transformations which bring the training acoustic models as close as possible to the test acoustic models rather than transformation maximizing the likelihood of the adaptation data. In this manner, even with very small amount of adaptation data, the parameters transformations are accurately estimated for means and variances. This adaptation strategy has shown a significant performance improvement in a large vocabulary speech recognition task, alone and combined with the MLLR adaptation.

1 – ADAPTATION USING TEST AND TRAINING DATA

1.1 – Introduction

To deal with inter-speaker variability, two classes of approaches have been studied. The first one consists in performing normalization in the feature space. This class contains the cepstral mean removal technique [1], the vocal track length normalization [2], a feature space normalization based on mixture density Hidden Model Markov (HMM) [3], and a signal bias removal estimated by Maximum Likelihood Estimation (MLE) [4].

The second class of approaches operates in acoustic model space. In this class a compact model for Speaker Adaptive Training (SAT) technique was introduced in [5]. This technique consists in modeling separately the speaker variation and removing its effect in the training data. Thus the variance of models is reduced and hence the overlap of the acoustic models. The most used techniques in the second class consist in adapting the speaker-independent models to a specific speaker so as to obtain a recognition accuracy as close as possible to the one obtained on speaker-dependent system. In this framework, many adaptation schemes have been proposed: in [6] Maximum *A Posteriori* (MAP) estimations

techniques were proposed. It attempts to obtain a Bayesian estimate of the model parameters using adaptation data available from the test speaker. In [7] the speaker-independent system is transformed to come closer to the test speaker by applying a linear transformation on the means of speaker-independent Gaussians. The transformation is estimated so as to maximize the likelihood of the test speaker's adaptation data.

Other adaptation schemes are based on the fact that the training data contains a number of training speakers, some of whom are acoustically closer, to the test speaker, than the others [8]. This technique uses the adaptation data to find a subset of the training speakers which are closer to the test speaker. And then, it compute and apply a linear transformation to map the acoustic space of each selected training speaker closer to the test speaker's acoustic space. The linear transformation is computed by using the MLLR procedure [9].

In this paper, we propose a method using training and test-data for acoustic model adaptation. There are two steps in this method: the first one performs an *a priori* adaptation using transcribed training data with MAP adaptation. The second one performs an *a posteriori* adaptation using test data with MLLR adaptation. Both modifications have different goals: the former allows a reduced variance in acoustic models whereas the latter allows a mapping of the acoustic models means to be closer to the test speaker's acoustic space.

In the next section we present the proposed adaptation method. We describe the goals and the strategies for the *a priori* and the *a posteriori* speaker adaptations. In section 1.3 we describe two strategies for training-speakers selection. Section 1.4 shows results for several recognition experiments in large vocabulary task framework.

1.2 – Adaptation process

Because of the inter-speaker variability modeling, the speaker-independent models have a relative large variance in comparison with the corresponding speaker-dependent models. By using the MLLR adaptation we only adapt the Gaussians means, so the resulting acoustic models still have a relative high variance and hence an high overlap among different speech units, resulting in

reduced discriminatory capabilities. To reduce variances, one way is to use the MAP (Maximum *A Posteriori*) adaptation [6]. But this process requires a relative large amount of adaptation data to re-estimate all Gaussians variances.

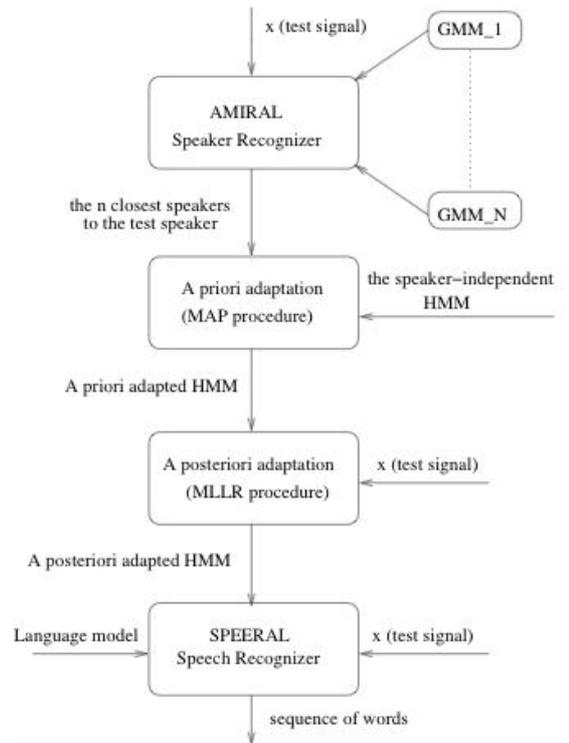


Figure 1: Adaptation process

In this paper, we propose a strategy resulting in adapted acoustic models with reduced variances. The adaptation is performed in two steps (see Figure 1). The first adaptation step, that we term the *a priori adaptation* is based on selecting a cluster of training speakers who “are” acoustically close to the test speaker. Then the speaker-independent acoustic models are adapted by using the transcribed training data corresponding to those selected speakers. This first adaptation is done with the MAP procedure, which transforms the means, variances and gains of Gaussians: let g a Gaussian having μ_g as mean and Σ_g as variance in the speaker-independent acoustic models. The new mean $\tilde{\mu}_g$ and variance $\tilde{\Sigma}_g$ of the Gaussian are given by:

$$\tilde{\mu}_g = \frac{\eta_g + \tau_g \mu_g}{C_g + \tau_g}$$

$$\Sigma_g = \frac{1}{c_g + \tau_g} + [\gamma_g + \tau_g[\Sigma_g + \mu_g \mu_g^{tr}]] - \mu_g \mu_g^{tr}$$

where :

$$c_g = \sum_t c_g(t)$$

$$\eta_g = \sum_t c_g(t) x_t$$

$$\gamma_g = \sum_t c_g(t) x_t x_t^{tr}$$

The parameter τ_g is usually chosen to be constant. $c_g(t)$ is the *a posteriori* probability of the Gaussian g at time t conditioned on all acoustic observations $x_t = 1 \dots T$.

This first processing step adapts all Gaussians parameters, however, only variances and gains Gaussians adaptations can effectively improve the modeling capabilities of the system for a specific test speaker. In fact, really adapting the Gaussians means compared to a specific speaker, only relying on training speakers requires a very large population of speakers. The selected training speakers cluster is then accordingly closer to the test speaker. By using a not so large training speakers population (120 speakers), the spectral variation caused by the inter-speaker variability in each speech unit is reduced, but the Gaussians means remains unadapted to the test speaker.

The second processing step consists in adapting the Gaussians means of the acoustic models resulting from the first adaptation step. This last adaptation is done by using the MLLR procedure [9]: the test data is decoded by using the reduced variances acoustic models (the *a priori* adapted models). Then the resulting frame/state alignment is used to estimate a global linear transformation, which is applied to the Gaussians means of the *a priori* adapted acoustic models. We term this second step adaptation the *a posteriori* adaptation.

1.3 – Speaker clustering

To perform the *a priori* adaptation we need to find a subset of the training speakers who are the closest to the test speaker. This is done with the LIA speaker recognizer, AMIRAL [10], based on Gaussian Mixture Models (GMM). For this task we used a GMM with 128 Gaussians for each training speaker. The system compares all the training speakers to the test speaker, and then order these speakers

from nearest to farthest. So, the transcribed training data of the nearest speakers are selected to adapt the speaker-independent acoustic models with the MAP procedure.

Another strategy to select the training speakers based on HMM instead of GMM was tested. Firstly, we constructed 120 training-speaker dependent HMMs. However, the data available from each training speaker are usually not sufficient to obtain robust estimations of the speaker dependent model parameters. So, we used MAP procedure [6] to adapt the speaker-independent models to each training speaker, and hence obtain 120 HMMs representing each of the training speakers. It was then required to find the subset of the closest training speakers to the test speaker. The test data are decoded using a speaker-independent system leading to frame/state alignment. Then the acoustic likelihood of test data, conditioned on this alignment, is computed using each of training speaker-dependent HMM. The top speakers are then selected as the acoustically closest training speakers to the test speaker.

The two strategies give always the same five first speakers. The experiments in this paper is performed using the GMM based strategy.

1.4 – Experimental results

In this section, we present the results of several recognition experiments. These experiments were conducted using SPEERAL [11], a large vocabulary speech recognition system, developed at the LIA. The lexicon size is about 20k words with 3.6% out-of-vocabulary words. This system uses a trigram language model. The baseline system is speaker and gender independent. The acoustic model contains 38 phonemes. Each phoneme is 3-state left-to-right context-independent CDHMM (Continuous Density HMM). Each state is a mixture of 64 Gaussians. The signal speech is parameterized using 13 coefficients, 12 mel-warped cepstral coefficients plus energy. The first and second order derivatives parameters are also used.

To estimate the acoustic and linguistic models, we have used a training data extracted from Bref [13], with 120 speakers (66 females and 54 males). The training data contain 66.5k sentences. The test data were provided for ARC B1 of AUPELF [12], with 20 speakers and 299 sentences. The sentences are articles published in the french newspaper “Le

Monde”. The *a priori* adaptation is performed by using the 5 closest training speakers to the test speaker. This adaptation is done with the MAP procedure. In the Table 1, we term this adaptation *Adapt. 1*. The *a posteriori* adaptation is performed by using the test signal with the MLLR procedure. In the Table 1, we term this adaptation *Adapt. 2*. Both adaptations will be compared to the MLLR-only one, applied on the test data. We term this last adaptation *Adapt. 3*. For MLLR, we used 1 global linear transformation with an *offset*.

	Baseline	Adapt. 1	Adapt. 2	Adapt. 3
WER	26.2	25.4	22.4	24.9

Table 1: Word Error Rate (%) Comparisons: *Adapt. 1: a priori adaptation with MAP, Adapt. 2: a posteriori adaptation with MLLR on models obtained in Adapt. 1, Adapt. 3: MLLR on the baseline acoustic models.*

We can see that the *a priori* adaptation using MAP (adapt. 1 in Table 1) doesn’t improve significantly the word error rate (only 3% relative gain with respect to the baseline system). However, this step is important because its conjunction with the MLLR procedure leads to a relative gain of 15% with respect to the baseline system (compare adapt. 2 and adapt. 1). This fact is due to a smaller variance of acoustic models in *a priori* adaptation (MAP). Then the MLLR better maps the means of acoustic Gaussians model to match the test data.

The word error rates (WER) by speaker shows that there is a relative large variation of recognition improvements with respect to test speakers. For example, the WER for one speaker was 41.1% without adaptation, 40.5% after the *a priori* adaptation and finally 27.9% after the *a posteriori* adaptation (1.5% relative gain by *a priori* adaptation compared to 32% by *a priori* and *a posteriori* adaptation). For the complete test, the average relative gain is about 15%. And, if some speakers were not improved, the recognition for these speakers is not degraded. This gain variation between test speakers can be explained by the fact that some test speakers miss significantly close training speakers, with respect to the whole training

speakers set. This problem should be solved by using a larger population of speakers.

In our experiments, the relative gain obtained by using the MLLR (1 global transformation) with respect to the baseline system is about 5% (from 26.2% to 24.9%). This gain is 3 times smaller than the one obtained by conjunction of the *a priori* and *a posteriori* adaptations. The relative gain obtained by the *a priori* and *a posteriori* adaptations with respect to the conventional MLLR is about 10%.

2 – STRUCTURAL ADAPTATION USING MAP AND GAUSSIANS MERGING TECHNIQUE ADAPTATION

In this section, the SMAPGM (Structural Adaptation using MAP and Gaussians Merging technique) will be presented. This technique uses a classification tree and a new adaptation method.

2.1 – Introduction

Due to complex inter-speaker variabilities, the performance of speaker-independent (SI) large vocabulary continuous speech recognition systems still lags behind that of speaker-dependent (SD) systems. Speaker-independent systems are typically constructed using speech samples collected from an as large as possible population of speakers [14].

Nevertheless, in the speaker-dependent case, the large amount of required training data for each test speaker reduces the utility and portability of such systems.

The main difficulty in speaker adaptation techniques is to adapt a large number of parameters with only a relative small amount of data. The MAP adaptation approach allows accurate estimation of HMM parameters for which enough adaptation data is available [6], and the unseen parameters are still unchanged. In this manner, the MAP approach leads to too much local adaptation. Hence the MAP approach can't be effective with relative small amount of adaptation data especially in unsupervised mode.

In order to reduce this problem, Shinoda and Lee proposed a structural maximum a posteriori (SMAP) approach [15], in which a hierarchical structure (tree) in the parameter space is assumed. The parameters transformation for each node in the tree are

estimated by using the MAP approach in which the a priori parameters are given by the parent node. The resulting transformation parameter, corresponding to each HMM parameter, is a combination of the transformation parameters at all higher levels. The weights in this combination depend on the amount of adaptation data at each node and on a fixed parameter.

In SMAP, the parameters estimation at each node in the tree is based on the assumption that the mismatch between the training and adaptation data is a Gaussian PDF.

The mean and the variance of this Gaussian mismatch PDF are estimated directly from the adaptation data by using the Maximum Likelihood criterion. In this manner, the estimation accuracy of the transformation parameters depends on the amount of the adaptation data. To avoid poor transformation parameters estimation accuracy due to an insufficiency of adaptation data we propose a new technique based on maximum a posteriori approach [6] and PDF Gaussian Merging. The basic idea behind this new technique is to estimate transformations which make the training acoustic models as close as possible to the test acoustic models rather than transformation maximizing the likelihood of the adaptation data. The test acoustic models are estimated using the MAP approach [6]. In this manner, even with very small amount of adaptation data, the parameters transformations are accurately estimated.

In this paper, like in SMAP [15], we assume that the models parameters are organized in tree containing all the Gaussian distributions. Each node in that tree represents a cluster of Gaussians. All the Gaussian distributions of a given cluster/node share a simple common affine transformation (diagonal matrix plus *offset*) compensating the mismatch between training and test conditions.

To estimate this affine transformation, we propose a new technique based on a Gaussian distributions merging and the standard MAP adaptation. This new technique is very fast and allows a good adaptation for both means and variances even with small amount of adaptation data in unsupervised mode. At each node, the transformation is obtained by combining three kinds of information: the adaptation data, the parameters transformation

at the parent node and the parent node adapted parameters.

Section 2.2 presents the whole adaptation process proposed in this work: the adaptation process in a given node in the tree, the combination of the mismatch information at different tree layers, the merging procedure, and the tree construction. Section 2.3 shows results for several recognition experiments in a large vocabulary task framework.

2.2 – Adaptation process

Gaussian distributions. The first step in the adaptation process is to build a classification tree structure representing the set of Gaussian distributions. Each node in the tree represents a subset of Gaussians and the root node represents the whole set. Let ν denote one node in the classification tree, and $G_\nu = \{g_{m_\nu}, m_\nu = 1 \dots M_\nu\}$ be the subset of Gaussian distributions associated to the node ν : $g_{m_\nu} = N(\mu_{m_\nu}, \Sigma_{m_\nu})$. In the following paragraphs, we describe the adaptation process for a node ν and show the strategy for combining information at different layers.

2.3 – Adaptation Process in a node

The goal of this work is to estimate for each node ν an affine transformation T_ν (diagonal matrix plus *offset*) shared by all Gaussian distributions in the subset G_ν . This affine transformation is then applied to only the distributions belonging to G_ν . Let $X = \{x_1, \dots, x_t\}$ denote a given set of T observation vectors for parameters adaptation.

Let $\tilde{g}_{m_\nu} = N(\tilde{\mu}_{m_\nu}, \tilde{\Sigma}_{m_\nu})$ be the Gaussian obtained by adapting the Gaussian $g_{m_\nu} = N(\mu_{m_\nu}, \Sigma_{m_\nu})$ using the standard MAP adaptation (see Formulas).

Let \tilde{G}_ν be the subset of MAP adapted Gaussians in the node ν :

$\tilde{G}_\nu = \{\tilde{g}_{m_\nu}, m_\nu = 1 \dots M_\nu\}$. Let $\tilde{g}_\nu = N(\tilde{\mu}_\nu, \tilde{\Sigma}_\nu)$

and $g_\nu = N(\mu_\nu, \Sigma_\nu)$ be the two Gaussians obtained by merging into one all Gaussians in

G_ν and \tilde{G}_ν respectively (see section 2.2). The affine transformation T_ν is then estimated as the one which matches the Gaussian g_ν to the

Gaussian \tilde{g}_ν . Each Gaussian $g_{m_\nu} = N(\mu_{m_\nu}, \Sigma_{m_\nu})$ is then adapted as follows :

$$\mu'_{m_v} = \frac{1}{\tilde{\Sigma}_v} \frac{1}{\tilde{\Sigma}_v} (\mu_{m_v} - \mu_v) + \tilde{\mu}_v$$

$$\Sigma'_{m_v} = \tilde{\Sigma}_v \tilde{\Sigma}_v^{-1} \Sigma_{m_v}$$

Where μ'_{m_v} and Σ'_{m_v} are the adapted parameters of μ_{m_v} and Σ_{m_v} respectively. This adaptation procedure can be performed iteratively. We have shown experimentally that the likelihood of adaptation data increases at each iteration.

2.4 – Merging Process

The merging process is based on the merging of pairs of Gaussian distributions until we obtain a single Gaussian. In this work the merging of two Gaussians uses the minimum loss likelihood criterion. Let $G = \{g_1, \dots, g_n\}$ denote a set of Gaussians to be merged into one representing the set G . Firstly, we choose two Gaussians $g_i = N(\mu_i, \Sigma_i)$ and $g_j = N(\mu_j, \Sigma_j)$ in G . Let c_i and c_j denote their associated counts. The Gaussian $g = N(\mu, \Sigma)$ obtained by merging g_i and g_j is given by the classic formula:

$$\mu = \frac{c_i \mu_i + c_j \mu_j}{c_i + c_j}$$

$$\Sigma = \frac{c_i \Sigma_i + c_j \Sigma_j + \frac{c_i \times c_j}{c_i + c_j} (\mu_i - \mu_j)(\mu_i - \mu_j)^{tr}}{c_i + c_j}$$

The count c associated with the new Gaussian g is the sum of the two counts c_i and c_j associated with the two Gaussians g_i and g_j . The two Gaussians g_i and g_j in G are then replaced by the Gaussian g . We repeat this merging procedure until we obtain one Gaussian representing the set G . The initial count c_{m_v} associated to a Gaussian g_{m_v} is the sum over all observation vectors of the a posteriori probabilities: $c_m = \sum_t \gamma_{m_v t}$.

2.5 – Adaptation Using Hierarchical Priors

In section 2.1. we have treated the problem of estimating an affine transformation T_v associated to the node v . The estimation of T_v was based only on the Gaussians belonging to this node and their associated observation vectors. To estimate the transformation T_v by using all Gaussians in the CDHMM and their associated observation vectors we use the adaptation with hierarchical priors.

Let $p(v)$ denote the parent node of v . Let g_v and $g_{p(v)}$ be the two Gaussians obtained by merging into one all the Gaussians in G_v and $G_{p(v)}$ respectively (the original Gaussians in the node v and $p(v)$). In the manner, let \tilde{g}_v and $\tilde{g}_{p(v)}$ denote the Gaussians obtained by merging into one all the Gaussians in \tilde{G}_v and $\tilde{G}_{p(v)}$ respectively (the MAP adapted Gaussians in the node v and $p(v)$) (see section 2.1).

On one hand we merge the Gaussians g_v and $g_{p(v)}$ to obtain one Gaussian $g_v^{p(v)} = N(\mu_v^{p(v)}, \Sigma_v^{p(v)})$ and on the other hand we merge the Gaussians \tilde{g}_v and $\tilde{g}_{p(v)}$ to obtain

one Gaussian $\tilde{g}_v^{p(v)} = N(\mu_v^{p(v)}, \Sigma_v^{p(v)})$. In this merging process the count associated to the Gaussians in the parent node $p(v)$ is a fixed parameter, and the count associated to the Gaussians in the node v is the sum of the counts associated to all Gaussians in that node ($\Sigma_m c_{m_v} = \Sigma_m \sum_t \gamma_{m_v t}$). The affine transformation T_v is then estimated as the one which matches the Gaussian $g_v^{p(v)}$ to the Gaussian $\tilde{g}_v^{p(v)}$. Each Gaussian $g_{m_v} = N(\mu_{m_v}, \Sigma_{m_v})$ is then adapted as follows :

$$\mu'_{m_v} = (\Sigma_v^{p(v)})^{1/2} (\Sigma_v^{p(v)})^{-1/2} (\mu_{m_v} - \mu_v^{p(v)}) + \mu_v^{p(v)}$$

$$\Sigma'_{m_v} = (\Sigma_v^{p(v)}) (\Sigma_v^{p(v)})^{-1} \Sigma_{m_v}$$

Where μ'_{m_v} and Σ'_{m_v} are the adapted parameters of μ_{m_v} and Σ_{m_v} respectively. These adaptation formula are then used instead of equations 1 and 2. In this manner the resulting transformation parameter, corresponding to each parameter, is a combination of mismatch information at all levels. In this combination the weight for each level changes autonomously according to the amount of adaptation data.

2.6 – Construction of the tree structure

The use of the tree structure has been largely studied in the contextual acoustic units estimation framework [16].

In this work we have used a binary tree. We assumed that all Gaussians in a state of the CDHMM belong to the same class and the tree leaves represent the CDHMM states. Each node in the tree is a collection of states which are collections of Gaussians. For classification, each state is represented by one Gaussian obtained by merging all Gaussians in that state. Hence, we construct a state classification tree using the loss likelihood minimization criterion for clustering. We used the *up to down* strategy as classification tree algorithm. Our classification tree algorithm is not optimal because, at each node with n states, we don't explore the 2^{n-1} two-cluster splits possible. Instead, we use an iterative procedure like k-means clustering with two centers.

2.7 – Experimental results

In this section, we present the results of several speech recognition experiments. These experiments were conducted using the same experimental set than in the previous section, expect that now the baseline system is gender-dependent with 3-states left-to-right context-dependent unit acoustic models.

In these experiments, we used two binary trees with six layers: one for the male acoustic models and the other for the female acoustic models. These classification trees are built once before the adaptation process. In the experiments, both mean vectors and covariances were adapted. All adaptation procedures were performed speaker per speaker in unsupervised mode.

We will call the proposed technique SMAPGM (Structural Adaptation using MAP and Gaussians Merging technique).

In Table 1 we can see that the SMAPGM technique gives an average relative gain about 16% with respect to the baseline system. It should be noted that part of the improvements of MLLR and SMAPGM can be cumulated.

In fact, by performing SMAPGM after MLLR the relative cumulated gain is about 18% with respect to the baseline system and by performing MLLR after SMAPGM the relative cumulated gain is about 19.5%. In these experiments, we have noted that the effect of the proposed method is more significant for speakers with higher word error rates.

	Male	Female	Average
Base	21.2	21.0	21.1
SMAPGM	18.0	17.7	17.8
SMAPGM+MLLR	16.6	17.4	17.0
MLLR+SMAPGM	17.1	17.5	17.3

Table 1: *Word Error Rate (%) for gender-dependent speech recognizer with different speaker adaptation techniques. SMAPGM designates the proposed technique: structural adaptation using MAP and Gaussians Merging technique.*

We have performed the same experiments with a better lexicon and language model. The baseline word error rate becomes 19%. After SMAPGM adaptation, the word error rate was 16.3% (a relative gain of 14% with respect to the baseline system, instead of 16% with the first system). When SMAPGM is performed after MLLR, the word error rate comes down to 15.9% (a relative gain of 16% with respect of baseline system, instead of 19.5% with the first system). The relative gain obtained by using SMAPGM seems to be larger for the baseline system with higher word error rate. In order to compare SMAPGM with SMAP [8], we realized experiments under the same conditions (with the same tree with six layers). The SMAP adaptation leads to a word error rate of 17.3% (a relative gain of 9%, instead of 14% for SMAPGM adaptation, see Table 2).

	Average	Relative gain
Baseline	19.0	
SMAPGM	16.3	14.2
SMAP	17.3	8.9

Table 2: *Word Error Rate (%) and relative gain in regard of baseline system for gender-dependent speech recognizer with SMAP and SMAPGM adaptations*

3 – CONCLUSION

In this paper, we have presented two new method for speaker adaptation. Their effectiveness was confirmed by experiments in a large vocabulary speech recognition task: a relative gain of 15% with regard to the baseline system was obtained in the case of the first technique and a relative gain of 16% in the case of the second technique.

BIBLIOGRAPHIE

- [1] T. Anastaskos, F. Kubala, J. Makhoul and R. Schwartz, "Adaptation to new microphones using tied-mixture normalization", in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp. 433-436, 1994.
- [2] H. Eide and H. Gish, "A parametric approach to vocal tract length normalization", in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp. 346-349, 1996.
- [3] Y. Zhao, "An Acoustic-phonetic-based Speaker Adaptation Technique Improving Speaker independent Continuous Speech Recognition", *IEEE Trans. Speech and Audio Processing*, vol. 2, no. 3, pp. 380-394, July 1994.
- [4] M. Rahim and B-H. Juang, "Signal bias removal by maximum likelihood estimation for robust telephone speech recognition", in *IEEE Trans. Speech and Audio Processing*, vol. 4, no. 1, pp. 19-30, January 1996.
- [5] T. Anastaskos, J. McDonough, R. Schwartz and J. Makhoul, "A Compact Model for Speaker- Adaptive Training", *Proc. ICSLP'96*, pp. 1137- 1140, Philadelphia, 1996.
- [6] J.-L. Gauvain and C.-H. Lee, « Maximum A Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains », in *IEEE Trans. on Speech and Audio Processing*, 2(2):291-298, April 1994.
- [7] J. R. Bellegarda, P. V. de Souza, A. Nadas, D. Nahamoo, M. A. Picheny and L. R. Bahl, "The Metamorphic Algorithm: A Speaker Mapping Approach to Data Augmentation", *IEEE Trans. Speech and Audio Processing*, vol. 2, no. 3, pp. 413-420, July 1994.
- [8] M. Padmanabhan, L. R. Bahl, D. Nahamoo and M.A. Picheny, "Speaker Clustering and Transformation for Speaker Adaptation in Speech Recognition Systems", in *IEEE Transactions on Speech and Signal Processing*, vol. 6, no. 1, pp. 71-77, January 1998.
- [9] C.J. Leggetter and P.C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models", in *Computer Speech and Language*, pp. 171-185, 1995.
- [10] C. Fredouille, J.-F. Bonastre and T. Merlin, "AMIRAL: A Block-Segmental Multirecognizer Architecture for Automatic Speaker Recognition", in *Digital Signal Processing*, 2000.
- [11] P. Nocera, G. Linares, D. Massonié, L. Lefort, « Phoneme lattice based A* search algorithm for speech recognition », Sept. 2002, Brno, TSD2002
- [12] J. Dolmazon, F. Bimbot, G. Adda, J. Caerou, J. Zeiliger, M. Adda-Decker, "Première campagne AUPELF d'évaluation des systèmes de Dictée Vocale", *Ressources et 'évaluation en ingénierie des langues*, pp. 279-307, 2000
- [13] L. F. Lamel et al., "BREF, a Large Vocabulary Spoken Corpus for French", in *EuroSpeech '91*, Genoa, Sept. 1991.
- [14] D. Matrouf, O. Bellot, P. Nocera, G. Linares, J.F. Bonastre, « A Priori and a posteriori Transformations for Speaker Adaptation in Large Vocabulary Speech Recognition Systems », *7th European Conference on Speech Communication and Technology*, Vol. II, p. 1245-1248, Aalborg DENMARK, Sept. 2001.
- [15] K. Shinoda and C.-H. Lee, « Unsupervised adaptation using structural Bayes approach », *In Proc IEEE ICASSP*, Seattle, Washington, USA, 1998.
- [16] R. Singh, B. Raj and R. M. Stern, « Automatic Clustering and Generation of Contextual Questions for Tied States in Hidden Markov Models », *In Proc IEEE ICASSP*, Phoenix, Arizona, USA, 1999.

GXCAST : UNE GENERALISATION DU PROTOCOLE XCAST

Ali Boudani,

Doctorant en informatique
ali.boudani@irisa.fr , + 33 2 99 84 25 37

Alexandre Guitton,

Doctorant en informatique
alexandre.guitton@irisa.fr , + 33 2 99 84 71 30

Bernard Cousin,

Professeur en informatique
bernard.cousin@irisa.fr , +33 2 99 84 73 33

Adresse professionnelle

Université de Rennes I ★ Campus de Beaulieu ★ 35 042 Rennes Cedex

Résumé : Dans ce papier, nous étudions une généralisation du protocole Xcast. Après avoir introduit les deux protocoles Xcast et Xcast+, leurs avantages et leurs inconvénients, nous présentons notre protocole GXcast : le format d'un paquet GXcast et l'algorithme de routage. La résistance au facteur d'échelle du protocole GXcast est comparée à celle d'autres protocoles *multicast*. Nous proposons notamment une fonction permettant de réduire les inconvénients de ce type de protocoles et évaluons le protocole GXcast en termes de surcoût et de délai. Les résultats de simulation confirment notre analyse.

Summary : In this paper, we study a generalization of the Xcast protocol. After having introduced both Xcast and Xcast+ protocols, their advantages and drawbacks, we present our protocol Gxcast: the GXcast packet format and the forwarding algorithm. The scalability of the GXcast protocol is compared to the scalability of other multicast protocols. We propose a function that allow to reduce the drawbacks of this kind of protocols and we evaluate the GXcast protocol in terms of overhead and delay. The simulation results confirm our analysis.

Mots clés : Routage explicite, analyse de performance, protocole *multicast*.

GXcast : une généralisation du protocole Xcast

1 – INTRODUCTION

Un unique protocole de routage *multicast* est incapable de servir les différents types d'applications *multicast*, comme l'indique la charte du groupe de travail *Reliable Multicast Transport* de l'IETF (2003). En effet, le nombre d'applications *multicast* ayant souvent des exigences multiples voire contradictoires ne cesse de croître. Les protocoles de routage *multicast* doivent présenter une certaine flexibilité selon les besoins des différentes applications. Dans un réseau où il y a un très grand nombre de groupes *multicast* de petite taille (technique SGM : *Small Group Multicast*, présentée par Ooms (2000)) dont les destinataires sont largement dispersés, le modèle de *multicast* traditionnel ne convient pas.

1.1 - Le protocole Xcast

Explicit Multicast (Xcast) a été proposé par Boivie *et al* (2000) pour résoudre le problème de passage à l'échelle des protocoles *multicast* et pour servir des groupes ayant peu de membres tout en minimisant la consommation de la bande passante. La source encode explicitement la liste des destinations dans l'en-tête Xcast d'un paquet au lieu d'utiliser une adresse *multicast* et envoie le paquet vers son routeur désigné (c'est-à-dire le routeur Xcast en charge de cette source). Chaque routeur du chemin analyse l'en-tête Xcast, classe les destinations selon leur prochain routeur au sens *unicast* et envoie une copie¹ vers chacun des prochains routeurs. Dans le cas où il reste une seule destination dans la liste, le paquet Xcast est transformé par l'algorithme X2U (Xcast à *unicast*) en un paquet *unicast*.

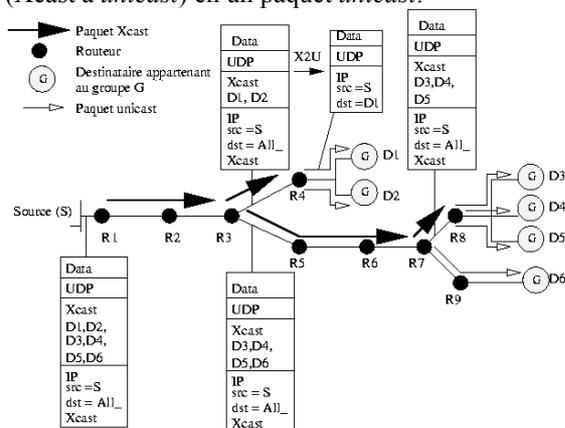


Figure 1 - La transmission d'un paquet Xcast.

Exemple : considérons le groupe représenté sur la figure 1, comportant une source *S* et six

destinataires *D1, D2, D3, D4, D5* et *D6*. La source *S* envoie un paquet Xcast contenant la liste des destinations (*D1, D2, D3, D4, D5, D6*) à *R1*. *R1* traite ce paquet comme un paquet Xcast quelconque : *R2* est le prochain routeur sur chacun des chemins *unicast* de *R1* à *Di*, donc le paquet Xcast entier est transmis à *R2*. *R2* envoie à son tour le paquet à *R3*. Lorsque *R3* reçoit le paquet, il en envoie une copie au routeur *R4* avec dans l'en-tête Xcast la liste (*D1, D2*) et une copie au routeur *R5* avec dans l'en-tête Xcast la liste (*D3, D4, D5, D6*). *R4*, recevant le paquet qui lui est destiné, enverra à son tour à *D1* le message Xcast contenant la liste (*D1*) et à *D2* le message Xcast contenant la liste (*D2*). *D1* pourra extraire du message qu'il recevra les données qui lui sont utiles. Le comportement est similaire pour les routeurs *R5* à *R9* et pour les cinq autres destinataires.

1.2 – Le protocole Xcast+

Xcast+ a été proposé par Myung-Ki *et al* (2001) pour résoudre le problème de passage à l'échelle de Xcast pour les groupes de taille moyenne. Il est basé sur le protocole Xcast (il utilise d'ailleurs un en-tête très similaire) et permet de réduire la liste des destinataires en utilisant habilement le protocole de gestion locale des groupes d'Internet (protocole IGMP décrit dans la RFC 3376 (2002)). En effet, un récepteur désirant faire partie du groupe (*S, G*) (un groupe, appelé session *multicast* dans la terminologie Xcast, est identifié par le canal (*S, G*) où *S* est l'adresse de la source et *G* l'adresse du groupe) émet un message *join* IGMP à destination du groupe (*S, G*). Quand le routeur désigné reçoit ce message, il envoie à la source *S* un message de demande d'enregistrement Xcast+ contenant l'adresse de la source *S*, l'adresse de groupe *G*, et sa propre adresse *DR*. Lorsque le *DR* associé à la source reçoit ce message, il maintient les adresses de tous les routeurs *DR* ayant des récepteurs appartenant au groupe *multicast* (*S, G*).

Lorsque la source envoie les paquets *multicast*, le *DR* de la source crée un paquet Xcast+ dans lequel il encode explicitement la liste des *DR* associés aux destinations dans l'en-tête Xcast+, il complète le paquet avec les données à envoyer et émet ce paquet vers le(s) prochain(s) routeur(s) concerné(s) (*M2X* : *multicast* à Xcast+). Le chemin suivi par le paquet Xcast+ est le même que celui suivi par le paquet Xcast. La seconde différence a lieu aux *DR* récepteurs : les paquets Xcast+ qui leur parviennent sont convertis en paquets *multicast* et envoyés aux réseaux dont les *DR* ont la charge (*X2M* : Xcast à *multicast*).

¹ Cette copie est souvent légèrement modifiée.

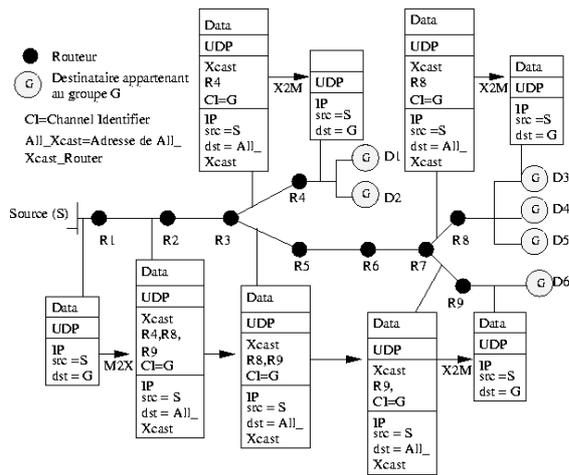


Figure 2 – Exemple de la transmission d'un paquet Xcast+.

Exemple : prenons le même exemple que celui du paragraphe 1.1 : considérons le groupe représenté sur la figure 2, comportant une source *S* et les six destinataires *D1*, *D2*, *D3*, *D4*, *D5* et *D6*. *S* envoie un paquet *multicast* ayant *G* comme adresse destination. Ce message parvient au routeur désigné de son réseau, *R1* dans notre cas. Lorsque *R1* reçoit ce paquet, il génère (en utilisant l'algorithme de transformation *M2X*) un paquet *Xcast+* avec la liste de destinations (*R4*, *R8*, *R9*) dans son en-tête *Xcast+* et il applique alors au paquet l'algorithme d'émission *Xcast+*. Cet algorithme est similaire à celui de *Xcast* à une nuance près : quand un routeur désigné, par exemple *R4*, reçoit un paquet qui lui est destiné, il applique à ce paquet l'algorithme de transformation *X2M* et envoie le paquet *multicast* ainsi généré aux réseaux dont il est le routeur désigné.

Alors que *Xcast* ne permet que de gérer efficacement les petits groupes, *Xcast+* permet la gestion de groupes *multicast* de taille moyenne. En effet, le facteur limitant est principalement le nombre d'entrées dans la liste des destinataires, et cette liste est réduite dans *Xcast+* (Myung-KI *et al*, 2001).

1.3 – Avantages et inconvénients de la technique Xcast

Par la suite, nous désignerons sous le nom générique *Xcast* les deux protocoles *Xcast* et *Xcast+* et nous appellerons technique *Xcast* le codage explicite d'une liste de destinataires dans les paquets ainsi que les algorithmes décrits précédemment.

L'utilisation de *Xcast* présente des avantages et introduit quelques inconvénients par rapport aux protocoles de routage traditionnels :

1) Avantages de la technique Xcast

a) Gestion des états de routage et des messages de signalisation : une des principales caractéristiques du protocole *Xcast* est qu'il élimine d'une part les états de routage *multicast* au sein des

routeurs et d'autre part la nécessité de mettre en place un mécanisme de signalisation spécifique au routage *multicast* entre les routeurs. Il élimine de plus le besoin de protocoles de routage *multicast* intra-domaines et inter-domaines. Cette caractéristique lui permet d'être capable de gérer de nombreuses sessions simultanément.

b) Ingénierie de trafic simplifiée : l'ingénierie de trafic *multicast* est transformée en ingénierie de trafic *unicast*. En effet, le routage *Xcast* est basé sur le routage *unicast*, les outils *unicast* pouvant donc être utilisés. De plus, les changements de topologie dans *Xcast* sont pris en compte naturellement sans qu'il n'y ait besoin d'établir une communication supplémentaire entre le protocole *unicast* et le protocole *Xcast*. Ainsi, le temps de réaction aux pannes est plus court avec *Xcast* qu'avec les protocoles *multicast* traditionnels. Outre ces deux avantages, on peut citer les deux suivants :

- *Xcast* ne nécessite pas de mécanisme complexe d'allocation d'adresses *multicast* puisque l'adresse d'une session est (*S*,*G*) et non pas *G* uniquement.
- *Xcast* ne nécessite pas que les chemins soient symétriques puisqu'il se base entièrement sur les chemins *unicast*.

2) Inconvénients de la technique Xcast

a) Fragmentation des paquets *Xcast* : un paquet IP circulant sur un réseau peut être amené à être fragmenté si sa taille dépasse la capacité du lien qu'il doit emprunter. La fragmentation est un mécanisme IP qui tronque un paquet en plusieurs paquets IP autonomes, c'est-à-dire que chacun est muni d'un en-tête IP valide, et qui partage les données parmi ces deux paquets. La figure 3 montre l'effet qu'aurait une fragmentation sur un paquet *Xcast*. On peut voir que seul le premier paquet résultant est un paquet *Xcast* valide, puisqu'il est le seul paquet à contenir un en-tête *Xcast* (il ne comporte cependant aucune données provenant de l'application). Les trois autres paquets ne seront pas traités comme des paquets *Xcast* : ils ne parviendront pas aux destinataires du groupe. Pour interdire à un paquet *Xcast* d'être fragmenté, le drapeau *DF* (*Don't Fragment*) de son en-tête IP doit être positionné à 1. Dans ce cas, si un lien est incapable de transmettre le paquet sans le fragmenter, le paquet est détruit.

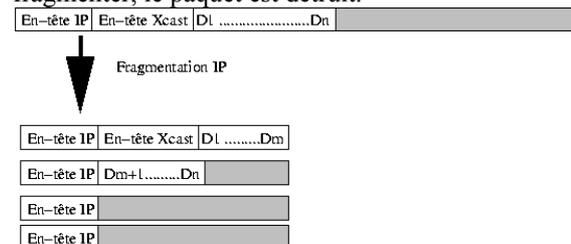


Figure 3 – Le problème de la fragmentation d'un paquet Xcast.

b) Surcharge introduite par l'en-tête Xcast : comme un paquet Xcast ne peut être fragmenté, le volume de données utiles qu'il peut contenir est restreint. Ce volume est d'autant plus petit que le nombre de destinataires est important, ce qui pose de nombreux problèmes dont la diminution notable de l'efficacité.

c) Changement des en-têtes des paquets à acheminer : un troisième problème est qu'à un paquet Xcast arrivant, un routeur Xcast doit associer plusieurs paquets Xcast dont les en-têtes sont différents : en effet, la liste des destinataires est différente pour chacun des prochains routeurs. Ce problème est atténué par la proposition d'utiliser une table d'indicateurs permettant de n'effectuer qu'un minimum de modifications dans l'en-tête afin de réduire l'impact sur le traitement et sur le calcul de la somme de contrôle des paquets à émettre.

3) Critiques de la technique Xcast

a) Utilisation d'*unicast* plutôt que du protocole Xcast : si le nombre de membres d'un groupe *multicast* est très limité, il est possible d'envoyer à chacun de ces membres un message *unicast*. Cette technique n'est pas toujours aisée à mettre en place, notamment lorsque la limitation de bande passante du « dernier mile » (*last mile*) rend nécessaire l'utilisation du *multicast* (Boivie et al, 2000).

b) Temps de réaction aux départs des membres : lorsqu'un membre désire quitter un groupe, il (ou bien son DR dans le cas de Xcast+) envoie un message à la source de ce groupe. Celle-ci prend en compte cette requête et cesse de lui faire parvenir des messages. Cependant, les paquets envoyés pendant le laps de temps entre lequel le membre a envoyé le message de désabonnement et où la source a pris en compte cette demande vont continuer d'arriver au destinataire. Une solution à ce problème a été proposée par Boudani et Cousin (2000).

2 – UNE GENERALISATION DE LA TECHNIQUE XCAST

GXcast (*Generalized Xcast*) est une adaptation de la technique Xcast conçue pour résoudre le problème de la fragmentation de la technique Xcast et pour permettre un plus grand nombre de membres par groupe.

2.1 – Description du protocole GXcast

GXcast utilise le même principe que Xcast mais il limite explicitement le nombre maximum n_m de destinataires autorisés dans un paquet Xcast dès la source. La source dans GXcast partitionne la liste initiale L de destinataires en plusieurs sous-listes de destinataires L_i . Chacune de ces listes L_i contiendra au plus n_m membres. Autant de paquets GXcast que de listes L_i seront envoyés, chacun contenant dans l'en-tête GXcast les membres contenus dans la liste

correspondante. Ce mécanisme est en fait un mécanisme de fragmentation à la source.

Exemple : considérons le groupe représenté sur la figure 4, comportant une source S et six destinataires $D1, D2, D3, D4, D5$ et $D6$. Dans un premier temps, décrivons le procédé d'adhésion d'un membre à un groupe. Chacun des destinataires envoie un message *join* IGMP qui parvient au routeur désigné de son sous-réseau. Dans notre exemple, $R4, R8$ et $R9$ reçoivent la demande d'adhésion IGMP. Ces routeurs désignés envoient alors des messages *join* vers la source S (message que $R1$ peut intercepter) qui ajoute effectivement le routeur désigné dans la liste des routeurs destinataires. À présent, étudions l'algorithme d'émission de la source. Pour couvrir les n membres en limitant le nombre de destinataires par paquets à n_m , il faut envoyer $\lceil n/n_m \rceil$ paquets.

Prenons comme exemple le cas où n_m vaut 2 et plaçons nous dans le cas du protocole Xcast+ qui considère $n=3$ membres². Le nombre de paquets à générer va être égal à 2, le premier paquet GXcast étant à destination des membres de la liste ($R4, R8$) et le second à destination du membre de la liste ($R9$). Ces paquets GXcast peuvent être considérés comme des paquets Xcast indépendants et subissent donc un traitement similaire.

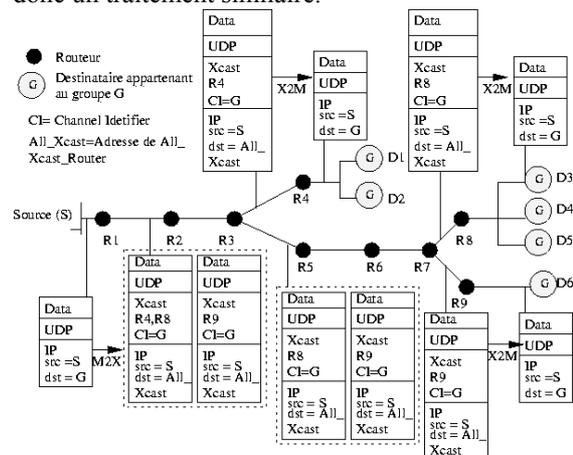


Figure 4 – Exemple de la transmission d'un paquet dans GXcast.

2.2 – Le format d'un paquet GXcast

Un paquet GXcast est très similaire à un paquet Xcast. Décrivons d'abord brièvement le format d'un paquet Xcast.

Un paquet Xcast est constitué de trois parties : un en-tête IP, un en-tête Xcast et une partie données. La source de l'en-tête IP est la source du groupe *multicast*, la destination de l'en-tête IP est l'adresse

² Ces petites valeurs de nm et de n ne sont données qu'à but pédagogique. Les valeurs réalistes de nm seront données ultérieurement. On peut néanmoins noter que, pour des petits groupes, GXcast a un comportement identique à Xcast+.

All_Xcast_Routers, le type de protocole de l'en-tête IP est XCAST_PROTO et le drapeau DF est positionné. Un champ *Destination* et un champ *Protocol* sont inclus dans l'en-tête Xcast pour indiquer respectivement l'adresse de groupe et potentiellement le protocole suivant l'en-tête Xcast (traditionnellement UDP).

Un paquet GXcast est lui aussi constitué de trois parties : un en-tête IP, un en-tête GXcast et une partie données. La différence entre un paquet GXcast et un paquet Xcast est minime :

- Le champ *Destination* de l'en-tête IP est l'adresse All_GXcast_routers.
- Le type de protocole de l'en-tête IP est GXCAST_PROTO.
- Le champ indiquant le nombre de destinataires dans un paquet GXcast doit pouvoir permettre de stocker un grand nombre de destinataires. Il semble que la limitation à 7 bits proposée dans Xcast soit insuffisante (en effet, le calcul du nombre maximal de destinataires n_{max} effectué dans le paragraphe suivant aboutit à une valeur de 134, ce qui est au-dessus de la valeur maximale représentable avec 7 bits).
- Le champ indiquant la taille de l'en-tête GXcast doit lui aussi être suffisamment grand.

Il semble raisonnable de considérer qu'un en-tête GXcast peut être stocké sur 16 octets³. Combiné à l'en-tête IP dont la taille est de 20 octets pour IPv4, la taille totale des deux en-têtes sans la liste des destinataires de GXcast atteint donc $E=36$ octets.

2.3 - Liens entre GXcast et l'unité de transfert maximale

L'unité de transfert maximale (MTU, *Maximum Transfer Unit*) est la taille du plus grand paquet pouvant circuler sur un lien sans subir de fragmentation. Comme un paquet GXcast ne peut être fragmenté au sens IP, cette taille va limiter le nombre de destinataires par paquet, c'est-à-dire la valeur de n_m . L'étude que nous proposons se base sur le fait que la MTU est connue *a priori*. Par la suite, nous utiliserons la valeur $M=576$ octets qui est la MTU minimale garantie pour le protocole IPv4⁴. Le dernier paramètre à fixer est $IP=4$ qui est la taille en octets d'une adresse IPv4⁵. De ces valeurs, on peut calculer le nombre de destinataires maximal par paquet n_{max} de la manière suivante :

$$n_{max} = \left\lfloor \frac{M-E-1}{IP} \right\rfloor = 134.$$

³ Rappelons que l'en-tête Xcast est, quant-à lui, codé sur 12 octets.

⁴ Elle est de $M=1280$ octets pour le protocole IPv6.

⁵ Cette taille est de $IP=16$ octets pour le protocole IPv6.

Cette expression intègre le fait qu'un paquet GXcast est censé contenir au moins un octet de donnée⁶.

2.4 – À propos du nombre de paquets envoyés

Rappelons que n_m désigne le nombre maximal de destinataires autorisé par paquet GXcast. Le nombre de paquets générés pour envoyer un volume D de données à un ensemble de n destinataires va dépendre de cette valeur. On s'intéresse alors à la fonction f^* qui va associer à un nombre de destinataires n et à un certain volume de données D la valeur n_m permettant d'envoyer un nombre minimal de paquets.

À titre d'exemple, considérons un ensemble de $n=100$ destinataires et une source désirant leur transmettre un volume $D=300$ octets de données. Dans le cas du protocole Xcast, la quantité de données disponibles par paquet est égale à $M-E-IP*n$, soit 140 octets. Xcast va donc nécessiter l'envoi de trois paquets pour transmettre la totalité des données. Plaçons nous à présent dans le cas du protocole GXcast et supposons que n_m soit égal à 50. La quantité de données disponibles par paquets est égale à $M-E-IP*n_m$, soit 340 octets. Chaque paquet GXcast peut donc contenir la totalité des données. Il va cependant être nécessaire d'envoyer deux paquets : le premier sera à destination des 50 premiers membres, le second à destination des 50 restants.

La valeur optimale $n_m=f^*(n,D)$ peut être calculée comme suit :

$$f^*(n,D) = \arg \min_{n_m \in \{1,134\}} \left\{ \left\lceil \frac{n}{n_m} \right\rceil \left\lceil \frac{D}{(M-E-IP.n_m)} \right\rceil \right\}$$

$\left\lceil \frac{n}{n_m} \right\rceil$ représente le nombre de paquets à envoyer pour atteindre les n destinataires en limitant à n_m le nombre de destinataires de chaque paquet. $\left\lceil \frac{D}{(M-E-IP.n_m)} \right\rceil$ est la quantité de données utiles qu'il est possible de placer dans un paquet comportant au plus n_m destinataires.

Cette définition de f^* présente deux désavantages principaux :

- le calcul associé est coûteux puisqu'il faut effectuer le calcul du minimum sur un ensemble de 134 valeurs,
- le calcul dépend de D .

Pour résoudre ces deux inconvénients, nous nous proposons de chercher une fonction f telle que :

- f est simple à calculer,
- f ne dépend que de n ,
- pour tout D , $f(n)$ s'approche de $f^*(n,D)$.

L'écart entre $f(n)$ et $f^*(n,D)$ sera alors, dans le pire des cas, le surcoût en nombre de messages par rapport à l'optimal. La fonction $\left\lceil \frac{n}{n_m} \right\rceil \left\lceil \frac{D}{(M-E-IP.n_m)} \right\rceil$ admet un minimum

⁶ Pour IPv6, on aboutit à une valeur $n_{max}=76$. En effet, pour IPv6 on a $E=56$.

pour $n_m = n_{\max}/2$. On propose donc la fonction suivante : $f(n) = n_{\max}/2$ ⁷. Le paragraphe suivant étudie le comportement de cette fonction selon différents critères.

3 – ÉVALUATION ET SIMULATION DU PROTOCOLE GXCAST

3.1 – Le nombre de paquets générés

Le nombre de paquets générés par GXcast va dépendre des trois paramètres n , D et n_m . Le choix pour n_m de la valeur $f(n)$ a été justifié dans le paragraphe 2.4 (en tant que minimum). La figure 5 présente le surcoût de GXcast par rapport à un protocole *multicast* traditionnel. Rappelons qu'un protocole *multicast* traditionnel envoie $\lceil D/(M-E) \rceil$ paquets pour faire parvenir D octets de données à un groupe n de membres⁸. On remarque tout d'abord que le surcoût engendré par GXcast est pratiquement indépendant de la taille D de données à émettre. On remarque de plus que ce surcoût est linéaire en fonction de n : pour un groupe constitué de 150 membres, il sera nécessaire d'envoyer environ 5 fois plus de paquets avec GXcast qu'avec un protocole traditionnel. Pour un groupe de 300 membres, il sera nécessaire d'envoyer 10 fois plus de paquets.

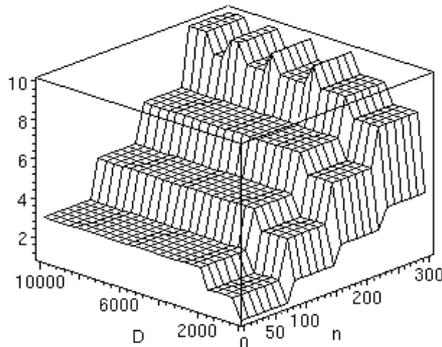


Figure 5 – Surcoût de GXcast par rapport à un protocole *multicast* traditionnel.

Pour transmettre à respectivement 150 et 300 destinataires un message de 1000 octets, un protocole basé sur des messages *unicast* aurait nécessité 150 et 300 paquets, soit respectivement 150 et 300 fois plus qu'un protocole *multicast* traditionnel⁹.

3.2 – Le temps de traitement global

⁷ Notons que $f(n)$ ne dépend pas de n .

⁸ On considère ici que la taille de l'en-tête associé au protocole *multicast* est aussi de E .

⁹ Rappelons que Xcast est incapable de gérer un groupe ayant plus de 134 membres.

Le temps de traitement global d'un protocole est la somme des temps de traitement des paquets qu'il envoie. Dans ce paragraphe, nous étudions l'évolution du temps de traitement global en fonction du choix du paramètre n_m . Nous nous intéresserons plus particulièrement au cas où $n_m = f(n)$. Pour un paquet GXcast comportant n_m destinataires, le temps de traitement est approché par $t_{n_m} = \tau + n_m \cdot \tau'$, où τ représente le temps de traitement des en-têtes IP et GXcast et τ' le temps de traitement associé à une entrée dans la liste des destinataires (recherche dans la table de routage, création des paquets par interface de sortie, etc.). On a donc le temps de traitement global défini par

$$t_G(n_m) = \lceil n / n_m \rceil t_{n_m} \approx \frac{n \cdot \tau}{n_m} + n \cdot \tau'$$

$t_G(n_m)$ est une fonction strictement décroissante. Son minimum est donc en $t_G(n_{\max})$. Cependant, choisir $n_m = n_{\max}$ est irréaliste. En effet, la formulation de $t_G(n_m)$ ne prend pas en compte les données à envoyer.

3.3 - Le délai supplémentaire moyen

Le délai d'un paquet est le temps qu'il lui faut pour atteindre sa destination à partir de la source. Il dépend en partie du nombre de paquets dans la file d'attente des routeurs et du temps de traitement de chacun d'eux. La différence de délai perçue par l'utilisateur final (c'est-à-dire le destinataire) joue un rôle important dans le choix des protocoles. Afin de mesurer le délai supplémentaire introduit au niveau du récepteur par l'utilisation de différentes valeurs de n_m , le protocole GXcast a été implémenté sous la plate-forme de simulation NS (Fall et Varadhan, 2001). Le réseau de test est un réseau aléatoire généré à l'aide de GT-ITM, algorithme proposé par Zegura *et al* (1996). Il possède 100 nœuds reliés entre eux par des liens bidirectionnels de capacité 20 Mb/s. On ne considère qu'un seul groupe *multicast* comportant une source et un ensemble de 18 destinataires qui s'abonnent et se désabonnent à des instants quelconques. Le nombre maximal de destinataires par paquet GXcast a été fixé à 1, 3, 6, 9 et 18 afin de pouvoir observer l'évolution du délai. La figure 6 présente le délai supplémentaire moyen par rapport à la valeur 18 de n_m . On remarque que le pourcentage de délai ajouté est assez faible, puisqu'il n'est que de 30% dans le pire des cas. Le délai ajouté pour la valeur $n_m = n/2 = 9$ est très faible puisqu'il ne représente que 5% du délai optimal.

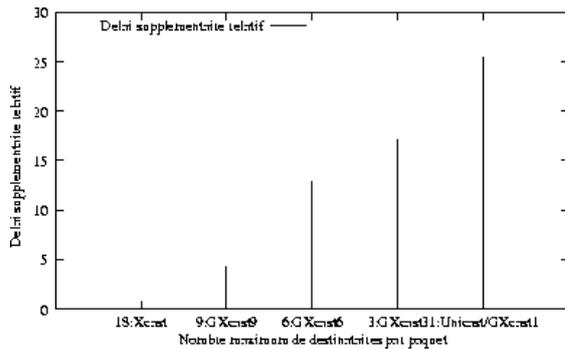


Figure 6 – Délai supplémentaire moyen en fonction de n_m .

4 – CONCLUSIONS

Les protocoles Xcast et Xcast+ permettent de gérer efficacement un grand nombre de petits groupes *multicast*. Une partie de leurs inconvénients vient de leur incapacité à gérer la fragmentation des paquets. De plus, une limite forte existe sur le nombre de membres du groupe *multicast*. Dans ce papier, nous avons proposé une extension à ces protocoles nommée GXcast. GXcast permet de résoudre le problème de la fragmentation et optimise des critères comme le nombre de paquets envoyés ou le temps de traitement des paquets au sein des routeurs. Une étude ultérieure effectuera l'analyse de GXcast en fonction de la topologie du réseau. À l'issue de l'étude de ce protocole, il semble néanmoins que GXcast puisse gérer un grand nombre de groupes de taille moyenne, dont les membres sont disséminés sur plusieurs centaines de sous-réseaux.

BIBLIOGRAPHIE

- Boivie, R., Feldman, N., Imai, Y., Livens, W., Ooms, D., Paridaens, O. (2000), *Explicit multicast (Xcast) basic specification*, IETF Internet Draft.
- Myung-KI, S., Yong-Jin, K., Ki-Shik, P., Sang-Ha, K. (2001), *Explicit multicast extension (Xcast+) for efficient multicast packet delivery*, ETRI Journal, vol 23, n° 4
- Fall, K., Varadhan, K. (2001), *The NS Manual*, UC Berkeley, LBL, USC/ISI, and Xerox PARC.
- Reliable Multicast Transport Working Group Web Site (2003), <http://www.ietf.org/html.charters/rmt-charter.html>

Ooms, D. (2000), *Taxonomy of Xcast/SGM proposals*, IETF Internet Draft.

Deering, S., Hares, S., Perkins, C., Perlman, R. (2000), *Overview of the 1998 IAB Routing Workshop*, IETF RFC 2902.

Cain, B., Deering, S., Kouvelas, I., Fenner, B., Thyagarajan, A. (2002), *Internet Group Management Protocol, Version 3*, IETF RFC 3376.

Boudani, A., Cousin, B. (2003), *SEM: a new small group multicast routing protocol*, Proceedings of the 10th International Conference on Telecommunications (ICT).

Zegura, E., Calvert, K., Bhattacharjee, S. (1996), *How to model an internet network*, Proceedings of INFOCOM.

***UNE APPROCHE POUR LE CONTROLE DE LA QUALITE
DES SYSTEMES DE FILTRAGE COLLABORATIF***

Lizbeth Gallardo-Lopez *

Docteurant en Informatique

gallardo-lopez@imag.fr + 33 4 46 51 46 29

Catherine Berrut,

Professeur en Informatique

catherine.berrut@imag.fr + 33 4 76 51 42 63

Nathalie Denos,

Maître de conférences en Informatique

nathalie.denos@imag.fr +33 4 76 51 49 15

Adresse professionnelle

Laboratoire CLIPS-IMAG Equipe MRIM ★ BP 53 ★ 38041 Grenoble Cedex 9

Résumé : Assurer la qualité des systèmes de filtrage collaboratif pendant leur exploitation est un sujet d'actualité qui répond au constat du manque d'utilisation à long terme de ces systèmes. Le but de cet article est de proposer une approche pour assurer cette qualité à travers un système de contrôle capable d'identifier les problèmes et de proposer des actions pour y remédier. Nous décrivons d'abord une méthodologie pour définir les bases d'un tel système de contrôle, et montrons comment elle permet d'aboutir à des indicateurs opérationnels destinés à alimenter un tableau de bord.

Summary : To tackle the well-known problem of limited user motivation to participate in collaborative filtering systems, a quality assurance policy must be defined and implemented for these systems. In this paper, we propose to add a control layer to collaborative filtering systems, in order to identify problems and adapt the system to remedy them. We describe our methodology to define such a control system and show how the quality indicators can be made operational in order to feed a control panel.

Mots clés : Information, représentation, accès, et traitement intelligent ; Filtrage Collaboratif ; Qualité; Contrôle

* Ce travail a bénéficié du soutien de CONACyT

Une approche pour le contrôle de la qualité des Systèmes de Filtrage Collaboratif

1. INTRODUCTION

De nos jours, d'importantes quantités d'informations sont à la disposition de chacun grâce au développement des technologies de l'information. Afin d'aider à la découverte de documents intéressants, des outils comme les systèmes de filtrage d'informations sont développés. Ces systèmes fournissent aux utilisateurs un flot continu de documents, sans qu'ils aient à exprimer explicitement ce qu'ils cherchent, contrairement aux systèmes de recherche d'information où une requête doit être produite à chaque fois. En contrepartie, il faut que le système connaisse les centres d'intérêt des utilisateurs (profils) et suive leur évolution au cours du temps. Cela est possible grâce aux retours d'évaluations que l'utilisateur donne sur les documents fournis par le système.

Les systèmes de filtrage en général requièrent la participation active des utilisateurs pour atteindre de bonnes performances : c'est en effet par leurs évaluations des documents reçus que les utilisateurs ajustent leur profil. Le filtrage dit « collaboratif » fait intervenir explicitement une communauté par une entraide mutuelle : chaque membre de la communauté reçoit des documents que d'autres membres ont jugés intéressants. Pour cela, il est encore plus important que chaque utilisateur donne régulièrement son point de vue sur les documents envoyés par le système car les performances du système dépendent du comportement de l'ensemble des utilisateurs : un utilisateur qui ne contribue pas à alimenter le système voit non seulement les performances baisser pour lui-même, mais fait également baisser les performances pour les autres ! Cette situation est un frein à une utilisation réelle, efficace et à long terme des systèmes de filtrage collaboratif, comme le montrent des évaluations de systèmes faites sur le terrain [Lueg98] [Resnick et al. 94] [Miller et al. 97].

Si l'on veut que leur travail d'évaluation ne soit pas perçu par les utilisateurs comme un pensum sans contrepartie, il faut munir le système de filtrage d'un contrôle de la qualité.

Notre objectif de recherche est de définir et mettre en œuvre un tel système de contrôle qui, à partir de données extraites du système de filtrage en cours d'exploitation, élabore des indicateurs de qualité. Ces indicateurs permettront d'identifier des situations problématiques et de suggérer des actions afin de garantir un niveau de performance acceptable. De façon plus générale, nous proposons une méthodologie permettant de définir de nouveaux indicateurs basés sur différents critères qualitatifs (utilisabilité, maintenance, fonctionnalité, etc.).

Dans cet article, nous décrivons d'abord les systèmes de filtrage collaboratif, et la motivation de notre recherche (section 2) ; puis l'état de l'art sur les travaux existants en matière d'évaluation des systèmes de filtrage collaboratif (section 3) ; notre approche théorique et une méthodologie d'évaluation et de surveillance d'un système de filtrage collaboratif (section 4).

2. FILTRAGE COLLABORATIF

Dans la plupart des communautés partageant des centres d'intérêt, les personnes échangent régulièrement des documents avec leurs amis ou collègues. Ces échanges véhiculent explicitement ou non un message tel que : « ça va t'intéresser » ou « ça va te plaire ». Les systèmes de filtrage collaboratif ou SFC, tentent de mettre en œuvre un tel processus pour intégrer plus efficacement un grand nombre d'utilisateurs et de documents. Ces systèmes sont donc destinés à des utilisateurs ayant des buts à long terme.

2.1 Modèle conceptuel

Pour mieux comprendre le fonctionnement des SFC, nous en donnons un modèle conceptuel formalisé via des entités et des processus manipulant l'information sur ces entités. Les principales entités sont les suivantes. Un *utilisateur* est une personne qui reçoit de l'information, grâce aux recommandations faites par le SFC. Un *document* est un support d'information susceptible de satisfaire les besoins d'un

utilisateur ou d'une communauté d'utilisateurs. Une *communauté* est un groupe d'utilisateurs partageant les mêmes goûts et/ou intérêts¹. Le *profil* d'un utilisateur est constitué de ses évaluations sur des documents, qui traduisent ses centres d'intérêts. Une *évaluation* est un jugement porté par un utilisateur sur un document, souvent basé sur une échelle discrète (1-5 par exemple).

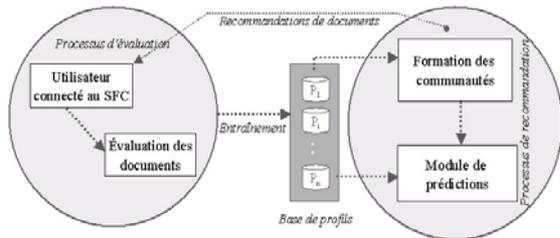


FIG. 1 Fonctionnement d'un système de filtrage

La figure FIG.1 illustre le fonctionnement des différents processus intervenant dans un SFC : un utilisateur étant connecté au système reçoit des *recommandations* de documents ; l'utilisateur, de son côté, décide d'évaluer ou non les documents proposés. L'*évaluation* va alimenter son profil stocké dans une *base de profils* qui sera donnée en entrée au *processus de recommandation*². Ce processus est constitué d'un module de *formation des communautés*, dont le principe est d'obtenir le voisinage le plus proche de chaque utilisateur³ et d'un module de *prédiction*, qui estime le degré de satisfaction qu'un document pourrait procurer à un utilisateur⁴.

¹ Il s'agit d'une communauté virtuelle calculée par le système : les utilisateurs ne se regroupent pas eux-mêmes, ils peuvent même ne pas se connaître.

² Le processus de recommandation est mise en œuvre par ce qu'on appelle un « moteur de filtrage », pour lequel il existe différents algorithmes possibles ([Resnick et al. 94] [Gokhale99] [Maltz et al. 95]).

³ L'algorithme dédié à la formation des communautés calcule la similarité entre paires d'utilisateurs en comparant les évaluations de documents se trouvant dans leurs deux profils.

⁴ L'algorithme de prédiction prend en entrée d'une part les évaluations que les utilisateurs ont données dans le passé sur un document, et d'autre part la similarité entre paires d'utilisateurs. La valeur de l'évaluation prédite combine les évaluations faites par les autres utilisateurs, en les pondérant selon leur similarité avec l'utilisateur en cours.

Les SFC peuvent s'appliquer au domaine commercial (les documents sont des produits à vendre), à la documentation scientifique, aux messages de forums de discussion⁵, etc. Tous suivent ce modèle conceptuel, mais dans leur mise en œuvre, chacun définit ses propres fonctionnalités et ses contraintes internes en prenant en compte l'objectif pour le contexte d'application : le type de documents manipulés, le type d'utilisateurs qui y auront accès, etc.

2.2 Motivation de recherche

Nous souhaitons analyser la qualité d'un SFC à partir du critère de « fonctionnalité ». La fonctionnalité signifie que le système doit accomplir correctement ses tâches [AFNOR95]. Nous pensons que pour garantir un bon niveau de fonctionnalité dans le SFC, il est nécessaire d'ajouter une couche de surveillance sous la forme d'un système de contrôle qui permet aux administrateurs de suivre le déroulement de l'exploitation du SFC. Ce système de contrôle doit fournir les indicateurs utiles pour estimer ou évaluer l'état de fonctionnement du SFC.

Un *indicateur* est une donnée objective qui décrit une situation ou un phénomène du strict point de vue quantitatif ; à ce titre, sa valeur évolue dans le temps. Dans un premier temps, nous nous intéressons aux situations problématiques suivantes:

La perte d'information due aux contraintes internes de chaque type de système de filtrage. Par exemple le choix du seuil en-dessous duquel une corrélation est considérée comme ne traduisant plus l'appartenance à une même « communauté, a un effet déterminant sur le processus de recommandation.

Les mauvais usages du système faits par des utilisateurs ayant une perception erronée du modèle conceptuel. Par exemple, quelqu'un qui évalue toujours les documents avec la même note n'a peut-être pas compris le fonctionnement du système, qui manque alors d'éléments discriminants pour lui fournir de bonnes recommandations.

Le « facteur d'usure » lié aux données qui au cours de l'exploitation deviennent obsolètes et qui sont tout de même prises en compte par le système, à tort. Par exemple, les profils des

⁵ Par exemple: MovieLens, MovieCritic, Amazon (commercial) ; Torii, Referral Web (scientifique) ; Phoaks, Grouplens (news groups).

utilisateurs qui ne participent plus au système, ne doivent plus être pris en compte, car finalement ils ne contribuent plus à la communauté.

Au vu de ces situations, un certain nombre de questions apparaissent : Est-il possible de définir et de mettre en œuvre un système de contrôle sur le SFC ? Ce système de contrôle serait-il capable de garantir aux utilisateurs un niveau de fonctionnalité acceptable ? Quelle méthodologie peut-on suivre pour définir ce système de contrôle ?

Pour contrôler, on a besoin d'observer et d'évaluer de façon permanente l'état qualitatif du système pendant son exploitation. Pour cela nous avons fait une étude sur l'état de l'art (section 3 ci-dessous), et nous avons constaté que la notion de contrôle n'existe pas, et par ailleurs que très peu de travaux sont consacrés à l'évaluation des SFC pendant leur exploitation.

3. TRAVAUX ACTUELS DANS L'EVALUATION DES SFC

Dans l'évaluation des SFC, nous distinguons deux catégories d'études : les évaluations en laboratoire qui ont pour but de mesurer la performance du seul moteur de filtrage ; et les évaluations sur le terrain, qui ont pour but d'étudier la performance globale du système.

3.1 Les évaluations réalisées en laboratoire

Ce type d'évaluation vise à comparer la performance de plusieurs moteurs (algorithmes) de filtrage. Il s'agit de mesurer automatiquement sa rapidité et sa correction. Parmi les techniques, on peut souligner celle de la « collection-test », consistant en une base d'évaluations fournies par des vrais utilisateurs. On divise la base en deux sous-bases : la première est plus grande et représente le passé du système de filtrage, et la deuxième représente les données de référence. La base du passé alimente le moteur pour son exécution, et les résultats ainsi obtenus sont comparés avec les données de référence. Deux types de métrique sont utilisés: les métriques pour mesurer l'erreur, notamment la moyenne de l'erreur absolue et l'écart type ; et les métriques pour évaluer la satisfaction de l'utilisateur, notamment le rappel et la précision aussi utilisées dans le domaine de la recherche d'information.

Cette approche présente plusieurs inconvénients soulignés par [Herlocker et al. 00]:

d'abord, il y a une grande diversité des techniques d'évaluation en laboratoire qui fournissent des résultats qui ne sont pas comparables car il n'existe toujours pas de standardisation des métriques employées pour l'évaluation. Voir [Breese et al. 98] [Herlocker et al. 00] pour une étude appliquée des métriques, et aussi [Miller et al. 97] [Goldberg et al. 00].

3.2 Les évaluations sur le terrain

Ce type d'évaluation fait intervenir de vrais utilisateurs [Fidel97]. Dans cette approche les utilisateurs sont mis dans un contexte d'utilisation prédéfini. Après avoir interagi avec le système de filtrage, ils sont amenés à répondre à des questionnaires. Le but est d'apprendre leur point de vue sur des aspects bien précis et basés sur des critères qualitatifs ciblés par l'évaluation du système de filtrage. Parmi ces critères on souligne la fonctionnalité, l'utilisabilité, la convivialité, etc. La dernière phase consiste à mettre en œuvre des méthodes statistiques pour le traitement des données, afin de tirer des conclusions sur la performance du système de filtrage.

Cette approche présente aussi des inconvénients. D'abord, le fait d'imposer un contexte prédéfini d'utilisation réduit la possibilité de mettre l'utilisateur face à un environnement réel. Elle est d'ailleurs difficile à mettre en œuvre, car souvent les utilisateurs ne sont pas disponibles. Enfin, selon les objectifs de l'évaluation, la préparation des questionnaires, la période d'observation, l'application des enquêtes et l'obtention des résultats peut prendre plusieurs mois.

3.3 Conclusion et synthèse de l'état de l'art

Le système de filtrage collaboratif, tel qu'on vient de le voir, est perçu comme une boîte noire autant par les administrateurs que par les utilisateurs. Les concepteurs et administrateurs du système n'ont que très peu d'informations sur la fonctionnalité du SFC, laquelle est basée uniquement sur le moteur de filtrage. Les utilisateurs, quant à eux, n'ont pas d'autres éléments que la pertinence des documents reçus et son expérience passée avec le système (notamment le coût-bénéfice ressenti au cours de l'exploitation) pour juger de la qualité du SFC.

De notre point de vue le SFC voit la qualité de ses réponses se dégrader non seulement à cause de la performance du moteur, mais aussi

à cause d'autres processus qui peuvent entraîner des détériorations. D'ailleurs, un constat, dans l'exploitation réelle des systèmes de filtrage collaboratif, est qu'ils rencontrent rarement l'adhésion des utilisateurs sur le long terme, même lorsque l'algorithme de filtrage est jugé de bonne qualité en laboratoire [Lueg et al. 97] [Maltz et al. 95] [Herlocker et al. 00]. Ces constats suscitent une fois de plus le besoin d'un système de contrôle.

4. UN SYSTEME DE CONTROLE SUR LE SFC

Nous répondons ici successivement à chaque question de la section 2.2.

Nous considérons que la définition et la mise en œuvre d'un système de contrôle sur un SFC est possible grâce à la simplicité du modèle conceptuel d'un SFC (défini dans la section 2). Chaque processus est bien délimité et on en connaît les données d'entrée, les données de sortie, ainsi que les contraintes manipulées à l'intérieur de chaque processus. Par exemple, *le processus de formation des communautés* reçoit en entrée la base des profils des utilisateurs, et retourne l'ensemble des communautés formées à partir de la corrélation entre paires de profils ; la contrainte, interne au processus, est le seuil au-dessous duquel les corrélations ne traduisent plus l'appartenance à la même communauté. Le système de contrôle peut donc utiliser ces données comme des paramètres pour évaluer la fonctionnalité de chaque processus et du SFC en général. Pour revenir à l'exemple, apprendre qu'un utilisateur n'appartient à aucune communauté car son profil est loin de tous les autres utilisateurs, permet d'identifier un problème simple. Une action corrective peut être d'informer l'utilisateur de ce fait et de lui proposer des documents jugés pertinents par la communauté la moins éloignée de son profil.

Nous pensons que si on garantit la fonctionnalité dans chaque processus, on garantit aussi le niveau de fonctionnalité de l'ensemble du système, et la satisfaction de l'utilisateur.

Nous proposons une méthodologie en trois étapes (voir figure 2) : la construction des indicateurs, la synthèse des indicateurs et l'exploitation d'un tableau de bord. Un tableau de bord est un outil de visualisation synthétique de l'état du SFC via les indicateurs. Son utilisation va aider les administrateurs à pren-

dre des décisions sur le SFC, afin de corriger, voire de prévenir les problèmes.

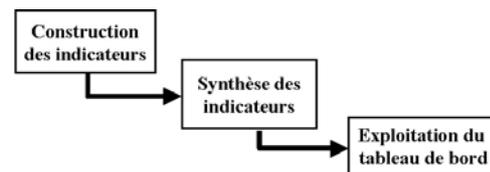


FIG. 2 Méthodologie pour obtenir un système de contrôle

La construction des indicateurs exige une connaissance détaillée des processus internes du SFC, ainsi que des besoins de l'utilisateur final. A partir de ces connaissances, on peut définir des indicateurs reflétant la performance des différents processus et du système dans sa globalité.

La synthèse des indicateurs consiste à définir des indicateurs globaux qui synthétisent un ensemble d'indicateurs basiques, afin de fournir un type d'information différent et/ou limiter la charge d'information pour les administrateurs.

L'exploitation du tableau de bord a pour but de résoudre les problèmes qu'il a permis de mettre en évidence. Dans cette phase nous envisageons la possibilité de définir des actions type comme réponse aux problèmes identifiés.

Dans la suite, toute l'analyse est basée sur un SFC minimal et générique, c'est-à-dire, sur les principaux processus du modèle conceptuel, sans tenir compte d'un SFC particulier. Pour cela, nous avons étudié différents systèmes de filtrage, quelques-uns disponibles sur Internet et notamment COCofil un SFC développé dans notre équipe (http://tips.imag.fr:22222/Portal/TIPS_EnterPortal.html). Nous avons cherché à construire des indicateurs génériques, afin de pouvoir les adapter sur différents SFC en prenant en compte les contraintes internes de chaque système.

4.1 Méthode de construction des indicateurs

Dans [Cérutti et al. 92], repris par [Selmer03], une méthode progressive de construction des indicateurs est proposée. Elle vise plus particulièrement le domaine de la gestion de production, mais les principes et les étapes suivies restent appropriés pour définir des indicateurs sur un SFC.

Cette méthode se décompose en sept étapes que nous détaillons ci-dessous. La figure 3

montre ces étapes et les illustre sur un exemple pris dans le contexte des SFC.

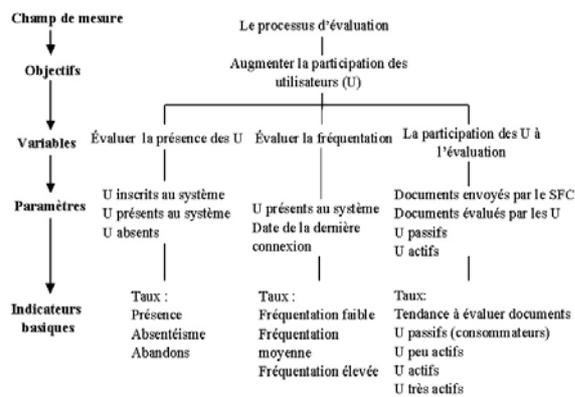


FIG. 3 Processus de construction des indicateurs

La définition du champ de mesure, consiste à définir le domaine auquel on s'intéresse et sur lequel va porter une mesure qualitative. La question qu'on se pose est « sur quoi est-il intéressant de faire le point ? ». Pour nous, il s'agit : du processus d'évaluation, du processus de recommandation et de l'évolution du corpus d'utilisateurs et de documents. Pour expliquer la méthode nous allons suivre l'exemple du *processus d'évaluation* (figure 3).

La définition des objectifs que l'on cherche à atteindre dans ce champ. Leur définition donne tout son sens à la mesure ; cependant, à ce stade de la réflexion, les objectifs n'ont pas besoin d'être quantifiés. Exemple : Dans le processus d'évaluation on cherche à augmenter la participation des utilisateurs.

L'identification des variables. On cherche les éléments qu'il est nécessaire de suivre pour se situer par rapport aux objectifs. Exemple : Pour augmenter la participation des utilisateurs il nous paraît pertinent de suivre la fréquentation du système par les utilisateurs, et leur participation à l'évaluation des documents.

Le choix des paramètres mesurables : les variables doivent être traduites en paramètres mesurables, qui peuvent être nombreux et parmi lesquels il est nécessaire de faire un tri selon l'importance du paramètre ou son accessibilité. Exemple : Que peut-on mesurer sur les variables choisies ? Nous avons choisi un ensemble de paramètres à partir de l'observation et de l'expérience en tant qu'utilisateur et administrateur de COCoFil. Nous avons limité la source des paramètres aux données stockées dans la base de données de ce système.

La sélection des indicateurs : Une fois la sélection des paramètres menée à bien, il est nécessaire de les transcrire en données quantifiables, et éventuellement de les combiner pour obtenir un indicateur global. On obtient ainsi un ensemble d'indicateurs, censés être représentatifs du phénomène que l'on veut observer. Exemple : Les indicateurs obtenus par combinaison des paramètres mesurables, sont présentés et classés par variable :

Présence des utilisateurs : Les indicateurs (1) et (2) permettent d'apprendre sans ambiguïté le *taux d'utilisateurs présents* et le *taux d'utilisateurs absents* du système pendant une période. En fonction du temps d'absence, on pourrait déduire l'ensemble des *utilisateurs en situation d'abandon* (3).

$$\text{Taux de présence dans SCF} = \frac{U \text{ présents}}{U \text{ inscrits}} \quad (1)$$

$$\text{Taux absentéisme} = \frac{Nb. U \text{ absents}}{Nb. U \text{ inscrits}} \quad (2)$$

$$\text{Taux d'abandon} = \frac{Nb. U \text{ en situation d'abandon}}{Nb. U \text{ inscrits}} \quad (3)$$

Fréquentation des utilisateurs : il s'agit de calculer le nombre de sous-périodes dans lesquels l'utilisateur a été présent pour une période donnée. Nous définissons ainsi deux seuils qui vont distinguer trois catégories d'utilisateurs : ceux de *faible fréquentation*, ceux de *fréquentation moyenne* et finalement ceux de *fréquentation élevée*. Pour chaque catégorie d'utilisateurs on obtient le taux correspondant.

Participation des utilisateurs au SFC : L'indicateur de *tendance à évaluer* (4) va nous servir pour mesurer la participation de chaque utilisateur. Nous pourrions alors calculer le *taux d'utilisateurs passifs* c'est à dire qui n'ont pas participé à l'évaluation des documents ; ainsi que le taux d'utilisateurs considérés comme *peu actifs, actifs et très actifs*.

$$\text{Tendance à évaluer} = \frac{Nb. docs. proposés et évalués}{Nb. docs. proposés} \quad (4)$$

La validation des indicateurs sélectionnés. Avant de mettre en œuvre les indicateurs, il est préférable de vérifier leur cohérence et leur pertinence. La lecture de l'indicateur doit permettre, sans ambiguïté, de faire le point sur le champ de mesure choisi.

L'élaboration du tableau de bord. Pour faire du tableau de bord un véritable outil de com-

munication, il est nécessaire d'organiser les indicateurs obtenus, puis de les présenter en offrant une bonne visualisation de leurs résultats. Le tableau de bord doit être daté, et peut présenter différents indicateurs sous forme de graphiques, par exemple en camembert, en courbe xy, en barres, etc. Nous envisageons un tableau de bord automatisé faisant partie intégrante du système de contrôle.

4.2 Synthèse des indicateurs

Les indicateurs sont trop nombreux pour être suivis individuellement par un administrateur. Nous proposons d'établir des indices plus globaux synthétisant plusieurs indicateurs, de façon à mettre en évidence une information différente ou simplement réduire le nombre d'indicateurs à surveiller. Ces indices donneront la tendance générale et permettront de tirer une éventuelle sonnette d'alarme en cas de dégradation des résultats.

Pour la construction de ces indices, nous utilisons des techniques issues du domaine de la production. Si les indicateurs sont de même nature, on utilise l'**approche multicritère** qui consiste à effectuer une moyenne pondérée en tenant compte de l'importance relative de chacun des indicateurs concernés. S'il s'agit d'indicateurs divers et variés, on utilise la méthode dite des **indices Vigier** [Cérutti et al. 92] qui permet de ramener l'expression de chaque indicateur à un nombre « sans dimension » compris entre 0 et 1. On distingue deux types d'indicateurs à synthétiser :

Les indicateurs à limite unilatérale comportant une frontière au-delà de laquelle il ne faut pas aller. Par exemple, le taux d'absentéisme des utilisateurs ne doit pas dépasser 20% des utilisateurs inscrits. La relation entre l'indicateur X mesuré sur le terrain et son indice de qualité S_X nous est donnée par l'équation (5) (voir figure 4).

$$S_X = \exp(-\exp(V)) \quad (5)$$

$$V = aX + b \quad (6)$$

X : valeur réelle de l'indicateur opérationnel

S_X : indice « sans dimension »

a et b : deux paramètres permettant de choisir la sévérité avec laquelle on va juger l'indicateur opérationnel X ; la sévérité est exprimée par la pente de la courbe.

Par convention la valeur limite donnée à l'indicateur opérationnel X pour être satisfaisant correspond toujours à un indice $S_X = 0,37$; une valeur médiocre correspond à $S_X = 0,15$ et une valeur bonne à $S_X = 0,9$.

Les indicateurs à limite bilatérale sont délimités par une zone à l'intérieur de laquelle l'indicateur doit se trouver pour être « bon ». Par exemple, le taux de croissance de la base d'utilisateurs doit être inférieur à 5% mais supérieur à 1%. La relation entre l'indicateur Y mesuré sur le terrain et son indice de qualité S_Y nous est donnée par l'équation (7) ; l'indice calculé est visualisé sur la courbe du graphique sur la figure 5.

$$S_Y = \exp(-|U|^n) \quad (7)$$

$$U = \frac{2Y - (Y_{\max} + Y_{\min})}{Y_{\max} - Y_{\min}} \quad (8)$$

$$n = \frac{\log\left(\log\left(\frac{1}{S_Y}\right)\right)}{\log|U|} \quad (9)$$

Y : valeur réelle de l'indicateur opérationnel

S_Y : indice « sans dimension »

Y_{\max} et Y_{\min} : limites supérieure et inférieure

n coefficient de sévérité pour juger l'indicateur opérationnel exprimé par la « largeur » de la courbe.

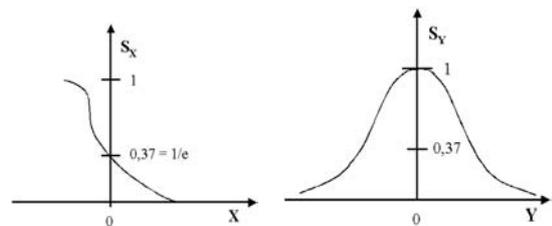


FIG. 4 Unilatérale **FIG. 5 Bilatérale**

Ainsi les indicateurs mesurés sur le terrain seront ramenés à des nombres sans dimension: S_X et S_Y . Finalement l'indice global que l'on pourra comparer avec la valeur de référence ($S=0,37$) est défini par l'équation (10). Où n = nombre d'indicateurs opérationnels ramenés à un indice « sans dimension ».

$$S_n = ((S_{X1})(S_{X2})... (S_{Xn}))^{1/n} \quad (10)$$

Pour l'illustrer, nous appliquons cette méthode à l'exemple de la section 4.1. Pour faciliter la

surveillance du processus d'évaluation, nous construisons deux indices globaux qui vont résumer les indicateurs de *fréquentation* et de *participation*⁶.

Fréquentation	Participation
Tx. de présence	Tx. d'U passifs
Tx. d'absentéisme	Tx. d'U peu actifs
Tx. d'abandon	Tx. d'U actifs et très actifs
Tx. d'U de fréq. faible	
Tx. d'U de fréq. Moyenne+élevée	

TAB. 1 Tri des indicateurs par variable

Nous devons d'abord fixer, pour chaque *indicateur opérationnel*, les 3 bornes (médiocre, satisfaisant, bon) qui vont nous permettre de les qualifier⁷. Nous construisons alors deux modèles pour calculer l'indice de qualité S_X associé à la valeur de l'indicateur opérationnel X . Selon que X est inférieur à la borne « satisfaisant » (mauvaise qualité) ou supérieur à cette borne (bonne qualité), l'un ou l'autre des modèles sera appliqué. Les bornes « médiocre » et « bon » serviront à fixer la sévérité de chacun de ces modèles. Illustrons cela avec l'indicateur *taux de présence*⁸ (voir le tableau 2).

Présence	Absentéisme	Abandon	Fréq. faible	Fréq. moyenne + élevée	Bornes
70	20	5	5	85	Médiocre
80	10	2	3	95	Satisfaisant
95	1	0.5	1	99	Bon
80,5	16,7	2,80	3,79	96,52	Opérationnel

TAB. 2 Bornes et indicateurs opérationnels

Modèle de bonne qualité : Un indicateur est considéré de bonne qualité quand son taux est supérieur à 80%.

Par convention, pour $X=80\% = 0,8$ on a $S=0,37$, soit $V = -\ln(\ln(0,37)) = 0$ (voir équation (5)) ; de même, pour $X=95\% = 0,95$ on a $S=0,9$

⁶ Les taux d'utilisateurs relatifs à une fréquentation « moyenne » et « élevée » sont regroupés car on considère que tous deux correspondent à l'idée d'une fréquentation satisfaisante. On procède de même pour les taux relatifs aux utilisateurs « actifs » et « très actifs »

⁷ Ces bornes dépendent du SFC qu'on est en train d'évaluer.

⁸ Pour les autres indicateurs on donne juste le résultat final.

et donc $V = -\ln(\ln(0,9)) = 2,25$. Avec la formule (6) on obtient deux équations : $0 = 0,8a + b$ et $2,25 = 0,95a + b$, qui donnent $a=15$ et $b=-12$, d'où $V=(15X-12)$. Ainsi, pour tout taux de présence supérieur à 80%, l'indice S_X correspondant sera déterminé par la formule (11) :

$$S_X = \exp(-\exp-(15X - 12)) \quad (11)$$

Modèle de mauvaise qualité : Un indicateur est considéré de mauvaise qualité quand son taux est inférieur à 80%.

Ici, pour $X=80\% = 0,8$ on a $S=0,37$, soit $V = -\ln(\ln(0,37)) = 0$ (voir formule (5)) ; de même, pour $X=70\% = 0,7$ on a $S=0,15$ et donc $V = -\ln(\ln(0,15)) = -0,64$. Avec la équation (6) on obtient le système : $0 = 0,8a+b$ et $-0,64 = 0,7a+b$, qui donne $a=6,4$ et $b=-5,12$, d'où $V = (6,4X-5,12)$. Ainsi, pour tout taux de présence inférieur à 80%, l'indice S_X correspondant sera déterminé par la formule (12) :

$$S_X = \exp(-\exp-(6,4X - 5,12)) \quad (12)$$

Avec les valeurs de la dernière ligne du tableau 2, on a comme taux de présence $X_I = 80,5 > 80$; on applique donc le modèle de bonne qualité, d'où l'indice de présence $S_{X_I} = 0,3954$. De même pour les autres indicateurs on obtient : l'absentéisme $S_{X_2} = 0,2153$; l'abandon $S_{X_3} = 0,3054$; la fréquentation faible $S_{X_4} = 0,2759$; et la fréquentation moyenne et élevée $S_{X_5} = 0,6556$.

L'indice global de la fréquentation est calculé par la formule (10), donc $S_5 = (0,3954 * 0,2153 * 0,3054 * 0,2759 * 0,6556)^{1/5} = 0,3423$. Comme $S_5 < 0,37$, la fréquentation n'atteint pas l'objectif espéré.

Pour ce qui concerne l'indice global de participation, nous avons procédé de la même façon et obtenu les indices suivants: les passifs $S_{X_1} = 0,2538$; les peu actifs $S_{X_2} = 0,3032$; les actifs $S_{X_3} = 0,7914$. Donc l'indice global de participation est donné par $S_3 = (0,2538 * 0,3032 * 0,7914)^{1/3} = 0,3934$. Comme $S_3 > 0,37$, la participation ne compromet pas le niveau de qualité de réponse du SFC.

Dans le tableau de bord automatisé, une première vue présentera les indices globaux, mettant en œuvre la surveillance du SFC, et une deuxième vue présentera les indicateurs basiques associés. Ainsi, quand un indice montre une dégradation, les administrateurs pourront naviguer vers les indicateurs basiques

afin de situer le problème. Le tableau de bord sera actualisé automatiquement de façon périodique (période au choix des administrateurs). L'exploitation de ces indices globaux et des indicateurs basiques est traitée dans la section suivante.

4.3 Exploitation du tableau de bord

La dernière phase consiste à exploiter le tableau de bord. Les constats obtenus par les indicateurs seront confrontés aux objectifs choisis dès le départ, et l'écart observé va permettre de cerner le problème. Exploiter un tableau de bord consiste donc à suivre une démarche de résolution de problème. Considérer que le terme « problème » se ramène au constat d'un écart permet d'engager l'action sur des bases claires et de formuler ce même « problème » avec clarté et précision. Comment passer de...à...? Comment obtenir un taux... X ? Comment réduire le taux...? etc.

Nous proposons de suivre les étapes proposées en figure 6 pour détecter les causes des écarts et a posteriori programmer des actions de correction. Quels sont les écarts à corriger prioritairement? La réponse à cette question implique une hiérarchisation des écarts, qui s'appuie sur l'identification des priorités soulignées par les administrateurs du système. Quelles sont les origines de l'écart? Quelles sont les actions à mettre en œuvre? Et on termine par l'évaluation de ces actions.

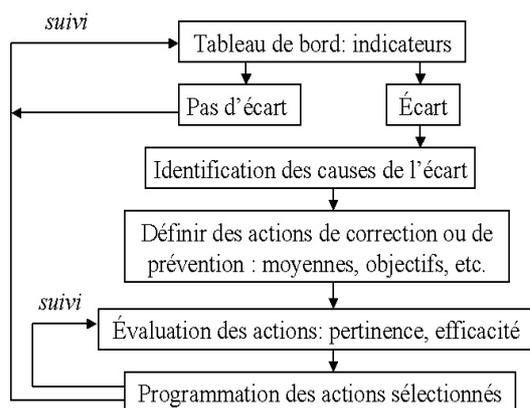


FIG. 6 Exploitation du tableau de bord

Supposons que nous venons de recevoir une alerte sur l'*indice de fréquentation* 34,23% qui n'atteint pas notre objectif de 37%. En étudiant les indicateurs basiques, nous remarquons un écart assez important entre le *taux d'absentéisme* 16,7% et notre seuil de satisfaction 10%. De même, dans une moindre mesure, pour le *taux d'abandon* (2,8% contre 2%) et le

taux de fréquentation faible (3,79% contre 3%). Ces trois indicateurs basiques sont donc responsables de la faible valeur de *fréquentation*, malgré les taux de *fréquence* 80,5% et de *fréquentation moyenne et élevée* 96,52% qui dépassent légèrement les seuils 80% et 95%. Par ailleurs, l'*indice de participation* (39,34% > 37%) montre que les utilisateurs présents participent assez. De tous ces constats, on peut conclure que l'accent doit être mis sur les utilisateurs qui sont absents, les utilisateurs se trouvant en possibilité d'abandonner le système et finalement sur les utilisateurs qui ont une fréquentation faible. Supposons maintenant qu'une des causes de l'absentéisme et de la faible fréquentation, est due au fait que le système n'informe pas suffisamment les utilisateurs de son état ou de ses activités. Nous pouvons alors définir certaines actions destinées à renseigner l'utilisateur sur un petit nombre d'indicateurs, tels que : la précision passée du SFC, un histogramme des évaluations faites par sa communauté, etc. On doit ensuite évaluer la pertinence de ces actions. Une boucle entre la mise en œuvre d'une action et son évaluation va se déclencher, jusqu'à ce que nous trouvions une action vraiment efficace, ou bien que nous appliquions une autre action. Nous envisageons de définir certaines actions type qui peuvent inspirer les administrateurs pour résoudre certains problèmes.

5. CONCLUSION ET PERSPECTIVES

Assurer la qualité d'un système de filtrage collaboratif pendant son exploitation est un sujet d'actualité qui répond au constat du manque d'utilisation à long terme de ces systèmes. Nous sommes convaincus qu'un système de contrôle pendant l'exploitation d'un SFC peut aider les administrateurs à maintenir un niveau de fonctionnalité acceptable et ainsi convaincre les utilisateurs de prolonger leur utilisation.

Nous avons développé dans cet article une méthodologie pour définir les indicateurs mesurant la « fonctionnalité » d'un SFC. Nous avons montré comment ces indicateurs, via la définition d'indices globaux, peuvent être visualisés dans un tableau de bord afin de permettre la surveillance de la qualité du SFC pendant son exploitation.

Les premières couches du système de contrôle CoFilControl ont été développées (mesure des indicateurs et visualisation sous forme

numérique dans un tableau de bord) en tant que contrôle de la plateforme de filtrage collaboratif COCoFil. Le calcul des indices globaux et leur visualisation graphique sont en cours de développement.

De nombreuses perspectives s'offrent pour concevoir un système de contrôle de la qualité d'un SFC. La méthodologie proposée ici, peut être appliquée sur des SFC à partir de critères de qualité différents, comme l'utilisabilité ; ou encore sur des processus différents de ceux traités dans cet article. Par ailleurs, nos indicateurs peuvent être adaptés à différents SFC, car ils ont été conçus de façon générique.

BIBLIOGRAPHIE

- Breese, J.S., Heckerman, D. and Kadie, C. Empirical analysis of predictive algorithms for collaborative filtering. Technical Report MSR-TR-98-12, Microsoft Research., Oct 1998.
- Cérutti, O. and Gattino, B. *Indicateurs et tableaux de bord*. AFNOR, 1992.
- CIIBA-AFNOR. *Management de la qualité du logiciel : Les référentiels*. AFNOR, 1995.
- Fidel, R. User's perception of the performance of a filtering system. *SIGIR*, pages 198–205, 1997.
- Gokhale, A. *Improvements to Collaborative Filtering Algorithms*. Thesis computer science department, WPI, May 1999.
- Goldberg, K. and Roeder, T. Eigentaste : A constant time collaborative filtering algorithm. Technical Report M00/41, UCB Electronics Research Laboratory, 2000.
- Herlocker, J., Konstant, J. and Riedl, J. Explaining collaborative filtering recommendations. In *CSCW*. ACM, 2000.
- Lueg, C. Considering collaborative filtering as groupware : Experiences and lessons learned. In *2nd International Conference on Practical Aspects of Knowledge Management (PAKM)*, 1998.
- Lueg, C., Social filtering and social reality. In *Delos Workshop on Collaborative Filtering*, Budapest, Nov 1997.
- Maltz, D. and Enhrlich, K. Pointing the way : active collaborative filtering. In *Conference on human factors in computing systems (CHI)*, Denver, Colorado USA, May 1995.
- Miller, B.N., Riedl, J.T. and Konstan, J.A. Experiences with grouplens : Making usenet useful again. In *Usenix Winter Technical Conference*, Jan 1997.
- Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. An open architecture for collaborative filtering of netnews. In *Conference on Computer Supported Co-operative Work*, pages 175–186.
- Selmer, C. *Concevoir le tableau de bord*. DUNOD, 2003.

J-RADEX¹ : UN SIMULATEUR DE SGBDTR² CONVIVIAL³

Jérôme Haubert

Etudiant Doctorant en Informatique

Jerome.Haubert@univ-lehavre.fr, +33 (0)2 32 74 43 84

Bruno Sadeg, Laurent Amanton,

Maîtres de conférences en Informatique

Bruno.Sadeg@univ-lehavre.fr, +33 (0)2 32 74 44 05

Laurent.amanton@univ-lehavre.fr, +33 (0)2 32 74 43 19

Roland Coma

Etudiant Doctorant en Informatique

Roland.Coma@univ-lehavre.fr, +33 (0)2 32 74 43 84

Adresse Professionnelle

LIH, UFR des Sciences et Techniques ★ Université du Havre
★ 25 rue Philippe Lebon ★ BP 540 ★ F-76 058 Le Havre Cedex

Résumé : L'utilisation d'un simulateur est très importante pour évaluer les performances des nouveaux protocoles. RADEX est un simulateur de SGBDTR puissant prenant en compte bon nombre des paramètres relatifs aux SGBDTR. Dans cet article, nous proposons notre contribution à l'évolution de ce simulateur pour le rendre plus convivial et pour y intégrer de nouveaux modules et concepts apparus récemment. Le but de notre contribution est de permettre à l'utilisateur de gérer un environnement de simulation correspondant à ses attentes.

Abstract : A simulator is generally used to test the performances of the new protocols. RADEX is a simulator of RTDBMSs. It is powerful since it takes into account the main parameters relative to RTDBMSs. In this article, we want to give our contribution in the evolution of this simulator by making it easier to use and by integrating some new modules and concepts. The aim is to provide the user a simulation environment corresponding to its expectation.

Mots clés : Base de donnée temps réel, simulateur, modules, interface d'entrée et de sortie, contrôle de concurrence, ordonnancement.

Key words : Real-time database, simulator, modules, input and output interface, concurrency control, scheduling.

¹Java Real-time Active Database Experimental system

²Système de Gestion de Base de Données Temps Réel

³Ce travail est subventionné par le Ministère Français de la Recherche dans le cadre d'une ACI-JC (no1055)

J-RADEx : un simulateur de SGBDTR convivial

L'utilisation d'un simulateur est très importante pour évaluer les performances des nouveaux protocoles. RADEx est un simulateur de SGBDTR puissant prenant en compte bon nombre des paramètres relatifs aux SGBDTR. Dans cet article, nous proposons notre contribution à l'évolution de ce simulateur pour le rendre plus convivial et pour y intégrer de nouveaux modules et concepts apparus récemment. Le but de notre contribution est de permettre à l'utilisateur de gérer un environnement de simulation correspondant à ses attentes.

1 - INTRODUCTION

La plupart des recherches sur les Systèmes de Gestion de Base de Données (SGBD), notamment temps réel, concerne la gestion des transactions. Les problèmes rencontrés sont alors le contrôle de concurrence, la validation des transactions distribuées ou encore l'ordonnancement des files d'attente de transactions. Pour résoudre chacun de ces problèmes, des protocoles ont été proposés. Pour les SGBD Temps Réel (SGBDTR), ces méthodes sont généralement testées et évaluées par le biais de simulations car peu de prototypes existent (Ramamritham, 1993). Chaque concepteur de protocoles évalue donc sa méthode dans un environnement de simulation particulier, ce qui peut aboutir à des interprétations différentes, voire à des résultats incompatibles (Huang, 1990).

Sivasankaran et al. (1994) ont proposé un simulateur de bases de données temps réel (*Real-time Active Database Experimental system* : RADEx) dont le modèle a ensuite été étendu par Hansson (1998). Ce simulateur est construit suivant un modèle permettant de tester des protocoles de contrôle de concurrence, des protocoles de validation, des méthodes de recouvrement, etc. En effet, chaque composant du modèle gère une partie de l'exécution des transactions sur la base de données.

Le simulateur RADEx est basé sur un modèle objet et a été écrit en langage C/C++. La première étape avant toute utilisation du simulateur est l'initialisation des paramètres de chaque gestionnaire (ou composant). Dans la version de base de RADEx, les valeurs sont lues dans un fichier et quelques paramètres et variables sont initialisés dans le fichier Makefile (qui construit le simulateur à partir des programmes sources sous Unix).

Comme on peut le constater dans (Sivasankaran, 1994), RADEx est un simulateur puissant. En effet,

l'utilisation d'un simulateur permet aux simulations d'être uniformes quel que soit le protocole testé. De plus, le modèle utilisé permet de tenir compte des principaux paramètres apparaissant dans les SGBDTR. Nous avons donc choisi de contribuer à l'évolution de ce simulateur.

Notre première contribution à cette extension est la réécriture de tous les composants de RADEx en langage Java. On augmente ainsi la portabilité du système. Cette nouvelle version de RADEx est nommée J-RADEx.

La deuxième étape de notre travail est la réalisation d'une interface graphique d'entrée et de sortie pour faciliter l'initialisation des paramètres, visualiser les résultats et simplifier l'utilisation du simulateur.

La troisième étape consiste à intégrer dans le simulateur de nouveaux protocoles de base pour la comparaison des résultats avec les nouveaux protocoles à tester.

Enfin, une quatrième étape ajoute de nouvelles notions apparues ces dernières années et pouvant être utilisées dans l'étude des performances (cf. section 3.3).

Le but de ce papier est donc de montrer les avantages conséquents du simulateur RADEx et de présenter une adaptation de celui-ci (J-RADEx) plus conviviale et plus complète. La suite du document est organisée de la façon suivante. La section 2 présente le modèle de base du simulateur. La section 3 présente le modèle utilisé pour J-RADEx, en particulier, les ajouts par rapport à RADEx. Enfin, la section 4 conclut ce document en donnant des perspectives à ce travail.

2 - LE SIMULATEUR RADEX DE BASE

Dans cette section, nous présentons l'architecture qui a servi de base à la conception du simulateur RADEx par Sivasankaran et al. (1994) (figure 1).

Initialement, le modèle utilisait une collection d'objets en multi-niveaux mais pour des raisons de simplicité, seuls deux niveaux sont utilisés. Le premier niveau définit tous les composants du modèle (les différents gestionnaires de la figure 1) et le second niveau représente les communications entre ces composants (représentées par des flèches sur la figure 1).

Le simulateur RADEx prend en compte un très grand nombre de paramètres relatifs aux SGBDTR centralisés et distribués. En effet, certains paramètres comme le nombre de CPU ou le nombre de disques

permettent de représenter la répartition des données sur plusieurs sites. On peut donc dire que le simulateur RADEx s'intéresse à au moins quatre types de SGBD : centralisé (temps réel ou non) et distribué (temps réel ou non).

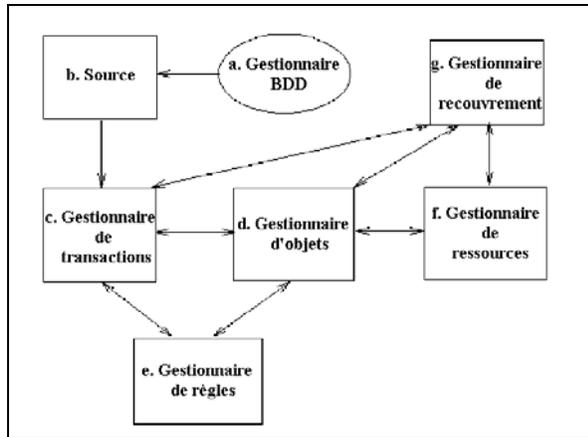


Figure 1 : Architecture du simulateur

Voyons maintenant en détail les différents composants de RADEx. Une liste exhaustive des paramètres inhérents à chaque composant est présentée dans Sivasankaran et al. (1994).

a - Gestionnaire de la base de données

Ce composant, appelé aussi *DBManager*, définit la base de données en elle-même (Sivasankaran, 1994). Par exemple, il détermine le nombre de classes d'objets dans le système, le nombre d'instances de chaque classe, etc. Chaque classe d'objets possède un certain nombre de méthodes ayant une durée d'exécution estimée et une probabilité d'exécution. La base de données est également définie par le nombre de pages disques dont elle dispose. Pour des raisons de simplicité, il a été décidé qu'à une classe d'objets correspond une page disque (Sivasankaran, 1994). Dans les SGBD distribués, le gestionnaire de la base doit notamment savoir si on autorise la duplication des données sur plusieurs sites.

Les paramètres du gestionnaire de base de données permettent donc de définir le contexte de la simulation ou, autrement dit, le contexte dans lequel vont s'exécuter les transactions (temps réel ou non).

b - Gestionnaire de la source

Le gestionnaire de la source peut être considéré comme un générateur de transactions puisqu'il simule la soumission des transactions au système (Sivasankaran, 1994). Le générateur crée régulièrement de nouvelles transactions qu'il transmet au gestionnaire de transactions. Il est donc responsable de l'initialisation des caractéristiques des transactions. On peut par exemple simuler l'exécution

de transactions apériodiques en utilisant la loi de Poisson pour le taux d'arrivée des transactions. Le gestionnaire de la source s'occupe également d'autres caractéristiques plus globales des transactions. Parmi celles-ci, on peut citer le nombre de classes de transactions, le type d'échéance,...

Nous avons donc d'un côté les informations concernant les transactions et de l'autre les informations concernant les accès aux données de ces transactions. Comme exemple de caractéristiques d'accès aux données, on peut citer la probabilité qu'une transaction modifie une donnée (*update transaction*) ou le nombre d'objets accédés par une classe de transaction donnée. Une fois ces caractéristiques déterminées, le gestionnaire de la source crée de nouvelles transactions qu'il soumet ensuite au gestionnaire de transactions.

c - Gestionnaire de transactions

L'exécution des transactions est simulée par le gestionnaire de transactions ou *Transaction Manager* (TM) (Sivasankaran, 1994). En particulier, le gestionnaire de transactions gère les événements relatifs aux transactions (*begin*, *commit* ou *abort*) qui sont transmis au gestionnaire de règles. Une autre tâche gérée par le gestionnaire de transactions est l'assignation des priorités aux transactions. C'est grâce à ce gestionnaire que l'on peut tester différentes politiques d'assignations de priorités. Une fois les priorités des transactions calculées, le gestionnaire de transactions exécute réellement la transaction en demandant au gestionnaire d'objets d'exécuter les accès aux objets en évoquant les méthodes d'accès à ces données.

Le schéma maître-esclaves (ou coordinateur-participants) est très utilisé pour représenter les SGBD distribués. Le coordinateur s'occupe de recevoir les transactions, de distribuer les sous-transactions sur les différents sites participants et de récupérer les résultats. Les participants reçoivent des parties de transactions du coordinateur qu'ils exécutent sur leur base de données locale. Les informations concernant les différents sites (coordinateur et participants) sont définies dans le gestionnaire de transactions. Mais, le travail principal du gestionnaire reste l'assignation des priorités.

d - Gestionnaire d'objets

Le gestionnaire d'objets (*Object Manager* (OM)) lance les méthodes relatives aux objets dont l'accès a été requis par le TM (Sivasankaran, 1994). Il est donc responsable du contrôle de concurrence entre les transactions. Par exemple, la méthode est exécutée (l'objet est accessible) ou un message de blocage ou d'abandon est retourné au TM (l'objet n'est pas accessible).

Le gestionnaire d'objets est relié à la fois au gestionnaire de ressources où les données sont effectivement manipulées et au gestionnaire de transactions qui le sollicite non seulement pour les accès aux données mais aussi pour la validation des transactions. En fonction de l'exécution de la transaction, le gestionnaire OM doit décider de valider (commit) ou d'abandonner (abort) la transaction. En terme de simulation, cela implique que le gestionnaire d'objets permet de tester à la fois des protocoles de contrôle de concurrence et des protocoles de validation.

e - Gestionnaire de règles

Ce gestionnaire pourrait être surnommé le contrôleur d'intégrité puisqu'il permet de vérifier que les transactions qui s'exécutent respectent les contraintes d'intégrité du système. Les transactions doivent respecter certaines règles en fonction des données auxquelles elles accèdent pour que la base reste cohérente (Delobel, 1982). Par exemple, le nombre de réservation dans un avion ne peut excéder le nombre de places. Ce module est en fait complexe et possède de nombreux paramètres.

Régulièrement, ce gestionnaire reçoit les événements des transactions en provenance du gestionnaire de transactions et du gestionnaire d'objets. Il vérifie ensuite que ces événements respectent les règles d'intégrité de la base. Si aucune règle n'est violée, un message autorise les gestionnaires TM et OM à continuer l'exécution de la transaction. Sinon, le gestionnaire de règles tente de trouver les conditions nécessaires pour que les règles soient respectées et, si de telles conditions existent, les opérations correspondantes sont transmises au TM comme s'il s'agissait d'une nouvelle transaction (Sivasankaran, 1994). Une autre tâche du gestionnaire de règles est le chargement des transactions périodiques.

f - Gestionnaire de ressources

Ce gestionnaire modélise les ressources physiques du SGBDTR (Sivasankaran, 1994). Il s'occupe notamment des pages disques, de la mémoire centrale et du (ou des) CPU. Lorsque, le gestionnaire d'objets effectue l'accès à une donnée, le gestionnaire de ressources doit vérifier que la donnée est localisée en mémoire centrale. Si ce n'est pas le cas, il doit charger la page correspondante depuis le disque en utilisant, si nécessaire, une méthode de remplacement des pages mémoires. Le gestionnaire de ressources permet donc de tester les algorithmes d'ordonnancement pour les files d'attente du processeur et des disques mais aussi des algorithmes de remplacement de pages mémoires. Les principaux paramètres de ce gestionnaire donnent donc des informations sur le nombre de processeurs, de disques, le temps d'accès disque, etc. Dans le cas

où une panne surviendrait pendant l'exécution, des échanges de messages avec le gestionnaire de recouvrement permettent de retrouver une base de données stable et cohérente.

g - Gestionnaire de recouvrement

Le gestionnaire de recouvrement n'a pas été étudié en détails dans la première version de RADEx (Sivasankaran, 1994) mais a été ajouté dans la version RADEx++ de Hansson (1998). Son rôle est la gestion des fichiers journaux qui sont des supports stables utilisés en cas de panne. Les événements des transactions sont régulièrement soumis au gestionnaire de recouvrement qui note les opérations correspondantes dans les fichiers journaux. Les politiques de recouvrement utilisées après une panne examinent ces fichiers pour rendre la base à nouveau cohérente et stable. Nous étudierons plus en détail ce gestionnaire dans des travaux futurs.

3 - LE SIMULATEUR J-RADEX

À partir du simulateur RADEx proposé par Sivasankaran et al. (1994), puis Hansson (1998), nous proposons quelques améliorations pour rendre le simulateur plus attractif et plus complet. Nous avons réécrit le simulateur en langage Java (J-RADEx), ce qui le rend plus portable. Nous avons également intégré une interface graphique pour permettre une utilisation plus aisée et plus conviviale du simulateur. On permet également à l'utilisateur de saisir les paramètres des différents composants en diminuant les risques d'erreurs. Dans un second temps, nous nous sommes intéressés à l'extension du modèle en étudiant notamment l'ajout de nouveaux protocoles et concepts récents permettant de comparer les résultats des protocoles à tester face à ces protocoles. Enfin, dans une dernière étape, nous avons ajouté de nouvelles notions introduites dans les SGBDTR ces dernières années.

3.1 - Présentation de l'interface utilisateur

Dans la version initiale du simulateur, les concepteurs utilisaient un système de fichiers pour les entrées/sorties. Le fichier d'entrée intègre l'ensemble des paramètres à initialiser et le fichier de sortie présente les résultats de la simulation. Nous allons remplacer ces fichiers par des interfaces pour une utilisation plus intuitive du simulateur.

3.1.1 - Interface d'entrée

L'interface d'entrée permet de fournir les valeurs de l'ensemble des paramètres de manière plus attractive. Deux modes de saisie des paramètres sont ainsi proposés à l'utilisateur : à partir d'un fichier (comme pour le simulateur RADEx) ou de manière interactive.

La première méthode doit d'abord vérifier le contenu du fichier avant d'initialiser les paramètres. La seconde méthode oblige l'utilisateur à entrer un par un les paramètres de chaque composant. La figure 2 illustre, par exemple, le gestionnaire de ressource et ses paramètres. Lorsque l'utilisateur a entré tous les paramètres d'un composant et qu'il valide (OK), le programme vérifie la cohérence des valeurs et informe l'utilisateur en cas d'erreur (figure 3).

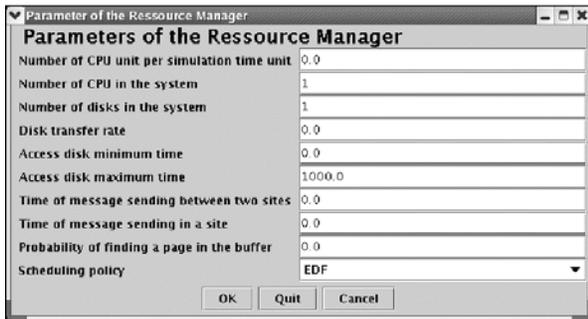


Figure 2 : Exemple de composant, le gestionnaire de ressource.

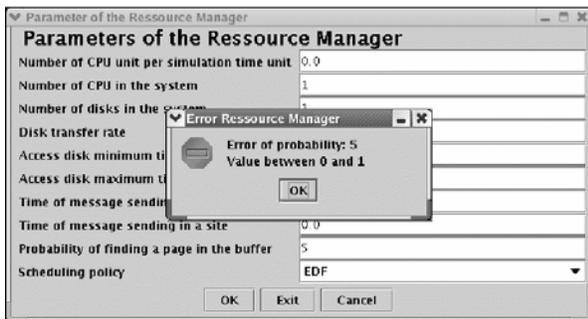


Figure 3 : Vérification des valeurs données par l'utilisateur.

L'utilisateur ne peut passer à l'étape suivante qu'une fois tous les composants initialisés. Un système de coloration des boutons permet de visualiser les composants initialisés ou non (dans la figure 4, les gestionnaires de la base et de la source sont initialisés mais pas les autres).



Figure 4 : Les gestionnaires non initialisés.

L'étape suivante est identique pour le mode graphique et le mode fichier. Pour des raisons de réutilisabilité, l'utilisateur peut sauvegarder les paramètres initialisés avant de lancer la simulation. L'avantage de ce type de méthode est que l'utilisateur peut lancer plusieurs simulations en modifiant simplement quelques paramètres et sans sortir du simulateur.

Ensuite, l'étape suivante consiste à choisir le type de protocoles qu'il désire tester (contrôle de concurrence ou validation ou autres...). La simulation peut maintenant être réalisée.

3.1.2 – Interface de sortie

Le fichier de sortie construit par RADEx est difficilement exploitable tel quel : la figure 5 présente une partie d'un fichier de sortie fourni par RADEx. Les renseignements sont fournis de manière brute et la comparaison des résultats n'est pas évidente.

```
* Run 1
Performance Averages:
=====
Miss Ratio of Class0 Transactions[13390/32014]: 0.41825
Number Of Aborts .....: 13390
Number Of Aborts on Entry: 293
Disk Work Wasted On Restarts: 0
CPU Work Wasted On Restarts: 0
Number Of Restarts .....: 16
Restarts On Validation...: 0
.....that complete: 0
[...]
Class Distributions:
1. Transaction Size (SIZ) in number of operations.
2. Number of Transactions of that size (Ns)
3. Number of Aborts for that size (As)
[...]
SIZ NS As AEs Rs Ts Ws [...]
4 1782 684 192 0 808.39 0.45 [...]
5 3933 1239 81 1 998.04 0.68 [...]
6 6397 2147 19 3 1251.97 0.99 [...]
7 7561 2984 1 3 1507.27 1.38 [...]
Normalized Miss Ratio.... : 46.251%
Normalized Restart Ratio : 0.071%
[...]
```

Figure 5 : Partie du fichier de sortie fournie par RADEx

Pour pallier ces inconvénients, nous avons élaboré une interface construisant des graphiques à partir du fichier de résultats fourni par RADEx.

Pour cela, nous avons séparé le fichier résultat en deux parties. La première partie concerne les résultats globaux de la simulation, ce qui correspond à la partie *Performance Averages* dans le fichier résultat. La seconde partie détaille les résultats suivant les classes de transactions, ce qui correspond au tableau *Class Distribution* dans le fichier.

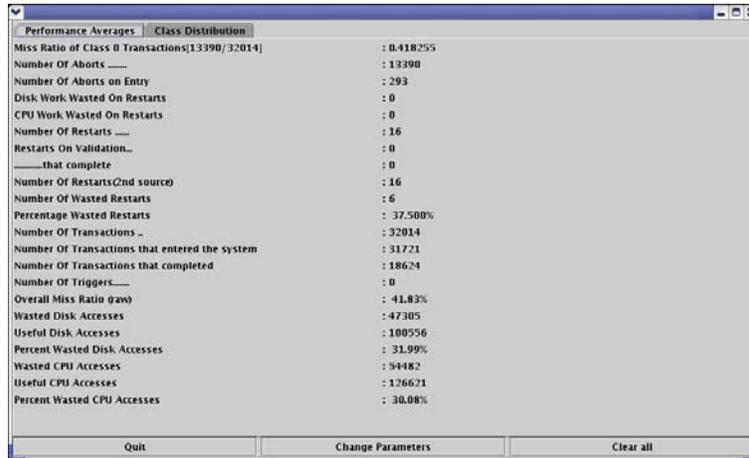


Figure 6 : Résultats globaux de la simulation

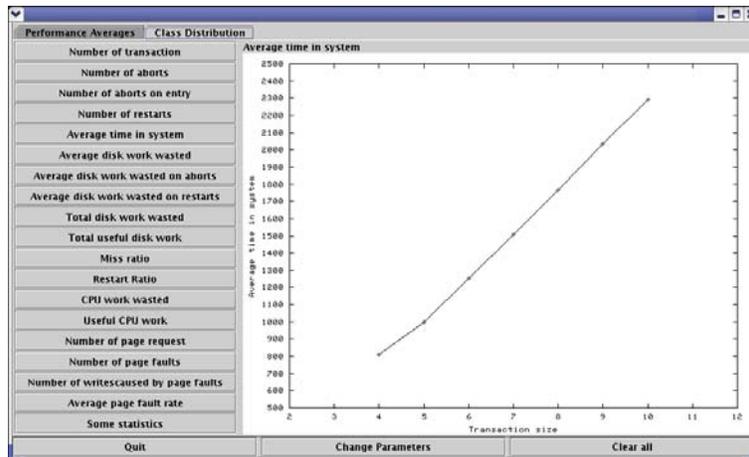


Figure 7 : Résultats en fonction des classes de transactions.

La première partie fournit simplement des résultats numériques indépendants alors que les colonnes du tableau de la deuxième partie permettent d'illustrer l'évolution des différents paramètres en fonction de la classe de transactions utilisée.

Les figures 6 et 7 illustrent respectivement les deux parties du fichier résultat. Ainsi, la figure 6 représente les résultats globaux de la simulation et la figure 7 permet de visualiser les courbes d'évolution de chaque paramètre correspondant aux différentes colonnes du tableau *Class Distribution* du fichier de sortie (les boutons de gauche permettent de passer d'un paramètre à un autre).

3.2 - Extension du modèle

La réalisation initiale de RADEx ne tient pas compte du gestionnaire de recouvrement à cause de sa complexité. En effet, ce gestionnaire implique une étude approfondie des différents types de pannes et des mécanismes de recouvrement permettant de rétablir la base. C'est pourquoi le gestionnaire de

recouvrement a été intégré au modèle étendu par Hansson (1998) : RADEx++. Ce dernier ajoute des composants et des paramètres par rapport à RADEx comme présenté en figure 8.

Dans les SGBD distribués, une notion très importante doit être prise en considération dans l'exécution de simulations : les communications. En effet, suivant les capacités du réseau (en terme de surcharge par exemple), les résultats des simulations peuvent être faussés. En général, les messages sont ordonnés sur le réseau par la méthode FCFS (*First Come First Serve*) dont les performances dans un contexte temps réel sont discutables (Cottet, 2000). Il est donc intéressant d'étudier les politiques d'ordonnement pour les appliquer aux communications et ainsi permettre à davantage de transactions de respecter leur échéance. C'est le but principal des SGBD temps réel (Ramamritham, 1993).

Les simulations comparent généralement les performances des nouveaux protocoles par rapport à ceux qui existent déjà et qui servent de base (comme

le 2PL⁴ et ses adaptations pour le contrôle de concurrence). Dans le simulateur RADEx que nous utilisons (datant de 1999), pour le contrôle de concurrence, seuls les protocoles 2PL et OCC⁵ ont été incorporés. Or, les performances de ces protocoles sont critiquables dans un contexte temps réel. Il est donc intéressant d'étudier quels sont les protocoles qui servent de base dans les SGBDTR, voire même d'intégrer régulièrement de nouveaux protocoles qui apparaissent dans la littérature.

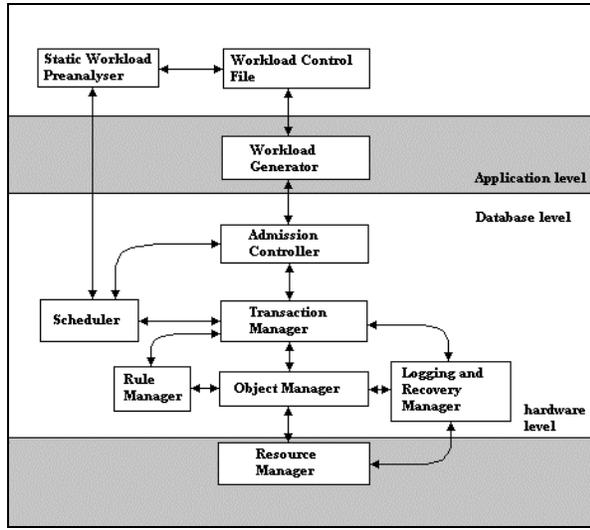


Figure 8 : Le modèle proposé par Hansson (1998)

3.3 - Introduction de nouveaux concepts

Depuis le travail d'implémentation de RADEx (1999), de nouveaux concepts sont apparus dans les SGBD distribués temps réel. Parmi ces concepts, certains sont intéressants pour évaluer les performances des nouveaux protocoles.

Pour les transactions à échéances non strictes (*soft* (Duvallet, 1999)), nous proposons d'étendre le simulateur en utilisant les notions d' ϵ -donnée et de Δ -échéance (Sadeg, 2000). Ces notions sont particulièrement adaptées aux applications multimédia et on les utilise de la façon suivante. Lorsqu'une transaction ne peut pas respecter son échéance principale, une échéance étendue lui est assignée (échéance+ Δ). Si l'échéance étendue d'une transaction ne peut pas être respectée, la transaction retourne des résultats partiels avec une imprécision ne dépassant pas ϵ . Pour intégrer ces notions au simulateur, il est nécessaire de modifier le gestionnaire de transactions, le gestionnaire de règles et le gestionnaire d'objets.

⁴ Two-Phase Locking

⁵ Optimistic Concurrency Control

Un deuxième concept que l'on peut intégrer au simulateur est le prêt des données non encore validées proposé dans le protocole de validation PROMPT (*Permits Reading Of Modified Prepared-data for Timeliness* (Haritsa, 2000)) pour les transactions distribuées. Lorsqu'un site attend la décision finale du coordinateur (*commit* ou *abort*), il peut prêter des données modifiées à d'autres transactions voulant accéder à ces données. Si la transaction "prêteuse" valide (resp. abandonne), les transactions ayant emprunté des données continuent de s'exécuter (resp. sont redémarrées). Certaines conditions ou paramètres permettent d'éviter notamment l'abandon en cascade (Haritsa, 2000). Pour intégrer cette notion au simulateur, il est donc nécessaire de modifier le gestionnaire d'objets.

Récemment, nous avons étudié des transactions distribuées particulières : les transactions pondérées (*weighted transactions* (Sadeg, 2003)). Dans ce modèle, les transactions se divisent en deux types de sous-transactions en fonction de leur poids : les sous-transactions obligatoires (ou vitales) et les sous-transactions optionnelles (ou non-vitales). Par exemple, une transaction peut se décomposer en une partie vitale qui contrôle un processus et une partie non-vitale qui affiche des résultats à l'écran. On retrouve cette notion dans les modèles de transactions étendus comme les transactions emboîtées. Des protocoles utilisant la pondération des transactions ont ainsi été proposés pour le contrôle de concurrence et/ou la validation des transactions distribuées (Sadeg, 2003). L'ajout de cette notion se fait en ajoutant de nouveaux paramètres au moment de la génération des transactions et en intégrant les protocoles de contrôle de concurrence et de validation adéquats dans les gestionnaires d'objets et de transactions.

Dans le simulateur J-RADEx, les concepts ci-dessus peuvent être considérés comme des options pour l'utilisateur lorsqu'il initialise les paramètres de la simulation. De cette façon, la simulation peut être aussi proche que possible du modèle désiré par l'utilisateur.

4 - CONCLUSION

La plupart des nouveaux protocoles sont testés par le biais de simulations. Malheureusement, ces simulations ne sont pas uniformes et peuvent avoir des résultats incompatibles dans certains cas. Il est donc intéressant de se pencher sur la réalisation d'un simulateur de SGBD (distribués et/ou temps réel). Un tel simulateur (RADEx) a été proposé par Sivasankaran et al. (1994).

Malgré ses capacités puissantes, le simulateur RADEx n'est pas très aisé à utiliser, notamment par la quantité de paramètres à initialiser pour lancer une simulation.

Les paramètres sont initialisés à partir de fichiers qui ne sont pas évident à construire : le fichier de données contient de nombreuses valeurs et l'initialisation de certains paramètres se fait dans le fichier Makefile (qui construit le simulateur à partir des programmes sources écrit en C/C++ sous Unix).

Dans ce papier, nous avons proposé d'étendre le simulateur RADEx de base (Sivasankaran, 1994) de la façon suivante : (1) en construisant une interface graphique pour la saisie des paramètres, l'affichage des résultats et plus généralement pour une utilisation plus interactive du simulateur, (2) en ajoutant de nouveaux modules comme la gestion du réseau dans un SGBD distribué et (3) en insérant de nouveaux protocoles et concepts apparus depuis la réalisation du simulateur. Pour réaliser ces modifications, nous avons choisi de réécrire le simulateur en langage Java (J-RADEx). Dans un premier temps, le simulateur de base a donc été réécrit en y intégrant l'interface graphique. Ensuite, nous l'avons augmenté pour qu'il prenne en compte les nouveaux modules et concepts.

Nos travaux futurs consistent d'une part à maintenir à jour le simulateur en y ajoutant de nouveaux concepts et protocoles qui apparaissent dans la littérature et d'autre part d'offrir des outils à l'utilisateur pour lui permettre de créer et d'intégrer lui-même ces propres concepts et protocoles. Nous pensons que le simulateur J-RADEx pourra être considéré comme un bon simulateur de SGBD distribués temps réel et qu'il sera utilisé pour tester et évaluer uniformément les nouveaux protocoles pouvant apparaître dans les différents domaines de recherche sur les SGBD distribués temps réel.

BIBLIOGRAPHIE

Cottet, F., Delacroix, J., Kaiser, C., Mammeri, Z. (2000), "Ordonnancement Temps Réel". Hermès ed.

Delobel, C., Adiba, M. (1982), "Base de données et systèmes relationnels". Dunod Informatique.

Duvallet, C., Mammeri, Z., Sadeg, B. (1999), "Les SGBD Temps Réel". *Technique et Science Informatiques*, vol. 18, n° 5, pp 479–517 (1999).

Hansson, J. (1998), "RADEx++". Technical report, University of Skövde.

Haritsa, J., Ramamritham, K., Gupta, R. (2000), "The PROMPT Real-Time Commit

Protocol". *IEEE Transactions on Parallel and Distributed Systems*, vol 11, n° 2.

Ramamritham, K. (1993), "Real-Time Databases". *Journal of Distributed and Parallel Databases*, vol 1, n° 2, pp 199-226.

Sadeg, B., Amanton, L., Haubert, J. (2003), "Trading Precision for Timeliness in Distributed Real-Time Databases". *5th International Conference on Enterprise Information Systems (ICEIS)*.

Sadeg, B., Saad-Bouzeffrane, S. (2000), "Gestion des transactions temps réel à échéance non stricte". *8th International Conference on Real-Time and Embedded Systems (RTS)*, pp 232-246. Téknéa ed.

Sivasankaran, R., Purimetla, B., Stankovic, J., Ramamritham, K., Towsley, D. (1994), "Design of RADEx - Real-time Active Database Experimental System". Technical report, University of Massachusetts.

**EXTENSION D'UN PROTOCOLE DE CONTROLE DE CONCURRENCE
DES TRANSACTIONS TEMPS REEL¹**

Jérôme Haubert,

Etudiant Doctorant en Informatique
Jerome.Haubert@univ-lehavre.fr, +33 (0)2 32 74 43 84

Bruno Sadeg, Laurent Amanton,

Maîtres de conférences en Informatique
Bruno.Sadeg@univ-lehavre.fr, +33 (0)2 32 74 44 05
Laurent.amanton@univ-lehavre.fr, +33 (0)2 32 74 43 19

Adresse Professionnelle

LIH, UFR des Sciences et Techniques ★ Université du Havre
★ 25 rue Philippe Lebon ★ BP 540 ★ F-76 058 Le Havre Cedex

Résumé : Le protocole SCC (*Speculative Concurrency Control*) est l'un des premiers protocoles de contrôle de concurrence des transactions temps réel. Il utilise la duplication des transactions et les estampilles des transactions pour résoudre les problèmes de conflits. Néanmoins, des problèmes subsistent dans son utilisation. Nous proposons une nouvelle extension de ce protocole pour résoudre ces problèmes et permettre à davantage de transactions de respecter leur échéance.

Abstract: The SCC protocol (*Speculative Concurrency Control*) is one of the first concurrency control protocol especially designed for real-time transactions. It is based on the transactions duplication and the use of the transactions timestamp to solve conflicts between transactions. Nevertheless, some problems remain in its principle. We propose a new extension of this protocol to solve these problems and to allow more transactions to meet their deadline.

Mots clés : Contrôle de concurrence temps réel, duplication, transactions fantômes, ordonnancement temps réel.

Key words : Real-time concurrency control, transaction duplication, shadow transaction, real-time scheduling.

¹Ce travail est subventionné par le Ministère Français de la Recherche dans le cadre d'une ACI-JC (n°1055)

Extension d'un protocole de contrôle de concurrence des transactions temps réel

Le protocole SCC (*Speculative Concurrency Control*) est l'un des premiers protocoles de contrôle de concurrence des transactions temps réel. Il utilise la duplication des transactions et les estampilles des transactions pour résoudre les problèmes de conflits. Néanmoins, des problèmes subsistent dans son utilisation. Nous proposons une nouvelle extension de ce protocole pour résoudre ces problèmes et permettre à davantage de transactions de respecter leur échéance.

1 - INTRODUCTION

Pour garantir la cohérence de la base de données, un gestionnaire de transactions doit résoudre les problèmes de conflits d'accès aux données. Un conflit apparaît lorsque deux transactions non encore validées désirent accéder à une même donnée avec des opérations incompatibles (écriture-lecture par exemple). Dans un Système de Gestion de Base de Données (SGBD), les problèmes de conflits sont résolus grâce à des protocoles de contrôle de concurrence (Bernstein, 1987). Ces protocoles peuvent être divisés en deux principaux groupes utilisant des méthodes duales. La première méthode, dite pessimiste, utilise des verrous pour empêcher les conflits potentiels : l'opération de lecture ou d'écriture est validée avant l'accès à la donnée. Au contraire, la seconde, dite optimiste, laisse les transactions s'exécuter en concurrence et ne vérifie la présence de conflits qu'une fois la phase de validation atteinte : les transactions en conflits sont alors abandonnées puis redémarrées selon certaines hypothèses. La section 2.2 présente plus en détail les méthodes pessimistes et optimistes pour le contrôle de concurrence.

Pour les SGBD Temps Réel (SGBDTR), le problème est plus compliqué : le SGBDTR doit respecter non seulement les contraintes d'intégrité de la base mais aussi les contraintes temporelles individuelles des transactions qui s'expriment par l'attribution d'une échéance à chaque transaction. Les transactions ne sont correctes que si elles sont validées avant leur échéance (Ramamritham, 1993). On distingue trois types de transactions temps réel selon l'importance attribuée à leur échéance (Duvallat, 1999) :

1. les transactions à échéances strictes critiques (*hard*) : une transaction qui rate son échéance peut

avoir des conséquences graves sur le système ou sur l'environnement contrôlé ;

2. les transactions à échéances strictes non critiques (*firm*) : si une transaction rate son échéance, elle devient inutile pour le système. Elle est donc simplement ignorée et abandonnée ;
3. les transactions à échéances non strictes (*soft*) : si une transaction rate son échéance, le système ne l'abandonne pas immédiatement car elle peut avoir une certaine utilité pendant un certain temps encore après l'expiration de son échéance, mais la qualité de service qu'elle offre est moindre.

Dans ce contexte, des adaptations des protocoles de contrôle de concurrence existant dans les SGBD non temps réel sont nécessaires pour tenir compte des échéances des transactions. De nombreux protocoles ont ainsi vu le jour (Abbott, 1988, Haritsa, 1992). Il s'agit généralement d'ajouter aux protocoles classiques un ordonnancement temps réel basé sur les échéances des transactions e.g. le protocole 2PL-HP² (Abbott, 1988).

Des études de performances ont montré que suivant le type de transactions temps réel utilisé, les protocoles de contrôle de concurrence pessimistes et optimistes ne fournissent pas les mêmes résultats. Pour les transactions de type *firm*, les protocoles optimistes offrent de meilleures performances grâce à leurs propriétés : non-bloquants et libres de tout interblocage (Haritsa, 1990). Mais ces résultats deviennent discutables lorsque les transactions sont de type *soft* (Huang, 1990) : le redémarrage des transactions et la possibilité d'exécution des transactions après échéances entraînent rapidement une surcharge du système.

Dans ce papier, nous nous intéressons à un protocole de contrôle de concurrence temps réel conçu spécialement pour les SGBDTR : le protocole SCC (*Speculative Concurrency Control*) proposé par Bestavros (1992). Ce protocole combine les avantages des protocoles pessimistes et optimistes puisqu'il détecte les conflits dès leur apparition (méthode pessimiste) mais il laisse les transactions s'exécuter en concurrence (méthode optimiste). Le protocole SCC est particulièrement adapté aux SGBDTR dans le sens où il réduit l'impact négatif du blocage et du redémarrage qui sont les inconvénients principaux des

² *Two Phase Locking High Priority.*

méthodes pessimistes et optimistes respectivement (Bestavros, 1992).

Des études ont montré que les performances de ce protocole sont meilleures que celles des méthodes pessimistes et optimistes classiques (Bestavros, 1993), même si les premières versions du protocole SCC ne prenaient pas en compte les échéances des transactions. Ensuite, une extension du protocole prenant en compte les échéances et la criticité des transactions a été proposée (Bestavros, 1995).

Dans ce papier, nous proposons une nouvelle extension du protocole SCC permettant de résoudre les deux principaux problèmes qui subsistent dans son utilisation. (1) Nous proposons une autre résolution des conflits de type w-w (*write-write*) et (2) nous proposons un ordonnancement temps réel des transactions pour augmenter les chances des transactions de respecter leur échéance.

Ce document est structuré de la façon suivante. La section 2 expose en détail le protocole SCC (et une de ses extensions : *value-cognizant SCC*) et montre les principaux problèmes de ces protocoles. La section 3 présente notre nouvelle méthode de résolution de conflits. Cette nouvelle méthode est discutée en section 5. Enfin, la section 6 conclut ce travail en donnant quelques pistes de recherche.

2 - LE PROTOCOLE SCC

L'intérêt principal du protocole SCC proposé par Bestavros (1992) est qu'il s'agit d'un protocole conçu spécialement pour les SGBDTR. Ses performances résident dans le fait qu'il combine les avantages des méthodes pessimistes et optimistes tout en se débarrassant de leurs principaux inconvénients.

2.1 - Notations

Dans la suite de ce papier, nous noterons T_i les transactions qui s'exécutent en concurrence. Chaque transaction peut être dupliquée et la transaction fantôme de T_i est notée T_i' . Les opérations effectuées par les transactions sont S (*Start*) pour démarrer, Rx (*Read x*) et Wx (*Write x*) pour lire et mettre à jour la donnée x . C et A sont utilisés lorsque la transaction est validée (*commit*) ou abandonnée respectivement.

2.2 – Méthodes pessimistes et méthodes optimistes

Les méthodes pessimistes empêchent les conflits avant même qu'ils n'apparaissent en posant des verrous sur les données avant tout accès (figure 1-a). Quand une transaction désire accéder à une donnée, elle doit d'abord obtenir un verrou sur cette donnée. Un verrou peut être posé sur une donnée si et seulement si la donnée n'est pas déjà verrouillée par une autre transaction avec un mode incompatible (lecture et écriture par exemple). Si le mode est

incompatible, la transaction requérant le verrou doit attendre que la donnée soit libérée. L'avantage de cette méthode est que les conflits sont détectés dès leur apparition, ce qui permet de garantir la validité de tous les accès aux données. Dans un contexte temps réel, ce type de protocoles possède l'inconvénient majeur d'être un protocole bloquant : les transactions peuvent attendre un verrou pendant un temps indéterminé.

Les méthodes optimistes, de leur côté, laissent les transactions s'exécuter en concurrence et ne vérifient l'apparition de conflits qu'une fois la phase de validation atteinte (figure 1-b). En fait, quand une transaction désire effectuer son opération de *commit*, elle doit vérifier qu'aucun conflit n'est apparu. On utilise pour cela une méthode de certification (Kung, 1981). Les conflits sont détectés soit avec les transactions en cours d'exécution, c'est la certification en avant, soit avec les transactions déjà validées, c'est la certification en arrière. Il est clair que la certification en arrière est difficilement applicable dans un contexte temps réel puisque seule la transaction validante peut être abandonnée en cas de conflit. En revanche pour la certification en avant, on peut utiliser les priorités des transactions pour déterminer le blocage ou l'abandon éventuel de certaines transactions. On peut citer comme exemple le protocole OCC-BC³ proposé par Haritsa et al. (1992). L'avantage de ce type de méthodes est l'absence de blocage dû aux verrous et l'exécution des transactions en concurrence. Par contre, le redémarrage des transactions peut entraîner une surcharge du système qui peut, à son tour, empêcher des transactions de se terminer avant échéance.

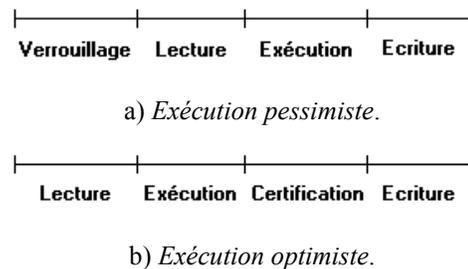


Figure 1 : Différentes phases des exécutions des transactions selon le type de protocole.

En ce qui concerne le protocole SCC, les conflits sont détectés dès leur apparition comme pour les méthodes pessimistes mais les transactions continuent à s'exécuter en concurrence comme dans les méthodes optimistes.

³ *Optimistic Concurrency Control with Broadcast Commit*

2.3 - Principe du protocole SCC

Le protocole SCC a été proposé pour résoudre les problèmes de conflits de type r-w et w-r. Les conflits de type w-w sont résolus par la méthode TWR⁴ (Bernstein, 1987, Bestavros, 1994). Cette méthode est basée sur les estampilles des transactions. L'hypothèse est la suivante : seule l'opération d'écriture de la transaction de plus grande estampille (la transaction la plus jeune) sera visible. Avec cette hypothèse, lorsqu'une transaction T_1 désire écrire sur une donnée alors qu'une transaction plus jeune T_2 a déjà écrit sur cette donnée, l'écriture de T_1 est simplement ignorée.

En ce qui concerne les autres types de conflits (r-w et w-r), le protocole SCC propose de dupliquer la transaction en lecture dès qu'un conflit est détecté. La copie de la transaction principale est appelée transaction "fantôme" (ou *shadow transaction*). On a alors deux transactions identiques à un détail près, la transaction principale continue de s'exécuter avec l'*image avant* de la donnée alors que la transaction fantôme reste bloquée avec l'*image après* de la donnée. Deux scénarios peuvent se produire :

1. La transaction qui accède à la donnée en lecture (T_2) se termine et valide avant que la transaction en écriture (T_1) ne soit terminée : la transaction fantôme (T_2') est alors simplement abandonnée (figure 2-a).
2. T_1 se termine et valide avant T_2 : T_2 doit être abandonnée. Sa transaction fantôme est alors libérée et s'exécute avec l'*image après* de la donnée (figure 2-b). La transaction est alors redémarrée depuis le point de conflit et non depuis le début de la transaction ce qui augmente ses chances de se terminer avant échéance.

Ces exemples simples montrent l'exécution de deux transactions. Mais, avec le protocole SCC de base, une transaction fantôme est créée à chaque apparition de conflit. En limitant le nombre de transactions fantômes par transaction principale, on obtient une classe de protocole SCC- kS (*k-shadow SCC*) où k désigne le nombre de transactions dupliquées autorisées par transaction, i.e. $k-1$ transactions fantômes (Bestavros, 1992). Parmi cette classe de protocoles, le protocole le plus étudié est le SCC-2S dans lequel on autorise une transaction fantôme par transaction principale. Dans ce cas, les conflits de type r-w et w-r doivent être considérés séparément :

1. Si une transaction T_2 désire accéder à une donnée en lecture alors que cette donnée est déjà accédée en écriture par une autre transaction T_1 (figures 2-a et 2-b), alors T_2 est dupliquée et la transaction fantôme est bloquée. La transaction fantôme sera débloquée si et seulement si la

transaction principale doit être abandonnée pour résoudre un conflit.

2. Si T_1 désire accéder à une donnée en écriture alors que T_2 accède déjà à la donnée en lecture, la transaction fantôme T_2' de T_2 doit être reconsidérée. Si le point de conflit courant est placé dans la transaction fantôme (figure 3), celle-ci doit être redémarrée depuis le début et est bloquée au nouveau point de conflit. Sinon, la transaction fantôme reste inchangée : le nouveau point de conflit sera pris en compte lorsque T_2' sera libérée. Si T_2 n'a pas de transaction fantôme, T_2' est créée et elle ré-exécute la transaction depuis le début jusqu'au point de conflit.

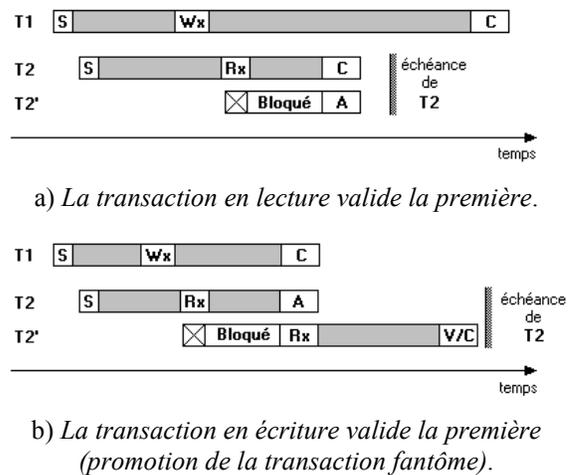


Figure 2 : Le protocole SCC.

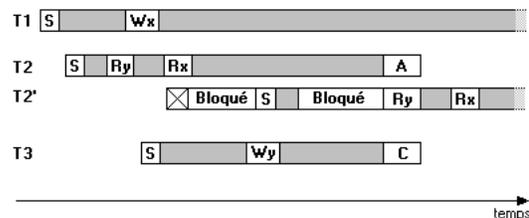


Figure 3 : Redémarrage et promotion de la transaction fantôme

Il est à noter que lorsque la transaction fantôme est libérée, elle devient transaction principale et peut être elle-même dupliquée.

2.4 - Une extension de SCC : Value-cognizant SCC

Bestavros et al. (1995) ont ensuite proposé une extension du protocole SCC de base qui prend en compte les échéances et la criticité des transactions. Ces paramètres permettent de calculer un coefficient de pénalité (*penalty gradient*). Les coefficients des transactions sont utilisés lors de la phase de validation

⁴ Thomas Write Rule

pour déterminer dans quelle mesure l'opération de *commit* peut être différée. Par exemple, considérons la figure 4 où la situation est initialement la même que sur la figure 2-b. Si on diffère l'opération de *commit* de T_1 , on peut laisser la transaction T_2 s'exécuter jusqu'à validation et ensuite valider T_1 . Le conflit se résout alors de lui-même. Les coefficients des transactions permettent donc de définir des heuristiques pour différer l'opération de *commit* des transactions si nécessaire.

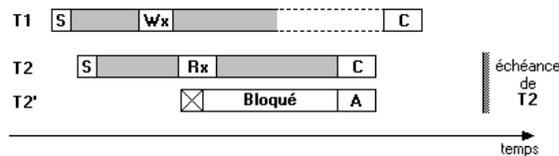


Figure 4 : Exemple d'exécution du protocole *Value-cognizant SCC*.

Les détails et les performances de ce type de protocoles peuvent être trouvés dans (Bestavros, 1995), mais nous pouvons simplement dire qu'il s'applique aux transactions à échéances non strictes (*soft*) et que les coefficients ne sont utilisés qu'au moment de la validation. Nous pensons que les échéances des transactions peuvent être utilisées dès la détection des conflits pour augmenter les chances des transactions de respecter leur échéance.

2.5 - Discussion de la méthode spéculative

Dans cette section, nous discutons l'utilisation du protocole SCC pour le contrôle de concurrence des transactions temps réel. Pour cela, nous allons différencier les types de conflits : les conflits de type r-w et w-r sont résolus de manière similaire alors que les conflits de type w-w sont résolus séparément.

Dans un premier temps, considérons les conflits de type w-w. Ils sont résolus par la méthode TWR qui utilise les estampilles des transactions pour éventuellement ignorer des opérations d'écriture (cf. section 2.1). Dans un contexte temps réel, l'hypothèse sous-jacente à la méthode TWR est difficilement applicable. En effet, si les estampilles des transactions représentent leur date d'arrivée dans le système, une transaction plus ancienne (T_1) avec un temps d'exécution très long et loin de son échéance peut être en conflit avec une transaction plus jeune (T_2) mais dont l'échéance est plus proche. Dans ce cas, TWR peut ignorer l'opération d'écriture de T_1 . Le problème n'est pas résolu de manière correcte puisque T_2 a plus de chance de se terminer (valider ou abandonner) avant T_1 et cette dernière doit alors réécrire sur la donnée.

Autrement dit, la méthode TWR perd la mise à jour de T_1 . Ceci est difficilement concevable dans les SGBDTR. En effet, les résultats des transactions

temps réel sont importants dès qu'elles ont effectué leur opération de *commit* (Ramamritham, 1993). Ainsi, la résolution des conflits w-w en utilisant les estampilles des transactions ne semble pas adaptée aux SGBDTR : TWR souffre du problème de perte de mise à jour.

Considérons maintenant les autres types de conflits (r-w et w-r). Avec le protocole SCC, si l'échéance de T_1 est plus proche que celle de T_2 , on peut aboutir à la situation présentée en figure 5 : la transaction T_2 ne peut pas respecter son échéance parce que l'opération de *commit* de T_1 ne peut pas être différée et le temps restant à T_2 pour s'exécuter n'est pas suffisant. Si les échéances des transactions sont de type *soft*, seule la qualité des résultats de T_2 va diminuer (Bestavros, 1995, Duvallat, 1999). Mais si les transactions sont de type *firm* (voire *hard*), T_2 ne fournit aucun résultat et peut engendrer des conséquences plus ou moins graves.

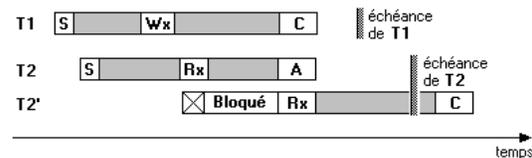


Figure 5 : Limites du protocole SCC.

En fait, la résolution des conflits du protocole SCC de base ne tient pas compte des priorités des transactions (échéance et/ou criticité). Une extension de ce protocole a été proposée mais les priorités des transactions ne sont prises en considération qu'au moment de la validation (*value-cognizant SCC*). Nous pensons que les priorités des transactions doivent être considérées dès la détection des conflits pour donner plus de chances aux transactions de respecter leur échéance.

Dans la section suivante, nous proposons une nouvelle extension du protocole SCC, nommée ESCC (*Extended SCC*) qui résout les deux problèmes que nous venons d'exposer (1) en utilisant une nouvelle technique de résolution des conflits w-w et (2) en ordonnant les transactions temps réel dès la détection d'un conflit.

3. EXTENSION DU PROTOCOLE SCC

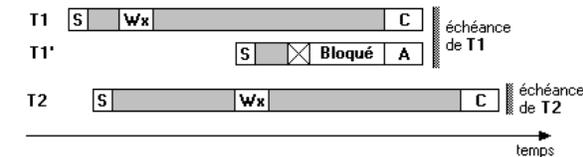
3.1 - Conflits w-w

Nous avons montré dans la section précédente que la résolution des conflits basée sur les estampilles des transactions fournit des résultats discutables dans les SGBDTR. Dans ce cas, la méthode TWR, souffrant du problème de perte de mise à jour, n'est pas adaptée. Pour les conflits w-w, nous proposons d'utiliser la même notion que pour les autres types de

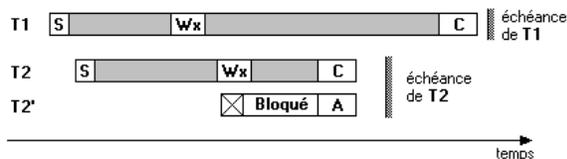
conflits : la duplication des transactions. De cette façon, une transaction fantôme est créée dès l'apparition d'un conflit. Le nombre de transactions fantômes par transaction peut être limité de la même manière que pour le protocole SCC-2S (Bestavros, 1992).

Examinons la possibilité d'appliquer la duplication des transactions aux conflits w-w. Considérons la figure 6. Lorsqu'un conflit de type w-w apparaît, on duplique la transaction dont l'échéance est la plus proche. En effet, si cette dernière doit être redémarrée, alors elle aura moins de chance de respecter son échéance. Deux cas se présentent alors :

- Soit le point de conflit se situe avant le début de la transaction fantôme (figure 6-a) : la transaction fantôme doit être redémarrée et se bloque au nouveau point de conflit.
- Soit le point de conflit est l'instant courant (figure 6-b) : il suffit de bloquer la transaction fantôme à l'instant courant.



a) T_1 est la plus prioritaire



b) T_2 est la plus prioritaire

Figure 6 : ESCC appliqué aux conflits w-w.

Ainsi, la duplication des transactions peut s'appliquer à tous les types de conflits.

3.2 – Autres conflits

La figure 5 montre les limites du protocole SCC pour résoudre les problèmes de conflits r-w et w-r entre des transactions concurrentes. En effet, une seule des transactions se termine avant échéance. Existe-t-il une méthode permettant aux deux transactions de se terminer avant leur échéance ?

Pour cela, nous avons besoin de faire une nouvelle hypothèse : une transaction peut lire les résultats d'une autre transaction non encore validée. Nous relaxons ainsi l'isolation, l'une des propriétés ACID⁵ des transactions. Nous discuterons cette hypothèse dans la section suivante. Ainsi, si T_1 et T_2 sont deux

⁵ Atomicité, Cohérence, Isolation, Durabilité

transactions accédant à une même donnée respectivement en écriture et en lecture, notons d_1 et d_2 leur échéance respective. Un conflit entre ces transactions implique la création d'une transaction fantôme pour T_2 : T_2' . Le protocole ESCC propose d'ordonner T_2 et T_2' en fonction des échéances de T_1 et T_2 de la façon suivante :

1. Si $d_1 > d_2$, alors T_1 a de grandes chances de se terminer avant T_2 (figure 2-a). L'écriture de T_1 n'a alors pas d'impact sur le déroulement de T_2 . De manière optimiste, on peut donc laisser T_2 se dérouler comme dans le protocole SCC de base. Dans le cas où T_1 se termine avant T_2 , on peut utiliser le protocole *value-cognizant SCC* pour différer l'opération de commit de T_1 et laisser T_2 se terminer en premier.
2. Si $d_1 < d_2$, alors on peut penser raisonnablement que T_1 va se terminer avant T_2 et les modifications qu'elle apporte devront être considérées par T_2 . L'exécution de SCC peut mener à la situation de la figure 5. C'est pourquoi on choisit ici de bloquer la transaction principale et d'exécuter la transaction fantôme (figure 7). Autrement dit, la transaction en lecture s'exécute avec l'*image après* de la donnée. Les deux transactions ont ainsi plus de chances de se terminer et de valider avant leur échéance.

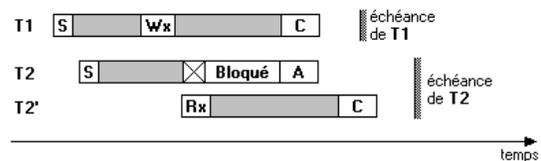


Figure 7 : ESCC appliqué aux conflits r-w et w-r.

En résumé, le protocole ESCC propose un ordonnancement des transactions (principales et fantômes) en fonction des échéances des transactions en conflit. Le protocole ESCC, en combinant la duplication des transactions avec un ordonnancement temps réel des transactions, permet donc à davantage de transactions de respecter leur échéance.

4 - DISCUSSION SUR LA METHODE

Comme nous l'avons signalé dans la section précédente, le protocole ESCC proposé ne respecte pas la propriété d'isolation des transactions. Cette propriété précise que les résultats d'une transaction ne sont visibles des autres transactions qu'une fois la transaction terminée. Dans un contexte temps réel, cette propriété peut être relaxée (Ramamritham, 1993) dans le sens où l'isolation peut différer l'exécution de certaines transactions et les empêcher de respecter leur échéance. Si on relaxe la propriété d'isolation, alors les transactions peuvent accéder à des données

non encore validées et peuvent ainsi fournir des résultats même partiels avant échéance. Autrement dit, dans un contexte temps réel, il peut être plus important d'obtenir des résultats partiels avant échéance plutôt que des résultats complets après échéance (Ramamritham, 1993, Gupta, 1996).

Le fait de relaxer l'isolation des transactions oblige à tenir compte de la validité des données. Lorsqu'une transaction T_1 accède à des données déjà utilisées par une autre transaction T_2 , la validation de T_1 est alors dépendante de celle de T_2 . Si T_2 se termine correctement et valide, alors T_1 peut continuer normalement son exécution. Mais, dans le cas où T_2 doit être abandonnée, T_1 doit également être abandonnée puisqu'elle a manipulé des données devenues invalides. Des heuristiques sont utilisées pour limiter les conditions d'accès aux données non encore validées (Gupta, 1996). Par exemple, on peut éviter l'abandon en cascade en ne permettant pas à une transaction accédant déjà à des données non validées d'autoriser à nouveau l'accès aux données qu'elle manipule. On peut également empêcher une transaction proche de son échéance de prêter des données pour limiter le nombre de redémarrage des transactions.

Une autre discussion que l'on peut faire sur le protocole ESCC est le nombre de transactions qu'implique la duplication. En effet, avec N transactions et en se limitant à une transaction fantôme par transaction principale, $2N$ transactions peuvent coexister. Mais, comme le montre la discussion précédente, la plupart du temps, une transaction et sa transaction fantôme ne s'exécutent pas en même temps. La transaction fantôme est souvent bloquée pendant que la transaction principale s'exécute sauf dans le cas où la transaction fantôme est redémarrée. On peut donc penser que la surcharge entraînée par cette méthode est négligeable par rapport aux performances en termes de respect des échéances des transactions.

Enfin, il est à noter que la méthode de résolution des conflits w-w du protocole ESCC est plus restrictive que la méthode TWR. En effet, avec le protocole ESCC, toutes les opérations d'écritures doivent être effectivement réalisées (aucune opération ne peut être ignorée). Cette restriction va probablement limiter les performances du protocole ESCC, mais rappelons que cette nouvelle méthode ne souffre pas du problème de perte de mises à jour.

5 - CONCLUSION

Dans cet article, nous nous sommes intéressés à un protocole de contrôle de concurrence (SCC) proposé par Bestavros et al. (1992, 1993, 1995) et qui est spécifique aux transactions temps réel. Notre étude de

ce protocole montre que malgré ses bonnes performances dans les SGBDTR, il possède quelques inconvénients, notamment pour les conflits de type w-w ou encore dans la prise en compte des échéances des transactions. En effet, il semble que le protocole SCC soit plutôt adapté aux transactions à échéances non strictes (*soft*), mais ses performances décroissent pour les transactions à échéances strictes (critiques ou non). Nous avons donc proposé quelques améliorations pour permettre au protocole SCC de pallier ces problèmes et permettre à davantage de transactions de se terminer avant échéance.

Les améliorations que nous proposons sont basées sur une nouvelle méthode de résolution des conflits de type w-w et la prise en compte des échéances des transactions dès la détection d'un conflit. Le protocole SCC amélioré est appelé ESCC. En fait, le protocole ESCC propose d'utiliser à la fois la duplication des transactions et un ordonnancement temps réel des transactions basé sur leur échéance pour résoudre tous les types de conflits. Des simulations sont actuellement en cours pour évaluer les performances du protocole ESCC par rapport au protocole SCC mais aussi pour évaluer la limitation apportée par la résolution de conflits w-w du protocole ESCC par rapport à la méthode TWR.

Nos travaux portent sur les SGBD distribués temps réel. La répartition des données implique de gérer à la fois le contrôle de concurrence localement sur chaque site mais aussi la validation globale des transactions distribuées. Le protocole ESCC comme le protocole SCC peuvent s'utiliser localement sur chaque site pour le contrôle de concurrence mais il faut tenir compte de la validation globale des transactions. Parmi les protocoles de validation, certains ne sont pas adaptés au contexte temps réel soit parce qu'ils sont bloquants soit à cause du grand nombre de messages échangés entre le coordinateur et les participants (Haritsa, 2000). La validation des transactions distribuées temps réel reste un domaine de recherche ouvert dans les SGBDTR distribués.

BIBLIOGRAPHIE

- Abbott R.K., Garcia-Molina, H. (1988), "Scheduling Real-Time Transactions: A Performance Evaluation". In *14th Very Large Data Bases Conference*, pp 1-12, Loas Angeles, California.
- Bernstein, P., Hadzilacos, V., Goodman, N. (1987), "Concurrency Control and Recovery in Database Systems". Addison-Wesley.

- Bestavros, A. (1992), "Speculative Concurrency Control". Technical Report TR-16-92, Boston University, Boston, MA.
- Bestavros, A., Braoudakis, S., Panagos, E. (1993), "Performance Evaluation of Two-Shadow Speculative Concurrency Control". Technical Report 1993-001, Boston University, Boston, MA.
- Bestavros, A., Braoudakis S. (1994), "Timeliness via Speculation for Real-Time Databases". In *14th IEEE Real-Time System Symposium*, Puerto Rico.
- Bestavros, A., Braoudakis, S. (1995), "Value-cognizant Speculative Concurrency Control". In *21st VLDB Conference*, Zurich, Switzerland.
- Duvallet, C., Mammeri, Z., Sadeg, B. (1999), "Les SGBD Temps Réel". *Technique et Science Informatiques*, vol 18, n° 5, pp 479–517.
- Gupta, R., Haritsa, J. (1996) "Commit Processing in Distributed Real-Time Database Systems". In *Natl. Conference on Software for Real-Time Systems*, Cochin, India.
- Haritsa, J., Carey, M., Livny, M. (1990), "On Being Optimistic about Real-Time Constraints". In 9th ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems.
- Haritsa, J. Carey, M., Livny, M. (1992), "Data Acces Scheduling in Firm Real-Time Database Systems". *Real-Time Systems*, vol 4, n°3, pp 203-241.
- Haritsa, J., Ramamritham, K., Gupta R. (2000), "Real-Time Commit Processing". In *Real-Time Database Systems*, pp 227–243. Kluwer Academic Publishers.
- Huang, J., Stankovic, J. (1990), "Concurrency Control in Real-Time Database System: Optimistic Scheme vs. Two-Phase Locking". Technical Report UM-CS-1990-066, University of Massachusetts.
- Kung, H., Robinson, J. (1981), "On Optimistic Methods for Concurrency Control". In *ACM Trans. on Database System*, vol 6, n° 2, pp 213-226.
- Ramamritham, K. (1993), "Real-Time Databases". *Journal of Distributed and Parallel Databases*, vol 1, n° 2, pp 199–226.

APPROCHE CONNEXIONNISTE DE LA PERCEPTION DU MOUVEMENT

Claudio CASTELLANOS SÁNCHEZ,

Doctorant en informatique à l'Université Henri Poincaré, Nancy, France
Boursier mexicain du Conseil National de Science et Technologie (CONACyT, 113287/169076),
México.

Tél.: + 33 3 83 59 20 56

Courriel : castella@loria.fr

Site InterNet: <http://www.loria.fr/~castella>

Bernard GIRAU,

Maître de conférences en informatique à l'Université Nancy 2, Nancy, France

Tél. : + 33 3 83 59 30 92

Courriel : girau@loria.fr

Frédéric ALEXANDRE,

Directeur de Recherche INRIA

Tél. : +33 3 83 59 20 53

Courriel : falex@loria.fr

Adresse professionnelle

LORIA INRIA-Lorraine, équipe Cortex ★ Campus Scientifique BP 239 ★ 54506 Vandœuvre-lès-Nancy ★ France

Résumé: La perception du mouvement visuel modélisé par des modèles connexionnistes fournit différents axes de recherche pour le développement de modèles de perception-action en temps réel appliqués à la perception visuelle dynamique du mouvement. Nous présentons dans cet article les bases d'une approche connexionniste bio-inspirée avec une adaptation particulière des filtres de Gabor spatio-temporels que nous utilisons en vue d'une approche plus modulaire et fortement localisée pour la perception visuelle du mouvement.

Summary : The visual perception of motion modeled by connectionist models offers various areas of research for the development of models of real-time perception-action applied to the dynamic visual perception of motion. We present in this article the bases of a bio-inspired connectionist approach with a particular adaptation of the spatiotemporal Gabor filters that we use within a more modular and strongly localized approach for the visual perception of motion.

Mots clés : Perception du mouvement, réseau de neurones bio-inspirés, flot optique, filtres de Gabor.

Keywords : Perception of motion, bio-inspired neural network, optical flow, Gabor filters.

Approche connexionniste de la perception du mouvement

1 – INTRODUCTION

Le traitement et l'interprétation de la perception du mouvement visuel a une importance croissante en robotique autonome. Les modèles connexionnistes y ont apporté leur puissance de généralisation et leur robustesse au manque d'information. Leur parallélisme intrinsèque conjugué à un traitement local nous fournit différents axes de recherche pour le développement de modèles de perception-action en temps réel appliqués à la perception visuelle dynamique du mouvement.

Nous proposons un rapide survol de l'état de l'art en neurophysiologie, des algorithmes proposés pour l'obtention du flot optique, et de quelques réalisations matérielles. Nous présentons ensuite les bases d'une approche connexionniste bio-inspirée avec une adaptation particulière des filtres de Gabor spatio-temporels. Finalement, nous envisageons une approche connexionniste plus modulaire et fortement localisée pour la perception visuelle du mouvement.

2 – ÉTAT DE L'ART

2.1 - Point de vue biologique

La perception du mouvement chez les animaux est indispensable pour survivre aux prédateurs en détectant leur forme 3D et leur profondeur d'une part, et en réalisant des jugements de coïncidence pour estimer la direction et la vitesse de déplacement d'autre part.

On sait actuellement que la détection et l'analyse du mouvement sont obtenues par une cascade d'opérations neuronales [Sekuler, 2002], à savoir: l'enregistrement des signaux de mouvement local dans des régions restreintes du champ visuel et leur intégration pour décrire de façon générale la direction et la vitesse des mouvements.

A - Le parcours des signaux depuis la rétine

De la rétine jusqu'au cortex cérébral de l'être humain, dix-sept régions différentes interviennent [Sunaert, 1999]. On peut organiser les principales en quatre étapes:

l'acquisition et la compression des signaux lumineux au niveau de la rétine, leur intégration binoculaire et temporelle au niveau du chiasma optique et des corps genouillés latéraux, respectivement; leur analyse corticale au niveau de l'aire visuelle primaire, et leur traitement secondaire dans les aires 20 et 21 du cortex temporal et l'aire 7 du cortex pariétal.

Dans tout ce chemin que parcourent les signaux lumineux on peut détecter deux voies [Rao, 2003]: la voie ventrale ou chemin cortical occipito-temporal (incluant les aires visuelles V1, V2 et V4 et l'aire inféro-temporale IT), qui passe par les cellules parvocellulaires, responsables de la perception des objets et de leur forme, et la voie dorsale ou chemin occipito-pariétal (incluant les aires visuelles V1, V2 et V3, l'aire temporelle moyenne -MT/V5- et l'aire pariétale postérieure PP), qui passe par les cellules magnocellulaires, responsables de la perception du mouvement et de l'espace. Nous nous sommes particulièrement intéressés à la voie dorsale et aux aires spécialisées dans l'analyse du mouvement.

La rétine répond fortement aux mouvements des contours, aux points lumineux et aux grilles, mais ses réponses ne sont sélectives ni à la direction ni à la vitesse de mouvement. Ces dernières caractéristiques sont issues des neurones de l'aire visuelle primaire.

B - L'analyse du mouvement dans le cerveau

Les zones du cerveau humain spécialisées dans la perception du mouvement sont [Sekuler, 2002]: l'aire visuelle primaire (V1) qui réagit au mouvement vacillant, l'aire temporelle moyenne (MT) et l'aire supérieure temporelle moyenne (MST) qui sont spécialisées dans la détection du mouvement cohérent autant que l'aire occipitale kynétique (KO) l'est pour les contours définis du mouvement, et, finalement, le sulcus supérieur temporel (STS) qui répond aux formes de mouvement dépendantes du mouvement biologique. Nous nous sommes notamment intéressés aux deux premières aires.

a - La détection de la direction et de l'orientation

V1 et MT semble extraire les formes du flot global [Bremmer, 1999] et MST les formes particulières (radiales, rotationnelles et leurs combinaisons) [Lappe, 1996]. Et, comme MST reçoit une entrée vestibulaire depuis MT, il semble être impliqué dans la perception du mouvement propre.

La première analyse corticale est effectuée dans V1 en assurant la sensibilité aux contrastes lumineux par des champs récepteurs allongés. Organisée en six couches de bas en haut, et en bandes de gauche à droite [Alexandre, 1990], les neurones de cette aire sont organisés en colonnes d'orientation (qui tourne progressivement de 10° d'une colonne à l'autre) et de dominance oculaire. Ces neurones émettent principalement leurs prolongements dans la direction verticale et sont accordés à une direction préférentielle de mouvement [Hubel, 1962] effectuant une analyse locale de l'énergie du mouvement qualifiée de filtre. Ces « filtres » enregistrent la présence des caractéristiques fondamentales des objets bougeant autour de leurs champs récepteurs.

Ces cellules sélectives à l'orientation spatiale peuvent être modélisées par des filtres spatio-temporels [Adelson, 1985] et leurs champs récepteurs comme un produit des interactions inhibitrices et excitatrices dans l'espace et dans le temps.

Toutes ces cellules sont sélectives à une direction préférentielle, à une fréquence spatiale, peu sensibles à la composition de la longueur d'onde du stimulus du mouvement et limitées à une extraction locale du mouvement. Or, ces cellules ne voient qu'une partie de la scène, leur réponse est ambiguë car pour chaque direction il existe une famille de cellules sélectives répondant aux mêmes changements de luminosité spatio-temporelles qu'une cellule locale et qui génère la même réponse.

Pour autant, le mouvement local de l'image rétinienne est extrait par les neurones de V1 ayant un champ récepteur semblable à une petite fenêtre spatialement délimitée où ils peuvent enregistrer la présence de mouvement dans une direction déterminée. Ce traitement fortement localisé et récurrent est notre source d'inspiration vers un modèle bio-inspiré pour la détection et l'estimation du mouvement.

b - La détection de la direction et de la vitesse

La perception visuelle du mouvement n'est pas complètement déterminée par les réponses locales dans le champ récepteur des neurones, elles sont de plus traitées pour obtenir la vitesse après les avoir collectées et combinées depuis V1 et en les regroupant dans MT. C'est cette combinaison des signaux qui résout l'ambiguïté des réponses individuelles des neurones.

MT intègre les signaux de mouvement issus de V1 sur des régions plus étendues de l'espace visuel, et construit une représentation rétinotopiquement organisée de l'espace visuel. Presque tous ses neurones sont sélectifs à la direction et à la vitesse d'un stimulus de mouvement qu'ils calculent de façon locale, explicite et non-ambiguë [Perrone, 2001]. Les champs récepteurs de ces neurones sont plus grands que ceux de V1 et augmentent avec l'excentricité. En outre, un neurone de MT répondra mieux si les signaux du mouvement sont renforcés par la luminosité, la couleur ou la texture. Par conséquent, ces neurones ont une certaine invariance car ils enregistrent l'information du mouvement par eux-mêmes. Ces neurones sont accordés à une direction préférentielle et activent un processus compétitif pour aboutir à un vote non-ambigu d'une vitesse particulière. Ce processus compétitif est dépendant de la fréquence spatiale, de la forme, du contraste et de la disparité des objets [Bradley, 1995].

La différence des réponses des neurones de MT et de ceux de V1 est que certains neurones de MT reçoivent en entrée les réponses des neurones de V1 avec différentes directions préférentielles et, comme une grande partie des neurones de MT répondent à une direction de mouvement donnée indépendamment de l'orientation du stimulus, cette aire élimine l'ambiguïté de la direction du mouvement. Ces neurones fonctionnent comme si leurs champs récepteurs étaient une combinaison de multiples sorties des neurones de V1 [Movshon, 1996].

Finalement, les neurones de MST reçoivent les réponses des neurones de MT. Ils sont sélectifs au mouvement complexe comme l'expansion et la contraction [Britten, 1998] et aussi sensibles à l'information non-rétinienne comme le mouvement des yeux.

D'autre part, la détection du contraste est suffisante pour l'identification de la direction du mouvement, de telle façon que les mécanismes visuels extrayant le mouvement sont étiquetés par la direction [Derrington, 1993].

2.2 - Point de vue mathématique

La détection et l'estimation du mouvement sont des analyses de type bas niveau. Par contre, sa segmentation et son interprétation sont des analyses de type haut niveau. Notre approche reste dans le premier type.

A - Le flot optique

Le flot optique est « la distribution 2D de vitesses apparentes pouvant être associées à la variation de la luminosité des formes de l'image » [Horn, 1980]. Il cherche alors à apparier les pixels de même intensité.

L'estimation du flot optique a été proposée comme un pas de pré-traitement dans plusieurs algorithmes de vision de haut niveau [Galvin, 1999].

En général, pour obtenir le flot optique on récupère, d'abord, toute l'information localement disponible de la vitesse (en utilisant un petit voisinage spatio-temporel) pour tout le champ visuel en réalisant un pré-filtrage (passe-bas ou passe-bande) en vue d'extraire la structure d'intérêt du signal et de rehausser la proportion signal-bruit. Ensuite, on extrait des mesures (les dérivées spatio-temporelles ou les surfaces de corrélation, par exemple) et on les intègre soit par régularisation, soit par mise en correspondance, soit par le calcul des moindres carrés. Finalement, on propage la vitesse des régions disposant d'une information complète vers celles n'ayant qu'une information partielle ou nulle de façon à récupérer le champ de flot optique correct [Barron, 1994].

En accord avec la conservation de l'intensité lumineuse et son équation [Horn, 1980] :

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \quad (1)$$

où $I(x,y,t) = I(x+\delta x, y+\delta y, t+\delta t)$ est la luminosité sur le point (x, y) du plan de l'image et en supposant qu'elle reste constante au temps t , on peut se rendre compte des principes fondamentaux de quelques approches d'estimation du flot optique [Galvin, 1999]: par mise en correspondance, basées sur le gradient,

stochastiques et spatio-temporelles (fréquentielles). Nous nous sommes surtout intéressés à cette dernière approche.

B - Des approches pour estimer le flot optique

Dans les approches de mise en correspondance, on suppose la conservation locale de la distribution d'intensité, laquelle est filtrée, tout d'abord, par un filtre passe-bande, puis traitée selon un schéma hiérarchique. Pour cela, on fait un appariement entre deux images consécutives d'une séquence donnée. Cette méthode définit un déplacement (lequel est une approximation de la vitesse) comme le meilleur ajustement parmi les régions voisines variant dans le temps [Anandan, 1989].

Dans les approches basées sur le gradient, on suppose, tout d'abord, la conservation de l'intensité de l'image dans un petit intervalle de temps et, en conséquence, on a une équation linéaire avec deux inconnues. Ensuite, on utilise le résultat d'une fonction de l'intensité de l'image plutôt que l'intensité de l'image elle-même. Finalement, on calcule la vitesse des dérivées spatio-temporelles des intensités de l'image en supposant que le domaine de l'image est continu (différentiable) dans l'espace et dans le temps [Nagel, 1987].

Les approches stochastiques (de relaxation) sont basées sur une contrainte concernant les discontinuités du mouvement. Elles manipulent les occlusions en modélisant explicitement les intensités discontinues par des processus stochastiques [Heitz, 1993] ou en utilisant les filtres de Kalman [Black, 1992].

Dans les approches fréquentielles, les fréquences spatio-temporelles sont mises en relation avec la vitesse du stimulus de mouvement, et le flot optique devient l'identification d'un plan d'énergie dans l'espace de la fréquence spatio-temporelle. On a trouvé que les mécanismes sensibles au mouvement basés sur l'énergie orientée spatiale et temporellement dans l'espace de Fourier pouvaient estimer le mouvement dans les endroits où les autres approches échouent. Ainsi, la détection du mouvement dans l'image revient ici à extraire l'orientation spatio-temporelle [Adelson, 1985 ; Heeger, 1987].

On connaît deux sous-approches, fondées sur la phase ou sur l'énergie. Dans le premier cas, on définit les composantes de la vitesse en termes de mouvement instantané au niveau de

la phase des contours de la sortie des filtres passe-bande de Gabor accordés à la vitesse [Fleet, 1990]. Dans le deuxième cas, ces filtres orientés spatio-temporellement peuvent être construits sur des filtres séparés, i.e. les filtrages spatiaux et temporels peuvent être appliqués séquentiellement à l'image d'entrée [Adelson, 1985]. Pour l'obtention de la vitesse, on aurait un schéma de codification de la population où la vitesse est exprimée comme le rapport des réponses des filtres d'énergie accordés aux différentes vitesses. En supposant une linéarité, les réponses de chaque filtre changent avec les différents contrastes mais leur rapport reste constant.

Cette approche est donc équivalente à l'approche de mise en correspondance [Adelson, 1986 ; Simoncelli, 1993]. L'utilisation des filtres spatio-temporels orientés qui sont sensibles à des stimuli en mouvement (comme les filtres de Gabor) a pour origine la recherche concernant la vision des mammifères, qui avait mis en évidence la présence de cellules simples dans l'aire corticale V1 [McLean-Palmer, 1985] et de cellules complexes dans l'aire MT [Newsome, 1983] se comportant comme des filtres passe-bande spatio-temporels. La réponse de ces cellules dépend de la vitesse d'un objet en mouvement ainsi que de sa forme et de ses caractéristiques fréquentielles [Hammond, 1981].

Parmi les avantages des méthodes énergétiques on pourrait énoncer [Spinei, 1998]: un lissage intrinsèque qui réduit les effets du problème d'ouverture, une meilleure robustesse au bruit et des résultats de qualité pour des séquences d'images naturelles. Par contre, un coût élevé en calcul et le manque de précision à proximité des frontières du mouvement en sont ses défauts.

Les principes de l'approche spatio-temporelle consistent à localiser le plan d'énergie non nul ou le plan de vitesse correspondant à un mouvement de translation 2D apparent sur un support spatial délimité par le biais de filtres spatio-temporels orientés.

2.3 – Modèles bio-inspirés

Étant donné que l'aire V1 est spécialisée dans la détection et l'estimation du mouvement visuel, et que MT et MST sont spécialisés dans sa segmentation et son interprétation, plusieurs chercheurs ont essayé de s'inspirer de leur

fonctionnement pour créer des modèles bio-inspirés.

Simoncelli et Heeger [Simoncelli, 1998] ont proposé un modèle séquentiel pondéré, rectifié et normalisé des aires V1 en avant et MT en aval.

Zemel et Sejnowski [Zemel, 1998] ont proposé un modèle bidirectionnel de MSTd segmentant les champs de mouvement des différents objets mobiles selon les formes cohérentes du mouvement de chaque objet. Tout d'abord, ils ont utilisé un pré-traitement tiré de Wang et al. [Wang, 1989], puis la modélisation de MT pour l'obtention du flot optique par l'algorithme de Nagel [Nagel, 1987], et finalement ils utilisent au niveau de MST une couche cachée qui est sélective aux formes.

Pack et al. [Pack, 2000] ont proposé un modèle qui distribue les cellules de MT et de MST en unités inhibées ou excitées avec une structure auto-récurrente pour MST en aval. Ils ont modélisé MT et MST en couples: MT → MSTd (calcul de la direction supposée, vitesse perçue, excitation aux petits mouvements et inhibition aux grands mouvements) et MT+ → MSTv (calcul du mouvement du fond par compétition, excitation et inhibition contraires au premier couple).

2.4 – Solutions matérielles

D'autre part, quelques chercheurs ont proposé des solutions matérielles comme Tsao et Chen [Tsao, 1994] qui ont proposé un schéma neuronal pour le calcul du flot optique fondé sur les filtres de Gabor et une méthode généralisée du gradient. Leur architecture VLSI est composée de trois couches tendant à estimer le mouvement comme un processus rétinien (et de LGN) effectué par un ensemble de filtres de Gabor lissant et rejetant le bruit de l'information mais conservant les composantes de la fréquence orientée spatialement dans les différentes directions et dans un canal déterminé (comme on peut voir dans les systèmes biologiques).

Stocker [Stocker, 2001] a proposé un système composé de deux réseaux: un pour la segmentation du mouvement et un autre pour la sélection du mouvement. Le premier cherche les discontinuités du mouvement et restreint le processus d'estimation collective du flot optique aux unités recevant l'information visuelle d'un même objet dans l'espace. Le

deuxième fournit une perception sélective du mouvement visuel en accord avec un mouvement préférentiel donné. Ces réseaux sont analogiques et récurrents. Le premier est de type « winner-take-all » (WTA) et le deuxième est entièrement connecté. Il modélise la perception visuelle du mouvement comme un problème d'optimisation.

Pirim [Pirim, 1998] a valorisé un système qui détermine l'existence et la localisation d'une zone en déplacement ainsi que sa vitesse et sa direction orientée. Le système a trois étapes: la première effectue un traitement temporel, la deuxième fait un traitement spatial suivant l'algorithme temps-échelle de Mallat [Pirim, 1998] et la troisième fait un calcul d'histogrammes pour trouver le maximum et les limites inférieures et supérieures de la zone du mouvement. Les paramètres de résolution spatiale et de la structure de l'image sont le résultat d'une analyse par des ondelettes de Gabor orientées décrites par Daugman [Daugman, 1988].

3 – APPROCHE RETENUE

En vue d'une réalisation bio-inspirée, connexionniste, fortement locale et embarquable sur un robot mobile qui suit une cible bougeant, nous proposons dans cet article une adaptation des filtres de Gabor spatio-temporels pour modéliser les cellules magnocellulaires qui peuvent être vues comme des senseurs de mouvement pour l'extraction des gradients spatio-temporels généralisés de l'intensité de l'image et de sa dérivée temporelle [Pollen, 1981].

3.1 - Les fondements biologiques

Il y a deux types de calcul dans les cellules magnocellulaires: la différentiation temporelle et l'intégration spatiale, sachant que « la vision périphérique est presque totalement impliquée dans la détection des dérivées temporelles et une grande partie de la bande passante dans le nerf optique est utilisée par l'information des dérivées temporelles » [Mead, 1989].

Le mouvement visuel est ressenti comme des stimuli visuels de mouvement, notamment des contrastes de luminosité orientés spatialement, extraits par les différents champs récepteurs des cellules magnocellulaires dans le cortex visuel. Les opérations mathématiques réalisées par ces champs récepteurs peuvent être

simplifiées par des intégrations spatiales pondérées des signaux.

En voyant les cellules corticales comme des détecteurs de caractéristiques, quelques modèles permettent un principe d'attente et de comparaison [Adelson, 1985], c'est-à-dire, premièrement, on extrait les caractéristiques visuelles d'une séquence d'images et, deuxièmement, on calcule le mouvement visuel basé sur le déplacement spatial de ces caractéristiques.

Les champs récepteurs de type filtres spatiaux 2D de Gabor proposés par Marcelja [Marcelja, 1980] ont été étendus à des filtres spatio-temporels 3D et appliqués comme des analyseurs spectraux locaux spatio-temporels. Ces champs ont des fonctions pondérées de lissage distinctes des masques simples pour la détection de barres. En outre, ils sont passe-bande et sélectifs à l'orientation mais ils ne sont pas spécifiques aux caractéristiques. En conséquence, ils ne sont ni de bons analyseurs spectraux ni de bons détecteurs de barres ni de bons détecteurs de contours. Ce sont des champs récepteurs passe-bande sélectifs à l'orientation.

3.2 - Le filtre de Gabor adapté

Soit $I(x,y,t)$ la forme de l'intensité dans l'image variant dans le temps en supposant que chaque point a une luminosité invariante. Soit aussi $I(x,y,t) = I(x - ut, y - vt)$ où (u,v) est le vecteur de mouvement d'une petite région de l'image, et où $I(x,y)$ est la trame d'image de l'échantillonnage au temps $t=0$. Grâce à la supposition d'une fréquence d'échantillonnage suffisamment haute pour assurer la localité des détections de mouvement, nous pouvons faire l'hypothèse d'une vitesse locale instantanée constante.

En appliquant alors le filtre de Gabor orienté, $G_\theta(x,y)$ avec $0 \leq \theta < 2\pi$, à $I(x,y)$ nous obtenons:

$$\left. \frac{d}{dt} \int \int G_\theta(\hat{x}-\hat{u}, \hat{y}-\hat{v}) dx dy \right|_{t=0} = \frac{d}{dt} \int \int I(x-ut, y-vt) G_\theta(\hat{x}-\hat{u}, \hat{y}-\hat{v}) dx dy \Big|_{t=0} \quad (2)$$

où

$$\hat{x}=(x-\xi)\cos\theta-(y-\eta)\sin\theta, \hat{y}=(x-\xi)\sin\theta+(y-\eta)\cos\theta \quad (3)$$

avec $(\xi, \eta) \in \Gamma$ un petit voisinage autour de (x,y) , et

$$\hat{u} = \frac{t-t'}{\tau-1} v_i \cos \theta, \quad \hat{v} = \frac{t-t'}{\tau-1} v_i \sin \theta \quad (4)$$

pour τ images consécutives et une vitesse v_i supposée.

(\hat{x}, \hat{y}) est l'endroit où le signal de Gabor orienté va être calculé et

$$G(x, y) = \exp\left(-\frac{x^2 + \gamma^2 y^2}{\sigma^2}\right) \cos\left(2\pi \frac{x}{\lambda} + \phi\right) \quad (5)$$

est la fonction de réponse à l'impulsion du filtre de Gabor modélisant la fonction des cellules ganglionnaires magnocellulaires où γ est l'excentricité du champ récepteur et σ sa taille; λ la longueur d'onde et ϕ la phase, tous d'après [Petkov, 1997].

La mesure obtenue par un seul filtre de Gabor ne peut pas déterminer le vecteur du mouvement 2D, il faut mettre un ensemble de filtres de Gabor ne différant qu'en orientation. Ils sont alors placés dans un vecteur appelé vecteur de senseurs de mouvement où chaque orientation est un senseur de mouvement.

De plus,

$$\frac{d}{dt} \left[\iint I(x-ut, y-vt) G(\hat{x}-\hat{u}, \hat{y}-\hat{v}) dx dy \right] \Big|_{t=0} = \frac{d}{dt} \left[\iint I(x, y) G(\hat{x}-\hat{u}-ut, \hat{y}-\hat{v}-vt) dx dy \right] \Big|_{t=0}$$

$$= - \iint I(x, y) (\nabla G(\hat{x}-\hat{u}, \hat{y}-\hat{v}) \vec{v}) dx dy \Big|_{t=0} = - \vec{v} \cdot \iint I(x, y) \nabla G(\hat{x}-\hat{u}, \hat{y}-\hat{v}) dx dy \quad (6)$$

où $\vec{v}=(u, v)$ est la vitesse 2D de la région Γ .

En notant

$$\bar{\Omega}_x(\hat{x}, \hat{y}) = - \vec{v} \cdot \iint I(x, y) \nabla G(\hat{x}-\hat{u}, \hat{y}-\hat{v}) dx dy,$$

$$\bar{\Omega}_x(\hat{x}, \hat{y}) = \iint I(x, y) \frac{\partial G_x}{\partial x} G(\hat{x}-\hat{u}, \hat{y}-\hat{v}) dx dy,$$

$$\bar{\Omega}_y(\hat{x}, \hat{y}) = \iint I(x, y) \frac{\partial G_y}{\partial y} G(\hat{x}-\hat{u}, \hat{y}-\hat{v}) dx dy \quad (7)$$

on peut récrire la dernière équation comme:

$$\bar{\Omega}_x(\hat{x}, \hat{y}) + u \bar{\Omega}_x(\hat{x}, \hat{y}) + v \bar{\Omega}_y(\hat{x}, \hat{y}) = 0 \quad (8)$$

pour une vitesse v_i à un instant t . Voilà l'adaptation fréquentielle fondée sur l'énergie de l'équation 1.

Daugman [Daugman, 1989] avait suggéré que la fonction sélective de la direction des neurones dans MT est représentée de façon binaire par un réseau neuronal WTA. Dans ce cas, les neurones de MT sont bizarrement accordés à différentes vitesses et directions.

Nous appliquons simplement un WTA aux différentes orientations pour chaque vitesse supposée.

4 - RÉALISATIONS

Nous avons testé notre version adaptée des filtres de Gabor spatio-temporels sur des séquences d'images naturelles et synthétiques. La figure 1 montre les filtres d'amplitude maximale obtenus dans chacune des 8 directions (en fixant les paramètres libres des équations 2, 3, 4 et 5) ainsi que leur somme vectorielle, dans le cas de séquences d'images synthétiques où une boule s'approche ou s'éloigne dans un couloir rectiligne.

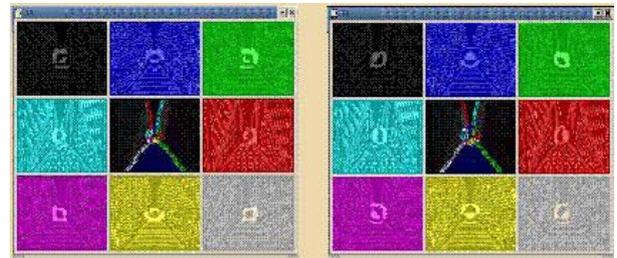


Fig. 1 - Perception du mouvement par filtres de Gabor adaptés. L'image du centre est la somme vectorielle des huit directions avec leur couleur respective et $t=3$, $v \in \{-2, -1, 0, 1, 2\}$ dans une image 2D de 5×5 et une taille de pas de 1×1 . Gauche : Boule s'éloignant. Droite : Boule s'approchant.

Nos travaux actuels consistent à inclure cette approche dans une architecture connexionniste utilisant un ensemble d'interactions latérales excitatrices et inhibitrices inspirées des colonnes corticales issues de V1, avec un schéma de connectivité fortement local et un rayon d'influence qui dépend de la réponse locale obtenue via l'équation 2. Notre but est d'arriver à proposer un modèle connexionniste complètement modulaire et fortement localisé en nous inspirant des résultats neurophysiologiques tirés des études du chemin des cellules magnocellulaires.

5 - CONCLUSION

La perception du mouvement est vitale chez l'être vivant. D'après les résultats neurophysiologiques de la vision chez les primates, nous avons exposé que: la rétine joue un rôle de filtrage des signaux lumineux, les corps genouillés latéraux font une intégration binoculaire, V1 une analyse spatiale, MT et MST une analyse temporelle. La forte réduction de l'information depuis la rétine et la grande quantité des interactions latérales dans

les aires supérieures font émerger la dynamique de notre perception du monde. Ces différents niveaux de traitement et leurs rétroactions respectives illustrent les coopérations existant entre colonnes rétinotopiquement proches, et que nous exploitons dans nos architectures connexionnistes. À ce niveau, nous poursuivons nos travaux afin d'intégrer les différentes étapes de la voie dorsale que parcourent les signaux lumineux avec un point de vue modulaire, purement connexionniste et fortement localisé.

BIBLIOGRAPHIE

- [Adelson, 1985] Adelson, E. H. and Bergen, J., Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A*, 2(7):284-299 (1985).
- [Adelson, 1986] Adelson, E. H. and Bergen, J., The extraction of spatiotemporal energy in human and machine vision. In *Proc. IEEE Workshop on Visual Motion*, pages 151-156, Charleston (1986).
- [Alexandre, 1990] Alexandre, F., Une modélisation fonctionnelle du cortex: la Colonne Cortical. Aspects visuels et moteurs. PhD thesis, Henri Poincaré, Nancy, France (1990).
- [Anandan, 1989] Anandan, P., A computational framework and an algorithm for the measurement of visual motion. *International Journal Computer Vision*, 2:283-310 (1989).
- [Barron, 1994] Barron, J. J., Fleet, D. J., and Beauchemin, S. S., Performance of optical flow techniques. In *Proceedings Conference on Computer Vision and Pattern Recognition*, pp. 236-242, Champaigne (1994).
- [Black, 1992] Black, M. J., Robust incremental optical flow. PhD thesis, Yale University, Yale (1992).
- [Bradley, 1995] Bradley, D., Quian, N., and Andersen, R., Integration of motion and stereopsis in middle temporal cortical area of macaques. *Nature*, 373:609-611 (1995).
- [Bremmer, 1999] Bremmer, F. and Lappe, M., The use of optical velocities for distance discrimination and reproduction during visually simulated self-motion. *Experimental Brain Research*, 127(1):33-42 (1999).
- [Britten, 1998] Britten, K. H. and van Wezel, R. J., Electrical micro-stimulation of cortical area MST biases heading perception in monkeys. *Nature Neuroscience*, 1:59-63 (1998).
- [Daugman, 1988] Daugman, J. G., Complete Discrete 2-D Gabor Transforms by Neural Networks for Image Analysis and Compression. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1169-1179 (1988).
- [Daugman, 1989] Daugman, J. G., Networks for images analysis: Motion and texture. In *IJCNN* (1989).
- [Derrington, 1993] Derrington, A. M. and Henning, G. B., Detecting and discriminating the direction of motion of luminance and color gratings. *Visual Research*, 33 (1993).
- [Fleet, 1990] Fleet, D. and Jepson, A. D., Computation of component image velocity from local phase information. *International Journal Computer Vision*, 5:77-104 (1990).
- [Galvin, 1999] Galvin, B., McCane, B., Novins, K., Mason, D., and Mills, S., Recovering Motion Fields: An Evaluation of Eight Optical Flow Algorithms. Technical report, Computer Science Department (1999).
- [Hammond, 1981] Hammond, P. and Reck, J., Influence of velocity on directional tuning of complex cells in cat striate cortex for texture motion. *Neuroscience Letters*, 19:309-314 (1981).
- [Heeger, 1987] Heeger, D. J., A model for the extraction of the image flow. *Journal of the Optical Society of America A*, 4(8):1455-1471 (1987).
- [Heitz, 1993] Heitz, F. and Bouthemy, P., Multimodal estimation of discontinuous optical flow using Markov random fields. *IEEE Transactions on Pattern Analysis and*

- Machine Intelligence, 15(12):1217-1232 (1993).
- [Horn and Schunk, 1980] Horn, B. K. P. and Schunk, B. G. Determining optical flow. Technical Report A. I. Memo No. 572, Artificial Intelligence Laboratory (1980).
- [Hubel, 1962] Hubel, D. H. and Weisel, T. N., Receptive fields, binocular interaction and functional architecture in the cats visual cortex. *Journal Physiology*, 160:106-154 (1962).
- [Lappe, 1996] Lappe, M., Bremmer, F., Pekel, M., Thiele, A., and Hoffmann, K. P., Optical flow processing in monkey STS: A theoretical and experimental approach. *The Journal of Neurosciences*, pages 6265-6285 (1996).
- [Marcelja, 1980] Marcelja, S., Mathematical description of the responses of simple cortical cells. *Journal of the Optical Society of America A*, 70/11:1297-1300 (1980).
- [McLean-Palmer, 1985] McLean-Palmer, J., Jones, J., and Palmer, L., New degrees of freedom in the structure of simple receptive fields. *Investigative Ophthalmology and Visual Science Supplement*, 26:265 (1985).
- [Mead, 1989] Mead, C. A., editor (1989). *Analog VLSI and neural systems*. Addison-Wesley, Reading, MA.
- [Movshon, 1996] Movshon, J. A. and Newsome, W. T., Visual response properties of striate cortical neurons projecting to area MT in macaque monkeys. *The Journal of Neurosciences*, 23:7733 (1996).
- [Nagel, 1987] Nagel, H. H., On the estimation of optical flow: Relation between different approaches and some new results. *Artificial Intelligence*, 33:299-324 (1987).
- [Newsome, 1983] Newsome, W. T., Gizzi, M. S., and Movshon, J. A., Spatial and temporal properties of neurons in macaque MT. *Investigative Ophthalmology and Visual Science Supplement*, 24:106 (1983).
- [Pack, 2000] Pack, C., Grossberg, S., and Mingolla, E., A Neural Model of Smooth Pursuit Control and Motion Perception by Cortical Area MST. Technical Report CAS/CNR-TR-99-023, Department of Cognitive and Neural Systems and Center for Adaptive Systems, Boston (2000).
- [Parker, 1998] Parker, A. J. and Newsome, W. T., Sense and the single neuron: Probing the physiology of perception. *Annual Review of Neuroscience*, 21:227-277 (1998).
- [Perrone, 2001] Perrone, J. and Thiele, A., Speed skills: measuring the visual speed analyzing properties of primate MT neurons. *Nature Neuroscience*, 4(5):526-532 (2001).
- [Petkov, 1997] Petkov, N. and Kruizinga, P., Computational models of visual neurons specialized in the detection of periodic and aperiodic oriented visual stimuli: bar and grating cells. *Biological Cybernetics*, 76:83-96 (1997).
- [Pirim, 1998] Pirim, P., Procédé et dispositif fonctionnant en temps réel pour le repérage et la localisation d'une zone en mouvement relatif dans une scène, ainsi que pour la détermination de la vitesse et de la direction du déplacement. Technical Report WO 98/05002, France (1998).
- [Pollen, 1981] Pollen, D. and Ronner, S., Phase relationships between adjacent simple cells in the visual cortex. *Science*, 212:1409-1411 (1981).
- [Rao, 2003] Rao, H., Zhou, T., Zhuo, Y., Fan, S., and Chen, L., Spatiotemporal Activation of the Two Visual Pathways in Form Discrimination and Spatial Location: A Brain Mapping Study. *Human Brain Mapping*, 18:79-89 (2003).
- [Sekuler, 2002] Sekuler, R., Watamaniuk, S. N. J., and Blake, R., Motion Perception. *Steven's Handbook of Experimental Psychology*, 1:121-176 (2002).
- [Simoncelli, 1993] Simoncelli, E. P., Distributed analysis and representation of visual motion. PhD thesis, Massachusetts

Institute of Technology, Cambridge,
M. A. (1993).

- [Simoncelli, 1998] Simoncelli, E. P. and Heeger, D. J., A Model of Neural Responses in Visual Area MT. *Visual Research*, 38(5):743-761 (1998).
- [Spinei, 1998] Spinei, A., Estimation du mouvement par triades de filtres de Gabor: Application au mouvement d'objets transparents. PhD thesis, Laboratoire des Images et des Signaux (1998).
- [Stocker, 2001] Stocker, A. A., Constraint Optimization Networks for Visual Motion Perception - Analysis and Synthesis. PhD thesis, Swiss Federal Institute of Technology Zürich, Zürich, Switzerland (2001).
- [Sunaert, 1999] Sunaert, S., van Hecke, P., Marchal, G., and Orban, G. A., Motion-responsive regions of the human brain. *Experimental Brain Research*, 127:355 (1999).
- [Tsao, 1994] Tsao, T.-R. and Chen, V., A neural scheme for optical flow computation based on Gabor filters and generalized gradient method. *Neurocomputing*, 6:305-325 (1994).
- [Wang, 1989] Wang, H. T., Mathur, B., and Koch, C., Computing optical flow in the primate visual system. *Neural Computation*, 1:92-103 (1989).
- [Watson, 1985] Watson, A. B. and Jr., A. J. A., Model of human visual motion sensing. *Journal of the Optical Society of America A*, 2(2):322-341 (1985).
- [Zemel, 1998] Zemel, R. S. and Sejnowski, T. J., A Model for Encoding Multiple Object Motions and Self-Motion in Area MST of Primate Visual Cortex. *The Journal of Neurosciences*, 18(1):531-547 (1998).

**CLASSIFICATION AUTOMATIQUE DE COMPORTEMENTS D'ÉLÈVES
APPRENANT L'ALGÈBRE**

David Renaudie,

Doctorant en informatique, équipe Apprentissage

david.renaudie@imag.fr, + 33 4 76 57 48 13

Adresse professionnelle

Laboratoire Leibniz-IMAG, 46 av. Félix Viallet, 38031 Grenoble Cedex 1

Résumé : Nous nous intéressons à l'analyse de comportements d'élèves effectuant des exercices d'algèbre, basée sur des données recueillies lors de l'utilisation d'un EIAH (Environnement Informatique pour l'Apprentissage Humain) pour l'algèbre, le logiciel Aplusix. Toutes les actions des élèves ont été enregistrées et constituent notre corpus de données initial. Par l'application de techniques de classification non supervisée à ces données préalablement « vectorisées », nous construisons des catégories d'élèves dont la réponse aux problèmes posés est comportementalement homogène. Nous présentons ici les résultats obtenus avec les différents prétraitements employés, ainsi que les variables de représentation retenues lors de l'analyse automatique de données réelles.

Summary : We focus on analysing the behaviors of students learning algebra, based on data gathered while using a learning environment software called Aplusix. Every single action performed by each student (key pressed, mouse moves, etc...) was recorded during sessions organized in French high schools, and constitute our base corpus. By applying unsupervised classification techniques to the data, which is pre-processed into a vector set, we discover clusters of students which have homogeneous behaviors. We present here the results obtained by using various pre-processing techniques and feature sets during the automatic analysis of real data.

Mots clés : représentation de l'information, fouille de données, classification non supervisée, clustering, EIAH.

CLASSIFICATION AUTOMATIQUE DE COMPORTEMENTS D'ÉLÈVES APPRENANT L'ALGÈBRE

Nous nous intéressons à l'analyse de comportements d'élèves effectuant des exercices d'algèbre, basée sur des données recueillies lors de l'utilisation d'un EIAH (Environnement Informatique pour l'Apprentissage Humain) pour l'algèbre, le logiciel Aplusix. Toutes les actions des élèves ont été enregistrées et constituent notre corpus de données initial. Par l'application de techniques de classification non supervisée à ces données préalablement « vectorisées », nous construisons des catégories d'élèves dont la réponse aux problèmes posés est comportementalement homogène. Nous présentons ici les résultats obtenus avec les différents prétraitements employés, ainsi que les variables de représentation retenues lors de l'analyse automatique de données réelles.

1 – INTRODUCTION

L'analyse de comportements d'élèves, étape centrale de toute étude didactique, est souvent manuelle et individuelle. Il en ressort une description « détaillée » des phénomènes mais au prix d'un temps considérable. Ainsi, les outils statistiques (Bronner, 2003) apportent une aide précieuse au didacticien en synthétisant le résultat des observations. Notre approche consiste à étendre ce type de démarche en appliquant des méthodes d'apprentissage machine afin d'automatiser davantage une partie de ces analyses, et ainsi obtenir des résultats de plus haut niveau faisant apparaître des catégories de comportements d'élèves homogènes.

La modélisation de l'élève par apprentissage machine est un domaine très actif, comme en témoigne la riche bibliographie s'y rapportant. Ainsi, Sison (1998) dresse un éventail des techniques employées et des systèmes développés. Certains de ces systèmes sont capables de construire une représentation des connaissances de l'élève sur un domaine donné à partir d'une seule prestation de cet élève (LISP Tutor (Anderson, 1985)); Proust (Johnson, 1990)), mais ils s'appliquent

seulement dans la cas où la représentation du résultat est suffisamment structurée pour le faire, à savoir des programmes informatiques. D'autres systèmes nécessitent plusieurs cas pour construire le modèle de l'élève, et s'appliquent à des domaines où la nature du résultat est plus factuelle (DEBUGGY (Burton, 1982) ; ACM, (Langley 1984) ; ASSERT (Baffes, 1996)). Ces systèmes sont conçus à partir de techniques d'apprentissage machine symboliques (apprentissage par induction supervisé ou non supervisé), et nécessitent souvent la constitution préalable d'une large base de connaissances par des experts du domaine pour fonctionner. Dans un contexte proche de notre travail, Beck (2000) utilise l'apprentissage par renforcement, mais cherche à prédire la justesse de la réponse et le temps de réaction de l'étudiant modélisé, alors que nous cherchons à découvrir et expliciter des descriptions de groupes d'élève.

Dans notre approche, pour décrire le comportement des élèves nous disposons à la fois de descripteurs quantitatifs décrivant leurs performances (temps, résultats, ...) mais aussi de descripteurs modélisant les connaissances symboliques caractérisant les manipulations mathématiques effectuées par les élèves sur les expressions algébriques. Sur l'ensemble de ces descripteurs nous appliquons des techniques de classification non supervisée, dans un but exploratoire, afin d'obtenir des groupes d'élèves présentant des comportements homogènes.

Le contexte de travail : le projet « Cognitive-Algèbre » et Aplusix

Notre travail se déroule dans le cadre du projet « Modélisation cognitive de comportements d'élèves en algèbre », une ACI financée par le ministère de la recherche. Il repose sur l'utilisation d'un EIAH en algèbre, le système Aplusix, dont la présentation détaillée peut être trouvée dans (Bouhineau, 2001). Ce logiciel permet à l'élève de manipuler facilement des expressions mathématiques, tout en « découpant » son travail en plusieurs étapes de calcul (il y en a 3 dans la Figure 0).

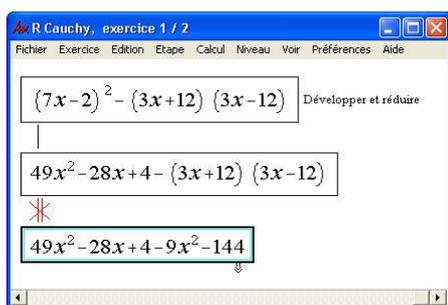


Figure 0 : le logiciel Aplusix

Chaque prestation d'élève utilisant Aplusix est systématiquement enregistrée dans un fichier recueillant la séquence des actions effectuées (frappes de touches, clics souris ...) ainsi que leurs instants d'occurrence, l'expression mathématique qui était alors affichée, et quelques renseignements additionnels. Un tel fichier, contenant tout l'historique des interactions élève-Aplusix, s'appelle un protocole. Enfin, Aplusix contient un module capable de « rejouer » un protocole, action par action ou étape par étape : le magnéscope. C'est l'outil d'analyse manuelle de protocoles.

En Septembre 2002, des enseignants en mathématiques d'une dizaine de collèges et lycées ont mis en place des sessions d'utilisation du logiciel Aplusix (voir (Bronner 2003)). Plusieurs listes d'exercices ont été mises au point pour chaque niveau (3e, 2nde, 1ère), chacune contenant une trentaine d'exercices de difficultés croissantes. Chaque exercice est traité par un nombre très variable d'élèves : de plus de 200 élèves sur les premiers exercices d'une série à moins de 10 pour les derniers. Notre corpus initial est constitué d'environ 11000 couples « élève/exercice ».

2 – LA CHAÎNE DE TRAITEMENTS

2.1 – La complétion de protocoles

Les protocoles bruts contiennent une information purement syntaxique (la succession des actions élémentaires et les expressions algébriques produites) qu'il est difficile d'exploiter directement. Un module de complétion de protocoles a été mis au point par nos partenaires au sein du projet Cognitique-Algèbre : à partir d'une analyse de l'expression algébrique, il explicite et déduit des informations de plus haut niveau. Les

fichiers obtenus sont désignés par protocoles complétés.

Parmi les champs rajoutés, nous avons ceux qui apportent des compléments d'information sur la justesse des expressions (par exemple l'équivalence mathématique entre l'expression en cours et la dernière expression de l'étape précédente), utiles pour les calculs des variables servant à la classification ou pour l'interprétation des résultats. D'autres descripteurs concernent la « qualification » (Gélis 1994). Cela consiste à identifier le type de transformation effectuée entre une expression initiale et une expression finale donnée, indépendamment de sa justesse mathématique. Ex : « $(10x=7x-4 \rightarrow 3x=4)$: *Reduction Additive Equivalente* » signifie qu'entre les expressions situées de part et d'autre du symbole « \rightarrow » a eu lieu une transformation qualifiée par « *Reduction Additive* », ici mathématiquement équivalente. Comme nous le verrons plus loin, ces champs additionnels vont nous être d'une grande utilité pour mieux identifier des groupes d'élèves aux comportements homogènes.

2.2 – Construction du jeu de données

Afin de pouvoir utiliser un algorithme de classification automatique, il nous faut « recoder » les informations des protocoles sous la forme d'une matrice lignes (élèves) colonnes (variables), donc décrire chaque protocole par un unique vecteur de taille fixe. Dès lors, il devient difficile de conserver l'aspect séquentiel des protocoles, et l'utilisation de descripteurs rendant compte de cette séquentialité pourrait être envisagée. Dans une première approche, les variables descriptives qu'il était possible de calculer directement à partir d'un protocole sont résumées dans la table 1.

Famille de variable	Nbre	Type	Explication
Nombre d'actions	14	Entiers naturels	Décomptes de nombres d'actions effectués par l'élève. Ex : actions dont l'expression résultante est équivalente, mal formée, etc...
Nombres d'étapes	10	Entiers naturels	Décomptes des nombres d'étapes, distingués selon l'état de l'expression au moment de la création.
Temps	9	Réels	Temps passés devant des expressions équivalentes, non équivalentes, etc...

Qualifications	30	Entiers naturels	Nombre d'instances de chaque qualification dans l'exercice, distribués dans des champs différents selon la justesse de la transformation (équivalente ou non)
----------------	----	------------------	---

Table 1 : Variables initialement retenues pour caractériser, de manière statistique, le comportement d'un élève sur un exercice donné.

De plus, nous normalisons ces variables car elles évoluent sur des domaines de valeurs très différentes, et cela a une influence considérable sur nos résultats vu que nous utilisons une distance euclidienne. Nous centrons puis réduisons chacune des dimensions selon les formules suivantes :

Soit $x_{i,j}$ la valeur de la variable numéro j de la donnée numéro i . La variable centrée-réduite $z_{i,j}$ correspondant vaut :

$$z_{i,j} = \frac{x_{i,j} - \bar{x}_j}{\sigma_j} \quad (1) \quad \text{avec} \quad \bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{i,j}$$

$$\sigma_j = \frac{1}{n} \sum_{i=1}^n (x_{i,j} - \bar{x}_j)^2$$

2.3 – La sélection des variables

Toutes les variables précédemment collectées ne sont pas de bons descripteurs représentatifs de la prestation de l'élève. Ainsi, Parmi les 63 variables présentées dans la table 1, nous avons choisi, dans un premier temps, d'en retenir 12. En effet, notre matrice de données est creuse, car certains de nos descripteurs initiaux ne sont calculés que lors d'évènements relativement rares, par exemple quand l'expression est indéfinie (ex : $\text{rac}(-1)$). Voici les 12 premières variables retenues

- #a_eqenc : nbe d'exp équiv à l'énoncé
- #a_equiv : nbe d'exp équiv à l'étape préc
- #a_expmf : nombre d'exp mal formées
- #a_modif : nbe d'actions modifiant l'exp
- #a_noneq : nbe d'exp non équivalentes
- #a_total : nombre total d'exp
- #e_suppr : nombre d'étapes supprimées
- #e_total : nombre d'étapes créées
- ~t_equiv : tps passé dans des exp équiv
- ~t_expmf : tps passé sur des exp mal formées

- ~t_noneq : tps passé sur des exp non équiv
- ~t_total : tps total passé sur l'exercice

(« exp »=expression, « equiv »=équivalentes, « tps »=temps).

De plus, parmi les 30 différentes variables portant sur les qualifications, nous avons choisi de ne retenir que celles portant sur des transformations fausses, et parmi ces dernières, seulement les 6 plus fréquentes sur l'exercice analysé (cf d.):

- #q_epcne_ANTIEFFONDREMENT : hausse subite du degré formel. Exemple : $x+3 \rightarrow x^2+3$

- #q_epcne_CALCULNUMERIQUE.

Exemple : $8+3 \rightarrow 12$

- #q_epcne_DEVELOPPEMENT.

Exemple : $-54x^2+18x+30x \rightarrow -54x^2+48$

- #q_epcne_DEVELOPPEMENTPOLY.

Exemple : $(9x-5)(-6x+2) \rightarrow -54x^2+18+30x-10$

- #q_epcne_EFFONDREMENT : baisse subite du degré formel. Exemple : $x^2+x-2 \rightarrow 3x-2$

- #q_epcne_REDUCTIONADDITIVE.

Exemple : $3x+4x \rightarrow 8x$

Le choix de ces variables est une phase importante de notre travail, et découle d'une expertise manuelle que nous avons acquise en manipulant les données. Nous travaillons actuellement pour confirmer notre analyse par d'autres mécanismes de sélection de variables, notamment basés sur des techniques d'apprentissage supervisé que l'on peut trouver dans la boîte à outils WEKA (Witten 1999). Nous présentons en partie 3.d quelques premiers travaux visant à étudier l'impact du choix de variables sur la classification obtenue.

2.4 – Choix du corpus d'apprentissage

Chaque élève ayant traité plusieurs exercices, nous disposons de plusieurs vecteurs numériques pour chaque élève. Lors d'expériences préliminaires, nous avons effectué des classifications en considérant comme représentant d'un élève un vecteur moyen sur tous les exercices traités. Cela menait à de mauvais résultats. Ceci est peu surprenant vu que cela revenait à mélanger les disparités de compréhension de l'élève (faiblesses dans tels types d'exercices et bon niveau dans d'autres).

En effet, un élève pouvait très bien avoir compris une technique (comme la factorisation) et être en difficulté sur une autre (comme le développement) : la nécessité de caractériser des groupes d'élèves sur des types de tâches différents est donc devenue incontournable ; et cela nous permettait, en outre, de décrire plus finement son profil. Ainsi, dans un premier temps, nous nous sommes concentrés sur un exercice précis et avons effectué notre classification sur tous les élèves l'ayant traité, avec l'objectif ultérieur de réitérer ce procédé sur d'autres exercices. Son choix a été guidé par deux facteurs : le nombre d'élèves l'ayant traité devait être le plus grand possible (afin d'avoir des statistiques significatives) et non trivial (un exercice trop facile donne peu d'indications sur le niveau de l'élève). Ainsi, nous avons choisi un exercice de 2^{nde}, que nous nommerons « exercice 1204 ». Son intitulé exact est : « développer puis réduire $(9x-5)(-6x+2)$ ». 218 élèves l'ont effectué, d'où un jeu de données de 218 vecteurs.

2.5 – Les outils de classification

Après avoir étudié plusieurs algorithmes de classification, nous avons retenu le SuperParamagnetic Clustering (SPC) développé par Blatt (1997). Inspiré par une analogie issue de la physique, cette technique présente l'avantage du multi-échelle : par opposition à beaucoup d'approches classiques (du type K-means (Diday, 1973) (Duda 2001), PAM/CLARANS (Kaufman 1990), DBSCAN (Ester 1996)...), le résultat fourni par le SPC n'est pas une partition des données en classes, mais plusieurs partitions ordonnées selon un paramètre dit de « température » (il n'est pas nécessaire de connaître ici le fonctionnement précis de cet algorithme. Le lecteur intéressé pourra se référer à (Blatt, 1997) pour plus de détails). C'est l'utilisateur qui choisit l'échelle à laquelle il veut observer les classes obtenues, à la manière des approches hiérarchiques (Lebart 2000). Cependant, cet algorithme surpasse ces approches hiérarchiques classiques en terme de temps de calcul et de résistance au bruit; de plus, son mode de fonctionnement par densité lui permet de détecter des clusters de formes arbitraires (Berkhin, 2002).

2.6 – La visualisation des résultats

Nous utilisons un module Matlab pour visualiser le résultat d'une classification par SPC. La courbe de la Figure 2 représente les tailles des différents clusters, pour chaque valeur du paramètre. Les paliers observés sont le signe de l'existence de groupements stables sur une plage de variation du paramètre : cela est caractéristique d'une classification de bonne qualité.

En pratique, nous choisissons la valeur de température à laquelle nous observons des paliers mettant en jeu des classes avec un effectif « convenable », en nous basant sur les cardinaux des groupes détectés : trop élevés, ils sont synonymes de faible homogénéité, et trop faibles, ils sont peu informatifs.

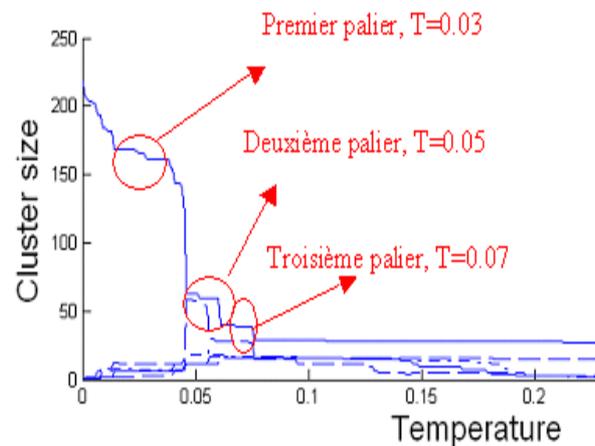


Figure 0 : Exemple de courbe fournie par le SPC. Le premier palier correspond à un seul groupe comprenant quasiment tous les individus. Le deuxième et le troisième palier sont plus intéressants car il y a plus de groupes de plus petite taille.

3 - RESULTATS

3.1 - Résultats avec les protocoles bruts, sans traits de qualification

La première de nos expérimentations avait pour objectif de constituer un résultat-témoin, auquel nous pourrions nous référer par la suite pour constater les améliorations. Nous n'avons donc utilisé que les 12 premières variables.

On obtient 4 principaux groupes totalisant 101 élèves :

a) Les succès rapides (48) : on constate, a posteriori, et ce sans avoir inclus la variable « résolu » dans nos 12 variables, qu'une majorité des élèves de ce groupe a fait juste, en un temps bien inférieur à la moyenne et en interagissant relativement peu avec le logiciel

b) Les succès actifs (15) : légèrement plus lents que le groupe a), ces élèves se caractérisent par un fort niveau d'interaction avec le logiciel, et leurs manipulations concernent essentiellement des expressions équivalentes à l'énoncé.

c) Les échecs pour faute de parcours (20) : ce groupe est très similaire au a) en termes de temps et d'actions, mais il se distingue par un très faible nombre d'expressions équivalentes. Les élèves de ce groupe ont souvent fait une erreur en cours de résolution

d) Les échecs lents (18) : ceux ci passent la majorité de leur temps devant des expressions mal formées ou non équivalentes. Ils font beaucoup d'erreurs (nombre d'actions équivalentes faible)

En examinant de manière plus détaillée le travail des élèves de chaque groupe avec le « magnétoscope » (cf. 1) on a constaté de fortes inhomogénéités comportementales dès que l'on s'éloignait des centroïdes (notamment dans le groupe a), c'est-à-dire que la présence de certains élèves était difficile à expliquer au sein de ces groupes. Cela est le signe de variables mal choisies.

En effet, en y regardant de plus près, on voit que les variables renseignées à partir des protocoles bruts tiennent compte de toutes les lignes, y compris celles relatives à des actions temporaires vides de sens algébrique. Par exemple, pour écrire $21x^2$, l'élève va frapper successivement sur les touches 2, 1, x, ^, 2 et passer par les expressions intermédiaires suivantes : $2 \rightarrow 21 \rightarrow 21x \rightarrow 21x^{\wedge} \rightarrow 21x^2$. Certaines de ces expressions sont mal formées et la majorité contient peu d'informations, donc toutes les variables de décomptes d'actions sont perturbées. Tous les élèves passent par ce style d'expressions transitoires : ceci explique la plus grande difficulté de cette expérience à dégager de nettes catégories d'élèves.

Aussi, nous avons introduit une étape de prétraitement qui vise à « condenser » les protocoles bruts de manière à supprimer toutes

ces actions intermédiaires et à garder de manière automatique des expressions plus pertinentes. Cela nous a conduit à concevoir un programme PERL effectuant un résumé de protocoles, afin d'effectuer une classification sur des données plus significatives.

3.2 – Mécanismes de résumé

Nous avons mis au point des heuristiques de transformations qui remplacent une ou plusieurs lignes de protocole en une nouvelle ligne (on parle alors de condensation) éventuellement vide (on parle alors de nettoyage). Ces transformations sont appliquées successivement sur un protocole brut, dans l'ordre précisé ci-après.

1. Nettoyage des actions non modifiantes (clics souris, déplacements de curseur...)

2. Nettoyage des « mono-backspaces » : lorsqu'un élève fait une erreur de frappe immédiatement corrigée, on ne garde que le résultat de sa correction.

Exemple : 8, 4, Bcksp, 2 -> 8, 2

3. Condensation des frappes successives : agrégation d'une série ininterrompue d'actions considérées comme frappes additives.

Exemple : 2, 1, x, ^, 2 -> « $21x^2$ »

4. Condensation des répétitions : agrégation d'une répétition d'actions identiques. Exemple : Backspace, Backspace, Backspace -> « Backspace (x3) »

5. Nettoyage des séries de Backspaces entre deux séries de frappes successives : celles-ci produisant systématiquement des expressions intermédiaires (souvent mal formées) constituant des points de départ pour de nouvelles séries d'éditations, elles ne sont pas significatives d'un point de vue algébrique.

L'application successive de ces 5 transformations à un protocole brut produit un protocole résumé. En moyenne, sur toutes les données traitées, on constate une réduction de la taille du protocole à environ un cinquième de la longueur de l'original.

3.3 - Résultats et analyse avec les protocoles résumés

En n'utilisant que les 12 premières dimensions, mais cette fois sur les protocoles résumés au lieu des protocoles bruts, on obtient 5 groupes

(totalisant 96 élèves) beaucoup plus homogènes en termes de comportements :

a) Les succès rapides (27) : Ceux-là sont plus rapides que le groupe du même nom trouvé en 3.a, et passent par moins d'expressions intermédiaires non équivalentes. Ils sont moins nombreux et beaucoup plus homogènes.

b) Les succès lents (12) : ces élèves agissent plus et sont plus lents que a), mais finissent par faire juste. Ils n'étaient pas détectés précédemment, ce qui conforte l'utilité du procédé de résumé de protocole.

c) Les succès actifs (16) : la description est identique à 3.2.1

d) Les échecs par faute de parcours (26)

e) Les échecs lents (16)

Pour caractériser les classes, nous avons visionné au « magnétoscope » les protocoles de quelques élèves de chaque catégorie obtenue. Dans le cas de l'ensemble d), on constate que dans un même groupe d'élèves ayant fait faux, les élèves font des erreurs radicalement différentes. Exemples :

- $9x^* - 6x \rightarrow -64x^2$ (erreur de calcul)
- $9x^* - 6x \rightarrow -54x$ (oubli de la puissance)
- $9x^* - 6x \rightarrow -54$ (oubli de la variable)

Or, en introduisant des variables qualitatives portant sur les transformations effectuées par l'élève on peut facilement améliorer la classification, en faisant apparaître de nouvelles catégories correspondant à des groupes d'élèves aux comportements plus homogènes. Dans cette optique, les 6 variables de qualifications supplémentaires décrites en 2.c constituent de bons candidats.

3.4 – Résultats obtenus avec protocoles résumés et variables de qualification

Cette fois-ci, nous utilisons pour la classification la totalité des 18 variables décrites en 2.c. On obtient un découpage encore plus fin et homogène que le précédent, concernant plus d'élèves (129), bien que ce chiffre reste encore largement améliorable :

a) Les succès rapides (29)

b) Les succès actifs (16)

c) Les succès lents (10)

d) Les échecs pour erreurs diverses (28)

e) Les échecs pour erreur de développement polynômial (17)

f) Les erreurs de développement (11)

g) Les erreurs d'antiefonnement (11)

h) Les erreurs de calcul (7)

On constate que les groupes « élèves ayant fait une erreur de parcours » et « mauvais laborieux » (groupes d et e du 3.4) se retrouvent maintenant répartis dans les 5 groupes d'élèves ayant échoué détectés ici (d, e, f, g, h), et que cette classification concerne aussi de nouveaux élèves, grâce à l'ajout des nouveaux traits. On voit donc émerger des groupes encore plus homogènes en termes de comportements, pour lesquels le type d'erreur commise est identifié.

Par ailleurs, nous avons étudié l'influence des traits de qualification sur la stabilité des catégories détectées, en comparant les résultats obtenus en utilisant ou non ces traits lors de la classification. Pour cela, nous avons utilisé la matrice des cardinaux des intersections ensemblistes des deux familles de clusters : chaque case (i,j) de cette matrice contient le nombre d'éléments communs entre le cluster i de la 1ère classification et le cluster j de la 2e. Nous constatons que le groupe des bons (a) est bien détecté dans les deux cas (93% d'élèves ensemble en utilisant les qualifications se retrouvent ensemble sans les utiliser) et que les bons actifs sont relativement stables (50%), de même que les élèves ayant fait faux pour des raisons diverses (82%). Par contre, les groupes d'élèves ayant fait des erreurs identifiées (e,f,g,h) sont totalement absents d'une classification sans traits de qualification, ce qui montre l'utilité de ces traits pour notre objectif de classification comportementale.

Cependant, il faut remarquer que l'on mélange ici des variables quantitatives (nombre d'actions, temps...) avec des variables plus qualitatives (qualifications). En agissant de la sorte, et ce à cause du fonctionnement même des algorithmes de classification (basés sur des distances euclidiennes dans des espaces de haute dimension), des individus sont regroupés dans une même catégorie s'ils ont le même profil en termes à la fois d'utilisation d'Aplusix (nombre d'actions, d'étapes), de temps (rapides, lents), et d'erreurs effectuées (qualifications fausses). Par conséquent, si d'un côté on observe un gain certain dans

l'ajout de ces variables de « qualifications fausses » (une structuration plus forte des données), il y a aussi un effet négatif : un reclassement brutal de certains élèves passant temporairement par une expression erronée, mais dont la qualification est identifiée et peu fréquente.

4 – CONCLUSIONS ET PERSPECTIVES

Nous avons présenté les résultats de travaux visant à constituer automatiquement des catégories d'élèves à partir de séquences d'actions. Le travail effectué vise en grande partie à transformer ces séquences afin de faciliter l'analyse et de pouvoir s'appuyer sur des techniques de classification automatique. Nous avons réussi à améliorer la pertinence des catégories produites en résumant les protocoles et en introduisant des variables plus qualitatives qui permettent de caractériser les transformations faites par les élèves. Plusieurs directions de recherche se dessinent.

Premièrement, nous comptons tester l'influence de la granularité avec laquelle on regarde la prestation de l'élève. Toute la difficulté réside dans la détection (et la prise en compte) des expressions clés de l'élève. La condensation effectue un premier filtrage, mais est peut-être encore trop « laxiste » en conservant certaines lignes « brouillons ». D'autre part, si on ne regarde que les transformations d'étape en étape (un grain plus gros), on risque de rater des expressions révélatrices de conceptions erronées de l'élève. En nous munissant d'un évaluateur de la qualité de la condensation, nous pourrions affiner nos mécanismes de condensation.

Deuxièmement, nous comptons accentuer nos efforts dans la constitution des variables servant à la classification. En premier lieu, utiliser des ensembles de variables descriptives différents selon le type d'exercice nous paraît désormais une nécessité, car les types d'erreurs à identifier dépendent fortement de la tâche à effectuer. De plus, les didacticiens du projet ont travaillé à la constitution d'une base de règles erronées (elles correspondent à des conceptions fausses connues du type : $(a+b)^2 \rightarrow a^2 + b^2$), qui est désormais intégrée à la base de règles correctes existant dans Aplusix. Ainsi, la phase de complétion est en train de s'enrichir d'un nouvel outil (encore à l'étude), capable de construire (si possible)

un diagnostic, i.e. une séquence de règles correctes ou erronées permettant « d'expliquer » le passage d'une expression initiale à une expression finale, et ce de manière plus précise que les qualifications. Des expériences préliminaires en cours, où nous avons basé notre classification sur des variables issues de ces diagnostics, donnent des résultats encourageants. En effet, de tels indicateurs à forte sémantique mathématique et didactique, semblent faire ressortir de plus nettes catégories d'élèves.

Notre but est de constituer à partir de notre corpus de données des « cartes topologiques d'élèves », différentes selon le type de tâche (développer, factoriser...), permettant de situer tout nouvel élève en fonction des difficultés rencontrées. Nous pensons que la classification automatique est un outil exploratoire idéal pour découvrir les variables pertinentes permettant de bien détecter les catégories d'élèves aux comportements homogènes, et ce pour chaque type d'exercice. Ces analyses fourniraient une base de connaissances servant de support à un tuteur artificiel, qui proposerait à l'élève de nouveaux exercices ciblés en fonction de son appartenance à une catégorie identifiée. Une application en e-learning sera envisageable à plus long terme, si nos travaux permettent d'identifier avec un bon taux de réussite le groupe de niveau de connaissances auquel appartient l'élève, et si des réponses didactiques appropriées sont mises au point pour chaque type de difficulté rencontrée.

BIBLIOGRAPHIE

- Anderson, J.R., Reiser, B. (1985), "The LISP Tutor". *Byte n°10*, p. 159-175.
- Baffes, P., Mooney, M. (1996), "Refinement-based student modeling and automated bug library construction". *Journal of AI in Education, Vol 7(1)*, p. 75-116.
- Beck, J. E., Woolf, B.P. (2000), "High-level Student Modeling with Machine Learning". *Intelligent Tutoring Systems*, p. 584-593.
- Berkhin, P. (2002), "Survey of Clustering Data Mining Techniques". *Technical Report*.

- Bouhineau, D., Nicaud, J.F., Pavard, X., Sander, E. (2001), "Un micromonde pour aider les élèves à apprendre l'algèbre". *6e journées francophones Environnements Interactifs d'Apprentissage avec Ordinateur*.
- Blatt, M., Domany, E., Wiseman, S. (1997), "Data Clustering using a model granular magnet". *Neural Computation* 9.
- Bronner, A., Bouhineau, D., Chaachoua, H., Huguet, T. (2003), "Analyse didactique de protocoles obtenus dans un EIAH en algèbre ». *Actes de EIAH Strasbourg*, p. 79-90.
- Burton R. (1982), "Diagnosing Bugs in a simple procedural skill». *Intelligent Tutoring Systems*, Eds D. Sleeman & L. Brown, London Academic Press.
- Diday E. (1973), "The dynamic cluster method in non-hierarchical clustering ». *Journal of Comput. Inf. Sci.* 2, p. 61-88.
- Duda, R.O., Hart, P.E., Stork, D.G. (2001), *Pattern Classification (2nd edition)*. John Wiley & Sons, p. 561-562
- Ester, M., Kriegel, H.P., Sander, J., Xu, X. (1996), "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". *2nd Int. Conf. on Knowledge Discovery and Data-mining*.
- Gélis, J.M. (1994), *Éléments d'une théorie cognitive et computationnelle de l'algèbre*. Thèse à l'Université Paris-Sud/ Orsay.
- Johnson, WL (1990), "Understanding and debugging novice programs". *AI n°42*, p. 51-97.
- Kaufman, L., Rousseuw, P.J. (1990), *Finding Groups in Data : An introduction to Cluster Analysis*. John Wiley & Sons.
- Langley, P., Ohlsson, S. (1984), "Automated cognitive modeling". *Second National Conference on AI*.
- Lebart, L., Morineau, A., Piron, M. (2000), *Statistique exploratoire multidimensionnelle* (chap 2.2). Dunod, Paris (3e éd).
- Sison, R., Shimura, M. (1998), "Student Modeling and Machine Learning". *Int. Journal of AI in Education*, n°9, p. 128-158.
- Witten, I.H., Frank, E. (1999), "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations". Morgan Kaufmann.

Remerciements: Bisson G, Gordon M.B, Nicaud JF

NSCSP : DEFINITION ET RESOLUTION PAR TRANSFORMATION

Alexis Anglada,

Doctorant en programmation par contraintes et décision

alexis.anglada@dassault-aviation.fr, + 33 1 47 11 48 19

Adresses professionnelles

DASSAULT AVIATION ★ 78 Quai Marcel Dassault ★ 92552 Saint Cloud Cedex, France.

Laboratoire informatique Paris 6 (LIP6) ★ Rue du capitaine Scott ★ 75015 Paris, France.

Résumé : L'expressivité des problèmes de satisfaction de contraintes discrets (CSP) a été augmentée ces dernières années par l'introduction de la gestion de l'incertitude et des préférences (cf. les cadres SCSP (CSP à base de demi-anneau) et VCSP (CSP valué)) permettant ainsi d'appréhender de manière plus flexible les problèmes réels. Cependant ces derniers comportent souvent des composantes continues. Pour prendre en compte cette spécificité, la communauté développe activement la théorie des CSPs continus (Numerical CSP noté NCSP). Dans ce poster

nous définissons une extension du paradigme des problèmes de satisfaction de contraintes flexibles aux domaines continus dans le cadre des demi-anneaux. Nous montrons comment résoudre les NCSPs flexibles ainsi définis par transformation en un NCSPs classiques.

Summary : Recent extension of the Constraint Satisfaction Problem (CSP) paradigm are the soft CSP framework and the numerical CSP (NCSP) framework. The first one addresses the management of data uncertainty and the expression of preferences. The second one addresses numerical problem solving. In order to address real problems we wish to combine these two paradigm. Therefore we will define in this poster an extension of the soft CSP paradigm to the continuous domains within the semirings' framework. We will show how to solve soft NCSPs by transforming them into NCSPs.

Mots clés : Intelligence artificielle, Programmation par contraintes, Flexibilité, Problème de satisfaction de contraintes numériques.

Key words : Artificial intelligence, Constraint Programming, Softness, Numerical constraint satisfaction problem.

NSCSP : Définition et résolution par transformation

1 - INTRODUCTION

Les problèmes de satisfaction de contraintes (CSP) permettent de modéliser de nombreux problèmes réels tel que le dimensionnement ou l'allocation de ressources. Or, dans les cas réels, on retrouve souvent des données incomplètes, des spécifications imprécises, des problèmes sur-contraints. Ces situations ne sont pas prises en compte par les CSPs classiques, aussi a-t-on cherché à introduire de la flexibilité, via les contraintes molles ou flexibles, dans la programmation par contraintes. Les travaux en domaine fini ont conduit à dégager deux cadres : les CSPs valués (VCSP [SFV97]) et les CSPs à base de demi-anneau (Semi-ring based CSP (SCSP) [BMR95]). Il a été démontré que tout VCSP peut s'exprimer par un SCSP dans [BMR+99]. Bien qu'ils soient en adéquation avec la modélisation de problèmes réels, les CSPs continus (NCSP pour Numerical CSP) n'ont pas bénéficié de cette évolution. Aussi nous montrerons dans un premier temps comment étendre les SCSPs au cas continu en généralisant cette théorie. Cette extension permettra d'exploiter toute l'expressivité de la combinaison des SCSPs et des domaines continus, nous noterons le cadre ainsi obtenu NSCSP pour numerical semi-ring based CSP. Comme l'ont montré les auteurs de [BCR02] et de [BGR00], il est souvent difficile de résoudre un SCSP ; pour faciliter la résolution des NSCSPs, nous proposerons, dans un deuxième temps, une transformation de ces derniers en NCSPs classiques. Et nous montrerons que les solutions du NCSP obtenu sont les mêmes que celles du NSCSP d'origine.

Ce papier s'organise comme suit : dans la partie 2, après avoir rappelé quelques notations et définitions, nous définirons les NSCSPs. Dans la section 3, nous définirons notre méthode de transformation et prouverons la validité des solutions obtenues. Avant de conclure, nous parlerons de notre implémentation dans la section 4 et nous y montrerons en exemple résolu de NSCSP.

2 - VERS LES NSCSPS

Après un bref rappel des définitions utiles au cadre des SCSPs, nous donnerons une définition des contraintes flexibles continus et définirons ce qu'est un CSP continu à base de c-demi-anneau (NSCSP).

2.1 - Rappels

Dans la suite de ce document nous noterons, X l'ensemble des variables d'un problème, D l'ensemble des domaines associés à celles-ci, C l'ensemble des contraintes. $x_i \in X$ est une variable, d_{x_i} est son domaine et c_i est une contrainte. Les SCSPs et les NSCSPs repose sur l'utilisation d'une structure mathématique : le c-demi-anneau.

Définition 1: Un c-demi-anneau est un tuple $(A, \oplus, \otimes, 0, 1)$ tel que :

A est un ensemble et $0 \in A$, $1 \in A$

\oplus , appelée opération additive, est commutative (i.e : $a \oplus b = b \oplus a$), associative ($a \oplus (b \oplus c) = (a \oplus b) \oplus c$), idempotent ($a \oplus a = a$), 0 est son élément neutre ($a \oplus 0 = a = 0 \oplus a$) et 1 son élément absorbant ($a \oplus 1 = 1 = 1 \oplus a$)

\otimes , appelée opération multiplicative, est commutative, associative, 1 est son élément neutre et 0 son élément absorbant.

\oplus est distributive par rapport à \otimes (i.e : $\forall a, b, c \in A, a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$)

Un c-demi-anneau est muni d'un ordre partiel (\leq_A) et ses opérations sont monotones sur cet ordre.

Pour définir un SCSP, il est nécessaire de garder à l'esprit qu'une contrainte flexible est une relation qui, à chaque tuple de valeurs des variables sur lesquelles elle porte, associe une valeur dans l'ensemble A , support du c-demi-anneau, cette définition fut introduite dans [BMR95].

Définition 2: Un SCSP est un tuple (X, D, C, A) avec (X, D, C) un CSP où les contraintes ont valeur dans A , un c-demi-anneau.

2.2 - NSCSP

Les NSCSPS sont , à l'instar des NCSPs pour les CSPs, une extension des SCSPs. Comme pour leur homologues classiques, les contraintes exprimées au sein d'un NSCSP le sont en extension, car les domaines des variables sont des intervalles et possèdent donc une infinité de valeur. Aussi définissons nous les contraintes à l'aide d'une fonction et non comme une relation. Cette définition des contraintes se rapproche de celle fonctionnelle vue dans [BMR02].

Définition 3: Une contrainte est un couple (def, con) où :

- $con \in X$
- def est une fonction

$$def: \prod_{i|x_i \in con} d_{xi} \rightarrow A$$

où A est un ensemble de valeurs (ex : $\{vrai, faux\}$, $[0,1]$, \mathbf{R}), $\prod d_{xi}$ le produit cartésien des domaines des variables sur lesquelles porte la contrainte. def est appelée fonction de satisfaction de la contrainte.

Un exemple de contrainte flexible :

$$con = \{x, y\}, d_x = d_y = [0, 10]$$

$$[1, 10] \rightarrow [0, 1]$$

$$def: x, y \rightarrow \begin{cases} 1 & \text{si } x < y \\ \frac{1 - |x - y|}{10} & \text{autrement} \end{cases}$$

On remarque que les définitions des contraintes en extension pour les domaines finis sont un cas particulier de cette définition. La définition d'un SCSP se généralise donc ainsi :

Définition 4: Numerical Semi-ring based CSP

Un NSCSP est un tuple $P = (X, D, C, A)$ où (X, D, C) est un NCSP et A un c-demi-anneau. C est l'ensemble des contraintes définies selon le couple (def, con) de la définition 3.

Définition 5: Solutions d'un NSCSP

Une solution d'un NSCSP est une instantiation globale de X associée à sa valeur dans A . Cette valeur s'obtient par combinaison des valeurs de satisfaction de toutes les contraintes du problème. Cette combinaison s'obtient par application de l'opération multiplicative au

résultats de toutes les résultats des fonctions de satisfaction.

Une solution optimale d'un NSCSP est une solution qui a la "meilleure" satisfaction au sens de \leq_A .

Les instances des SCSPs montrés dans [CR00] s'étendent aux NSCSPs en gardant les mêmes c-demi-anneaux si les contraintes sont exprimées avec des fonctions de satisfaction comme prévu dans la définition 3.

3 - RESOLUTION PAR TRANSFORMATION

Les CSPs ainsi que les NCSPs sont résolu à l'aide d'algorithmes de consistance. Pour les SCSPs, les auteurs de [BGR00] ont explicité les propriétés nécessaires de l'opération multiplicative du demi-anneau pour assurer la terminaison des algorithmes de consistance locale. Dans [BCR02] les auteurs définissent une autre méthode de résolution des SCSPs: l'abstraction. L'idée est d'exprimer un SCSP par un ensemble de CSP et de résoudre ces CSPs de manière itérative. Il existe des solveurs efficace de NCSP et nous voulons nous servir de cet avantage pour résoudre les NSCSPs. Nous proposons donc une technique de transformation inspirée de ce travail d'abstraction et des technique de réification utilisé dans le cadre des CSPs hiérarchiques. Dans un premier temps nous expliciterons notre méthode de transformation des NSCSPs en NCSPs. Et dans un deuxième temps, nous montrerons l'équivalence entre les solutions du NSCSP et celles du NCSP qui en est issu.

3.1 Définition

Pour transformer un NSCSP en NCSP, il faut exprimer les contraintes flexibles via des contraintes dures. Le cas évident est celui où A le c-demi-anneau est $(\{vrai, faux\}, \vee, \wedge, faux, vrai)$ car toutes les contraintes sont déjà des contraintes dures, en effet ce c-demi-anneau est celui de (N)CSPs classiques. Pour la suite de cet article nous supposons que le c-demi-anneau est soit $(\{vrai, faux\}, \vee, \wedge, faux, vrai)$ soit que $A \subseteq \mathbf{R}$; ainsi nous serons sûrs de pouvoir effectuer notre transformation car A est un ensemble exprimable par un intervalle comme prévu dans un NCSP. De plus nous posons comme notation : con_j l'ensemble des variables sur lesquelles porte la

contrainte c_j , def_j la fonction de définition de cette contrainte.

Nous supposons que nous avons une expression analytique pour chaque fonction de satisfaction des contraintes du problème.

Il existe deux cas de transformation d'une contrainte flexible selon sa fonction de satisfaction:

- la fonction de satisfaction à une expression analytique unique
- la fonction de satisfaction est "définie par morceaux"

Commençons par le cas de transformation le plus simple : une expression analytique unique. Pour obtenir une contrainte dure, il suffit d'introduire une variable supplémentaire, noté z_j , de domaine A, l'ensemble du c-demi-anneau, et dont la valeur sera donnée par la fonction de satisfaction selon l'expression :

$$z_j = def_j (con_j)$$

Prenons, dans le cadre des CSPs flous ou fuzzy CSPs (FCSP) $A = [0,1]$, la contrainte flexible suivante :

$$c_1 : con_1 = \{ x,y \}, dx = [1,10] = dy$$

$$def_1 : [1,10]^2 \rightarrow [0,1]$$

$$x,y \rightarrow \frac{1-|x-y|}{10}$$

Elle s'exprime par la contrainte dure suivante :

$$z_1 = \frac{1-|x-y|}{10}$$

$$\text{avec } \begin{cases} d_{z_1} = [0,1] \\ d_x = [1,10] \\ d_y = [1,10] \end{cases}$$

Notons qu'ici notre hypothèse $A \subseteq \mathbb{R}$ permet bien d'avoir un domaine continu pour z_1 représentable par un intervalle.

La transformation se complique pour le cas où la fonction de satisfaction est "définie par morceaux". Il est nécessaire d'avoir un mécanisme de contrainte conditionnelle. Nous exprimerons une contrainte conditionnelle par l'expression régulière suivante : $H \rightsquigarrow C$ où H est une condition s'évaluant par vrai ou faux et C une contrainte dure. $H \rightsquigarrow C$ est une contrainte conditionnelle au sens où les

opérateurs de réduction des domaines associés à C ne sont utilisés lors de la résolution que si H est vrai. H est toujours évalué et aucun opérateur de réduction n'y est associé. Avec ce mécanisme, une fonction de satisfaction "par morceaux" est transformée en autant de contraintes conditionnelles qu'il y a de "morceaux" dans sa définition.

Exemple, toujours avec un cadre FCSP :

$$c_2 : con_2 = \{ x,y \}, dx = [1,10] = dy$$

$$def_2 : [1,10]^2 \rightarrow [0,1]$$

$$x,y \rightarrow \begin{cases} 1 & \text{si } x \leq y \\ \frac{1-|x-y|}{10} & \text{si } x > y \end{cases}$$

s'exprime par :

$$x \leq y \rightsquigarrow z_2 = 1$$

$$x > y \rightsquigarrow z_2 = \frac{1-|x-y|}{10}$$

$$\text{avec } d_{z_2} = [0,1] \quad dx = dy = [1,10]$$

On a ainsi défini la transformation d'une contrainte flexible de la définition 3 en contrainte dure. Mais cela ne suffit pas pour avoir un NCSP équivalent au NSCSP de départ. Pour finir, il faut ajouter une contrainte qui fera l'agrégation des valeurs de satisfaction des contraintes flexibles. Cette contrainte introduit une nouvelle variable de domaine A qui représente la valeur de satisfaction d'une solution. Cette contrainte s'exprime ainsi :

$$sat = \bigotimes_{j=1}^m z_j$$

où \otimes est l'opération multiplicative du demi-anneau A du SCSP considéré, m le nombre de contraintes flexibles transformées et z_i les variables de domaine A introduites par cette même transformation. Dans le cadre des NSCSPs, comme dans celui des SCSPs, les solutions souhaitées sont souvent les solutions optimales. Pour les obtenir grâce à notre transformation, il est nécessaire d'optimiser la valeur de la variable *sat*.

3.2 Équivalence des solutions

Pouvoir transformer et ainsi résoudre un NSCSP n'est efficace que si les solutions trouvées après transformation sont bien celles définies par le NSCSP transformé.

Proposition 6 (équivalence des solutions)

Les solutions d'un NSCSP et celles du NCSP obtenu par la transformation définie dans la section 3.1 sont équivalentes.

Preuve :

Toutes instantiation globales des variables du NSCSP est un tuple $T = (t_1, \dots, t_n)$ associé à une valeur de satisfaction $k \in A$ défini par

$$k = \bigotimes_{j=1}^m def_j(con_j) \text{ dans le SCSP.}$$

Dans le NCSP obtenu par abstraction, toute instantiation globale est un tuple

$$T' = (t'_1, \dots, t'_n, z_1, \dots, z_m, sat)$$

Avec $sat = \bigotimes_{j=1}^m z_j$ et z_j obtenu par la transformation.

A toute instantiation dans le NSCSP correspond une seule instantiation dans le NCSP abstrait, car les fonctions def_j , qui définissent les contraintes, donne une seule valeur pour chaque tuple de valeurs de con_j , donc les valeurs t_i des variables définissent de façon unique les valeurs z_j des variables de satisfaction.

D'où pour tout tuple T , on a un tuple T' avec $t_i = t'_i$ et $k = sat$ par définition de sat et k . On a ainsi une bijection entre les tuples du NSCSP et ceux du NCSP obtenu. D'où toute solution du NSCSP existe dans le NCSP obtenu par transformation et sa valeur de satisfaction est celle attribuée à la variable sat . De plus, si k est optimal alors l'image de la solution dans le NCSP est optimale et si une solution est optimale dans le NCSP, alors sat est optimale et k est optimal, donc la solution est optimale dans le NSCSP. Nous avons donc l'équivalence des solutions du NSCSP avec celle du NCSP obtenu par transformation ainsi que pour les solutions optimales.

4 IMPLÉMENTATION ET EXEMPLE

4.1 Implémentation

L'équipe Contraintes et Décision de DASSAULT-AVIATION développe un logiciel basé sur un solveur de NCSP: Constraint Explorer(CE). CE a pour objectif la modélisation et la résolution des problèmes de

conception. Cet outil est le logiciel utilisé et amélioré dans le projet RNTL CO2 (acronyme de COnception par COntraintes). Une description du projet CO2 et de CE est faite dans [Zim01].

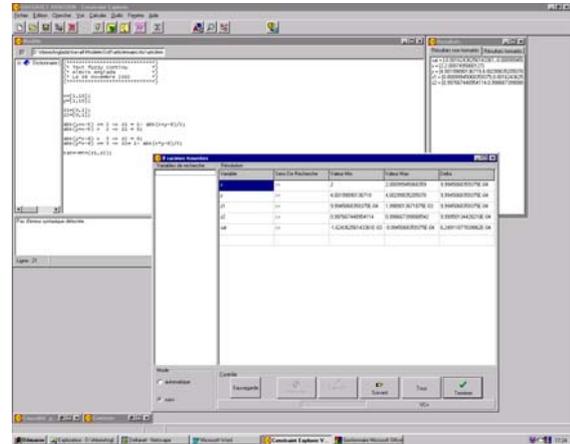


Figure 1 : Interface de Constraint Explorer

La figure 1 présente une vue de l'interface du logiciel de l'équipe Contrainte et Décision.

Ce logiciel fournit un mécanisme de contrainte conditionnelle à la fois dans son langage et dans son algorithme de résolution. La phase de propagation du solveur est assuré par un algorithme de Hull-consistance. Un algorithme de backtrack chronologique avec un ordre statique est à la base de la résolution. L'exploration de l'espace de recherche s'effectue par découpe dichotomique des domaines des variables. Le logiciel fournit aussi un mécanisme d'optimisation. Cette fonctionnalité est assurée par un algorithme de minimisation. Ce dernier fonctionne par réintroduction dans la recherche de la dernière solution trouvée comme borne supérieure de l'optimum. Ce schéma d'optimisation est dérivé de ceux expliqué dans [PM95] et de l'algorithme de branch and bound. Il a été adapté pour les domaines continus et implémenté dans CE comme expliqué dans [Let01].

4.2 Exemple

Nous allons maintenant présenter un exemple de NSCSP et certaines de ses solutions. Nous nous plaçons dans le cadre FCSP, le c-demi-anneau utilisé est $([0,1], \max, \min, 0, 1)$. On a bien $[0,1] \subseteq \mathbb{R}$, nous pouvons donc utiliser notre transformation. Notre problème est de trouver un couple de $x, y \in [1, 10]^2$ tel que leur somme soit proche de 5 et leur produit de 8.

Les préférence sur les valeurs de $x + y$ et $x*y$ sont exprimé par les courbes de la figure 2.



Figure 2: Courbes de préférence. Les abscisses représentent les valeurs de $x+y-5$ à droite et $x*y-8$ à gauche.

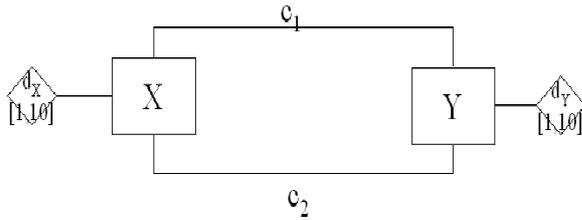


Figure 3: Exemple de NSCSP

Notre problème est schématisé par le NSCSP de la figure 3 avec c_1 et c_2 définies ci-dessous:

$$c_1 : con_1 = \{x, y\}$$

$$def_1 : [1,10]^2 \rightarrow [0,1]$$

$$x, y \rightarrow \begin{cases} 0 & \text{si } |x+y-5| > 2 \\ \frac{1-|x+y-5|}{2} & \text{si } |x+y-5| \leq 2 \end{cases}$$

$$c_2 : con_2 = \{x, y\}$$

$$def_2 : [1,10]^2 \rightarrow [0,1]$$

$$x, y \rightarrow \begin{cases} 0 & \text{si } |x*y-8| > 3 \\ \frac{1-|x*y-8|}{3} & \text{si } |x*y-8| \leq 3 \end{cases}$$

Les domaines de x et y sont $d_x=d_y=[1,10]$.

Par transformation, nous obtenons le NCSP suivant :

$$d_x = d_y = [1,10], d_{z_1} = d_{z_2} = [0,1]$$

$$|x+y-5| \leq 2 \rightsquigarrow z_1 = \frac{1-|x+y-5|}{2}$$

$$|x+y-5| > 2 \rightsquigarrow z_1 = 0$$

$$|x*y-8| \leq 3 \rightsquigarrow z_2 = \frac{1-|x*y-8|}{3}$$

$$|x*y-8| > 3 \rightsquigarrow z_2 = 0$$

$$sat = \min(z_1, z_2)$$

La première ligne exprime le domaine de chaque variable.

Nous avons résolu ce NCSP grâce à notre implémentation en arrêtant la découpe dichotomique des domaines quand leur taille est de l'ordre de 1% de leur borne inférieur. La table 1 présente quelques exemples de résultats. La figure 4 présente une courbe 3D de la satisfaction globale d'une solution, en fonction des valeurs des variables x et y , issue des résultats obtenu grâce à notre implémentation. Nous avons ensuite cherché les solutions optimales qui correspondent au sommet de cette courbe. Leurs valeurs, obtenues avec le critère précédent de terminaison de découpe des domaines, sont présentées dans la table 2.

x	y	z_1	z_2	sat
[1,1.01]	[5.59,5.6]	[0.2,0.203]	[0.2,0.203]	[0.2,0.203]
[1.41,1.43]	[4.56,4.59]	[0.5,0.506]	[0.5,0.506]	[0.506,0.506]
[1.93,1.96]	[3.62,3.67]	[0.7,0.708]	[0.7,0.708]	[0.705,0.705]

Table 1: Exemples de solutions

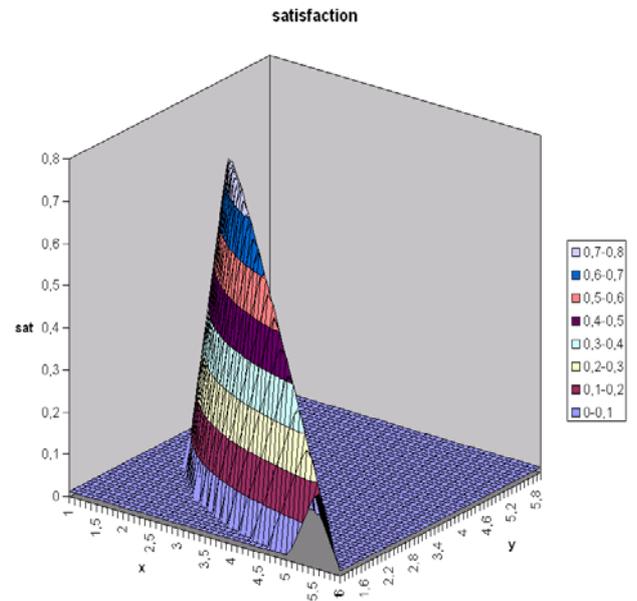


Figure 4: Courbe de satisfaction globale des solutions en fonction des valeurs de x et de y

x	y	z_1	z_2	sat
[2,6257; 2,6521]	[2,7744; 2,8023]	[0,7860; 0,7868]	[0,7860; 0,7874]	[0,7860; 0,7865]
[2,6522; 2,6789]	[2,7467; 2,7743]	[0,7860; 0,7872]	[0,7860; 0,7882]	[0,7860; 0,7869]
[2,6521; 2,6537]	[2,7743; 2,7759]	[0,7860; 0,7868]	[0,7860; 0,7874]	[0,7860; 0,7865]
[2,6790; 2,7058]	[2,7193; 2,7466]	[0,7860; 0,7875]	[0,7860; 0,7886]	[0,7860; 0,7870]
[2,6789; 2,6814]	[2,7466; 2,7491]	[0,7860; 0,7872]	[0,7860; 0,7883]	[0,7860; 0,7869]
[2,7058; 2,7331]	[2,6922; 2,7194]	[0,7860; 0,7875]	[0,7860; 0,7886]	[0,7860; 0,7870]
[2,7058; 2,7086]	[2,7194; 2,7222]	[0,7860; 0,7874]	[0,7860; 0,7886]	[0,7860; 0,7869]
[2,7331; 2,7606]	[2,6654; 2,6922]	[0,7860; 0,7874]	[0,7860; 0,7885]	[0,7860; 0,7869]
[2,7331; 2,7358]	[2,6922; 2,6949]	[0,7860; 0,7874]	[0,7860; 0,7885]	[0,7860; 0,7869]
[2,7606; 2,7884]	[2,6388; 2,6654]	[0,7860; 0,7871]	[0,7860; 0,7878]	[0,7860; 0,7867]
[2,7606; 2,7626]	[2,6654; 2,6674]	[0,7860; 0,7870]	[0,7860; 0,7879]	[0,7860; 0,7867]
[2,7884; 2,8023]	[2,6257; 2,6396]	[0,7860; 0,7864]	[0,7860; 0,7868]	[0,7860; 0,7863]

Table 2: Solutions optimales

5 - CONCLUSION ET TRAVAUX FUTURS

Nous avons étendu le paradigme des SCSPs au domaine continu et défini une résolution par transformation. Nous pouvons dorénavant utiliser la puissance d'expression de toutes les instances des SCSPs définies, car les c-demi-anneaux restent les mêmes. Aussi pensons nous développer un ou plusieurs c-demi-anneaux spécifiques au domaine continu. L'incertitude apporte des arguments qualitatifs aux solutions, nous pensons appliquer ces théories à la décision. Dans le domaine de la décision, l'approche multicritère joue un rôle important pour être au plus près des problèmes réels. Il serait intéressant de trouver un c-demi-anneau de dimension N pour traiter le multicritère ou d'avoir un c-demi-anneau permettant une agrégation "intelligente" des critères en un seul. Ces voies sont en cours d'exploration. D'autres peuvent être entreprises dans la théorie élargie pour les deux contextes discret et continu, voire sur des SCSPs mixtes

BIBLIOGRAPHIE

- [BGR02] Stefano Bistarelli, Philippe Codognot, and Francesca Rossi. (2002) "Abstracting soft constraints: Framework, properties, examples." *Artificial Intelligence*, Vol. 139, n° 2.
- [BGR00] Stefano Bistarelli, Rosella Gennari, and Francesca Rossi.(2000) "Constraint propagation for soft constraints: Generalization and termination conditions." *Principles and Practice of Constraint Programming*, p. 83–97.
- [BMR95] S. Bistarelli, U. Montanari, and F. Rossi. (1995) "Constraint solving over semirings." *IJCAI-95*, p. 624–630.
- [BMR+99] S. Bistarelli, U. Montanari, F. Rossi, T. Schiex, G. Verfaillie, and H. Fargier. "Semiring-based cps and valued cps : Framework, properties and comparaison. ", *Constraints*, Vol. 4, n° 3, p. 199–240.
- [BMR02] S. Bistarelli, U. Montanari, and F. Rossi. (2002) "Soft concurrent constraint programming." *European Symposium on Programming*, p. 53–67.
- [CR00] Philippe Codognot and Francesca Rossi. (2000) "Tutorial notes : Solving and programming with soft constraints : Theory and pra-tice." *ECAI2000*.
- [SFV97] Thomas Schiex, Hélène Fargier, and Gérard Verfaillie. (1997),"Problèmes de satisfaction de contraintes valués. ", *Revue d'intelligence artificielle*, Vol. 11,n° 2, p. 339–373.
- [PM95] Steven Prestwich and Shyan Mudambi.(1995), "Improved branch and bound in constraint logic programming." *Principles and Practice of Constraint Programming CP'95*, First International Conference, Cassis, France, September 19-22, p. 533–548.
- [Let01] Grégory Letribot (2001), "Optimisation et études paramétriques dans le cadre de CSP numériques", *Rapport de stage de DESS d'Intelligence Artificielle de l'Université Paris VI*, Dassault-Aviation.

***ANALYSE HORS LIGNE D'APPLICATIONS TEMPS REEL FORTEMENT COUPLEES
COMPORTANT DES TACHES A DUREES VARIABLES***

Stéphane Pailler,

Doctorant en Informatique
stephane.pailler@ensma.fr, + 33 5 49 49 83 36

Annie Choquet-Geniet,

Maître de conférences HDR en Informatique
ageniet@ensma.fr + 33 5 49 49 80 68

Adresse professionnelle

LISI - ENSMA ★ Téléport 2 – 1 Avenue Clément Ader
BP 40109 ★ F-86961 Futuroscope Cedex

Résumé : Nous proposons une méthodologie d'ordonnancement hors ligne d'applications temps réel fortement couplées. Nous mettons en évidence l'importance des variations de la durée d'exécution des tâches qui peuvent mettre en péril la sûreté de fonctionnement du procédé contrôlé. Nous proposons alors une méthode pour tenir compte de ces variations en intégrant explicitement dans le processus de validation du système les instructions conditionnelles du code des tâches. Après avoir adapté le modèle temporel de tâches à ce contexte, nous modélisons ces applications à l'aide de réseaux de Petri autonomes fonctionnant sous la règle de tir maximal et munis d'ensemble terminaux. Nous définissons deux concepts d'ordonnançabilité : l'ordonnançabilité locale et l'ordonnançabilité globale d'où découle la notion de graphe d'ordonnancement. Puis, nous montrons comment extraire ces graphes d'ordonnancement à partir du graphe d'ordonnançabilité globale.

Summary : We propose an off-line scheduling methodology for highly coupled real time applications. We show that variations in the computation time of tasks may hazard the safeness of the controlled process. Thus, we have chosen to take conditional instructions of tasks' code explicitly into account, in order to reduce the potential failures. After the adaptation of the task's temporal model to this context, we model these applications using autonomous Petri nets which run under the earliest firing rule with terminal marking set. We define two concepts of schedulability: the local schedulability and the global one and we define the concept of scheduling graph. Finally, we show how to obtain a scheduling graph from the graph of global schedulability.

Mots clés : Ordonnancement hors ligne, réseau de Petri, durées variables, graphe d'ordonnancement, instructions conditionnelles.

Analyse hors ligne d'applications temps réel fortement couplées comportant des tâches à durées variables

Un nombre sans cesse croissant de procédés autonomes ou assistés (système embarqué des voitures, centrale nucléaire, aéronautique, sonde spatiale...) sont contrôlés par des systèmes temps réel interagissant avec leur environnement. Pourtant, ces systèmes ne sont pas dépourvus d'erreurs qui peuvent mettre en danger des vies humaines ou occasionner une perte d'une valeur économique conséquente. C'est pourquoi, il est nécessaire de développer des méthodes permettant de minimiser ces risques en validant les applications temps réel qui contrôlent ces procédés critiques.

Dans la plupart des cas, ces erreurs proviennent d'un non-respect des contraintes temporelles. En effet, une application temps réel se définit comme une application multi-tâches, chacune d'elle étant assujettie à des contraintes temporelles inhérentes au procédé contrôlé. Par conséquent, le problème majeur consiste en un choix d'une politique d'ordonnancement permettant de distribuer au(x) processeur(s) les instances des tâches de façon à respecter chacune des contraintes temporelles.

Il existe deux méthodes pour y parvenir: la première consiste en l'implantation d'un algorithme d'ordonnancement qui choisira à chaque instant la tâche qu'il est nécessaire d'exécuter, c'est l'ordonnancement en ligne. La seconde consiste en une analyse préalable de l'application et des séquences d'ordonnancement possibles sur un intervalle de temps approprié: c'est l'ordonnancement hors ligne. L'analyse d'ordonnancement s'appuie sur le modèle temporel de tâche [LIU,1973],[STANKOVIC, 1998] permettant de caractériser chaque tâche à l'aide des paramètres temporels suivants: R_i , la date de

première activation de la tâche i , D_i son échéance relative de fin d'exécution, T_i sa périodicité d'activation, C_i sa durée d'exécution calculée au pire cas sur l'architecture cible. Nous adopterons par la suite, la notation $\langle R_i, C_i, D_i, T_i \rangle$ pour définir une tâche τ_i .

Beaucoup de travaux ont permis la réalisation d'outils de validation dans des contextes d'ordonnancement hétérogènes: contraintes temporelles strictes ou souples, applications fortement couplées (présence de primitives temps réel comme le partage de ressources et les communications), réduction du temps de réponse... Pourtant, rares sont les outils effectivement utilisés dans le monde industriel. En effet, les variations de la durée d'exécution d'une tâche dues principalement à la structure du code de la tâche mais également au non déterministe du processeur qui l'exécute (en raison des pipelines, caches, exécutions spéculatives etc.), remet en cause bien trop souvent la pertinence des outils d'analyse d'ordonnancement classique. La durée d'exécution C_i habituellement prise en compte dans le modèle temporel provient du calcul d'une borne supérieure de durée WCET (Worst Case Execution Time) qui doit répondre à deux principaux critères: la fiabilité, ce qui certifie qu'aucune exécution de la tâche considérée ne sera de durée supérieure au WCET, et la précision qui permet réduire sa surestimation [PUSCHNER, 2000], [PUSCHNER, 1989], [HEALY, 1998]. Malgré cela, dans un contexte avec primitives temps réel, il peut arriver que l'exécution d'une tâche soit très inférieure au WCET, ce qui peut conduire à une faute temporelle. L'exemple de la figure 1 illustre ce type d'anomalie mise en évidence par [Graham, 1969].

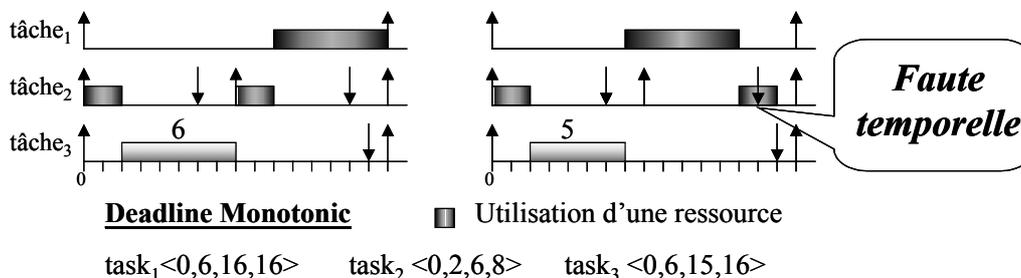


Figure 1 : Exemple d'une anomalie d'ordonnancement en raison d'une réduction de la durée d'exécution de la tâche τ_3 .

De plus, le fait de considérer uniquement le chemin d'exécution de durée maximale, oblige également à prendre en compte toutes les primitives temps réel présentes dans chacun des comportements de la tâche. Par conséquent, les tests d'ordonnabilité sont effectués sur un modèle sur-contraint.

Pour palier ce problème, nous suggérons une méthode qui étend le modèle temporel de Liu Layland pour permettre d'isoler chaque chemin d'exécution des tâches, chacun d'eux pouvant comporter des primitives temps réels. Puis, nous utilisons une modélisation par réseaux de Petri pour analyser l'ordonnabilité de ces applications fortement couplées.

Dans une première partie, nous introduisons un nouveau modèle temporel et nous redéfinissons le problème d'ordonnement associé à cette extension. Puis dans une seconde partie nous réutilisons la modélisation par réseaux de Petri autonomes, fonctionnant sous la règle de tir maximal et ensembles terminaux de [CHOQUET-GENIET, 1996], [GROLLEAU, 2000]. Enfin, dans une dernière partie, nous montrons comment extraire un graphe d'ordonnement décrivant un comportement valide de l'application à partir du graphe des marquages.

1 – EXTENSION DU MODELE DE TACHE

Nous considérons des applications fortement couplées composées de tâches périodiques à départ simultané (les dates de réveil R_i sont toutes égales à 0). Notre objectif consiste à isoler les différents chemins d'exécution possibles de chaque tâche. Pour cela, nous prenons en compte explicitement les instructions conditionnelles (IF THEN ELSE) du code des tâches et associons à chaque comportement induit une pire durée d'exécution. Nous pouvons alors étendre le modèle de Liu Layland en considérant non pas une durée d'exécution C_i mais un *multi-ensemble* de durée E_i pour chaque tâche. Ce multi-ensemble s'apparente au modèle multiframe de [MOK, 96] sans toutefois la connaissance préalable de la durée d'exécution de la tâche pour chaque instance successive. Nous proposons ici une méthode permettant de tenir compte de toutes ces durées pour chaque instance. On peut noter que dans le cas où une tâche ne possède pas d'instruction conditionnelle, ce multi-ensemble est réduit à une unique durée qui correspond au modèle temporel classique.

Comme nous nous sommes placés dans le cadre d'une étude hors ligne, nous devons tenir compte de l'indécision de chaque test conditionnel pour l'analyse d'ordonnabilité. Pour cela nous introduisons la notion de *graphe d'ordonnement*: un graphe d'ordonnement est

un graphe où chaque branche correspond à un ordonnancement de l'application en ne considérant pour chaque instance de tâche qu'un seul chemin d'exécution. Nous devons reformuler le problème d'ordonnement d'une application comportant des durées d'exécution variables, en définissant les deux concepts suivants:

- une application est dite *localement ordonnable* si tous les chemins d'exécution de chaque tâche sont indépendamment ordonnables.

- une application est dite *globalement ordonnable* s'il existe au moins un graphe d'ordonnement valide. On constate dans ce cas, qu'à chaque instruction conditionnelle d'une tâche à un instant t , il existe une branche d'ordonnement valide à partir de t , quel que soit le résultat du test conditionnel.

Nous pouvons noter ici que l'ordonnabilité globale implique l'ordonnabilité locale mais l'inverse est vrai uniquement dans un contexte sans primitives temps réel. Pour notre méthodologie d'analyse d'ordonnabilité hors ligne, nous devons donc considérer uniquement l'ordonnabilité globale qui permet de s'abstraire de l'incertitude des tests conditionnels. Nous cherchons donc à obtenir un graphe d'ordonnement.

2 MODELISATION PAR RESEAU DE PETRI

Le méthode d'analyse d'ordonnabilité que nous proposons reprend celle présentée dans [GROLLEAU, 2000], [CHOQUET-GENIET, 1996], [GROLLEAU, 1999]. Elle consiste en une modélisation d'une application par réseau de Petri coloré fonctionnant avec la règle de tir maximal. On peut ainsi obtenir tous les ordonnancements valides grâce à la construction du graphe des marquages.

Cette modélisation se décompose en deux parties:

- Un système de tâche qui est construit de façon classique à partir des blocs fonctionnels du code des tâches. On peut noter qu'une transition correspond à une action de durée unitaire et que chacune d'elles est en concurrence pour obtenir le jeton de la place processeur, nécessaire à leur franchissement. Par conséquent à chaque temps, une unique transition du système de tâches est franchie.

- Un système d'horlogerie discrétisé basé sur [FOHLER, 1994], [KOPETZ, 1992] qui comprend une horloge externe (RTC) qui capte le temps dans chaque place $Time_i$, cette dernière agissant comme une horloge locale pour réactiver de façon périodique la tâche i correspondante.

La figure 2 résume les différentes caractéristiques de cette modélisation. La règle de tir maximal impose au réseau de Petri le franchissement à chaque unité de temps, d'une transition du système de tâches (sauf si la charge total est nulle). Il apparaît immédiatement que seules les séquences d'ordonnancement conservatives peuvent être produites. Or, pour augmenter les capacités d'ordonnancement de notre modèle, nous devons

également considérer les séquences non conservatives. Pour cela, nous introduisons une tâche supplémentaire nommée *tâche oisive* (τ_0) qui simulera l'inactivité du processeur. Ainsi, à chaque franchissement d'une transition de la tâche oisive, le processeur reste inactif: les séquences d'ordonnancement non conservatives sont ainsi produites.

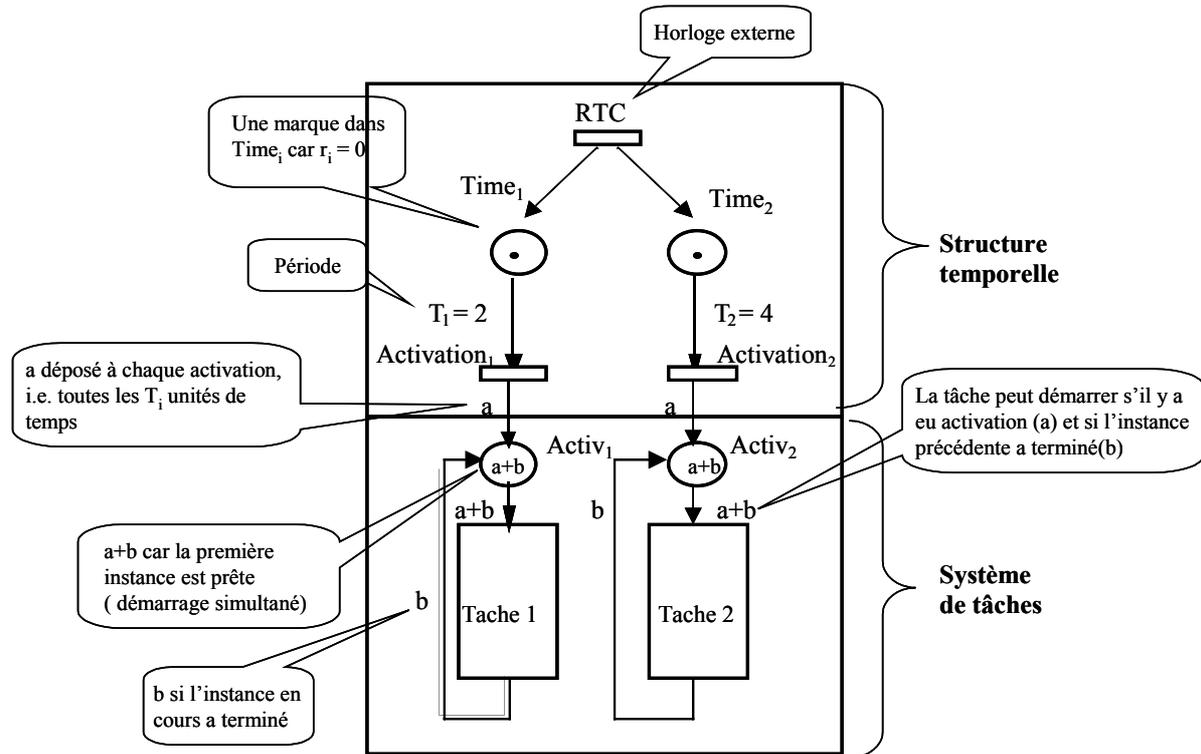


Figure 2 : Modélisation par réseau de Petri pour une application comportant 2 tâches avec respectivement des périodes de $T_1=2$ et $T_2=4$. La place processeur n'est pas représentée.

Les paramètres temporels de la tâche oisive dépendent directement du facteur de charge processeur (noté U). Considérons la métapériode P comme le plus petit multiple commun des périodes des tâches alors, puisque les séquences d'ordonnancement sont cycliques et de période P (on rappelle que les tâches démarre à l'instant 0), on peut définir le nombre exacte de temps d'inactivité processeur: $P(1-U)$. On peut donc définir la tâche oisive τ_0 par $\langle R_i=0, C_i=P(1-U), D_i=T_i=P \rangle$.

Toutefois, l'utilisation du modèle temporel étendu rend le facteur de charge processeur indéterministe puisqu'il dépend des chemins d'exécution empruntés et donc des choix des instructions conditionnelles. Pour calculer la durée de la tâche oisive, nous avons choisi de considérer les chemins d'exécution les plus longs pour chaque tâche, ce qui maximise le facteur de charge processeur et par conséquent réduit la durée de la tâche oisive.

Lorsqu'une tâche s'exécute en passant par un chemin de plus courte durée, la durée de la tâche oisive est alors incrémentée de la différence de durées. Pour permettre ce dynamisme de la durée de la tâche oisive, nous avons choisi de la modéliser par: une seule transition en concurrence avec les transitions des autres tâches, pour représenter un temps d'inactivité processeur à chaque franchissement et, une place comportant autant de jetons qu'il doit y avoir de temps oisif sur P . La figure 3 illustre la modélisation d'une tâche oisive.

Nous nous intéressons ici à l'exploitation du graphe des marquages induit par la modélisation par réseau de Petri, et plus particulièrement à l'extraction d'un graphe d'ordonnancement valide. Dans un premier temps, l'exploitation directe du réseau de Petri permet d'obtenir un graphe des marquages de profondeur P (le système est dans le même état à

une profondeur de t et à $t + k \times P$ ($k > 0$) du fait de la cyclicité des séquences d'ordonnement).

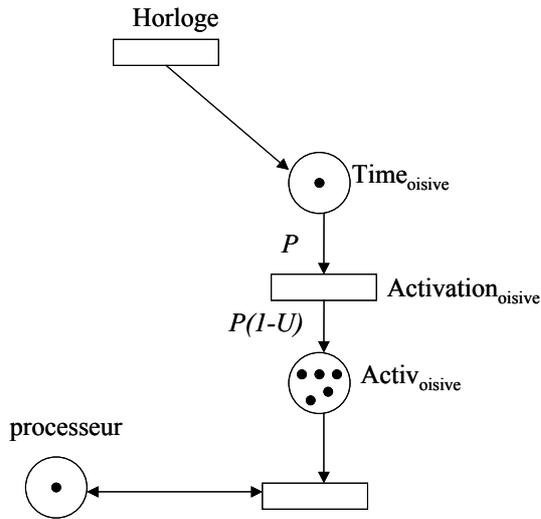


Figure 3 : Exemple d'une modélisation par réseau de Petri d'une tâche oisive

3 - ANALYSE HORS LIGNE DU GRAPHE DES MARQUAGES

Ce graphe des marquages comporte des états sans successeurs, c'est à dire des branches ne permettant pas d'arriver à la hauteur P en raison d'une erreur d'ordonnement (échéance non respectée). Il est donc nécessaire d'épurer le graphe pour obtenir au final un *graphe d'accessibilité* qui contient toutes les séquences d'ordonnement valides pour chaque chemin d'exécution des tâches: nous sommes alors dans un contexte d'ordonnement local. Or, nous nous intéressons à l'ordonnement global pour extraire un graphe d'ordonnement certifiant l'existence d'un chemin d'exécution quelque soit le résultat des instructions conditionnelles. Nous devons donc une nouvelle fois épurer le graphe d'accessibilité pour supprimer toutes les branches d'ordonnement issues d'un test conditionnel à une hauteur h dans le graphe qui ne permettent pas d'emprunter une succession d'états jusqu'à la hauteur P correspondant à l'alternative de ce test de hauteur h . A chaque instant où un choix est fait sur le chemin d'exécution à emprunter, le chemin correspondant à l'alternative de ce choix doit être possible au même instant. Nous obtenons alors un *graphe d'accessibilité global*.

La plus grande difficulté réside ensuite dans l'extraction d'un graphe d'ordonnement à partir du graphe d'accessibilité global. En effet, la taille d'un graphe d'ordonnement est d'autant plus grande qu'il y a de chemins d'exécution possibles pour chaque tâche.

Propriété : Considérons n tâches définies par $\langle R_i, E_i, D_i, T_i \rangle$ pour $1 \leq i \leq n$, ne comportant pas de primitives temps réel :

- Le nombre de comportements possibles pour une tâche τ_i sur la métapériode P est

$$\langle E_i \rangle_{T_i}^{\frac{P}{T_i}}$$

- Le nombre de comportements possibles dans un graphe d'ordonnement est

$$\prod_1^n \langle E_i \rangle_{T_i}^{\frac{P}{T_i}}$$

$\langle \cdot \rangle$ représentant le nombre d'éléments de E_i .

Le graphe d'accessibilité globale est composé de l'ensemble des graphes d'ordonnement possibles, on se rend donc facilement compte de l'explosion combinatoire rencontrée lors de l'énumération des graphes d'ordonnement valides. Malgré la présence de primitives temps réel qui diminue fortement le nombre de graphes d'ordonnement dans le graphe d'accessibilité global, il est de toute évidence nécessaire d'éviter sa construction dans son ensemble. C'est pourquoi, nous adoptons des techniques d'optimisation par contraintes [GROLLEAU, 1999] permettant de réduire la construction du graphe d'accessibilité globale pour la recherche d'un arbre d'ordonnement. On peut citer les contraintes de successeurs qui empêche la préemption des blocs indépendants des tâches, ainsi que des contraintes *a priori* qui permettent de ne pas construire des branches d'ordonnement qui ne vérifient pas certains critères de qualités d'ordonnement comme un seuil maximum de temps de réponse ou de gigue. Toutefois, ce dernier type de contrainte nécessite d'être redéfini pour s'appliquer non pas à une séquence mais à un graphe d'ordonnement. En effet, en considérant une contrainte de temps de réponse pour une tâche qui comprend plusieurs chemins d'exécution, le temps de réponse peut s'appliquer soit à l'un des chemins (le plus court ou le plus long par exemple) soit à la moyenne des différents chemins. Le résultat obtenu sera bien évidemment totalement différent. Par défaut nous considérons uniquement le temps réponse maximal de tous les chemins d'exécution de la tâche, mais nous travaillons actuellement sur les différentes possibilités qu'offre ce type d'optimisation sur le choix des graphes d'ordonnement.

Une autre voie que nous explorons également consiste à s'intéresser plus particulièrement aux placements des temps d'inactivités processeur. En effet, il pourrait être avantageux de les regrouper pour offrir une période de traitement indépendante au processeur, comme par exemple la simulation d'une tâche serveur pour le traitement de tâches aperiodiques, ou pour un traitement de fond non critique.

La figure 4 montre un exemple de graphe d'accessibilité globale sur lequel est appliqué une

contrainte de successeur, puis un graphe d'ordonnancement extrait.

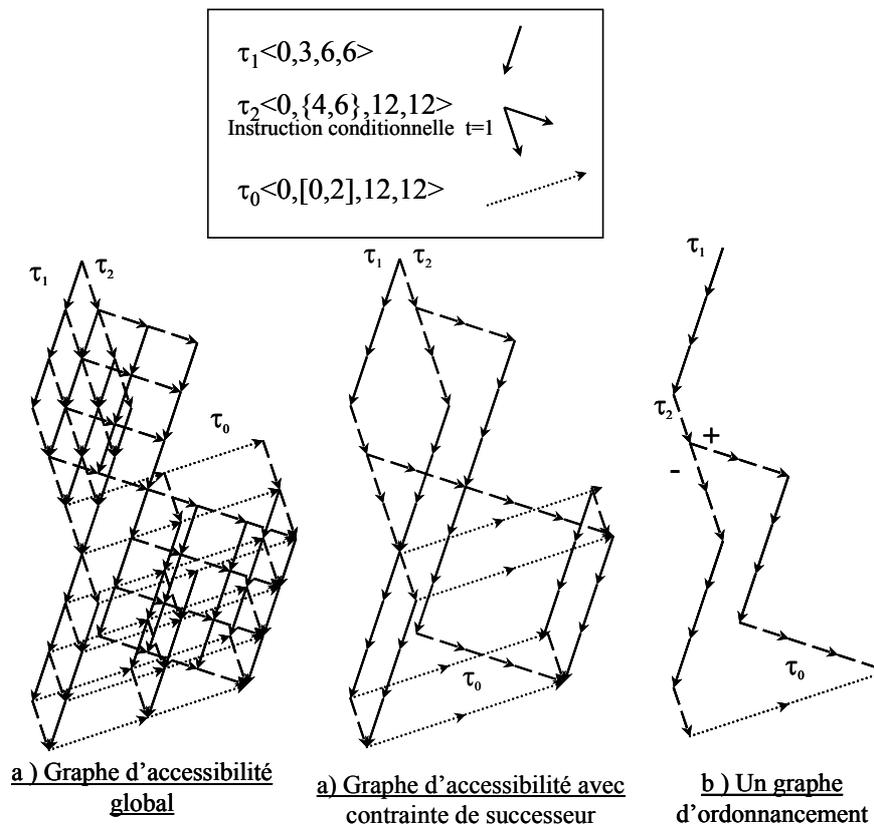


Figure 4 : Le graphe d'accessibilité (a) étant de taille très importante il est nécessaire de lui appliquer des contraintes lors de sa construction, comme des contraintes de successeur(b), pour pouvoir en extraire un graphe d'ordonnancement(c).

4 – CONCLUSION

Nous avons proposé une méthodologie pour l'analyse d'ordonnabilité d'applications temps réel fortement couplées, qui utilise une modélisation par réseau de Petri. Nous avons montré que les variations de la durée d'exécution doivent être prises en compte pour éviter certains problèmes d'instabilité, malgré les avancées dans le domaine de l'estimation des WCET. L'indéterminisme des processeurs, l'utilisation de compilation optimisée et le mode programmation actuelle augmente le nombre de comportement d'exécution pour chaque tâche et rend impossible la mesure exacte des WCET [PUSCHNER, 2002]. C'est pourquoi nous proposons une méthodologie qui tire avantage des études du WCET en intégrant dans la modélisation par réseaux de Petri, les instructions conditionnelles du code des tâches. Après avoir étendu le modèle temporel de tâche de Liu Layland, nous avons déduit deux concepts d'ordonnabilité : l'ordonnabilité locale et

globale. Nous avons par la suite introduit la notion de tâche oisive à durée dynamique qui est directement dépendante de la fluctuation des différents chemins d'exécution des tâches. Après avoir montré comment modéliser ce type d'applications par réseaux de Petri, nous en avons extrait un graphe des marquages. Il a été ensuite nécessaire de filtrer ce graphe pour supprimer les séquences d'ordonnancement non valides de façon à ne garder qu'un graphe d'accessibilité globale formé de l'ensemble des graphes d'ordonnancement valides. Nous avons montré par la suite comment en extraire un graphe d'ordonnancement en appliquant des contraintes permettant de réduire la construction du graphe d'accessibilité.

L'étape suivante de cette étude consistera à étendre les possibilités d'extraction des graphes d'ordonnancement tout en optimisant leur recherche.

BIBLIOGRAPHIE

- [CHOQUET-GENIET, 1996] A. CHOQUET-GENIET, D. GENIET et F. COTTET, “*Exhaustive computation of the scheduled task execution sequences of a real-time application*”. Proc. of the 4th International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems p. 246–262 (Uppsala, Sweden), 1996.
- [FOHLER, 1994] G. FOHLER, “*Flexibility in statically scheduled hard real-time systems*”. Thèse, University of Wien, April 1994.
- [GROLLEAU, 2000] E. GROLLEAU et A. CHOQUET-GENIET, “*Off line computation of real-time schedules by means of Petri nets*”. Workshop On Discrete Event Systems WODES2000, Discrete Event Systems: Analysis and Control, Kluwer Academic Publishers, p. 309–316, Ghent, Belgium, 2000.
- [GRAHAM, 1969] R. GRAHAM, “*Bounds on multiprocessing timing anomalies*”. SIAM Journal of Applied Mathematics 17, no. 2, p.416-429, 1969.
- [GROLLEAU, 1999] E. GROLLEAU, “*Ordonnement temps réel hors-ligne optimal à l’aide de réseaux de Petri en environnement monoprocesseur et multiprocesseur*”. Thèse, ENSMA – Université de Poitiers, Novembre 1999.
- [HEALY, 1998] C. HEALY, M. SJÖDIN, V. RUSTAGI et D. WHALLEY, “*Bounding loop iterations for timing analysis*”. 4th IEEE Real-Time Technology and Applications Symposium (RTAS '98), Discrete Event Systems: Analysis and Control, IEEE, p. 12–21, 1998.
- [KOPETZ, 1992] H. KOPETZ, “*Sparse time versus dense time in distributed real-time systems*”. 12th Int. Conf. On Distributed Computing Systems, p.460–467, Japon, June 1992.
- [LIU, 1973] C. LIU et J. LAYLAND, “*Scheduling algorithms for multiprogramming in real-time environment*”. Journal of the ACM 20(1), p.46–61, (1973).
- [MOK, 1996] A. MOK et D. CHEN, “*A multiframe Model for Real Time Tasks*”. in Proceedings of the 17th Real-Time System Symposium, pp. 22-- 29, Dec. 1996.
- [PUSCHNER, 2002] P. PUSCHNER, “*Is worst-case execution-time analysis a non problem? - towards new software and hardware architectures*”. 2nd Intl. Workshop on WCET Analysis 14th Euromicro Conference on Real- Time Systems, June 2002.
- [PUSCHNER, 2000] P. PUSCHNER et A. BURNS, “*Guest editorial: A review of worst-case execution-time analysis*”. Real-Time Systems 18, p.115–128, 2000.
- [PUSCHNER, 1989] P. PUSCHNER et C. KOZA, “*Calculating the maximum execution time of real-time programs*”. Journal of Real-Time Systems 1(2) p. 159–176, 1989.
- [STANKOVIC, 1998] J. STANKOVIC, M. SPURI, K. RAMAMRITHAM et G. C. BUTTAZZO, “*Deadline scheduling for real-time systems*”. vol. ISBN 0 7923 8269 2, 1998.

***COMPARAISON D'APPROCHES DE SIMULATIONS DISTRIBUEES A EVENEMENTS
DISCRETS D'ENTITES SPATIALISEES***

Gauthier Quesnel,

Doctorant en informatique

quesnel@lil.univ-littoral.fr , + 33 3 21 46 56 79

Raphael Duboz,

Doctorant en informatique

duboz@lil.univ-littoral.fr + 33 3 21 46 56 79

Eric Ramat,

Maître de conférences en informatique

ramat@univ-littoral.fr + 33 3 21 46 56 73

Adresse professionnelle

Laboratoire d'Informatique du Littoral Mission de la Recherche Blaise Pascal

50, rue Ferdinand Buisson - BP 719 62228 CALAIS cedex FRANCE

Résumé : Dans cet article, nous voulons répondre à la question suivante : quelle technique choisir pour la simulation à événements discrets de particules interagissant localement dans un espace continu à trois dimensions, et ceci dans un cadre distribué ? Pour cela, nous comparons les performances de trois méthodes classiques de simulation distribuée : une approche optimiste et deux approches pessimistes (faible et forte). Cette étude nous a amené à choisir la technique pessimiste faible comme une réponse possible.

Summary : In this article, we adress a particular issue: which technique to choose for a distributed discret event simulation of interacting particles in a three-dimensional continuous space? To answer, we compare three techniques of distributed simulation: strong synchronous pessimistic, weak asynchronous and optimistic. This study lead us to choose the weak asynchronous one.

Mots clés : Modélisation centrée individus, Simulation distribuée, Événements discrets, MPI.

Comparaison d'approches de simulations distribuées à événements discrets d'entités spatialisées

1 INTRODUCTION

Ce travail se situe dans une perspective d'étude de la dynamique des écosystèmes marins à petite échelle. On s'intéresse en effet plus particulièrement à l'interaction entre des cellules de phytoplancton (algues microscopique) et des animaux zooplanctonique (Ramat, 2003). Les interactions identifiées sont principalement les captures du phytoplancton par les prédateurs. Nous avons développé et simulé en monoprocesseur des modèles (Duboz 2001) de ce système pour des tailles de population de l'ordre de 10^5 phytoplanctons et 100 animaux. Ces quantités nous ont permis d'étudier la dynamique du système sur des échelles de temps et d'espace réduites.

La deuxième étape est donc de développer des modèles de simulation capable d'atteindre des tailles de population bien supérieures aux tailles actuelles (de l'ordre de 10^6 phytoplanctons et 10^3 prédateurs. Ce facteur 10 est impossible à atteindre en monoprocesseur faute de mémoire et de puissance.

Le travail présenté dans cet article est une première étape indispensable permettant de valider un modèle de simulation distribué.

Afin de réduire la complexité du problème, nous avons éliminé les interactions entre proies et prédateurs, ce qui réduit le système à un ensemble de particules effectuant une marche aléatoire simulant ainsi un mouvement brownien. Nous nous sommes focalisés sur deux éléments : la gestion du déplacement des prédateurs (que l'on nommera par la suite particules) et la synchronisation des événements liées à ces déplacements.

Ce dernier point est fondamental. En effet, dans le système que l'on cherche à développer, les particules interagissent et ces interactions nécessitent obligatoirement une synchronisation temporelle. A $t=0$, chaque particule est positionnée dans l'espace. Elles se déplacent et se situent donc à chaque instant à un endroit précis dans cet espace. Deux particules peuvent alors interagir si elles se situent au même instant dans la même zone.

Cette contrainte n'est pas toujours vérifiée dans les modèles de la littérature (Miguet 1995).

En simplifiant le modèle, nous avons fait disparaître, les calculs complexes qui sont réalisés au sein des particules (calculs liés au comportement et à la dynamique interne des prédateurs). Nous pallierons à ce biais en ajoutant fictivement une charge de calcul au sein des particules. Cette charge de calcul fera l'objet d'une étude afin de déterminer son impact sur les stratégies adoptées.

Le système simplifié possède un précieux avantage : on connaît parfaitement son comportement ce qui nous permettra de valider nos algorithmes.

Dans la première partie, nous abordons la description de notre modèle de mouvement brownien. En seconde partie, nous étudions trois techniques de distribution de calcul, deux pessimistes et une optimiste. Dans la troisième partie nous montrons les résultats des simulations. Enfin, dans la dernière partie, nous engagerons une discussion sur les stratégies possibles.

2 LE MODÈLE

Le système que l'on étudie est un ensemble de particules situés dans un espace à trois dimensions et pourvu d'un mouvement brownien. Nous avons modélisé ce mouvement de la manière suivante :

- à un instant t , la particule se situe à une position représentée par un triplet (x, y, z) dans un espace à trois dimensions.
- la particule possède une direction (x, Y, Z) dont chacune des composantes est initialisées aléatoirement entre -1 et +1.
- la particule possède une vitesse de déplacement générée aléatoirement selon une distribution gaussienne bornée de moyenne 1, d'écart type 0.2 et de limites 0 et 2. Ces valeurs sont inspirées de la vitesse observée des prédateurs.
- la particule se déplace pendant un temps Δt généré aléatoirement selon la même distribution que la vitesse.

À l'issue de Δt , les particules possèdent une nouvelle position et les composantes du mouvement sont régénérées aléatoirement.

Ce modèle nous permet d'identifier les événements du système qui sont tout simplement les instants de changement de direction. Cette approche s'inscrit dans le cadre de la simulation à événement discret. Entre deux changements, la particule se déplace sur une droite. Si on replace le discours dans le cadre de la simulation à événements discrets, l'avancement du temps est dirigé par les événements.

Le temps progresse donc dans la simulation au fur et à mesure que les événements sont traités. Pour gérer la chronologie des événements, ceux-ci sont triés par ordre croissant dans une liste appelée *échéancier*.

Au départ de la simulation, toutes les particules sont positionnées aléatoirement dans l'espace et possèdent un vecteur de direction. Le couple « *particule - temps de déplacement* » est ensuite inséré dans l'échéancier. L'échéancier est ensuite dépilé, afin de sélectionner la première particule devant se déplacer.

Suite au déplacement, deux cas de figure sont à envisager. En effet, nos particules sont plongées dans un espace borné et la stratégie de distribution que l'on va adopter, comme dans la majorité des modèles de ce type (Miguet 1995), implique un découpage de l'espace. Il y a donc deux types de frontières : celles liées à la limite de l'espace modélisé et celles liées au changement de sous espaces. Il y a donc deux politiques de déplacement lorsqu'une particule atteint une frontière. Dans le premier cas, nous itérons le processus de génération des composantes de déplacement jusqu'à ce qu'ils soient orientés dans l'espace valide. Dans le deuxième cas, la particule devra se situer dans un nouveau sous-espace après avoir franchi la frontière. Dans ce cas nous ne changeons pas les composantes du déplacement, seul change le temps de déplacement qui est diminué du temps de déplacement déjà effectué pour arriver sur la frontière.

3 LES TECHNIQUES

La stratégie de distribution adoptée concerne tout d'abord le découpage de l'espace en sous-espaces en y associant les particules appartenant à ces sous-espaces. Cette stratégie

a tendance à repartir à la fois la charge de calcul et le besoin en terme de stockage d'information.

La question qui se pose alors est la suivante : quel est l'algorithme de simulation distribuée optimal dans notre cas ? Nous rappelons que l'aspect synchronisation globale des événements, connu sous l'appellation causalité que nous développerons dans le cadre général, est vital dans l'hypothèse où les particules doivent interagir.

Nous allons dans la suite développer trois algorithmes de simulation distribuée (Fujimoto 1999), puis nous étudierons leurs conséquences sur notre modèle.

3.1 La causalité

Les systèmes physiques obéissent toujours au *principe de causalité*. Il peut être énoncé de la façon suivante : le futur n'influence jamais le passé. La causalité impose donc un ordre des transitions d'états dans les systèmes physiques. Lamport (Lamport 1978) a défini une méthode basée sur l'utilisation d'estampilles pour permettre aux processus de respecter la causalité. L'ordre dans lequel le simulateur produit les événements doit être cohérent pour que le simulateur soit un reflet juste du système physique simulé. Par exemple, si la transition A du système physique intervient à la date 3 et à une influence sur la transition B intervenant à la date 5, le simulateur doit générer l'événement modélisant la transition A avant celui concernant la transition B.

La distribution d'une simulation sur plusieurs machines pose un problème pour le respect du principe de causalité. En effet, les calculs étant répartis, l'avancement du temps simulé peut être différent sur chaque machine. Le problème principal est alors de définir des techniques de synchronisation des calculs. Dans ce qui suit, nous présentons trois techniques possibles. Nous les évaluons en terme de temps de calcul par rapport à une approche monoprocesseur afin de choisir la plus adaptée à notre problème.

3.2 Approche pessimiste : synchrone forte

La première approche, la plus simple, repose sur le principe suivant : chaque simulateur prend la décision de faire avancer la simulation de son sous-espace uniquement quand il est sûr de pouvoir le faire. Cette technique est dite pessimiste et fait partie des premiers algorithmes de synchronisation

(Chandy, 1978). Elle sera reprise dans l'approche synchrone faible.

Dans le cas de l'approche synchrone, un simulateur peut traiter tout événement qui n'a pas de conséquence sur les simulateurs jusqu'à une certaine date appelée barrière de synchronisation. Dans notre cas, nous sommes obligés de considérer que tout événement peut avoir une conséquence d'où l'algorithme adopté. Un processus central gère une horloge globale. Ce processus est appelé coordinateur et donne la main au simulateur qui possède l'évènement dont la date d'occurrence est la plus petite globalement. Cette technique s'inspire directement des travaux de Zeigler et des simulateurs abstraits associés (Zeigler 2000). Dans cette approche, il n'existe aucun parallélisme. Elle offre la possibilité de quantifier le coût en temps engendré par les communications entre les simulateurs, puisque globalement le simulateur synchrone possède un comportement identique au simulateur mono-processeur.

3.3 Approche pessimiste : asynchrone faible

La technique asynchrone pessimiste a pour principe de faire avancer chaque simulateur d'une manière asynchrone tant qu'aucun simulateur ne viole le principe de causalité.

Le principe est le suivant : chaque simulateur possède un échéancier qui fournit la date minimale (date jusqu'à laquelle un simulateur peut traiter les événements sans conséquence sur ces voisins). Celle-ci est envoyée à tous les simulateurs. Si toutes les dates reçues sont supérieures à sa date minimale, le simulateur traite son prochain événement et envoie à tous ses voisins sa date minimale. Le simulateur ne pourra recommencer à dépiler son échéancier seulement quand il aura reçu des dates supérieures à son temps courant des autres processus.

La technique est légèrement différente dans notre implémentation afin de mieux s'adapter à nos besoins (par exemple, utilisation d'un coordinateur pour éviter les DeadLock). Chaque simulateur possède une horloge et calcule une date jusqu'à laquelle aucun message ne peut être envoyé à un autre simulateur. Ces dates sont envoyées au coordinateur, celui-ci sélectionne la plus petite et demande aux simulateurs d'avancer jusqu'à cette date. Si le simulateur détenant la date maximum de déplacement autorisé, modifie sa prochaine date de sortie de particules, il l'envoie

au coordinateur qui demandera à tous les nœuds d'envoyer leurs dates maximums afin de sélectionner le simulateur qui détiendra la prochaine date de déplacement de particules.

Le calcul de la date minimale avec sortie d'une particule de l'espace, s'effectue comme suit :

Début

Pour toute particule contenue dans l'espace du simulateur **faire**

- Calcul pour chaque axe (abscisse, ordonné, profondeur), la date à laquelle la particule pourrait sortir en utilisant des vecteurs direction parallèle aux axes
- Sélection de la date minimale de sortie des trois dates.

Fin pour

Sélection de la date minimale de toutes les particules

Fin

Cette technique assure une intégrité du temps (l'horloge ne peut recevoir une date inférieure à celle courante) même si son fonctionnement reste asynchrone.

3.4 Approche optimiste : le Time Warp

Dans une approche optimiste, chaque processus décide de faire avancer la simulation, sans se préoccuper de l'état des autres. Dans ce cas, le principe de la causalité peut être corrompu. Pour rétablir ce principe, Jefferson (Jefferson 1985) propose une solution : le *Time Warp*.

Lorsqu'un processus s'aperçoit que le principe de la causalité n'est plus vérifié, c'est à dire que la date courante de la simulation est postérieure à la date du dernier événement reçu, il annule tous les messages déjà expédiés. Pour cela, il faut gérer un historique des états du système, mais aussi le contenu de tous les messages reçus et expédiés afin de pouvoir avertir les autres processus de faire des annulations. Ce retour en arrière dans le temps est appelé "BackTracking". Pour cela, nous utilisons des *anti-messages*, qui possèdent exactement les mêmes caractéristiques que les messages d'origine à l'exception d'un flag.

Ces messages sont générés dès qu'un problème de causalité survient. Ils sont

envoyés aux simulateurs pour annuler certains messages. Par exemple, un simulateur A envoie un message M1 au simulateur B qui arrive dans le futur de celui-ci. Si B envoie un message M2 dans le passé de A, celui-ci repasse à l'état sauvegardé où il se trouvait à la date d'arrivée du message M2. Le processus A doit annuler tous ses envois de messages qu'il a fait dans le futur, comme le message M1 en utilisant les *anti-messages*.

Après la description des trois méthodes de synchronisation, nous allons valider leurs modèles à l'aide d'un modèle déterministe.

3.5 Validation

La modélisation du mouvement brownien nous permet de valider notre modèle distribué par comparaison entre les résultats de simulation de la diffusion particulaire discrète avec un modèle mathématique déterministe (Duboz 2003). Nous plaçons toutes les particules au centre de l'espace au point (0, 0, 0). Nous simulons un certain temps t , ce qui permet, à l'aide d'une mesure de la distance moyenne parcourue par les particules, de calculer le coefficient de diffusion (Duboz 2003). Pour visualiser les résultats de simulation, nous découpons l'espace en sous espaces suivant l'axe des abscisses x et comptons le nombre de particules dans chaque sous espace afin d'obtenir une courbe de répartition des particules selon x . Nous cherchons à montrer que ces mesures respectent le modèle mathématique suivant (Fick 1855) :

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2}$$

où c est la concentration de particules, D le coefficient de diffusion et x l'axe sur lequel a lieu la diffusion.

À partir de cette valeur, nous pouvons calculer analytiquement la courbe de Fick donnée par l'équation suivante :

$$c(x, t) = \frac{c_0 e^{-x^2/4Dt}}{2\sqrt{\pi Dt}}$$

où c_0 représente la concentration initiale, $c(x, t)$ la concentration en un point à un instant donné et D le coefficient de diffusion.

Le résultat se présente sous la forme d'une gaussienne comparable à celle obtenue par simulation (figure 1).

Le résultat que nous obtenons montre effectivement que notre modèle distribué dans les cas synchrone fort et faible suit bien la courbe de Fick, validant ainsi notre modèle distribué par rapport au modèle monoprocesseur.

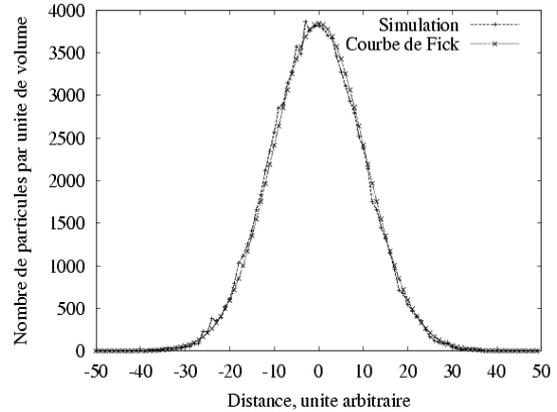


Figure 1 Nombre de particules suivant l'axe des x après une simulation de 30 secondes. La courbe en pointillés correspond au modèle analytique, elle se confond presque parfaitement avec la courbe en trait plein correspondant à la simulation particulaire.

4 LES RESULTATS

Après avoir vérifié empiriquement la validité de nos implémentations, nous allons décrire l'architecture matérielle et logicielle utilisée avant d'en analyser les performances.

4.1 Matériels et logiciels

La réalisation de ce simulateur est basée sur l'utilisation d'un cluster de PC et d'une bibliothèque de programmation réseau.

Nous avons utilisé la bibliothèque de communication réseau MPI (Message Passing Interface) qui nous permet d'assurer une correcte implémentation des approches synchrone et asynchrone :

- Les messages sont envoyés et reçus dans un file d'attente FIFO qui nous assure que si le simulateur A envoie deux messages M1 au temps $tA1$ et M2 au temps $tA2$ à B, le simulateur B les réceptionnera dans l'ordre M1 au temps $tB1$ et M2 en $tB2$.
- Les messages peuvent être envoyés de manière bloquante ou non bloquante.

- La sûreté, MPI vérifie en effet l'intégrité des paquets qu'il transfère sur le réseau.
- La rapidité de l'envoi et de la réception de messages est un point très important pour notre simulateur (l'ordre de grandeur du nombre de messages avoisine couramment le million lors d'une simulation), MPI est une bibliothèque de haut niveau très proche de la communication par socket.

Le deuxième point fondamental dans le type de travaux que l'on mène concerne les algorithmes de génération aléatoire. Étant au cœur de la validité de nos résultats, ce point n'est pas à négliger. De plus il faut prendre des précautions dès que l'on est dans un cadre distribué (L'Ecuyer 1998).

Le générateur de nombres aléatoires utilisé dans le simulateur est celui fourni par la bibliothèque Glib (bibliothèque de fonctions C utilisée dans le cadre du projet GNU¹). Ce générateur est basé sur le *Mersenne Twister* développé par M. Matsumoto et T. Nishimura (Matsumoto 1998). Son principal avantage est d'avoir une très longue période ($2^{19937}-1$ tirages indépendants). Dans notre système, un générateur est placé sur chaque nœud fournissant ainsi les nombres aléatoires pour le déplacement des particules. Cependant, son utilisation dans un environnement distribué n'est pas une bonne solution du fait que chaque nœud doit fournir une suite de nombres aléatoires non corrélés entre eux. Une des solutions est d'utiliser le générateur de nombres de P. L'Ecuyer (L'Ecuyer 2002) qui permet d'utiliser la même graine pour chaque génération et de démarrer la génération après avoir effectué un saut dans la séquence. Il faut néanmoins garantir que la séquence utilisée soient suffisamment longue pour éviter les chevauchements (choix du saut réalisé au démarrage).

Le cluster de PC utilisé est un cluster équipé de 8 nœuds, incluant chacun un biprocesseur AthlonXP 1800+, associé à 1 Go de mémoire vive. Les cartes réseaux sont des cartes Gigabits fournissant une bande passante 1Gb/s. Le système d'exploitation utilisé est un RedHat 8.0.

¹ API disponible sur <http://www.gnu.org>

4.2 Analyse des performances

La comparaison des performances est réalisée entre la version mono-processeur, qui sert de référentiel et les stratégies synchrone forte et synchrone faible. La stratégie asynchrone ne fait pas partie de ce comparatif. En effet, la charge due au backtracking s'avère trop coûteuse dans notre application et élimine d'entrée cette technique des choix possibles d'implémentation.

La figure 2 nous montre deux choses. La stratégie synchrone faible est plus coûteuse en temps total de simulation que la version mono-processeur ce qui était prévisible. Le surcoût observé est tout simplement l'œuvre des communications entre le coordinateur et les simulateurs. Il faut noter tout de même que ce coût est relativement faible ce qui nous autorise à dire que cette stratégie est intéressante dans le cas où les données attachées aux particules et aux sous-espaces manipulés par le simulateur ont une taille importante.

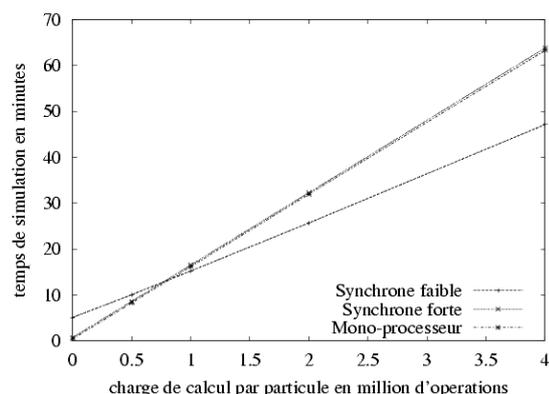


Figure 2 Variation du temps de simulation en fonction de la charge de calcul par particules ($30 \cdot 10^3$ particules simulées). L'approche synchrone faible est plus efficace à partir d'une charge d'environ 0,7 millions d'opérations par particules.

Le deuxième constant concerne l'évolution du temps de simulation dans le cas synchrone faible. Si la charge de calcul est négligeable voire nulle, il existe néanmoins un coût lié à la communication et à la synchronisation qui est incompressible. Ce qui implique que cette stratégie est moins bonne que notre stratégie de référence (mono-processeur) pour des charges de calculs inférieures à 0,7 million d'opérations. Au delà de cette valeur seuil, la stratégie synchrone faible est meilleure. Le constat reste

vrai quelque soit le nombre de particule (voir figure 3). On observe cependant une variation de ce seuil : plus le nombre de particules est élevé, plus la valeur de ce seuil est élevé.

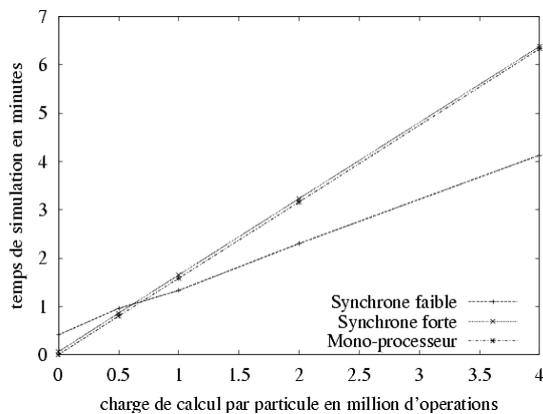


Figure 3 Variation du temps de simulation en fonction de la charge de calcul par particules ($3 \cdot 10^3$ particules simulées). L'approche synchrone faible est plus efficace à partir d'une charge d'environ 0,7 millions d'opérations par particules.

La figure 4 représente la variation du temps total de simulation en fonction de la charge de calcul par particule dans le cas synchrone faible en faisant varier le nombre de simulateur. Faire varier le nombre de simulateur consiste tout simplement à diviser l'espace en un nombre plus élevé de sous-espace. Le passage de 3 à 6 diminue sensiblement le temps de simulation. Mais on observe cette tendance qu'à partir d'un certain seuil de charge. Si on augmente le nombre de simulateurs (dans notre exemple à 9), on s'aperçoit que ce seuil s'éloigne rapidement. L'explication est simple. Si on mesure le nombre total de messages échangés entre les simulateurs et entre coordinateur et simulateurs, on observe une augmentation linéaire. En effet, plus le nombre de sous-espaces est élevé plus le nombre de messages est élevé.

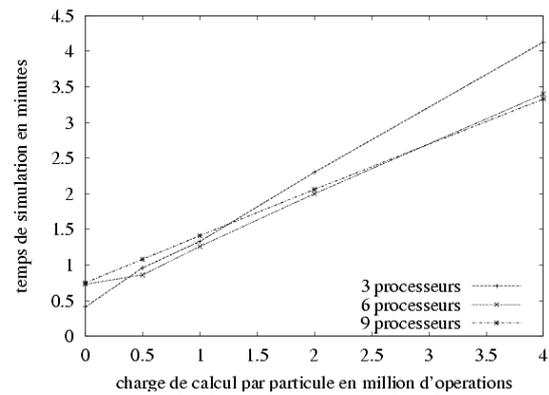


Figure 4 Variation du temps de simulation en fonction de la charge de calcul suivant l'approche asynchrone faible en utilisant 3, 6 et 9 processeurs.

5 CONCLUSION ET PERSPECTIVES

Dans cet article, nous avons comparé trois techniques de synchronisation pour des simulations à événements discrets avec pour objectif de choisir la mieux adaptée à notre problème.

Nous avons réalisé un simulateur de mouvement brownien distribué selon différentes stratégies (synchrone forte et faible, et asynchrone).

Les résultats que nous avons obtenus nous permettent de penser que l'utilisation de la technique synchrone faible semble être la meilleure stratégie. En effet, cette stratégie est la seule de nos deux approches permettant de gagner en temps de simulation par rapport à la technique mono-processeur. Cependant, il faut prendre en compte, que la technique synchrone faible ne devient intéressante qu'à partir d'une certaine charge de calcul par particule. L'approche synchrone forte n'a comme principal intérêt que l'augmentation du nombre de particules qu'il est possible de simuler sans perdre de temps par rapport à la technique mono-processeur et la simplicité de son algorithme.

Dans notre futur système proie-prédateur, les prédateurs réalisent un grand nombre de calculs en interne pour leur comportement (perception, digestion ...), mais aussi pour calculer leur mouvement dans un espace 3D (calculs de trajectoires, intersections ...). Ces coûts nous font penser que la technique pessimiste est la meilleure pour notre modèle proie-prédateur.

Lors du passage du modèle de particules à un modèle d'agent situés (Ferber, 1995), nous serons confrontés à des problèmes liés au mécanisme de perception des prédateurs. En effet, la perception correspond à une interaction distante pouvant se produire entre deux individus localisés sur des simulateurs différents. Dans l'exemple du mouvement brownien, nous n'avons pas été confronté au problème d'équilibrage de charge, car la distribution des particules a tendance à rester uniforme et on sait que la charge de calcul est due au nombre de particule par simulateur. En revanche, dans le cas d'agent situés dotés d'un comportement, la répartition des individus dans l'espace peut ne plus être homogène. Dans ce cas, le modèle de distribution devra être capable de gérer le répartition de charge. Il existe de nombreux travaux traitant de ce problème (Miguet 1995) (Fonlupt 1998). Il reste néanmoins à l'adapter à notre système.

BIBLIOGRAPHIE

- (Chandy, 1978) Chandy K. M., Misra J., "Distributed Simulation : A Case Study in Design and Verification of Distributed Programs.", IEEE Transaction on Software Engineering, SE-5, No. 5, p. 440-452 (1978).
- (Duboz, 2003) Duboz R., Ramat É., Preux P., Amblard F. et Deffuant G. "Utiliser les modèles individus-centrés comme laboratoires virtuels pour identifier les paramètres d'un modèle agrégé.", MOSIM'03, p. 353-357 (2003).
- (Duboz, 2001) Duboz R., Ramat É. et Preux P., "Towards a coupling of continuous and discrete formalisms in ecological modelling - Influences of the choice of algorithms on results", ESS01, 13th European Simulation Symposium, p. 481-487 (2001).
- (Ferber, 1995) Ferber J. "Les systèmes multi-agents : Vers une intelligence collective", Editions InterEditions (1995).
- (L'Ecuyer, 1998) L'Ecuyer P. "Random Number Generation", Chapter 4 of the Handbook on Simulation, Jerry Banks Ed., Wiley, p. 93-137 (1998).
- (L'Ecuyer, 2002) L'Ecuyer P., Simard R., Chen E. J. et Kelton W. D., "An Objected-Oriented Random-Number Package with Many Long Streams and Substreams", Operations Research, Vol. 50, No. 6, p. 1073-1075 (2002).
- (Fick, 1855) Fick A. "Über diffusion", Annalen der Physik und Chemie, Vol. 94, p. 59-86 (1885).
- (Fonlupt, 1998) Fonlupt C., Marquet P., et Dekeyser J., "Data-parallel load balancing strategies", Computing 24, p. 1665-1684 (1998).
- (Fujimoto 1999) Fujimoto R., "Parallel and distributed simulation", Winter Simulation Conference, 122-131 (1999).
- (Jefferson, 1985) Jefferson D. R. "Virtual time", ACM Transactions on Programming Languages and Systems, Vol. 7, No. 3, p. 404-425 (1985).
- (Lamport, 1978) Lamport L., "Time, clocks and the ordering of events in a distributed system.", Communications of the Association of the Computing Machinery, Vol. 21, No. 7, p.558-565 (1978).
- (Leroudier, 1980) Leroudier J., "La simulation à événements discrets", Editions Hommes et Techniques (1980).
- (Miguet, 1995) Miguet S. et Pierson J. M. "Dynamic load balancing in a parallel particle simulation", In High Performance Computing Symposium, pages 420-431, Montreal, Canada (1995).
- (Matsumoto, 1998) Matsumoto T. et Nishimura T., "Mersene Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator", ACM Trans. on Modeling and Computer Simulation Vol. 8, No. 1, p.3-30 (1998).
- (Misra, 1986) Misra J., "Distributed discrete-event simulation.", Computing surveys, Vol. 18, No. 1, p.39-65 (1986).
- (Ramat, 2003) Ramat E. et Preux P. "Virtual laboratory environment (VLE): a software environment oriented agent and object for modeling and simulation of complex systems", Simulation Modelling Practice and Theory, 11, p.45-55 (2003).
- Rakotoarisoa, (1991) Rakotoarisoa H. et Mussi P., "Parseval : Parallélisation sur réseaux

de transputers de simulations pour
l'évaluation de performances.”,
Technical Report 131, INRIA (1991).

Zeigler (2000) Zeigler B., Kim D. et Praehofer H.
“Theory of modeling and simulation:
Integrating discrete event and
continuous complex dynamic
system”, academic press (2000).

CONSTRUCTION DE SPECIFICATIONS MULTI-VUES UML ET B

Dieu Donné Okalas Ossami,

Doctorant en Informatique

okalas@loria.fr +33 3.83.59.20.16

Jeanine Souquières,

Professeur en Informatique - Université Nancy 2

Jeanine.Souquieres@loria.fr +33 3.83.59.20.12

Jean-Pierre Jacquot,

Maître de conférences en Informatique – Université Nancy 1

Jean-Pierre.Jacquot@loria.fr + 33 3.83.59.20.10

Adresse professionnelle

LORIA campus scientifique ★ BP 239 ★ F-54506 Vandœuvre-Lès-Nancy Cedex

Résumé : On sait aujourd'hui transformer les diagrammes UML et les expressions OCL en B, mais pas le contraire. Le manque de retour de B vers UML laisse penser que l'évolution individuelle des deux représentations pourrait conduire à ce qu'elles expriment des exigences contradictoires. Par ailleurs, le fait que UML et B appartiennent à deux paradigmes de modélisation différents fait que la transformation systématique de B en UML ne peut se faire sans perte d'informations. Pour surmonter ces problèmes, nous proposons de structurer la spécification en vues: une vue UML et une vue B. Dans une telle approche, l'utilisateur ne travaille plus sur deux spécifications indépendantes, mais sur une des deux représentations d'une même spécification. La structuration en vues, permettra au concepteur de faire usage du meilleur des deux: clarté architecturale pour UML, pouvoir d'expression et outils de preuve pour B. Ce papier présente les concepts et la démarche générale de construction de spécifications multi-vues UML et B.

Summary: UML diagrams and OCL expressions can be transform into B, but not the opposite. The lack of return from B towards UML lets think that the individual evolution of both representations could lead to express contradictory requirements. Moreover as UML and B belong to two different modeling paradigms, the systematic transformation of B into UML cannot be done without losing information. To overcome these problems, we propose to structure the specification into views: A UML view and a B view. In such an approach, the user does not work any more on two independent specifications, but on one of the two representations of the same specification. By structuring into views, the specifier is able to use the best of both formalisms: architectural clearness for UML, expression power and proof tools for B. This paper presents the concepts to construct UML and B multi-views specifications.

Mots clés : B, UML, intégration UML et B, spécifications multi-vues, méthodes formelles.

Keywords : B, UML, UML and B integration, multi-views specifications, formal methods.

Construction de spécifications multi-vues UML et B

1 - MOTIVATIONS

Le développement du logiciel devient de plus en plus complexe. Cette complexité est due à la diversité des contraintes (temps réel, sûreté, etc.) à prendre en compte, à l'hétérogénéité des données manipulées, à la variété des fonctionnalités (flexibilité, maintenabilité, interopérabilité, etc.). Pour prendre en compte tous ces facteurs, la construction d'un logiciel est souvent l'œuvre de plusieurs spécialistes (analystes, programmeurs, testeurs, décideurs, etc.) qui interagissent à travers divers documents (cahier des charges, spécifications, documents de conception, plans de tests, programmes, etc.). Ces documents définissent les différentes facettes du futur logiciel et sont rédigés pour partie en langue naturelle et pour partie dans des langages artificiels dont la syntaxe et la sémantique sont définies formellement, comme par exemple les formalismes UML, B, Z, etc.

De manière générale, la construction de spécifications occupe une place importante dans le cycle de développement du logiciel. Dans la pratique industrielle, la construction des documents de spécification fait souvent appel à des notations semi-formelles UML (diagrammes de classes, diagrammes d'état-transition, diagrammes de cas d'utilisation, etc.). Ces notations combinent une structure à base de graphismes avec des contraintes OCL et des annotations en langue naturelle sous forme de commentaires. La rapidité de conception, la facilité de lecture et la clarté architecturale des modèles UML en font une méthode populaire et très utilisée. La difficulté liée à la conception de modèles UML fiables réside dans l'immatunité ou le manque d'outils de validation mathématique des propriétés modélisées. Conscients de cet handicap, les industriels sont sensibilisés à la nécessité de mettre au point et d'intégrer les méthodes formelles comme éléments à part entière du cycle de développement de logiciels fiables. L'expérience a montré qu'une erreur de spécification aussi infime soit elle détectée en cours d'exploitation, peut non seulement entraîner des surcoûts de correction

considérables, mais aussi provoquer des dégâts dont les conséquences sont parfois irréversibles sinon dramatiques comme la perte de vie humaines.

Les méthodes formelles sont basées sur une approche rigoureuse et répondent au double objectif non seulement de qualité et de sûreté de fonctionnement, mais aussi de faciliter le contrôle de conformité de logiciels avec leurs spécifications. Elles possèdent une notation avec une syntaxe et une sémantique précises et sont souvent supportées par des outils de preuves automatiques ou interactives. Mais le fait que leur notation soit à base de concepts mathématiques et de la logique a conduit à ce que les spécifications écrites avec ces notations sont difficiles à lire et à comprendre quand elles ne sont pas accompagnées d'une bonne documentation. Dans ces conditions, leur utilisation pour construire des systèmes de grande taille est coûteuse en temps et n'est réservée qu'aux seuls initiés. C'est à ce titre que de nombreuses recherches ont été menées sur le couplage d'UML et les méthodes formelles comme Z ou B. Le but de ces recherches est d'étudier des mécanismes de modélisation automatisables capables de combiner la rapidité de conception, la facilité de lecture et la clarté architecturale des modèles UML d'une part, avec la fiabilité, le pouvoir d'expression et la sémantique des méthodes formelles d'autre part. Ces travaux ont permis de mettre au point un certain nombre d'outils de génération de spécifications B [Ledang 2003, Snook 2002, Laleau 2000], Z [RoZ], etc. à partir de diagrammes UML.

Ces outils permettent de transformer les diagrammes UML et certaines expressions OCL en B, mais pas le contraire. Le manque de retour de B, Z, etc. vers UML laisse penser que l'évolution individuelle des deux représentations pourrait conduire à ce qu'elles expriment des exigences contradictoires. Par ailleurs, la qualité d'un outil d'aide au développement, ne dépend pas seulement de sa capacité à projeter un langage source en un langage cible, mais aussi de sa faculté à analyser la pertinence des constructions induites et à fournir une assistance à

l'utilisateur non spécialiste du langage. C'est dans ce cadre que nous étudions une nouvelle démarche d'intégration d'UML et B basée sur la structuration en vues. Dans une telle approche, l'utilisateur ne travaille plus sur deux spécifications indépendantes, mais sur une des deux représentations d'une même spécification. La structuration en vues, permet au concepteur de faire usage du meilleur des deux: clarté architecturale pour UML, pouvoir d'expression et outils de preuve pour B.

L'objectif de ce papier est double. D'une part, il s'agit de dresser une analyse sur le processus d'intégration d'UML et B, c'est à dire la traduction systématique de diagrammes UML en B. D'autre part, il s'agit d'apporter en s'appuyant sur cette analyse, des éléments de réflexion sur notre nouvelle façon de penser l'intégration de ces deux formalismes (la décomposition de la spécification en vues UML en B). Pour ce dernier point, notre but est de montrer l'intérêt et la nécessité d'une telle démarche. Le travail présenté dans le papier décrit la démarche générale de construction de spécifications multi-vues UML et B.

Le reste du papier est structuré comme suit. Dans le paragraphe 2 nous présentons brièvement les formalismes UML et B. Le paragraphe 3 présente les avantages et les limites de cette démarche. Le paragraphe 4 présente l'idée générale de notre approche. Le paragraphe 5 est dédié à la présentation des concepts importants supportés par notre approche de construction de spécifications multi-vues UML et B. Finalement, nous concluons et donnons des suites à ce travail dans le paragraphe 6.

2 – BREVE PRESENTATION D'UML ET B

2.1 - UML

UML [OMG 2001, Rumbaugh 1991, Rumbaugh 1998] est une technique de modélisation standardisée et très répandue dans l'industrie du logiciel. Elle propose un ensemble de notations graphiques de modélisation par objets. Ces notations permettent de visualiser, de construire et de documenter les systèmes à l'aide de diagrammes. UML possède neuf types de diagrammes représentant chacun un aspect

particulier du système (aspects fonctionnels, statiques, dynamiques).

Parmi les différents types de diagrammes UML, nous nous intéressons plus particulièrement aux diagrammes de classes et d'état-transition. Ces deux types de diagrammes se sont montrés suffisants pour représenter la structure et la dynamique d'un système. Dans ce papier nos efforts se focalisent sur le diagramme de classes.

2.1.1 – Eléments du diagramme de classes UML

Un diagramme de classes UML est constitué d'un ensemble de classes reliées entre elles par des associations et des hiérarchies de généralisation (cf. Fig. 1). L'association modélise comment sont liés les objets des classes participantes. Elle est caractérisée par un nom et deux ou plusieurs extrémités sur lesquelles figurent les cardinalités et les noms de rôle. Il existe deux cas particuliers pour les associations: la composition et l'agrégation. La généralisation est une relation dans laquelle une des classes impliquées est identifiée comme classe générale et les autres comme des spécialisations de celle-ci.

L'élément central dans la modélisation orientée objet avec UML est la classe. Une classe est en général représentée par un rectangle divisé en trois compartiments. Le compartiment du haut spécifie le nom de la classe et les compartiments du centre et du bas définissent respectivement la liste des attributs et les opérations. Selon le niveau de détails souhaité, les compartiments du centre et du bas peuvent être omis.

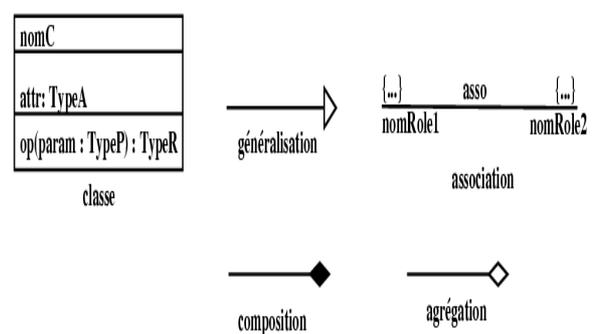


Fig. 1 : Eléments structuraux du diagramme de classes UML

L'ensemble des diagrammes UML servant à décrire les différents aspects d'un système forment le modèle de celui-ci (dans notre cas, diagramme de classes, d'état-transition). Chaque diagramme constitue une des représentations partielles (vue) du modèle UML.

2.2 – La méthode B

La méthode B [Abrial 1996] est une méthode de spécification formelle qui comprend toutes les phases de développement logiciel, de la spécification jusqu'à l'implémentation. Elle permet d'exprimer des propriétés sous la forme de prédicats de la logique de premier ordre. Les opérateurs utilisés par la notation B sont ceux de la théorie des ensembles de Zermelo-Frankel. Les propriétés d'un système à prouver en B sont essentiellement des propriétés d'invariance. Contrairement à UML, la notation B n'est pas orientée objets.

Structure d'une machine B

L'élément de granularité de base d'une spécification B est la machine abstraite. La figure 2 présente la structure syntaxique d'une machine abstraite B. Cette notion est similaire au concept de classe ou de module dans les langages de programmation classiques. La machine abstraite peut être successivement raffinée (*raffinement*) jusqu'à l'obtention d'un composant implantable (*implantation*).

Le concept central étant l'encapsulation, le changement d'état d'une machine ne peut se faire qu'au travers des opérations. Une machine B est structurée en trois parties principales: l'entête, la partie déclarative et la partie opérationnelle. L'entête spécifie le nom et la liste des paramètres éventuels de la machine. Elle inclut la partie composition de la machine qui permet de décrire les différents liens entre les machines. Ces liens se traduisent par la déclaration de noms de machines dans les clauses *SEES*, *USES*, *EXTENDS*, *IMPORTS* et *INCLUDES*. Chaque clause possède une sémantique et des règles de visibilité précises. La partie déclarative modélise l'état de la machine au travers de divers types de données (variables, constantes, ensembles) et de contraintes (cf. Fig. 2, *C*, *P*, *I*, etc.) que ces données doivent toujours vérifier. La partie opérationnelle est constituée de l'initialisation et des opérations. Elle est basée sur le langage des substitutions généralisées. Une substitution généralisée est une construction mathématique

abstraite construite à partir d'une substitution basique $x := e$, qui correspond à un changement d'état, par l'intermédiaire d'opérateurs comme CHOICE S_1 OR S_2 , PRE P THEN S , etc.

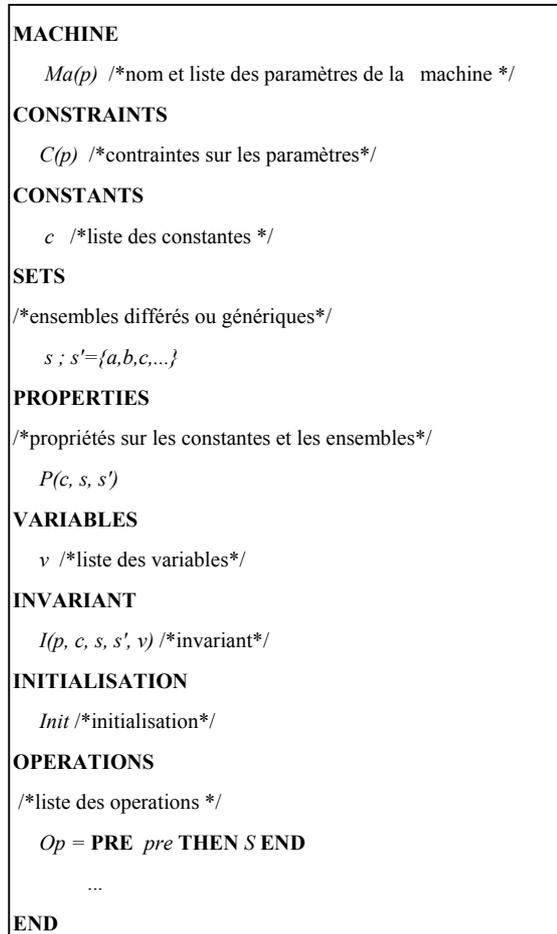


Fig. 2 : Structure d'une machine abstraite B

Dans la figure 2, les clauses *CONSTRAINTS*, *PROPERTIES*, *INVARIANT*, etc. regroupent un ensemble de prédicats (*C*, *P*, *I*, etc.). Le prédicat minimal est le prédicat de typage qui permet de déterminer le type (le domaine de valeurs) de chacune des données de la machine. $X(j)$ signifie que la donnée j apparaît dans le prédicat X . Les prédicats *C*, *P* et *I*, *Init*, *pre* et *S* doivent être écrits de telle sorte qu'ils garantissent la consistance interne de la machine [Lano 96].

3 – INTEGRATION D'UML ET B : LIMITES ET AVANTAGES

3.1 – Avantages de l'intégration d'UML et B

L'intégration d'UML et B est une technique qui consiste à représenter les connaissances d'un domaine d'application à l'aide de diagrammes UML augmenté de contraintes OCL [Warmer 1999] et de traduire ces diagrammes en B, puis de mettre en œuvre des raisonnements sur ces représentations. Les avantages d'une telle démarche sont nombreux:

- masquer la modélisation formelle B à l'utilisateur par la manipulation de graphismes,
- utiliser UML comme point de départ de la modélisation B des modèles orientés objets,
- valider les modèles UML à l'aide des outils de preuve (*atelierB*, *bToolkit*) de la méthode B,
- donner une sémantique aux modèles UM à l'aide de B.

L'utilisation des outils d'animation automatique de preuves permet d'explorer les aspects fonctionnels du système modélisé. En terme de modélisation UML, cela signifie que la dynamique du système peut être analysée en terme de vérification de pré- et post-condition d'opérations et d'invariants sur les données

(attributs) et les relations entre les objets. Le fait que B définisse un mécanisme de raffinement permet non seulement un développement incrémental, mais aussi de maintenir les propriétés de processus tout au long du développement. Pour ce dernier point,

3.2 – Limites de l'intégration d'UML et B

Malgré les nombreux avantages cités dans le paragraphe 3.1, la démarche de génération de spécifications B à partir de diagrammes UML possède quelques limites:

- on sait transformer UML en B, mais pas le contraire. Les modifications opérées sur la spécification B dérivée ne sont pas prises en compte au niveau UML, ce qui pose le problème de la cohérence entre les deux représentations,
- le processus de développement induit est séquentiel (cf. Fig. 3): concevoir le modèle UML (étape 1), le traduire en B (étape 2), poursuivre le processus avec la complétion et/ou le raffinement des squelettes de spécifications B induites (étape 3), intégrer une activité de validation (étape 4). Un tel processus ne permet pas de garantir la consistance des textes B produits avec le modèle UML initial à cause du manque de liens dynamiques entre les deux représentations.

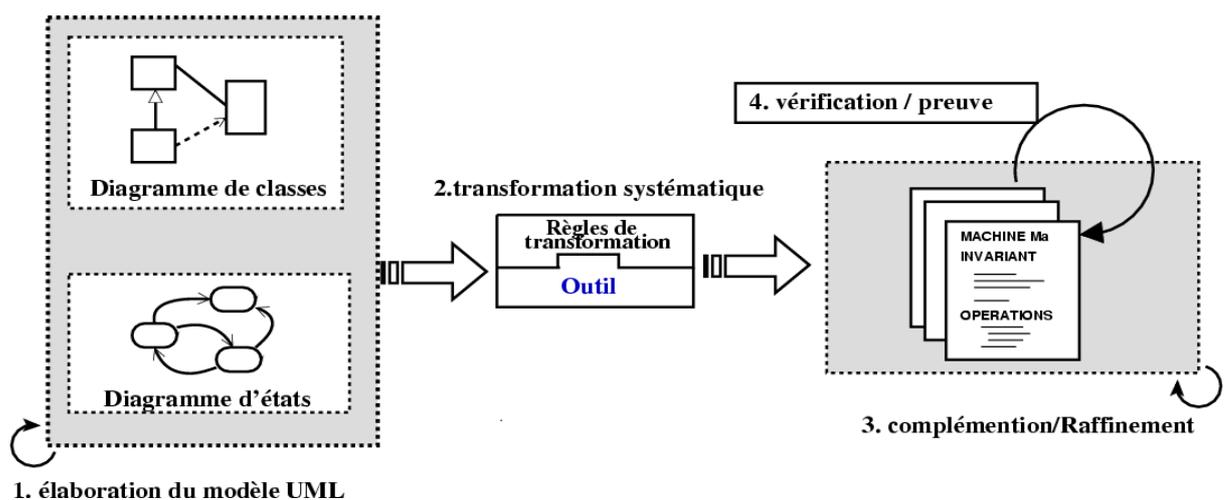


Fig. 3 : Processus de dérivation UML vers B

Il s'agit de s'assurer qu'il n'existe pas de contradictions entre les différentes abstractions du système.

La majorité des approches de couplage d'UML avec les méthodes formelles Z [Dupuy 2000, France 2000], B [Nguyen 1998, Meyer 2001] ou LOTOS [Wang 97] suivent le processus présenté dans la figure 3.

3.3 – Vers une structuration en vues UML et B

Pour surmonter les limites de la traduction d'UML en B, une idée consiste à poursuivre la construction de la spécification en B avec l'écriture des raffinements et en déduire la transformation inverse des textes B produits en diagrammes UML. Mais dans un développement à large échelle, cette solution peut vite s'avérer limitée pour une raison essentielle:

- la notation B contrairement à celle d'UML n'est pas orientée objets. Cette différence implique l'existence d'incompatibilités entre les deux formalismes.

La spécification B déduite automatiquement d'UML, est éloignée de celle qu'on aurait écrite directement en B ou en UML. Dans ces conditions, il est plus simple d'écrire directement les spécifications en B ou en UML.

Cette observation montre clairement que la transformation entre UML et B ne peut pas être un processus symétrique. Dans ce contexte, nous proposons de structurer la spécification en différentes vues (vue UML et vue B). Les transformations ne sont plus déduites systématiquement, mais dépendent de l'aspect décrit du contexte de modélisation. Dans une telle approche, le concepteur ne travaille plus sur deux spécifications indépendantes, mais sur une des deux représentations d'une même spécification. Nous appelons cette démarche : *construction de spécifications multi-vues UML et B*.

4 – NOTRE APPROCHE : CONSTRUCTION DE SPECIFICATIONS MULTI-VUES UML ET B

La *construction de spécifications multi-vues UML et B* est une extension des travaux de

Ledang [Ledang 2002] et de Meyer [Meyer 2001] qui ont défini un cadre théorique et pratique à la dérivation des spécifications B à partir des diagrammes UML et d'annotations OCL. L'extension de ces travaux consiste à considérer UML et B comme deux vues partielles d'une même spécification et à étudier les mécanismes de consistance entre toutes ces différentes vues (diagrammes de classes, diagrammes d'état-transition et spécification B induite).

Pour construire cette spécification, l'utilisateur utilise les "*opérations de construction*" qui font évoluer la spécification sur les deux représentations (UML et B). La traçabilité et la cohérence entre les deux représentations sont assurées par des liens dynamiques existants entre les deux textes. Ces liens se traduisent par la transformation bidirectionnelle qui permet de ce fait une plus grande flexibilité et la possibilité de faire évoluer la spécification tout en maintenant la cohérence entre les deux vues.

Cette démarche pose les bases d'une nouvelle technique d'intégration d'UML et B. Elle constitue une réponse aux problèmes (incohérence, traçabilité, etc.) liés à l'évolution individuelle de différents documents de spécification d'un système. L'idée consiste à composer les étapes de *l'élaboration du modèle UML* et de la *complétion des spécifications B* (cf. Fig. 3) en une seule et unique étape: *la construction de la spécification*.

Le scénario d'usage envisagé est le suivant. Le spécifieur se positionne sur une vue et construit les représentations de cette vue en effectuant les opérations de spécifications sur les objets qu'il souhaite modéliser (classe, attribut, opération, variable, machine, raffinement, etc.). L'outil modifie alors systématiquement les deux représentations en fonction des informations saisies et des contraintes de propagation définies par les règles de transformation.

Nous pensons que la construction simultanée et multi-vues d'une spécification dans un développement conjoint est un facteur important permettant d'une part de faciliter l'appropriation d'un système, d'autre part, d'autoriser une validation directe par l'expert non spécialiste du langage, de la modélisation adoptée. Une telle dynamique permet de suivre pas à pas les raisonnements effectués afin de

valider la représentation adoptée. La figure 4 présente l'architecture générale de cette

démarche.

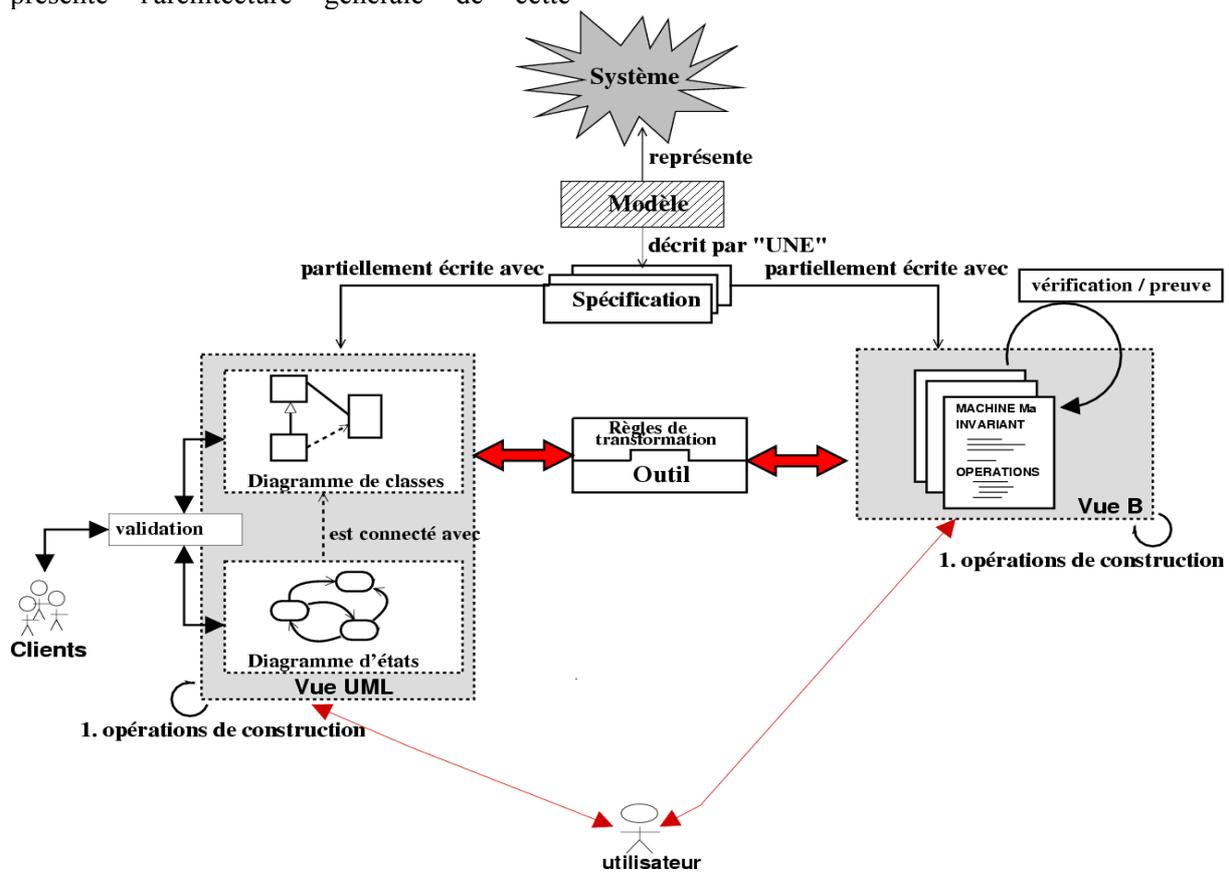


Fig. 4 : Assistance à la construction de spécifications multi-vues UML et B extrait de [Okalas 2003]

La construction de spécifications multi-vues UML et B vise à :

- utiliser UML et B de façon complémentaire,
- proposer un cadre de transformation et un environnement de développement interactif de spécifications par combinaison de plusieurs formalismes,
- faciliter la diffusion de B,
- documenter dynamiquement les projets B,
- apporter une réponse aux problèmes (incohérence, traçabilité, etc.) liés à l'évolution individuelle de différents documents de spécification d'un même système,
- concevoir un outil d'aide au développement de spécifications multi-vues UML et B.

La problématique générale à laquelle nous nous intéressons peut se traduire ainsi:

étant donné

1. un ensemble de langages (UML et B dans notre cas) définis chacun par une syntaxe et une sémantique données,
2. un ensemble de propriétés *Prop* d'un modèle *m* à construire,

déterminer

1. un ensemble d'opérations *O* qui permettent de décrire les propriétés *Prop* du modèle entièrement ou partiellement sur une ou sur les deux vues dans une dynamique de développement simultané UML et B,
2. un ensemble de règles permettant de calculer dynamiquement les correspondances entre les différentes représentations induites (i.e. UML et B) de la spécification,

- un mécanisme de contrôle de cohérence par construction entre les différentes vues.

5 – CONCEPTS IMPORTANTS POUR LA CONSTRUCTION DE SPECIFICATIONS MULTI-VUES UML ET B

Ce paragraphe est consacré à l'ensemble des concepts qui doivent être supportés par la construction de spécifications multi-vues UML et B. Ces concepts sont relatifs aux notions de *vues*, *d'opérations de construction* et de *transformation*. Les concepts sont illustrés par des exemples. En parallèle, les constructions UML et B sont présentées.

5.1 – Les vues

La structuration en vues UML et B s'inscrit dans une tendance générale en matière de développement logiciel. Il s'agit de maîtriser la complexité de la spécification en décrivant certains aspects particuliers d'un système dans un langage dédié. Jackson et Zave [Jackson 1993] ont montré que l'utilisation de vues et de notations adaptées à chaque aspect de l'application pouvait clarifier la spécification. UML intègre la notion de décomposition en vues en offrant un éventail de diagrammes pour représenter différents aspects d'un même système.

La difficulté de la structuration en vues dans le cadre d'un développement conjoint semi-formel et formel (i.e. UML et B) réside dans l'utilisation de deux techniques de représentations différentes (graphique et textuelle) avec deux notations différentes appartenant à deux paradigmes de modélisation différents. Une telle combinaison est très intéressante. En effet, la décomposition d'un système en plusieurs vues ne peut-être réellement bénéfique que si les vues décrivent des concepts de nature différente. Le contraire serait une simple juxtaposition de notations.

Dans notre travail, une vue est considérée comme un moyen de construction et de visualisation des constructions partielles UML et B induites. Ces constructions sont réalisées au travers d'éléments dont la syntaxe et la sémantique sont définies par les méta-modèles respectifs des deux formalismes.

Définition (Vue)

Dans la construction de spécifications multi-vues UML et B, une vue v_i est considérée comme un tuple (M, m)

Où

- $v_i := V_U | V_B$ désigne les vues UML et B,
- M est le méta-modèle (langage) dans lequel la description (modèle) partielle est écrite. Dans notre contexte, M désigne le méta-modèle UML ou le langage B. Chaque modèle m est construit avec les outils syntaxiques et sémantiques offerts par son méta-modèle respectif,
- $M ::= m_U | m_B$ désigne respectivement le modèle UML et la spécification B. m_U et m_B correspondent respectivement aux représentations partielles UML et B sur lesquelles l'utilisateur travaille.

Nous avons volontairement omis de présenter le détail de chacun des modèles m_U et m_B pour des raisons de lisibilité. Toutefois, nous les présenterons à chaque fois que c'est nécessaire.

Les éléments des modèles m_U et m_B entretiennent deux types de relations: les relations calculées et les simples correspondances. La figure 5 présente ces différents liens.

- Les relations calculées (T sur la figure 5) sont déduites automatiquement par l'application systématique des règles de transformation qui permettent de passer systématiquement d'UML à B et inversement,
- Les simples correspondances (C sur la figure 5) sont des relations non déductibles automatiquement. Par exemple, la post-condition d'une opération ne peut être déduite automatiquement. Par contre, elle doit faire référence à l'opération correspondante afin d'en assurer la traçabilité.

La figure 5 présente une classe *Customer* qui modélise le fichier client d'une société. Cette classe possède trois attributs *name*, *category* et *discount*. *name* caractérise le nom du client, *category* modélise le type de client (particulier, professionnel, etc.) et *discount* spécifie le pourcentage de réduction attribué à chaque catégorie de clients. Dans la suite, nous

utiliserons cette classe pour illustrer notre propos. L'exemple est petit, mais il est suffisant pour illustrer les idées présentées dans ce papier.

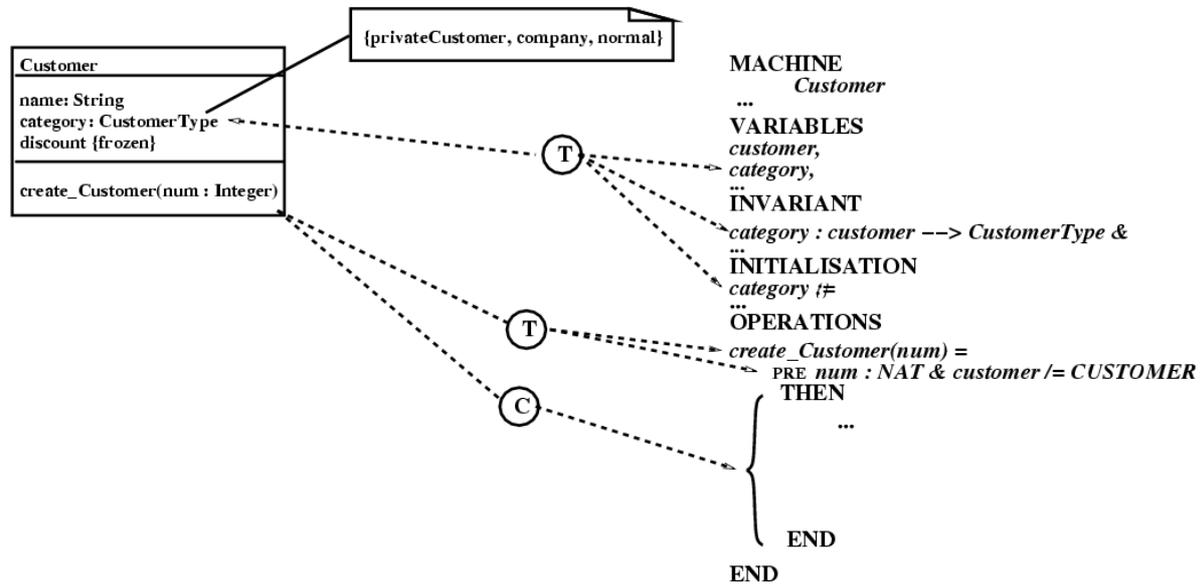


Fig. 5 : Exemple de relation entre m_U et m_B

5.2 – Opérations de construction

L'activité de construction d'une spécification peut être modélisée à travers un ensemble d'opérations atomiques: les *opérations de construction*. Ces opérations caractérisent la logique du développement de l'utilisateur. Un de nos objectifs est d'identifier et de caractériser ces opérations. En terme de modélisation, effectuer une opération de construction sur un ou plusieurs éléments d'un modèle m (m_U ou m_B dans notre cas) correspond à la modification de ce dernier. Il passe alors de m_{old} à m_{new} où m_{old} et m_{new} sont respectivement l'ancienne et la version modifiée de m . Dans un développement conjoint, cette modification est implicitement celle de toutes les représentations liées à l'élément édité. Dans cette perspective, la recherche sur la caractérisation des opérations de construction d'une spécification multi-vues peut se ramener à :

- un problème de calcul de la différence Δ entre deux versions d'un modèle m sur lequel l'utilisateur travaille:

$$\Delta = m_{new} \setminus m_{old}$$

- l'étude de la propagation T de Δ sur les autres vues de la spécification

Pour traiter ces problèmes de façon générique, nous proposons de mettre en œuvre une stratégie de construction dont la première étape consiste à définir systématiquement tous les éléments (ou endroits) à créer, modifier ou supprimer sur la vue source pour accomplir la description du concept en cours d'édition. Dans le contexte de modélisation d'une variable en B par exemple, ce sera la clause VARIABLES pour la déclaration de la variable, la clause INVARIANT pour l'écriture de l'invariant relatif à la variable et la clause INITIALISATION pour l'initialisation de celle-ci. La deuxième étape repose sur la propagation des modifications sur la ou les vues cibles. Par exemple le diagramme de classes UML si la modification est effectuée en B. Il s'agit de déterminer systématiquement à chaque saisie de l'utilisateur, le type de ou des éléments UML ou B correspondant afin d'en déduire les représentations UML ou B adéquates. L'idée sous-jacente à la propagation réside dans le principe que l'exécution d'une opération de construction O_i sur une vue correspond à l'exécution systématique de zéro, une ou plusieurs opérations O_0', \dots, O_i' qui

ajoutent, suppriment ou modifient les éléments des différentes représentations de la spécification.

5.2.1 – Caractérisation de Δ

Dans [Alanen 2003] Alanen et Pores présentent le résultat Δ de la différence entre deux versions d'un modèle UML en terme d'opérations et non comme un ensemble d'éléments. Une des raisons de ce choix est que la différence entre deux modèles n'est pas toujours un modèle, tout comme le résultat de la différence entre deux entiers positifs n'est pas toujours un entier positif. Il est donc plus intuitif de représenter Δ en termes d'opérations qui ajoutent, suppriment ou modifient les éléments d'un modèle. Nous adoptons cette représentation pour caractériser les opérations de construction génériques qui permettent de construire simultanément les différentes représentations UML et B de la spécification. Dans notre étude, nous distinguons trois types d'opérations de construction de base qui définissent Δ :

1. les opérations de création, désignées par $new(e, t)$: création d'un élément e de type t , tel que:

$$new(e, t) = [new(e, t) \mid e \in m_{new} \setminus m_{old}]$$

2. les opérations de suppression, désignées par $delete(e, t)$: suppression d'un élément e de type t , tel que:

$$delete(e, t) = [delete(e, t) \mid e \in m_{old} \setminus m_{new}]$$

3. les opérations de modification, désignées par $modify(e, e')$: modification d'un élément e qui devient e' .

La modification d'un élément est implicitement celle du modèle, puisqu'un modèle est constitué d'éléments respectant certaines propriétés. Il existe plusieurs façons pour modifier un modèle. Nous en distinguons trois principales:

- a. ajout, suppression ou modification ($add/delete/modify$) des données comme

par exemple les attributs, les variables, les ensembles ou les constantes B.

- b. ajout, modification ou suppression ($add/delete/modify$) des opérations des classes UML, des composants B (machines, raffinements, etc.).

- c. ajout, modification ou suppression ($add/delete/modify$) des classes, association ou des composants B (machines, raffinements, etc.).

Sur la base de ces opérations de base, on peut définir $\Delta = [new, delete, modify, O]$ comme la collection d'ensembles d'opérations atomiques $add, remove, set, etc.$ qui sont des sortes de raffinement de $new, delete, modify$ et O . Les opérations $new, delete, modify$ ajoutent, suppriment ou modifient un élément de façon globale. Tandis que $add, remove, set, etc.$ ajoutent, modifient ou suppriment une caractéristique particulière d'un élément. O désigne toutes les autres opérations dont $skip$ (opération sans effet).

Exemple (édition d'une variable B)

Du point de vue d'un utilisateur B, éditer une variable consiste à effectuer une saisie clavier pour la déclarer, écrire un invariant correspondant au domaine de valeurs qu'elle peut prendre et affecter une valeur initiale à celle-ci. Du point de vue de l'outil, ces banales saisies clavier sont interprétées comme une exécution de l'opération $modify$ qui modifie la machine dans laquelle la nouvelle variable doit être ajoutée. Une telle opération peut se raffiner en trois opérations atomiques add permettant de construire les éléments nécessaires à la description complète d'une variable en B comme illustré dans la figure 6. Elle peut entraîner la génération de un ou plusieurs éléments UML (cf. Fig. 6). La déduction du type des ces derniers dépend de plusieurs facteurs (cet aspect n'est pas traité dans ce papier) comme par exemple l'invariant $I(v)$ associé à v , les opérateurs ($\in, =, >, ->, \neq, \leq, \geq, etc.$) utilisés pour construire cet invariant, etc.

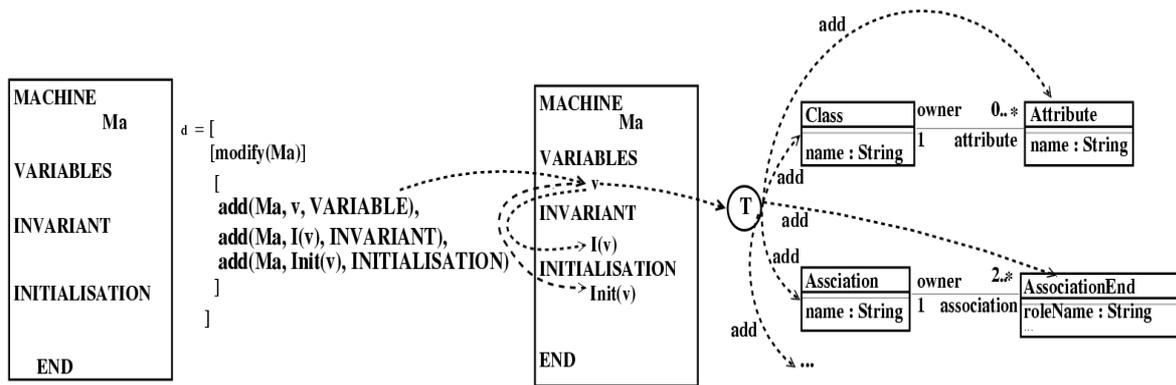


Fig. 6 : Ajout d'une variable dans une machine B et ses effets potentiels en UML.

Sur la figure 6 $d = \Delta$

L'exécution d'une opération $O_i \in \Delta$ est conditionnée par le contexte de modélisation et les règles de construction relatives aux concepts des deux formalismes UML et B. Par exemple, une variable ou un attribut ne peut être ajouté à un composant (classe ou machine, etc.) si celui-ci contient déjà une donnée de même nom.

5.3 – La transformation

Pour être efficace, la construction de spécifications multi-vues UML et B a besoin de la transformation. Dans une dynamique de développement simultané, elle est induite par les opérations de construction. Le rôle de la transformation est double. D'une part, elle permet de calculer et de maintenir les liens dynamiques entre les deux représentations. Ces liens sont déduits à partir des règles de passage de B à UML et inversement. D'autre part, il s'agit de fournir un mécanisme de contrôle de consistance garantissant la qualité et la cohérence par construction des descriptions produites. Cette section est dédiée à la présentation des propriétés que doit avoir la transformation dans le cadre du développement de spécifications multi-vues UML et B.

La transformation est un processus de traduction automatique d'une ou plusieurs constructions d'un modèle m_i vers une ou plusieurs constructions d'un modèle m_j . Elle est modélisée par une collection de règles r_0, \dots, r_i qui associe un ou plusieurs éléments $e_i \in m_i$ avec un élément ou plusieurs éléments $e_j \in m_j$.

5.3.1 – Propriétés de la transformation

Bidirectionnalité:

C'est permettre un va et vient dynamique entre les différentes représentations afin de maintenir la cohérence et la traçabilité entre les éléments de ces représentations.

Consistance:

Le contrôle de consistance concerne deux points:

1. assurer que les constructions UML ou B induites sont respectivement valides par rapport aux deux langages,
2. assurer que la spécification induite reste toujours cohérente au regard des manipulations opérées sur tout aspect e_i : les modifications opérées sur une vue v_i sont automatiquement propagées sur toutes les vues de la spécification: *cohérence par construction*.

Flexibilité:

Pour modéliser une donnée en B, l'utilisateur est parfois contraint de saisir certaines informations dont il ne dispose pas toujours.

Par exemple, une variable B v exige outre sa déclaration, un invariant (invariant de typage au minimum) et une valeur initiale. L'absence de l'une de ces informations induit une erreur. UML autorise dans ce cadre la modélisation partielle d'un attribut sans en préciser le type ou la valeur initiale. C'est par exemple le cas

des attributs dérivés qui sont des données calculées.

Dans ce cadre, un des objectifs de nos recherches est de proposer des mécanismes de construction capables de faire le lien entre la perception du développement UML de l'utilisateur et les exigences du développement formel B. Nous proposons dans ce contexte, de prédéfinir des mécanismes flexibles de construction permettant d'une part de conserver la facilité d'utilisation d'UML, et d'autre part de tenir compte de la rigueur de B. Prédéfinir dans ce contexte peut par exemple vouloir dire:

1. Prévoir des mécanismes de construction par défaut permettant de substituer aux valeurs manquantes des valeurs par défaut tout en prenant le soin de l'indiquer au spécifieur. Une telle substitution permet de ne pas contraindre l'utilisateur à spécifier des concepts non encore élucidés. Un tel mécanisme doit être capable de contrôler la pertinence des constructions déduites, d'en interdire ou de les compléter si besoin; l'utilisateur ne décrit pas nécessairement les informations implicites et/ou considérées comme évidentes (voir figures 7 et 8).

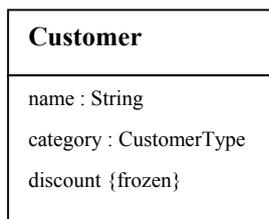


Fig. 7 Classe Customer

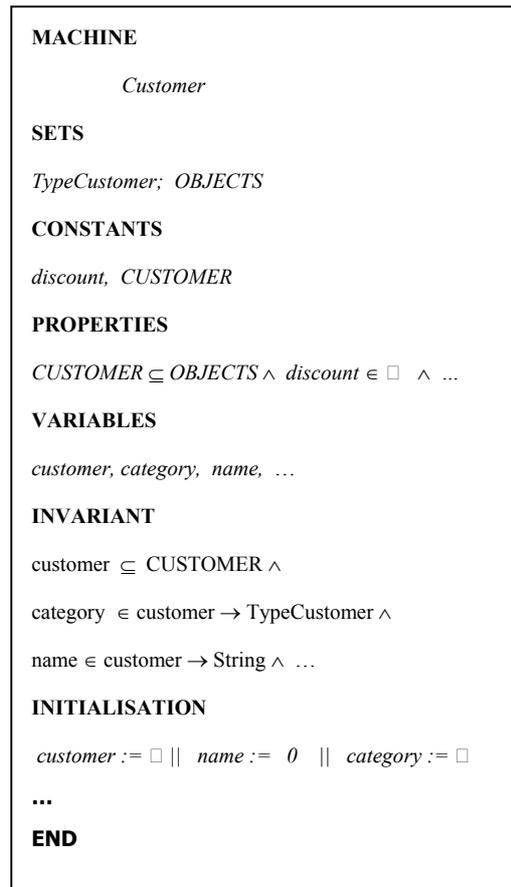


Fig. 8 Machine Customer

\square désigne non seulement des valeurs par défaut, mais aussi les endroits que l'utilisateur devra expliciter lors d'une exploitation concrète de la spécification (utilisation du prouveur par exemple).

2. Prévoir des mécanismes de représentation partielle des aspects n'ayant pas de forme de représentation équivalente sur telle ou telle vue. Par exemple, l'invariant d'un attribut peut parfois être une formule complexe qui ne peut toujours pas être projetée en UML comme en témoigne la propriété de la constante *discount* sur la figure 9 où l'on souhaite attribuer une remise allant de 0 à cent pour cent à chaque catégorie de client. Le lien existant entre cette propriété et l'attribut *discount* (cf. Fig.7) est une simple correspondance.

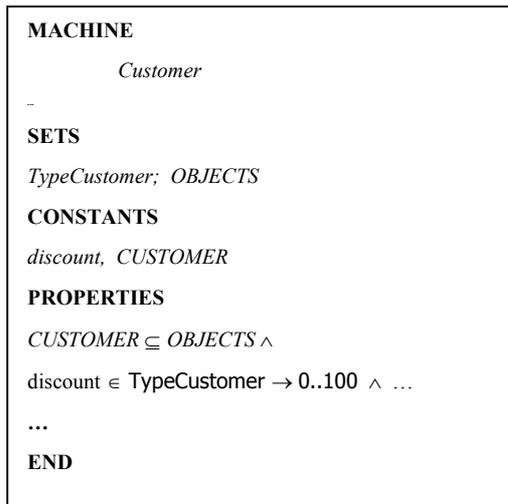


Fig. 9 Machine Customer

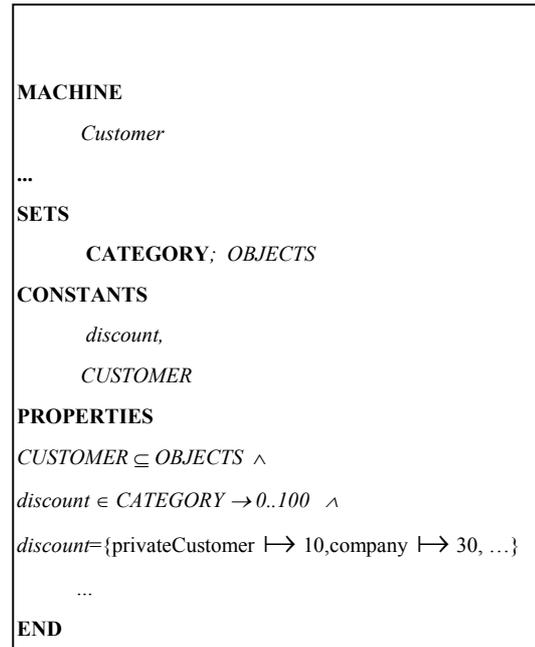


Fig. 11 Machine Customer modifiée

Persistence des informations:

Quand un modèle a été augmenté d'informations supplémentaires sur une vue cible v_j , après la modification de l'information initiale sur la vue source v_i , la projection de cette modification sur v_j ne doit pas entraîner la perte des informations supplémentaires de v_j .

Par exemple, on souhaiterait renommer le type *CustomerType* de l'attribut *category* en *CATEGORY* (cf. Fig. 10). L'utilisateur voudrait affiner la description B de la constante *discount* en associant à chaque catégorie de clients, un pourcentage de réduction précis (cf. Fig. 11). Cette modification ne doit pas entraîner la perte de la contrainte déjà exprimée sur *discount*. Grâce au mécanisme de contrôle de consistance et de cohérence par construction, cette modification doit être propagée partout où *CustomerType* est utilisé.

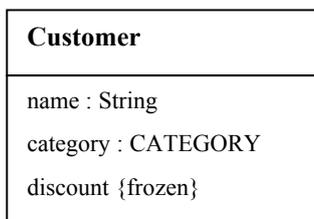


Fig. 10 Classe Customer modifiée

Assistance au développement :

L'assistance au développement est le mécanisme par lequel l'outil propose une réaction à l'utilisateur comme par exemple, il

1. lui propose la liste d'éléments qui peuvent être liés au concept manipulé. Par exemple, la déclaration d'une donnée dans une machine B est une opération qui est implicitement liée à l'écriture d'une propriété et/ou d'un invariant et/ou à l'initialisation relative à cette donnée,
2. lui indique les effets des actions qu'il opère sur la spécification,
3. lui signale les endroits où son attention est particulièrement attendue,
4. lui indique les tâches nécessaires à exécuter pour atteindre tel ou tel but.

6 - CONCLUSION ET PERSPECTIVES

Le processus d'intégration de B dans la construction de spécifications orientées objets UML et les mécanismes de transformation automatiques d'UML vers B sont susceptibles d'être améliorés malgré les avancées considérables de la recherche dans ce domaine. En effet, en plus de la formalisation d'UML en

B, il est actuellement important d'explorer les mécanismes flexibles et performants qui rendent cette formalisation transparente et attractive.

Dans ce papier, nous avons proposé une nouvelle forme d'intégration de la méthode B dans le processus de construction de spécifications orientées objets UML par construction systématique de spécifications multi-vues. Nous en avons présenté les concepts essentiels. Dans ce cadre, plusieurs aspects doivent encore être étudiés et approfondis. Nous pensons notamment à la formalisation des actions de spécification pouvant être prises en charge dans un tel processus. Une telle formalisation pose de solides bases à l'analyse des différents liens intra et extra-vue pour chaque aspect modélisé. Nous nous intéressons également à l'approche incrémentale afin de permettre un prototypage rapide par aspects.

BIBLIOGRAPHIE

- Abrial J.R. *B Book –Assigning Programs to Meanings-*, Cambridge University Press, ISBN 0-521-49619-5, 1996
- Alanen M., and Porres I. *Difference and Union of Models*. In <<UML>> 2003 Conference. Vol. LNCS 2863. Springer, San Francisco, California, USA, October 20-24, 2003.
- Dupuy S. *Couplage de notations semi-formelles et formelles pour la spécification des systèmes d'information*. PhD thesis, Université Joseph Fourier-Grenoble 1, Grenoble(F), septembre 2000.
- France R.B., Grant E. and Bruel J-M. *UMLtranZ: An UML-Based Rigorous Requirements Modeling Technique*. Technical report, Colorado State University, Ft. Collins, Colorado, January 2000.
- Warmer J., A. Kleppe *The Object Constraint Language: Precise Modelling with UML*. Addison-Wesley 1999, ISBN 0-201-37940-6.
- Okalas O. D., Souquières J., Jacquot J-P. *Assistance à la construction de spécifications multi-vues UML et B*. Poster. In MAJECSTIC'03, Marseille (F), 29-31 octobre, 2003.
- Wang E.Y., Richter H.A. and Cheng B.H.C. *Formalizing and integrating the dynamic model within OMT**. In ICSE'97:19th International Conference on Software Engineering Boston (USA), July 1997.
- OMG The Object Management Group. *OMG Unified Modeling Language Specification, version 1.3*, March 2001.
- Rumbaugh J., Blaha M., Premerlani W., Eddy F. and Lorenzen W. *Object-Oriented Modeling and Design*. Prentice Hall Inc. Englewood Cliffs, 1991.
- Rumbaugh J., Jacobson I and Booch G. *The Unified Modeling Language Reference Manual*. Addison-Wesley, 1998. ISBN 0-201-30998-X.
- Laleau R., Mammar A. *A Generic Process to Refine a B Specification into a Relational Database Implementation*. In ZB 2000: Formal Specification and Development in Z and B, LNCS 1878. Springer-Verlag York(UK), August/September 2000.
- Lano K. *The B Language and Method: A Guide to Practical Formal Development*. FACIT. Springer-Verlag, 1996. ISBN 3-540-76033-4.
- Ledang H., Souquières J. and Charles S. *ArgoUML+B: Un outil de transformation systématique de spécifications UML vers B*. Actes de la conférence AFADL'2003, INRIA, janvier 15-17, 2003, Rennes, France.
- Ledang H. *Traduction Systématique de Spécifications UML vers B*. PhD thesis, LORIA -Université Nancy2, novembre 2002.
- Meyer E. *Développements formels par objets: utilisation conjointe de B et d'UML*. PhD thesis, LORIA -Université Nancy2, mars 2001.
- Nguyen H.P. *Dérivation de Spécifications Formelles B à Partir de Spécifications Semi-Formelles*. PhD thesis, Conservatoire National des

Arts et Métiers (CNAM), décembre 1998.

Snook C., M. Butler M. *Tool-Supported Use Of UML for Constructing B specifications*. DSSE Technical Report, April 2002. Available at <http://www.ecs.soton.ac.uk/~mjb/>

Jackson M., Zave P. *Conjunction as composition*. ACM Transactions of Software Engineering and Methodology. Vol 2(4), p. 379-411, 1993.

RoZ Disponible à l'adresse: <http://www-lsr.imag.fr/Les.Groupes/PFL/RoZ/index.html>

***THE AUDITIVE CONSISTENCE PROPERTY IN DISTRIBUTED MUSIC PERFORMANCE:
A CONDUCTOR BASED SYNCHRONISATION***

Nicolas Bouillot,

Doctorant en services systèmes pour le multimédia
CNAM-CEDRIC, 192 rue St Martin, 75141 Paris Cedex 03
bouillot@cnam.fr, + 33 1 58 80 85 47

Résumé : La croissance des systèmes de transmission de flux multimédia interactifs nous laisse envisager la possibilité de fournir à des musiciens géographiquement éloignés un médium permettant de jouer de la musique ensemble en temps réel. Cependant, le streaming de flux multimédia introduit des délais de bout en bout. Parmi les musiciens distants, ces délais provoquent ce que nous appelons une incohérence auditive. Comme nous allons le montrer, cette incohérence empêche les musiciens de jouer de la musique collectivement. Dans cet article, notre mécanisme orienté chef d'orchestre fournit la propriété de cohérence auditive entre différents musiciens choisis. De plus, ce mécanisme cache complètement la latence introduite par le réseau aux musiciens et rend possible le jeu musical collectif distribué.

Abstract: With the growth of interactive multimedia streaming on Internet, we expect to provide a tool making remote musicians play together in real-time and across the Internet. However, real-time streaming deals with delays, producing among musicians what we call an auditive inconsistency. As we will show, this inconsistency disables the collective musical practice. In this paper, our conductor driven scheme provides the auditive consistency property among chosen participant of the musical performance. This synchronization scheme hides the network latency to the musicians and enables the distributed collective musical practice.

Key words: Musical interactivity, real-time, PCM audio streaming, synchronisation, distributed system.

Mots clés : Interactivité musicale, temps réel, flux audio PCM, synchronisation, système distribué.

The auditive consistence property in distributed music performance: a conductor based synchronization

1 - INTRODUCTION

1.1 - The distributed virtual concert project

This project comes from the collaboration between IRCAM's and CNAM-CEDRIC's research laboratories. It may enable a real-time orchestra of remote musicians.

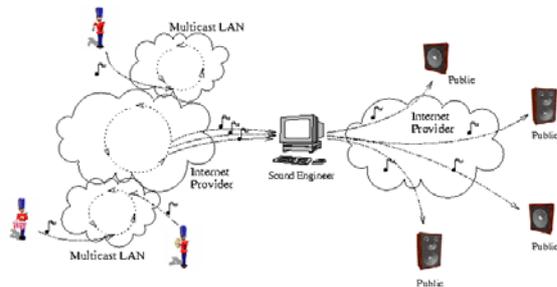


Fig 1. The distributed virtual concert

Figure 1 shows the general architecture of the distributed concert project. Remote musicians play together in real time, hearing each other through PCM audio streams. The different parts of the music (played by others) are heard after synchronization in the side fills of these streams. This allows them to play a real time and collective musical piece. In order to preserve local rhythms and audio quality in the streaming engine, latencies between musicians are kept constant. The public can hear the concert thanks to a sound engineer who performs remote control of the audio streams parameters as volume, localization in space, etc (see locher (2003)). In this way, he controls the streaming engine that achieves a traditional and spatial mixing. Spatial mixing consists in placing sources on a three dimensional space (then the sound is multi-channel). We focus here on the synchronization scheme among musicians and particularly on the way to play music according to the network constraints.

1.2 - The distributed auditive consistence problem

We will first show some differences between the traditional music practice in a band, i.e. when all musicians are in the same room, and the musical practice across the network. Then, we will define the distributed auditive consistency property.

Traditionally, the musical interaction is helped by various visual signs and conventions predetermined on the piece of music played (such as a sequence of chords with a theme for a piece of Jazz). At the same time, the musical contents added by each musician inform the others on the possible evolution of the piece. For example, groups of percussions sometime use rhythmic sentences to call each other to change rhythms. These kinds of interactions are possible due to the instantaneous hearing of the sound produced by each musician. For instance, if a bassist and a drummer play rhythms at the same time, everybody in the room will hear them together. We will next differentiate the direct sound produced by each musician and the side fills given by the environment. This allows us to consider the environment latency as the latency between the direct sound and the side fills. In a room, the environment latency exists due to the sound transmission delay in the air but it is not audible. In our work, we consider 20ms as the auditive delay perception threshold. As this latency cannot be heard musicians could synchronously feel the time as structured musical units (bars, quarter notes, triplets...). This common musical language gives them the opportunity to play synchronously their own part on the global performance. The environment keeps then the property we call the consistent auditive restitution. We can thus say that playing music with interaction is possible in a synchronous environment.

In our context of distributed "Live" music where the musicians are physically remote, networks and operating systems are asynchronous with audible and different latencies. Sound card, network, drivers and application provision the global latency. Let us show now why latencies impact on the consistent auditive restitution. Consider on figure 2 two musicians (Alice and Bob) who want to play across a network with perceptible delays.

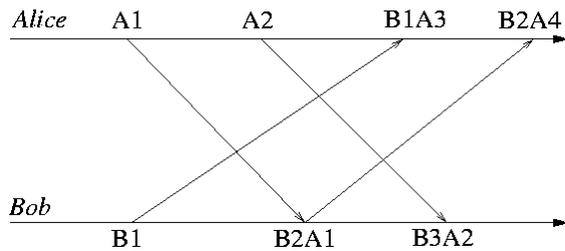


Fig 2. The auditory consistency problem

First, Alice is playing a sound at the time we call “A1”. When this sound is returned to Bob at “B1”, he is playing another sound. At this time, Bob hears the sounds «A1» and «B1» simultaneously. But he has to send to Alice his sound produced at “B1”. It is returned to Alice when she is producing the “A2” sound. This makes Alice hear the sound “A2” and “B1” simultaneously. As “B1” is mixed with different sounds for Bob and Alice, we can say that they are not living the same experience of the musical piece. This is what we call an inconsistent auditory perception of the music. It is then difficult to get a common rhythmical language. Suppose that Alice is a drummer and Bob is a bassist. If Bob feels himself rhythmically synchronized with the drums pattern, Alice will feel Bob out of the rhythm. Therefore she will adjust her rhythm in order to be synchronous with Bob but that will cause a de-synchronization at Bob’s side-fills, with a snowball effect.

1.3 - Solutions

To avoid this problem, we propose to add a mechanism that provides the consistent restitution. We distinguish two kinds of solutions. Bouillot (2003) has described the first one. This solution consists in adding consensually delays in each musician’s side fills, making each musician hear himself with a constant latency. There are situations where musicians are able to play with some delays, like organists interacting with a choir in church. But the musicians have to learn to play with this latency, limiting the spectrum of musical style possible. The second solution is presented in this paper (section 3) and it is called the “conductor synchronization scheme”. This solution gives to the musicians a synchronous side-fill with a zero-local latency perception. As a side effect, each site can hear only his own produced stream with the one coming from the conductor.

On section 2, a state of the art is given with short comparisons with our synchronization

scheme. We will next present our “conductor” based solution. Then we will present the current status of the prototype in section 4 and to finish, we will conclude in section 5.

2 - STATE OF THE ART

Several experiments of real time multimedia performances on Internet have already been achieved. Most of them use the MIDI standard (Musical Instrument Digital Interface). In Eliens (1997), latency between the production and the consumption of sounds is not designed to be constant so, variations are perceived while hearing. In the piano lesson system of Young (1999) and Fujinaga, a single musician is teaching while the pupils are listening. Then, there is no need here to play in a synchronized way. The team of Goto (1996, 1997) developed VirJa, a tool of virtual session of Jazz in which someone can play music with two processors, but there is no mechanism of synchronization of streams specified. An experiment of remote music was born based on RMCP (Remote Music Control Protocol) used in VirJa: *OpenRemoteGIG* (Goto (2002)) allows playing with remote musicians with MIDI streams and constant latency (an entire chord sequence). In this system, the musicians from various places will hear all the musicians but none of them hears the same resulting music. The example describes in Goto (1997) shows us this shift: the player improvises while listening to other sounds delayed by the constant period of the repetitive chord progression (a 12-bar blues chord progression in the text). Because the progression is repetitive, the delayed performance can fit the chords. In this way, musicians can play in a synchronous way. They will hear themselves instantaneously but simultaneously with the other delayed by the 12-bar latency. We can say that this work is done without a distributed auditory consistency and with a big latency reducing the interactivity between musicians. Despite the interest that such performances represents, the MIDI protocol allows to put aside some aspects of the transmission of PCM audio streams. MIDI takes less bandwidth thanks to its descriptive format but it also decreases the field of the transported sounds. Thus, it is difficult to make a direct analogy with the constraints raised by PCM audio streams. Among all of these papers no one

deals with the distributed auditive consistency problem or something equivalent.

Xu (2000) and Cooperstock (2001) have tested the transport in real time of PCM audio streams created on fly. The authors used a recording studio to sample a performance that took place in another country. The musicians were located at the same place, raising no problem on interactions between musicians.

The closest work done by another team is the “SoundWire” project. They experienced the streaming of professional-quality audio across the Internet2 network (see Chafe (2000)). Few LAN and WAN experiments were deployed in the United States, where round trip time was about a 2 factor slower than the speed of the light. In these experiments, the main musical goal was to see how musicians could play with latency introduced by the network. As these latencies were closed to the one perceptible by the ear, the musicians could progress together in the musical piece but not rhythmically synchronous (see SoundWire (2002) Live WAN test audio examples). The reason is that they did not include any musical synchronization in their streaming engine. In comparison with our experimental context, we are working on Internet with bigger round trip times. But even with the smallest delays possible (the light speed), “*the theoretical round trip time across USA and back is approximately 40msec*”. We take 20msec as the perceptible latency between two clicks delayed. This threshold is stated also by simulated delay tests from Schuett (2002) audio examples. We hear that with a 20ms delay, musicians can play together but with a 30ms delay, we can hear a difference between the side-fills of the two musicians. This let us imagine perceptible delays with bigger distance. Schuett’s thesis (2002) tries to “*define the level of delay at which effective real-time musical collaboration shifts from possible to impossible*”. The author talks about a leader-follower relationship between musicians. However, this relationship is different from the one presented in our work. In the Schuett’s study, two performers are subject to symmetric delays added with a digital mixing console. He remarks that the musicians can play together if one of them follows rhythmically the other one. This relationship is based on the musicians’ behavior. We point out that in our work, we

synchronize the audio streams thank to the streaming engine and the time stamping. Remote performers are then subject to delays that could be asymmetric. This is here the synchronization scheme which is conductor driven.

3 – THE SYNCHRONISATION SCHEME

3.1 – Justification of such an architecture

Playing music in a band is itself a hard task. Then we want to minimize the difficulties added by the streaming engine. Playing music with remote musicians avoids social interactions as looking to the other musicians, hearing them instantaneously... In order to provide an easier practice of the remote playing, we believe that musicians have to be in a synchronous musical environment. This immersion is possible if we include synchronization mechanisms in each local musician’s side-fills. We thus have to consider things like: “what does each musician should hear?” In Bouillot (2003), we supposed that each musician hears each other to keep a musical interaction. But it depends on the kind of music, especially when the entire band must communicate. This approach provides a good interactivity among the musicians but the side effect of keeping each stream in the side-fills is that the local sound is delayed too. The musical practice becomes then more difficult, but still possible. The idea we present here is quite different: the musicians will hear only two streams, the one from the conductor and the one from themselves. This is acceptable with some kinds of music where there is a little interaction between musicians. This is the case of many written music where the most important interaction is the one with the conductor. Symphony is a good example: in a symphonic orchestra, musicians are looking at the conductor and at the same time they are hearing their neighbors to play music. We can find some other examples as the Mugam of Azerbaijan, one of the liveliest Orient’s classical musical styles. Bois (1993) explains that this music requires « *a singer, [...] and two musicians (sazande) playing on the luth (târ) and the spike-fiddle (kemânche). The târ converses directly with the sung phrases, while the kemânche sometimes sustains the singer sometimes the târ player* ». Then, the singer needs to hear the târ player. The târ player needs to hear the singer but the kemânche

player stream is not necessary at the other's side fills to play in a synchronous way. At the question “what do musicians hear? » our conductor synchronization scheme answers that each musician must hear his own produced sound and the conductor sound. In this way, we can make easier the distributed musical practice. To compare that with our previous work, this avoids the local latency and also the interaction between musicians. With this conductor driven scheme, we can distribute the Mugam's musicians by associating the târ player with the singer as the conductor and the kemânche player as a musician. Thus the kemânche player can play without local latency.

3.2 - Description

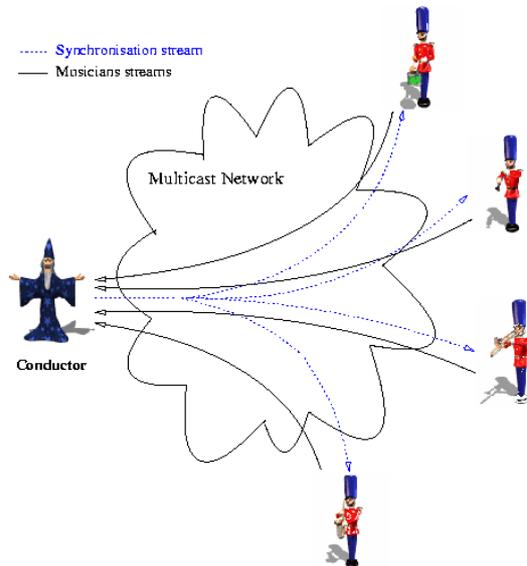


Fig 3. The conductor architecture

The idea behind the conductor driven architecture is quite simple. The conductor broadcast his own sound, for instance a beat, time stamped with the local sound card's clock. Helped by the streaming engine, the musicians can hear the sound sent by the conductor. It is easy now for all of them to play synchronously with the beat heard on the side-fills. As musicians got a global reference (the conductor's stream), it is able to mix all the streams in a synchronous one. In order to do that, each musician's sample are time stamped with the stamp of the conductor's sample perceived simultaneously. The conductor (or someone else) could receive each stream and mix them in the synchronous way by playing simultaneously the samples having the same timestamp. To keep

interactivity in the musical piece, the conductor can mix his own beat with his voice to indicate to musicians how the musical piece will change.

Let's call n the number of musicians, $m_i (0 \leq i < n)$ a musician, c the conductor, t_x the x 's local sound card clock, s_x^j a sample produced by x at $t_x = j$ and $t_x(s_y^j)$ the time when s_y^j is processed (produced, consumed or both) by t_x . If $x = y$, then s_y^j is produced by t_x . We can thus say that say $t_x(s_x^j) = j$ and but that $t_x(s_y^j) \neq j$ (the sample comes from another source. So it is here stamped by another clock). We call now $e_{m_i}(s_{m_i}^j)$ the timestamp associated to the sample $s_{m_i}^j$ (in the musician m_i 's output stream). According to the PCM audio streams semantic, sound card sample production and time stamping respect local order. Timestamps and sample produced at the sound card's clock are thus incremented, i.e.

$$t_x(s_x^j) + 1 == t_x(s_x^{j+1})$$

and

$$e_{m_i}(s_{m_i}^j) + 1 == e_{m_i}(s_{m_i}^{j+1})$$

These considerations suppose that the remote sound cards are running at the same sample rate. If not, a conversion can be added.

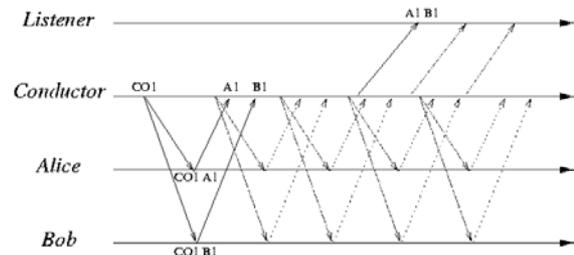


Fig. 4. The streams in the conductor driven scheme

As Figure 4 shows, in the conductor architecture, the conductor broadcast his own sound. Each musician receives thus all the s_c^j samples as a stream (Figure 4 shows us the sample CO1 broadcasted to the musicians). With that stream, each musician gets samples (with the timestamps associated) to play out. The main objective for them is now to timestamp their own stream with the values

that allow the conductor to mix them and get a resulting mix synchronized. Locally to m_i , the input stream coming from the m_i 's instrument and the stream coming from the conductor are processed by the same clock (t_{m_i}). Then for each musician m_i , for each locally produced sample, the timestamp associated is $e_{m_i}(s_{m_i}^k)$ where:

$$e_{m_i}(s_{m_i}^k) = j \text{ with } t_{m_i}(s_{m_i}^k) = t_{m_i}(s_c^j)$$

On figure 4, the A1 and B1 timestamp are set to CO1. After the timestamp calculation, each musician will send his produced sample with the associated timestamps to the conductor, who will mix the streams as:

$$t_c(s_{m_0}^{k_0}) = \dots = t_c(s_{m_{n-1}}^{k_{n-1}})$$

$$\text{where } e_{m_0}(s_{m_0}^{k_0}) = \dots = e_{m_{n-1}}(s_{m_{n-1}}^{k_{n-1}})$$

The conductor mixes these samples to hear them or to send the mix to listeners (Figure 4).

If musicians play synchronously with the conductor stream (the global reference), the mix will be synchronous. To keep an interaction between the conductor and the musicians, the conductor has to hear the resulting music, which includes a rhythmic de-synchronization between his side-fills and his direct sound. Therefore, he can send a pre-recorded sound (a beat for example) and hear the mix only, avoiding the problem. Another solution is to take two people as the conductor: one plays the synchronization stream and the other one hears the mix to give the indications. In this way, we expect a minimal latency from musicians to the conductor.

We can notice now that musicians do not perceive the network latency: they are just playing with a received stream. However, the conductor can hear all of them and perform the interactivity by giving them indications.

3.3 - Bandwidth requirement

A mono PCM stream with a 44100Hz sample rate sent with RTP (Schulzrinne (1998)) represents 0,7Mb/s. As we use IP multicast in our streaming engine, the conductor need to send 1 synchronization stream and need to receive n streams from the musicians. The conductor thus needs a 0,7Mb/s upload

bandwidth and a $0,7 \times n$ download bandwidth. Each musician needs 0,7Mb/s as upload bandwidth (his own produced sound) and 0,7Mb/s as download bandwidth (the conductor stream). Nowadays, these bandwidth requirements are too big to place the conductor on a DSL connection at home but not to play through academic Internet providers, which is our actual experimentation context.

3.4 – Combination with previous work

The conductor architecture fits well with the music where there are little interactions between musicians. It provides a null latency perception to the musicians. Bouillot (2003) focus on interactions between all the musicians and includes latency in the perception of the local produced sound. These two kinds of interactions have advantages and disadvantages but are not incompatible. We can take the advantages of both, thanks to an appropriate distribution of the roles between each musician in a band, according to the expected interactions. For example, the resulting music of a group of musicians that interacts with the self-synchronization algorithm (see Bouillot (2003)) music could be the synchronization stream in the conductor architecture. The other musicians can then play with this synchronization steam in a “conductor” oriented way. Therefore, by coupling both synchronizations, the solution to the auditive consistence problem can be generalized with two levels of added difficulties at the musicians’ instrumental practice.

4 – CURRENT STATUS

We have not implemented yet the conductor driven scheme in our prototype. However, as we will show, our streaming engine is modular and can be easily extended with the conductor driven scheme.

4.1 – The Streaming engine

We developed the streaming engine as part of the jMax (Déchelle (2000)) visual language. Figure 5 shows a jMax patch example. This visual language works with a message oriented semantic. For instance, a modification on the slider will send an integer to the division object, which will process the division by 127. By implementing reception and emission of audio streams as a function of this language,

we can easily route the different audio streams, as generated music, microphone input stream, speakers output stream or received streams. jMax allows us to process sound in real-time and gives us a modular approach to configure the streaming engine.

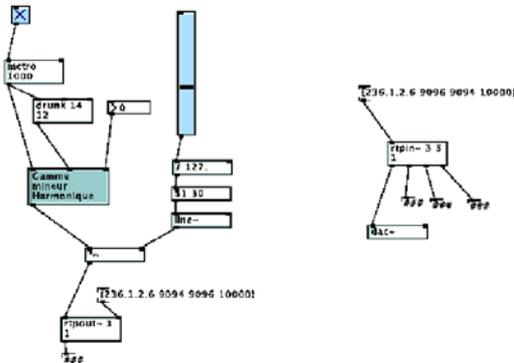


Fig 5. The rtp objects in a jMax visual program

In our first prototype, the reception of audio streams has been achieved with the *rtpin* object developed in jMax (figure 5) and the emission in the *rtpout* object. The sound samples are produced at a constant rate of 44100Hz and consumed by the *rtpin* objects (at the same rate). The transport of audio streams is done using the RTP protocol (Schulzrinne (1998)), through the RTP library called UCL Common Code Library version 1.2.8 developed by the Computer Science Department of the University London College University. Each *rtpout* object stamps the samples in the RTP's timestamp field. The timestamp is incremented per sample. Helped by the RTP's *ssrc* field, we get the music source identification. Each *rtpin* object consumes simultaneously a sample coming from each *rtpout* remote object.

4.2 – Implementing synchronization

In our prototypes, we deployed the synchronization described in Bouillot (2003). We tested successfully this kind of synchronization on LAN (jMax 2.5.1) and on MAN (with a second prototype deployed on jMax 4.1) where the musicians played a distributed Blues in a synchronous way with a synchronized start. Drums/bass were located at the IRCAM center and guitar/saxophone at the CNAM University with a distance of one kilometer across two Internet providers.

In our second prototype we took only one object for both reception and emission. With this implementation, we have local sound card

clock, time stamping, streaming and a shared space to read and write samples on the sound cards. Then, we can easily implement our conductor driven synchronization scheme.

5 - CONCLUSION AND FUTURE WORKS

We have seen that the major interest here is to provide a way for distributed musicians to play without any network latency perception. In all the other distributed musical systems, the instrumental playing deals with delays, including an alteration in the instrumental practice. Although our conductor driven architecture focuses the musicians on the conductor, it makes the distributed musical practice easier for musicians and keeps the auditive consistency among people interacting together. We can then provide a more generic solution to the distributed way of playing music with a combination of the conductor architecture and the self-synchronization algorithm described in a previous work (see Bouillot (2003)). This new solution would make the system's architect define interaction between musicians, allowing each of them to one of both synchronizations schemes discussed before. However, this combination introduces some alterations on the traditional way of playing music. This introduces two kinds of musical experimentations. The first one is to try to project existing kinds of music on our solution. The second is to work with composers and musicologists to develop a new kind of musical interaction, which would be aware of these synchronization schemes. As a side effect, the solution to the auditive consistency problem could fit well with some distributed virtual reality problems.

Added to the developments, we will achieve tests with bigger distances with more sites. The work of experimentation will be supplemented by a network provisioning. The guarantee of constant latency between the musicians and the conductor has a basically statistical nature.

In the future, it will be necessary to take account of the losses and to choose a strategy among the mechanisms presented in Perkins (1998), Bolot (1999) and Rosenberg (1998) or in the literature where the main challenge is to compensate losses without retransmissions.

The last work in progress is the skew existing between the clocks of the different sound cards. This skew occurs between the emission

time stamping clock (the sound card) and the reception clock. The problem is a many-to-many one. Orion (2000), Akester (2002) and Fober (2002) present a solution but in a one-to-one context.

However our priority is to make tests on larger network as MAN and WAN. On one hand, this will allow to dimension our prototype and on the other hand, to determine more precisely the kind of interactions that we will provide to the remote musicians.

REFERENCE

- Akester R., Hailes S. (2002), "A New Audio Skew Detection and Correction Algorithm". In proceedings of the *International Computer Music Conference*, ICMC (2002).
- Bois Pierre (1993), "The Mugam of Azerbaïdjan". In *Sakine Ismailova, Anthologie du Mugam d'Azerbaïdjan, Vol. 5 booklet*. CD n. w260049, INEDIT collection. Maison des Cultures du Monde
- Bolot J.C., Fosse-Parisis S., Towsley D. (1999), "Adaptive FEC-Based Error Control for Internet Telephony". *INFOCOM*, vol 3, pages 1453-1460 (1999).
- Bouillot N. (2003), "Un algorithme d'auto synchronisation distribuée de flux audio dans le concert virtuel réparti". Proceedings of the *conférence française sur les systèmes d'exploitation CFSE'03*, La Colle Sur Loup, France (2003).
- Chafe C., Wilson S., Leistikow R., Chisholm D., Scavone G. (2000), "A Simplified approach to high quality music and sound over IP". *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00)*, Verona, Italy, December 7-9, 2000
- Cooperstock J., Spackman S. (2001), "The Recording Studio That Spanned a Continent". *IEEE International Conference on Web Delivering of Music*, WEDELMUSIC, Florence Italie (2001).
- Déchelle, F., Borghesi, R., Orío, N., Schnell, N., « The jMax environment: an overview of new features », *ICMC: International Computer Music Conference*, Allemagne, 2000.
- Diot P., Huitema C., Turletti T. (1995), "Multimedia Applications should be Adaptive". Proc. *HPCS'95*, Mystic (CN) (1995).
- Eliëns A., van Welie M., van Ossenbruggen J., Schönhage B. (1997), "Jamming (on) the Web". Proceedings of *WWW6* (1997).
- Fober D., Orlarey Y., Letz S. (2002), "Clock Skew Compensation over a High Latency Network". Proceedings of the *International Computer Music Conference*, 548-552 (2002).
- Goto M., Hidaka I., Matsumoto H., Kuroda Y., Muraoka Y. (1996), "A Jazz Session System for Interplay among All Players". *ICMC Proceedings*, pages 346-349 (1996).
- Goto M., Neyama R. (2002), "Open RemoteGIG: An Open-to-the-public Distributed Session System Overcoming Network Latency". *IPSI JOURNAL*, vol 43, pages 299-309, IN JAPANESE (2002).
- Goto M., Neyama R., Muraoka Y. (1997), "RMCP: Remote Music Control Protocol, design and applications". *ICMC Proceedings*, pages 446-449 (1997).
- Locher H-N., Bouillot N., Becquet E., Dechelle F., Gressier-Soudan E. (2003), "Monitoring the Distributed Virtual Orchestra with a CORBA based Object Oriented Real-Time Data Distribution Service". In proceedings *DOA'03 International Symposium on Distributed Objects and Applications*. Catagna, Italy. November., 2003.
- Orion, Hodson, Colin (2000), "Skew Detection and Compensation for Internet Audio Applications". (2000).
- Perkins C., Hodson O., Hardman V. (1998), "A survey of packet-loss recovery techniques for streaming audio". *IEEE Network Magazine* (1998).
- Rosenberg J., Schulzrinne H. (1998), "An RTP Payload Format for Generic Forward Error Correction". *Internet-Draft draft-ietf-avt-fec-03.txt* (work in progress) (1998).
- Schuett N. (2002), "The effect of latency in ensemble performance". Thesis, CCRMA, department of music. Stanford University.
- Schulzrinne, Casner, Frederick, Jacobson (1998), "RTP: A Transport Protocol for Real-

Time Applications”. RFC 1889 (1998).

SoundWire home page (2002), <http://ccrma-www.stanford.edu/groups/soundwire/>

Xu A., Cooperstock J. (2000), “Real-Time Streaming of Multichannel Audio Data over Internet”. *AES 108th convension*, Paris (2000).

Young J.P., Fujinaga I. (1999), “ Piano master classes via the Internet ”. Proceedings of the *International Computer Music Conference*, pages 135-137 (1999).

VERSIONS ON-LINE DU PROBLEME DU STABLE MAXIMUM

Bruno Escoffier,

Doctorant en informatique
escoffier@lamsade.dauphine.fr

Vangélis Th. Paschos,

Professeur en informatique
paschos@lamsade.dauphine.fr

Adresse professionnelle

Lamsade, Université Paris IX Dauphine ★ Place du maréchal de Lattre de Tassigny ★
75775 Paris cedex 16

Résumé : Dans un problème on-line, l'instance du problème n'est pas entièrement connue au départ mais est "révélée" en plusieurs étapes, le but étant de construire, au cours de ce processus où l'on découvre l'instance, une solution réalisable la meilleure possible. Des modèles on-line pour le problème du stable ont été étudiés (voir notamment "*On-line maximum-order induced hereditary subgraph problem*" M. Demange, X. Paradon, V. Th. Paschos. SOFSEM 2000, p. 327-335) : l'instance y est alors initialement vide, et à chaque étape des sommets apparaissent jusqu'à l'obtention du graphe final. Nous allons ici étudier l'approximabilité de nouveaux modèles on-line : le premier modèle consistera à changer le mode de révélation de l'instance. Dans le deuxième, on étudiera le modèle classique mais en le relâchant par l'introduction d'une possibilité de backtracking. Cette étude sera faite sous l'angle de l'approximation : on recherchera ainsi des résultats positifs (algorithmes à garanties de performance) et des résultats négatifs (bornes sur les garanties possibles).

Summary : In an on-line problem, the instance is not completely known at the beginning, it's given in many stages. The goal is to build, during this process, a feasible solution as good as possible. An on-line model specifies how the instance is revealed and how an algorithm can construct its solution. Some on-line models for the max independent set problem have been studied (see "*On-line maximum-order induced hereditary subgraph problem*" M. Demange, X. Paradon, V. Th. Paschos. SOFSEM 2000, p. 327-335) : in such models, the instance is empty at the beginning. At each stage, a vertex (or a set of vertices) appears, and so on until the graph is completely revealed. Here, we will study approximability of new versions of the on-line independent set problem : in the first one, we change the way the instance is given; in the second one, we consider the 'classical' model but we introduce a relaxation : a possibility of backtracking. We will focus on finding approximation properties of such problems, i.e. both positive results (approximation algorithms) and negative ones (bounds on approximability).

Mots clés : optimisation combinatoire, algorithmique on-line, ensemble stable, algorithmes approchés, garanties de performance.

Versions on-line du problème du stable maximum

1 - INTRODUCTION

1.1 - Problème du stable et rapport d'approximation

Rappelons tout d'abord qu'un stable d'un graphe $G=(V,E)$ (où V est l'ensemble des sommets et E l'ensemble des arêtes) est un sous-ensemble S de sommets deux à deux non adjacents, c'est-à-dire tel que: $(x,y) \in S^2 \Rightarrow (x,y) \notin E$. Le but du problème est alors, étant donné un graphe, de trouver un stable de ce graphe de cardinal maximal.

Le problème du stable maximum est NP-difficile, et nous allons nous placer dans le cadre de la recherche de résolutions approchées. Un algorithme approché A pour un problème d'optimisation fournit, pour toute instance du problème, une solution réalisable (ici un stable). On mesure alors la qualité de la solution fournie par A sur le graphe G en regardant le rapport $\frac{|A(G)|}{\alpha(G)}$, où $|A(G)|$ désigne

le cardinal de la solution fournie par A et $\alpha(G)$ désigne le cardinal d'une solution optimale (que l'on nomme souvent nombre de stabilité du graphe).

On dit qu'un algorithme garantit $f(G)$ (où f est une fonction de l'ensemble des instances G à valeurs dans $[0,1]$) si pour toute instance G , $\frac{|A(G)|}{\alpha(G)} \geq f(G)$. On s'intéresse bien sûr particulièrement à des garanties d'algorithmes polynomiaux, et souvent à des garanties fonctions uniquement de la taille de l'instance.

Pour le problème du stable, la meilleure garantie polynomiale connue (en fonction du nombre n de sommets du graphe instance G) est en $O\left(\frac{\log^2(n)}{n}\right)$ (voir [Boppana, 1992]).

Rappelons également qu'il existe une version pondérée du problème du stable : chaque

sommet a un poids, le but étant alors de trouver un stable de poids maximal (le poids d'un sous-ensemble de sommets étant bien entendu défini comme la somme des poids des sommets). Du point de vue de l'approximation polynomiale, il est intéressant de noter que l'on dispose également d'un algorithme offrant une garantie en $O\left(\frac{\log^2(n)}{n}\right)$ (voir [Halldorsson, 2000a]).

1.2 - Problèmes on-line

L'algorithmique dite "on-line" est apparue pour modéliser des situations où l'instance d'un problème n'est pas entièrement connue quand commence la résolution effective. Il est par exemple fréquent que l'on doive commencer cette résolution dès réception des premières informations (problèmes en-ligne, avec contraintes de temps de réponse,...).

De nombreux problèmes d'optimisation combinatoire classiques ont été étudiés dans des versions on-line (on spécifie alors simplement des règles concernant la manière dont les instances sont révélées et la manière de construire une solution), voir par exemple [Ausiello,2001] pour le problème du voyageur de commerce, [Halldorsson, 2000b] et [Halldorsson, 1994] pour la coloration, [Demange, 2000], [Halldorsson, 2002], [Paradon, 2000] et [Paschos, 2001] pour le stable.

Considérons par exemple la situation suivante : une entreprise reçoit, régulièrement, des contrats pour des commandes de matériels. Cette entreprise, disposant de moyens (techniques, humains,...) limités, ne peut honorer tous les contrats, et est ainsi contrainte d'en refuser certains (son but étant d'en accepter le plus possible). Quand elle reçoit une proposition, l'entreprise doit, évidemment, donner sa réponse dans un intervalle de temps limité (soit très rapidement, soit à la fin du mois par exemple).

Modélisons cette situation par un graphe dont les sommets sont les contrats. Imaginons que

les incompatibilités se résument à des paires de contrats que l'entreprise ne peut honorer simultanément, ces paires constituant les arêtes du graphe. Le problème de l'entreprise est alors de trouver un stable de taille maximale dans ce graphe.

Cet exemple permet de définir le modèle le plus naturel pour le stable on-line. Un modèle consiste à définir comment l'instance est révélée à l'algorithme, et quels sont les droits de celui-ci pour construire sa solution. Ici, nous considérons donc que le graphe initial est vide. A chaque itération, on a un ensemble de sommets qui se rajoutent au graphe courant (les nouveaux contrats dans l'exemple). Un algorithme doit alors choisir de manière irrévocable quels sommets il met dans sa solution (quels contrats il accepte). Ce modèle contient un cas particulier, celui où les sommets arrivent un par un (c'est le cas où l'on doit choisir dès réception du contrat).

1.3 - Résultats précédents et objectifs de l'étude

C'est le modèle précédent qui a été étudié dans [Demange, 2000]. L'analyse y est faite sous l'angle de la recherche d'algorithmes à garanties de performance (en étudiant toujours le rapport du cardinal de la solution fournie par un algorithme on-line par le nombre de stabilité du graphe). Il est à noter que, contrairement au problème classique (que l'on appellera désormais off-line), la recherche de garantie pour des algorithmes non polynomiaux est intéressante : en effet, le fait de faire des choix irrévocables sans connaître la totalité de l'instance débouche sur le fait qu'aucun algorithme on-line ne va pouvoir être optimal.

Le résultat principal concerne l'existence d'un algorithme (non polynomial) garantissant $\frac{1}{\sqrt{nt}}$ (où t est le nombre d'itérations nécessaires à la révélation de l'instance), et d'un algorithme polynomial garantissant $O\left(\frac{\log(n)}{n\sqrt{t}}\right)$. On obtient ainsi des garanties

intéressantes si le nombre d'itérations est faible (en particulier s'il ne dépend pas de n). Détaillons l'obtention de ces résultats, puisque nous en réutiliserons l'idée par la suite. Le

problème principal lors de la recherche d'un algorithme offrant une garantie de performance est la cécité de celui-ci ; ne connaissant pas la suite de l'instance, il se peut que, dès qu'il choisit un sommet, tous les sommets suivants soient reliés au sommet choisi et ainsi l'algorithme ne peut augmenter sa solution. Il semble donc que, pour pouvoir obtenir une garantie de performance intéressante, un algorithme ne doit prendre des sommets que s'il peut en prendre au moins un certain nombre. Considérons un algorithme pour le problème du stable classique (off-line), et considérons l'algorithme on-line qui, à chaque fois qu'un sous-ensemble de sommets S_i arrive :

- fait agir l'algorithme off-line sur S_i ;
- renvoie cette solution si sa taille est plus grande qu'un certain seuil, attend l'itération suivante sinon.

C'est en réglant de manière optimale le seuil, et en prenant pour algorithme off-line un algorithme exact (non polynomial) puis le meilleur algorithme polynomial connu que l'on obtient les deux résultats précédents.

Les auteurs montrent également qu'aucun algorithme (quelle que soit sa complexité) ne peut garantir mieux que $\frac{1}{\sqrt{n-1}}$, prouvant ainsi que l'algorithme proposé ne peut être sensiblement amélioré si t est constant (on ne peut faire mieux qu'une garantie en $O\left(\frac{1}{\sqrt{n}}\right)$).

L'objectif ici va être de définir et d'analyser, toujours en terme de recherche de garanties de performance, deux nouvelles approches du problème du stable on-line.

Dans la première, on va modifier le mode de révélation de l'instance. L'instance initiale va cette fois ci être un graphe complet, et ce sont les arêtes qui vont disparaître, jusqu'à l'obtention du graphe final. On aboutira pour ce modèle à des garanties similaires à celles trouvées pour le premier modèle.

Ensuite, on reprendra le modèle initial en le relâchant. Une première idée de relaxation, proposée dans [Halldorsson, 2002], consiste à permettre à l'algorithme de garder à chaque

itération plusieurs solutions réalisables. Nous introduirons ici une possibilité de backtracking: l'algorithme pourra revenir sur certaines de ces décisions, cette nouvelle liberté étant compensée par l'instauration d'un système de pénalisation. Nous verrons ainsi comment on peut améliorer les garanties de performance précédentes

2 - MODELE DE DISPARITION DES ARETES

On va donc considérer ici le modèle suivant : le graphe initial est complet, et à chaque itération un ensemble d'arêtes disparaît. L'algorithme peut, à chaque itération, rajouter dans sa solution les sommets adjacents à (au moins) une arête supprimée. On considère bien entendu que quand l'algorithme choisit de mettre un sommet dans sa solution, ce choix est irrévocable.

Soit FA un algorithme pour le problème off-line.

Théorème 1 : *Si FA garantit $\rho(n)$ (décroissant en n), alors, en supposant que le nombre d'itérations est connu au départ, il existe un algorithme on-line $OLTA$ garantissant $\frac{\rho(n)}{\sqrt{nt}}$. De plus, si FA est polynomial, $OLTA$ l'est aussi.*

En particulier, en utilisant un algorithme FA exact, on obtient une garantie de $\frac{1}{\sqrt{nt}}$. On obtient également une garantie polynomiale en $O\left(\frac{\log(n)}{n\sqrt{t}}\right)$. Ces résultats sont particulièrement intéressants si le nombre d'itérations est faible (constant notamment).

Schéma de la preuve :

L'algorithme $OLTA$ est une adaptation de celui présenté dans [Demange, 2000]. Le problème est en effet similaire ici : l'algorithme est dans une situation de cécité, la notion de seuil est encore pertinente. On va donc adopter une stratégie analogue : à chaque itération, on utilise un algorithme off-line et, si la solution fournie par celui-ci est plus petite que le seuil, on ne prend aucun sommet et on attend l'itération suivante. En réglant le seuil optimalement, on va alors pouvoir montrer que

soit on aura une solution de taille conséquente, soit le graphe aura un nombre de stabilité faible, et aboutir ainsi au résultat.

Théorème 2 : *Même s'il n'y a que deux itérations, aucun algorithme ne peut garantir mieux que $\frac{1}{\sqrt{n/2-1}}$.*

Cette borne supérieure est de l'ordre de la garantie trouvée précédemment lorsque le nombre d'itérations est constant, ce qui indique que dans un tel cas on ne peut faire réellement mieux que l'algorithme à seuil $OLTA$.

Schéma de la preuve :

Il s'agit ici de "piéger" tout algorithme on-line A . Pour cela, on va révéler le graphe G (comportant n sommets v_1, v_2, \dots, v_n) en deux itérations. La première consiste à supprimer toutes les arêtes reliant deux sommets parmi $V = \{v_1, v_2, \dots, v_p\}$, où $p = \lfloor \sqrt{2n} \rfloor$, de manière à ce que V soit un stable.

La deuxième itération dépend des choix faits par l'algorithme A à l'issue de la première. Si A a choisi au moins un sommet (nécessairement dans V), on supprime les arêtes non adjacentes à un sommet de V . Dans ce cas, $|A(G)| \leq p$ et $\alpha(G) \geq n - p$. Sinon, on supprime une seule arête, entre v_1 et v_n par exemple. Alors $|A(G)| \leq 2$ et $\alpha(G) \geq p$. La valeur de p permet de montrer que, dans chacun des deux cas, $\frac{|A(G)|}{\alpha(G)} \leq \frac{1}{\sqrt{n/2-1}}$.

Remarques :

- Si l'on s'intéresse aux garanties en fonction du degré Δ du graphe, on peut facilement obtenir, en définissant un algorithme (polynomial) de type glouton, une garantie de $\frac{1}{\Delta}$.
- Nous avons supposé ici que le nombre d'itérations est connu au départ par l'algorithme. Si tel n'est pas le cas, on peut alors montrer qu'aucun algorithme ne peut garantir mieux que $\frac{2}{n-1}$, ce qui est donc très faible.

3 - RELAXATION PAR UNE POSSIBILITE DE BACKTRACKING

Dans cette partie, nous reprenons donc le modèle initial où l'on part d'un graphe vide et où les sommets arrivent par paquets. Nous allons relâcher ce modèle en considérant que les choix faits par l'algorithme ne sont plus irrévocables. Un algorithme pourra ainsi choisir un sommet apparu à une itération précédente, ou sortir de la solution un sommet qu'il avait précédemment choisi.

Néanmoins, cette relaxation est trop forte puisqu'il suffit alors d'attendre la dernière itération et l'on est ramené au problème off-line. Ceci nous a conduit à l'idée d'instaurer des pénalités : si l'algorithme choisit un sommet révélé à une itération précédente, celui-ci ne vaudra plus une unité pleine mais aura une valeur entre 0 et 1, susceptible d'évoluer au cours du processus de révélation du graphe.

Plus précisément, nous allons définir un paramètre réel $k \in]1, \infty[$. La valeur d'un sommet apparu à l'itération i et choisi à l'itération $j \geq i$ sera $\frac{1}{k^{j-i}}$. Le but du problème est alors de construire un stable de valeur maximale, cette valeur étant bien entendu définie comme la somme des valeurs des sommets du stable.

On mesurera ici encore la qualité de la solution d'un algorithme on-line *OLA* sur une instance G en étudiant le rapport $\frac{val(OLA(G))}{\alpha(G)}$ (où $val(S)$ désigne la valeur d'un stable S).

Maintenant, l'algorithme a donc face à lui à l'itération i un graphe valué de la manière suivante :

- Les sommets qui viennent d'arriver valent 1.
- Les sommets révélés à l'itération précédente valent $\frac{1}{k}$.
- ...
- Les sommets révélés à la première itération valent $\frac{1}{k^{i-1}}$.

Il peut choisir n'importe quels sommets de ce graphe. Il semble alors naturel d'utiliser maintenant un algorithme off-line pour le

problème du stable pondéré. Soit *WFA* un tel algorithme :

Théorème 3 : Si *WFA* garantit $\psi(n)$ (décroissant en n), alors il existe un algorithme on-line *WOLA* garantissant $\frac{\psi(n)}{t(1-1/k)-1/k}$.

De plus, *WOLA* est polynomial si *WFA* l'est.

On obtient donc une nette amélioration des garanties de performance par rapport au modèle non relâché. On obtient en particulier un algorithme (non polynomial) en $\frac{1}{t}$, donc à rapport constant si t est constant.

Schéma de la preuve :

L'algorithme *WOLA* consiste simplement à faire agir à chaque itération *WFA* sur le graphe valué dans son ensemble. Si la solution est meilleure que la solution courante, alors on prend cette nouvelle solution comme solution courante (ce qui est permis étant donné que les choix ne sont plus irrévocables).

L'obtention de la garantie de performance est un peu longue, mais elle consiste à :

- majorer la valeur d'un stable de poids maximal dans les graphes intermédiaires en fonction de la solution fournie par *WOLA* (ce qui est immédiat) ;
- majorer le nombre de stabilité du graphe en fonction des stables de poids maximal des graphes intermédiaires.

On peut maintenant légitimement se demander si l'on ne pourrait pas améliorer sensiblement cette garantie. La réponse est négative (du moins si le nombre de sommets est "assez grand"), puisque l'on a le théorème suivant :

Théorème 4 : Si $n \geq \frac{t(t+1)}{2}$, aucun algorithme ne peut garantir strictement mieux que $\frac{1}{t(1-1/k)}$.

La borne supérieure est donc quasiment égale à la garantie trouvée précédemment.

Schéma de la preuve :

Il va ici encore falloir piéger tout algorithme on-line *OLA*. Nous allons construire notre instance de la manière suivante : le premier sous-ensemble de sommets sera une clique de taille t , le deuxième une clique de taille $t-1$, ..., l'avant dernier une clique de taille 2, et le dernier une clique formée des $n-t(t-1)/2$ sommets restants (d'où l'hypothèse sur n dans l'énoncé du théorème). Pour les arêtes, on relie à chaque itération tous les sommets de la nouvelle clique à tous les sommets de la solution courante. On peut alors montrer que dans ce graphe :

- il existe un stable de taille t ;

$$- \text{val}(\text{OLA}(G)) \leq 1 + \frac{1}{k} + \dots + \frac{1}{k^{t-1}} \leq \frac{1}{1-1/k}$$

Ces deux propriétés permettent d'obtenir le résultat annoncé.

Remarque :

On pourrait envisager une relaxation plus limitée en considérant que le choix de mettre un sommet dans la solution demeure irrévocable (mais en se permettant de prendre des sommets que l'on avait précédemment refusés). Néanmoins, dans un tel modèle, le problème de cécité de l'algorithme reste entier et l'on ne parvient à améliorer que peu sensiblement la garantie obtenue dans le modèle non relâché.

4 - CONCLUSION

L'étude de problèmes on-line montre comment ces problèmes relativement nouveaux constituent un cadre naturel pour un prolongement de la théorie de l'approximation classique. On y recherche ainsi des résultats positifs (des garanties de performance) et négatifs (des bornes sur les garanties possibles), le but étant bien entendu de minimiser l'écart entre la meilleure garantie trouvée et la "meilleure" borne. On y retrouve également la notion clé de réduction en approximation permettant d'obtenir une garantie de performance pour un problème en fonction de celle d'un autre (ici entre les versions on-line et off-line de ce problème).

Il pourrait être intéressant d'élargir cette notion de réduction en approximation en réduisant un problème on-line à un autre problème off-line, ou même à un autre problème on-line.

Une deuxième voie pourrait être d'appliquer les modèles vus ici sur le stable (la relaxation notamment) à d'autres problèmes d'optimisation on-line, comme la coloration.

Enfin, il serait particulièrement intéressant et utile d'essayer d'unifier les différentes approches dans l'étude des problèmes on-line, les différents modèles, de proposer par exemple une sorte de classification permettant de définir clairement des problématiques, de manière à éviter un éparpillement dû au grand nombre de modélisations possibles.

BIBLIOGRAPHIE

[Ausiello, 2001] Ausiello G., Feuerstein E., Leonardi S., Stougie L., Talamo M.: "Algorithms for the on-line travelling salesman problem". *Algorithmica*, Vol 29, N°4, p. 560-581 (2001).

[Boppana, 1992] Boppana R., Halldorsson M. M.: "Approximating maximum independent sets by excluding subgraphs". *BIT*, Vol 32, N°2, p. 180-196 (1992).

[Demange, 2000] Demange M., Paradon X., Paschos V. Th. : "On-line maximum-order induced hereditary subgraph problem". *SOFSEM*, p. 327-335 (2000).

[Halldorsson, 2000a] Halldorsson M. M. : "Approximations of weighted independent set and hereditary subset problems". *Journal of graph algorithms and applications*, Vol 4, N°1, p. 1-16 (2000).

[Halldorsson, 2000b] Halldorsson M. M. : "Online coloring known graphs". *Electronic Journal of Combinatorics*, Vol 7, (2000).

[Halldorsson, 1994] Halldorsson M. M., Szegedy M. : "Lower bounds for on-line graph coloring". *Theoretical Computer Science*, Vol 130, N°1, p. 163-174 (1994).

[Halldorsson, 2002] Halldorsson M. M., Iwama K., Miyazaki S., Taketomi S. : "Online independent sets". *Theoretical Computer Science*, Vol 289, N°2, p. 953-962 (2002).

[Paradon, 2000] Paradon X. : “Algorithmique on-line”. Thèse de doctorat, Université Paris Dauphine (2000).

[Paschos, 2001] Paschos V. Th. : “on-line independent set by coloring vertices”. Proceedings of the 16th symposium on computer and information sciences (2001).

***DETECTION AUTOMATIQUE DE FRONTIERES DES PHRASES -
UN SYSTEME ADAPTATIF MULTI-LANGUES***

Anna Pappa,

Doctorante - Informatique

ap@ai.univ-paris8.fr, + 33 1 49 40 64 12

Gilles Bernard,

Professeur en Sciences de Informatique

gb@ai.univ-paris8.fr + 33 1 49 40 64 12

Hind Oukerradi,

Doctorante - Informatique

hind@ai.univ-paris8.fr +33 1 49 40 64 12

Adresse professionnelle

Laboratoire Intelligence Artificielle, Groupe CSAR, Université Paris 8 ★ 2, rue de la liberté ★ 93526
St Denis Cedex

Résumé : Cet article présente un système robuste adaptatif multi-langues, qui reconnaît les phrases dans un texte, en balisant les phrases et les signes de ponctuation. Il a été testé avec succès sur différents corpus de langues naturelles différentes (comme le français, le grec et l'arabe) avec un taux de réussite qui dépasse 99%. Le système est basé sur un algorithme de segmentation sous forme de règles. Ces règles sont issues des statistiques, elles-mêmes basées sur l'analyse distributionnelle [Harris, 1954]. Le traitement ne nécessite aucune connaissance préalable ni aucun dictionnaire. Sa portabilité et son adaptabilité le rendent précieux pour toute application ou analyse sur le langage naturel telles que MT, IE, TA, DS¹.

Summary : This article presents a robust system of phrase punctuation recognition. It is an adaptive multilingual one. Its accuracy exceeds 99% in different linguistic corpora (French, greek and Arabic). The system is a rule-based segmentation algorithm. The rules are statistics issued themselves based on distributional analysis [Harris, 1954]. No previous knowledge or dictionary are previously. System's portability as well as adaptability are precious for any further application or analysis like machine translation, information extraction, text alignment and document summarization.

Mots clés : TALN², ponctuation, règles, corpus multi-langues.

¹ (MT) Machine Translation, (IE) Information Extraction, (TA) Text Alignment, (DS) Document Summarization

² Traitement Automatique du Langage Naturel

Détection automatique de frontières des phrases - Un système adaptatif multi-langues

Cet article présente un système robuste adaptatif multi-langues, qui reconnaît les phrases dans un texte, en balisant les phrases et les signes de ponctuation. Il a été testé avec succès sur différents corpus de langues naturelles différentes (comme le français, le grec et l'arabe) avec un taux de réussite qui dépasse 99%. Le système est basé sur un algorithme de segmentation sous forme de règles. Ces règles sont issues des statistiques, elles-mêmes basées sur l'analyse distributionnelle [Harris, 1954]. Le traitement ne nécessite aucune connaissance préalable ni aucun dictionnaire. Sa portabilité et son adaptabilité le rendent précieux pour toute application ou analyse sur le langage naturel telles que MT, IE, TA, DS³.

1. INTRODUCTION

La grande majorité des applications de traitements du langage naturel présuppose le découpage de textes en phrases. Cette tâche est depuis très longtemps automatisée, on parle alors de reconnaissance de frontières des phrases⁴. La phrase est considérée comme l'unité centrale des processus du traitement du langage naturel, comme par exemple l'étiquetage. On reconnaît comme phrase la suite des mots qui se trouvent entre des signes de ponctuation dits majeurs tels que le point, le point d'exclamation, le point d'interrogation et d'autres qui précèdent ou suivent ces signes.

Formellement la ponctuation désigne l'ensemble des signes qui permettent l'interprétation des textes écrits. Traditionnellement la ponctuation est le module de la

compétence langagière le plus difficile à maîtriser, d'une part à cause de son caractère écrit, d'autre part à cause de différents styles appliqués par les auteurs dans la littérature. Les marqueurs de ponctuation jouent un rôle important pour indiquer les relations structurelles dans le discours (écrit), comme par exemple les relations rhétoriques particulières qui résident dans un texte [Dale, 1991]. Avant de procéder à la construction de systèmes de génération ou d'analyse automatique des documents écrits, il faut s'assurer que ces systèmes incorporent un modèle adéquat pour ces aspects textuels.

Des études linguistiques sérieuses sur le rôle de la ponctuation ont commencé récemment. [Nunberg, 1990] présente le début d'une théorie linguistique sur la ponctuation où les signes de ponctuation sont des indicateurs de surface pour plusieurs catégories syntaxiques.

Le problème de la détection automatique des phrases se pose à cause de l'ambiguïté de certains signes de ponctuation. Par exemple un point peut être utilisé pour déclarer la fin d'une phrase mais aussi pour exprimer une abréviation ou un acronyme (ex. : E.D.F. ou U.S.A.), ou même un nombre décimal, par exemple : 3.14 (écriture anglosaxone).

Les phrases⁵ qui suivent montrent l'ambiguïté des signes de ponctuation comme le point :

- a) Là où cela n'engage à rien, M.Filoché est pour les 37,5 annuités.
- b) Les pays membres de l'U.E. seront confrontés à une réforme concernant leur régime de retraites.
- c) Ce qui figure dans l'avant-projet de loi Fillon est considéré comme entièrement maintenu... et donc avalisé par la CFDT.

³ (MT) Machine Translation, (IE) Information Extraction, (TA) Text Alignment, (DS) Document Summarization

⁴ Dans la bibliographie anglosaxonne la phrase est sentence et les propositions (principales et secondaires) sont des clauses.

⁵ Ces exemples sont extraits du corpus traité composé des journaux quotidiens, hebdomadaires ou mensuels ainsi que des textes littéraires.

- d) Alors T., mère de deux enfants, ouvrit sa porte.
- e) « Est-ce qu'il sera là ? » demanda-t-il à son tour.
- f) (...) appuyés par des hélicoptères, qui, - une fois n'est pas coutume ! - n'ont tiré ni missiles, ni roquettes, ni balles, (...)

Cette ambiguïté varie selon le type de texte ou de corpus. Des statistiques effectuées à partir d'un corpus composé d'articles du *Wall Street Journal* montrent que plus de 47% des points utilisés sont des points qui se trouvent dans les abréviations, contrairement à 10% pour le corpus *Brown* [Church, 1991]. Ceci démontre que si l'on coupe un texte en phrases sans tenir compte des particularités de l'ambiguïté on n'aurait que 53% des phrases correctement découpées pour le premier et 90% pour le second. Nous pouvons deviner que la plus fréquente source d'ambiguïté est le point d'une abréviation., ex. : Chap. 1.

2. TRAVAUX EFFECTUES

Les travaux concernant la détection automatique des phrases représentent un premier pas pour une analyse morphologique ou syntaxique et ne font pas l'objet d'une recherche. [Amman et al. 1997] ont regroupé les approches utilisées pour la ponctuation. Les techniques utilisées (souvent des listes grammaticales et de longues listes d'abréviations) visent à reconnaître les cas les plus courants. Dans leur majorité ces techniques sont orientées vers la détection des phrases dans de corpus ou dans une langue de type donné, ce qui rend difficile de les adapter à un autre type de textes ou à une autre langue sans avoir à modifier l'algorithme. De plus, puisque la détection de phrases est juste une première étape dans le traitement automatique du langage naturel, elle ne doit pas demander trop de ressources ni de calcul au niveau exécution. Donc, le développement d'un système basé sur de grandes listes d'abréviations, ou de lexiques spécialisés avec des informations qui ne s'utiliseront plus par la suite, n'est pas la meilleure solution.

Plusieurs travaux mentionnent l'utilisation d'un détecteur de phrases, mais aucune information sur sa conception ou sur sa performance n'est donnée [Voutilainen, 1993]. L'approche la plus simple consiste à l'usage

des règles qui reconnaissent des suites de caractères (par exemple : point – espace – lettre majuscule), accompagnées de longues listes d'exceptions pour les abréviations. Une différente approche a été proposée par [Muller, 1980] : au lieu des listes d'abréviations, une analyse morphologique filtre les mots ayant les mêmes suffixes, pour lesquels la probabilité d'être une abréviation est minime ou nulle. De telles approches demandent plusieurs heures de travail pour construire et renouveler les listes de règles et d'abréviations. De plus, elles sont orientées à un type particulier de corpus.

L'approche avec le plus grand taux de détection [Riley, 1985] est basée sur un corpus annoté de 25 millions de mots. Sa performance pour le corpus *Brown* atteint 99,8%. Pour chaque mot du dictionnaire, ce modèle demande le calcul de probabilités qu'il soit le premier ou le dernier mot d'une phrase. Cette information est inutile aux analyses qui peuvent suivre morphologiques ou syntaxiques.

De plus la détection n'est qu'un moyen pour permettre à divers outils d'effectuer leur analyse des phrases découpées, donc le coût de calcul doit être minime. C'est pour cette raison que [Reynar et al. 1997] proposent un modèle basé sur la plus grande entropie qui ne nécessite aucun calcul ni le coût d'une information compliquée. L'information utilisée par ce modèle est fondée sur le segment (*token*) qui contient le signe de la ponctuation candidat pour la fin d'une phrase, ainsi que les autres *tokens*, juste avant et juste après celui-là. Pour la construction d'une liste d'abréviations extraite du corpus annoté, un algorithme simple est utilisé. L'approche de la plus grande entropie atteint 97,7% de solution pour le corpus *Brown*, avec utilisation d'une liste manuellement construite d'abréviations, d'appellations et d'acronymes. Si on enlève cette information supplémentaire, le taux est de 97,5%.

Une autre approche est le système *SATZ* [Palmer et al, 1997], qui utilise un réseau neuronal pour la désambiguïsation des frontières d'une phrase. Il est basé sur des probabilités d'appartenance d'un mot à une partie du discours principale (*prior part-of-speech*). Par exemple, le mot *portes* peut être un verbe mais il est plus probable qu'il soit un nom selon les textes utilisés. Ce système utilise un dictionnaire de 30000 entrées lexicales et

une information sur 6 différents *tokens* concernant leur contexte: 3 *tokens* avant et 3 après la fin candidate, et sa performance est de 98,5%, sur un corpus composé d'articles du *Wall Street Journal*. Le système SATZ peut être utilisé pour d'autres types de corpus ou d'autres langues naturelles après un apprentissage. Le problème demeure du fait qu'il existe des types de corpus, ou même des langues, pour lesquelles des dictionnaires spécifiques contenant des informations sur les principales parties du discours ou sur la façon de les détecter automatiquement n'existent pas. Les auteurs insistent sur le fait que la performance du système n'est pas influencée par la diminution de la taille du dictionnaire. Même si cette information est vraie, et peut être utile par la suite à un autre traitement du langage, il y a de traitements pour lesquels elle ne s'y applique pas, comme l'alignement des phrases [Santos, 2001] (*sentence alignment*).

Il faut préciser que le système SATZ ainsi que l'approche de l'entropie maximale, ne distinguent pas les différents usages des signes de ponctuation qui montrent la fin d'une phrase, qu'ils utilisent les mêmes règles partout. Or, il y a des ambiguïtés qui ne sont pas traitées. Par exemple, un point peut être utilisé pour une abréviation mais ce n'est pas le cas d'un point d'interrogation ou d'un point d'exclamation. Pour la langue grecque nous pouvons citer le travail de [Stamatatos, 2000], qui décrit un système de découpage en phrases basé sur la technique de *transformation-based learning* introduite par [Brill, 1993]. Cette théorie, dont les applications comme par exemple le *text chunking* [Ramshaw, 1995] sont largement répandues dans le domaine du TALN, consiste à extraire automatiquement de la connaissance linguistique sous forme de règles à partir des corpus annotés.

3. METHODOLOGIE

Notre méthode est fondée sur :

- des paramètres simples comme la lettre finale d'un mot ou sa longueur dont le coût de calcul est minime,
- des règles de désambiguïsation pour les signes de ponctuation. Ces règles sont extraites des informations tirées des statistiques sur des corpus non annotés et sont applicables pour les trois langues testées (français, grec, arabe).

Avant de procéder à la détection des frontières des phrases d'un texte (pour toutes les langues testées), nous définissons un ensemble de segments (*tokens*), ex. : caractères, nombres, signes de ponctuation. Pour chaque langue, ces signes sont différents, à part quelques signes comme le point final. Le point d'exclamation, qui sont communs aux trois langues.

La procédure de la segmentation suit la règle suivante : deux segments sont séparés par des espaces. Un segment qui est suivi par un des signes de ponctuation décrits ci-dessous, est considéré comme fin probable de la phrase (en respectant la forme particulière de chaque langue).

- Point .
- Point d'exclamation !
- Point d'interrogation ?
- Points de suspension ...

Une autre frontière probable est quand les signes mentionnés ci-dessus se trouvent juste avant ou après des suites de caractères doubles comme les parenthèses (), les crochets {}, les guillemets, comme dans l'exemple : « super! ». Le tableau qui suit montre le découpage effectué dans une partie du corpus.

Les tableaux contiennent des extraits des corpus (français, arabe, grec), les phrases sont découpées et balisées ; les balises <s> </s> désignent la phrase (sentence), et les balises <po> </po> la ponctuation de la fin de phrase.

1	<s>Ainsi avons-nous analysé la Théorie physique <po>.</po> </s>
2	<s>La première tache question que nous rencontrions est celle-ci<po>:</po> </s>
3	<s>quel est l'objet d'une théorie physique <po> ?</po> </s>

Tableau 1 : Extrait du corpus français:

Les corpus sont composés de textes des styles divers: journaux, magazines, littérature, etc. et ils sont non annotés. Aucune connaissance préalable ou dictionnaire ne sont utilisés pour notre algorithme.

Il faut préciser que le corpus français atteint les 100 millions des mots, les corpus grec et arabe, qui sont moins volumineux néanmoins pas moins significatifs, atteignent progressivement le million.

3.1. Particularités de langues naturelles

En effet, il n'y a pas de différences quant à l'utilisation de la ponctuation dans les langues étudiées au sein de notre recherche. Les différences se résument à la graphie et aux particularités morphologiques que nous décrivons brièvement. Les particularités morphologiques de la langue arabe ont été prises en compte - en traitant les textes codés en UTF-8, afin de finaliser l'algorithme de la détection de phrases et désambiguïser l'utilisation de signes de ponctuation.

Le mot arabe graphique (il existe aussi la notion du mot arabe phonique mais ce n'est pas le cas de notre recherche), est facile à identifier: c'est ce qui s'écrit en un seul bloc entre deux blancs [Kouloughli, 1994]. En arabe, on ne distingue pas minuscules ou majuscules, mais il existe néanmoins trois formes graphiques: initiale, médiale et finale [Siamak, 2001]. Ceci est intéressant pour la reconnaissance des acronymes car nous pouvons les identifier suivant les mêmes règles que pour les langues occidentales: "Majuscule - Point - Majuscule - Point" etc. Mais il y a encore des problèmes d'ambiguïté qui persistent, car l'unicode ne permet pas de distinction (au niveau programmation) des trois formes mentionnées plus haut. Mais dans le corpus traité nous n'avons pas trouvé d'acronymes non identifiables.

1	<s>وا نول هاج مه قيل اي ريبم اب ان نوم هتي نيدل اف </s> <po>[1] قينلا وئيس</po> </s>
2	<s>قنيدم ءاوض او ذيبنلا ل ع ج دوق ، عبط اب كيسك ملام نم عطتقا ام يسني ققار بلا راونا نالكين يمودل او يتياه يل ع قنيدل او ابوك دي يفتو مضو او غارالكين حاي ت ج او اي ببولوك نم ام نب عزت او نيبي ييل يفل <po>...</po> </s>
3	<s>قلطا سرغنوكلا لىا هوجم قلاس ريفو 1823 ماع ال يتلا قديق عل ورنوم سميج يكريم ال سي يزل مبس ل م ح ت ل ا ز ت <po>.</po> </s>

Tableau 2 : Extrait du corpus arabe.

Il faut juste mentionner que pour le grec moderne le signe de ponctuation point virgule (;) correspond au point d'interrogation et signifie donc la fin d'une phrase (voir exemple n° 2 dans le tableau 3 ci-dessous). Il faut aussi mentionner que le double point « : », étant donné qu'il peut signifier la fin d'une proposition, est marqueur de fin de phrase. Ceci est le cas pour plusieurs langues comme le français, l'italien, l'espagnol.

1	<s>Η φύση του νου σας, είναι στην πραγματικότητα καθαρή.</po>.</s>
2	<s>Η αντανάκλαση εμφανίζεται ακόμα κι αν ο ήλιος ή το φεγγάρι δεν έχουν αυτή την πρόθεση <po>:</po> </s>
3	<s>Πώς μπορούν τ' αποτυπώματα, τα οποία δεν έχουν μορφή να καθορίσουν κάτι φυσικό όπως τα σώματά μας.</po>.</s>

Tableau 3 : Extrait du corpus grec.

3.2. Les paramètres et les règles

Nous allons décrire le choix des paramètres pour la reconnaissance des fins des phrases en prenant comme modèle la langue grecque, sachant que les informations équivalentes ont été extraites des autres langues testées telles le français et l'arabe (les particularités de chaque langue faisant partie d'un module spécifique où, selon le codage, le module s'active ou non). La procédure est empirique et exploite les caractéristiques du grec moderne [Chatzivasiliou, 1995]. Les informations extraites des statistiques qui nous ont permis de fabriquer les règles de reconnaissance sont résumées ci-dessous :

- La longueur (en caractères) du dernier mot avant la fin de la phrase a peu de probabilité d'être petite (1 ou 2 caractères).
- La grande majorité des mots en grec moderne ont comme caractère final le ζ ou le ν, ou les voyelles.
- Les signes de ponctuation qui se trouvent à côté d'une possible fin de phrase sont significatifs quant à leur utilisation.

Type de signe de ponctuation	Exemples
Initial	(, [, {, «
Final),], }, »
Aucun	

Type de caractère	Exemples
français	fréquent minuscule e, s, t, n, a, etc.
	très peu fréquent minuscule w, k, j, etc.
	fréquent majuscule I, II, V, etc.
	très peu fréquent majuscule W, K, J, etc.
grec	fréquent minuscule α, ε, η, etc.
	très peu fréquent minuscule μ, κ, λ, etc.

fréquent majuscule	A, E, H, etc.
très peu fréquent majuscule	M, K, Λ, etc.
nombre	0, 1, 2, etc.
spécial	%, #, \$, etc.

Tableau 5 : Informations statistiques sur les caractères finaux en français et en grec ;

Les informations extraites des statistiques effectuées sur les corpus à partir des segments (comme l'exemple donné en figure 1) composent les paramètres suivants :

SG : la longueur (en caractères), le type du caractère, contient-il un point.

SD : la longueur (en caractères), le type du caractère, contient-il un point.

SPF : type de signe de ponctuation qui se trouve en position finale (à la fin d'un mot).

SPI : type de signe de ponctuation qui se trouve en position initiale (au début d'un mot).

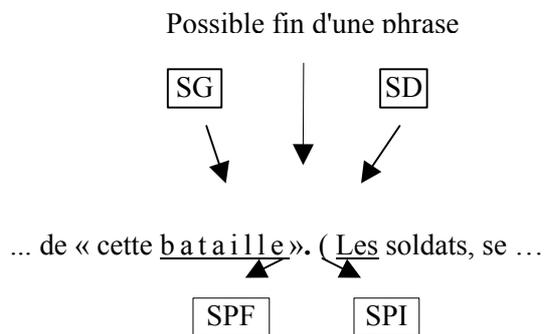


Figure 1 : Exemple des segments utilisés

Le tableau qui suit donne les informations extraites de l'exemple donné en figure 1.

SG	longueur : 8 type du premier caractère : minuscule, peu probable pour se trouver comme caractère final d'un mot type du dernier caractère : minuscule, final très probable contient-il un point : non
SD	longueur : 3 type du premier caractère : majuscule,

	probable final type du dernier caractère : minuscule probable final contient-il un point : non
SPF	type de signe de ponctuation : final - couple peut encadrer un paragraphe.
SPI	type de signe de ponctuation : initial - couple.

Tableau 6 : Explicatif des segments utilisés.

Le processus de la désambiguïsation des signes de ponctuation, ainsi que l'extraction automatique de règles que nous décrivons, est une variante de la théorie des règles de transformation. La différence consiste dans le fait que nous utilisons de corpus non annoté et que nous avons adopté le même algorithme à trois langues différentes.

Selon notre méthode, la reconnaissance des frontières d'une phrase peut se diviser en trois parties : dans la première nous effectuons une série de statistiques pour évaluer le contexte des signes de ponctuation. L'analyse distributionnelle nous permet de dégager des règles qui évaluent ensuite le contexte d'un signe particulier. Dans un premier temps nous considérons que tous les signes de ponctuation qui peuvent être signes de la fin d'une phrase indiquent une fin de phrase. La précision atteinte est la plus basse de notre système.

En deuxième partie nous appliquons des règles de type 1 :

SI contexte

ALORS éloigner signe de fin de phrase

Le contexte décrit la condition qui active la règle qui se compose soit d'une combinaison entre SG et SD soit entre SPF et SPI. L'exemple de la figure 1 devient :

SG	longueur : 8 type du premier caractère : minuscule, peu fréquent comme caractère finale d'un mot type du dernier caractère : minuscule, final très probable contient-il un point : non
ET	

SD	longueur : 3 type du premier caractère : majuscule, probable final type du dernier caractère : minuscule probable final contient-il un point : non
OU	
SPF	type de signe de ponctuation : final
ET	
SPI	type de signe de ponctuation : initial

Tableau 7 : Exemple du contexte

Une fois les règles de type 1 appliquées, le système procède à l'application des règles de type 2 (troisième partie) :

SI contexte
ALORS introduire signe de fin de phrase

Le contexte décrit la condition qui active la règle comme elle est définie plus haut. La procédure de règles peut se résumer comme le montre la figure 2 ci-dessous : tout d'abord nous appliquons les règles qui transforment un signe de fin de phrase en un simple signe de ponctuation, puis toutes les règles qui transforment un simple signe de ponctuation en un signe de fin de phrase.

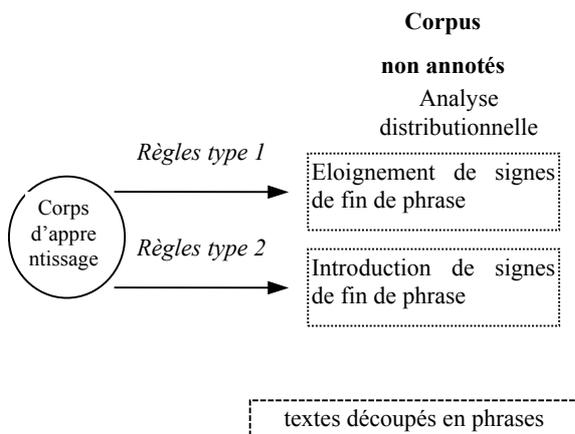


Figure 2 : Procédure de désambiguïsation

Pour chaque combinaison entre contexte et

signe possible comme fin de phrase, les informations suivantes (clauses Prolog) sont extraites du corps d'apprentissage

Le SIGNE DE PONCTUATION est un des quatre possible comme fin de phrase (point, point d'interrogation, point d'exclamation et points de suspension).

Le CONTEXTE est soit une combinaison entre SG - SD soit entre SPF - SPI.

Le C1 est un entier qui indique le nombre de fois qu'un signe de ponctuation particulier *n'est pas fin de phrase*.

Le C2 est un entier qui indique le nombre de fois qu'un signe de ponctuation particulier *est fin de phrase*.

Ensuite si l'on veut inclure une de ces clauses dans les règles de type 1 il faut respecter les contraintes suivantes :

$$C1 > C2$$

ET

$$C2 < SCFP * VERITE$$

Le *SCFP* représente la totalité de Signes Candidats pour Fin de Phrase, et *vérité* est un nombre entre 0 et 1 qui indique le degré de crédibilité des règles générées. Plus la vérité est grande, plus il y a des règles générées (moins de précision). Et plus la vérité est petite, plus les règles sont précises. Les tests effectués montrent une vérité de l'ordre de 0,01 (ou 99%) de crédibilité. Les contraintes pour les règles de type 2 sont respectivement :

$$C1 = 0$$

ET

$$C2 > 0$$

La définition des contraintes a été basée sur les statistiques. Dans un premier temps les contraintes pour les deux types de règles étaient symétriques mais l'analyse a montré que les contraintes de type 2 devaient être augmentées afin d'atteindre plus de précision.

Le système de découpage en phrases produit deux types d'erreur :

erreur positive : quand un signe de ponctuation est considéré comme fin de phrase, mais c'est faux, et

erreur négative : quand un signe de ponctuation qui est fin de phrase n'est pas reconnu par le système.

Le tableau qui suit regroupe quelques résultats pour les trois langues. La première colonne indique le choix des langues, la deuxième les 4 signes qui indiquent la fin d'une phrase (P pour le point, PI pour le point d'interrogation, PE pour point d'exclamation et PS pour points de suspension). La troisième colonne EP indique l'erreur positive, la suivante l'erreur négative (EN) et la dernière donne le taux de précision.

Langue	Signe	Phrases	EP	EN	Précision en %
français	P	1516868	3189	858	99,7
	PI	80847	1	642	99,2
	PE	41812	3	353	99,1
	PS	50009	1631	128	96,4
total		1689536	4824	1981	99,5
grec	P	56672	168	99	99,5
	PI	3028	0	29	99,1
	PE	1566	0	17	98,9
	PS	1873	63	6	96,3
total		63139	231	151	99,4
arabe	P	32721	96	59	99,5
	PI	1118	0	12	98,9
	PE	856	0	7	99,1
	PS	1055	32	2	96,7
total		35750	128	80	99,4
TOTAL		1788425	5183	2212	99,5

Tableau 8 : Résultats avec l'erreur.

4. CONCLUSION

La procédure de découpage en phrases constitue un pré-traitement pour toute analyse du langage naturel. Nous avons développé un système robuste qui reconnaît les frontières d'une phrase et qui s'adapte à différentes langues. Il a été testé avec succès sur des corpus des langues naturelles comme le français le grec et l'arabe. Nous n'utilisons aucune connaissance préalable (dictionnaire, liste d'abréviations, etc.), et les corpus ne sont pas annotés. Le taux de découpage sans faute atteint en moyenne pour les trois corpus 99,5%. La désambiguïsation des signes de ponctuation tels que le point, s'effectue grâce à un moteur de règles simples extraites des

statistiques basées sur l'analyse distributionnelle des segments différents. Les balises produites par le système au cours du traitement offrent un outil précieux pour toute analyse syntaxique postérieure.

BIBLIOGRAPHIE

- Akman V., Say B. (1997). *Current Approaches to Punctuation in Computational Linguistics*. In Computers and the Humanities. 30(6): 457-469.
- Brill E. (1993). *Automatic grammar induction and parsing free text : A transformation-based approach*. In Proceedings of the DARPA Speech and Natural Language Workshop, pages 237-242.
- Church K., Liberman M. (1991). *Rapport sur ACL/DCI*. In Proc. of the 7th Annual Conference of the UW Centre for the New OED and Text Research: Using Corpora, pages: 80 – 91.
- Dale R. (1991). *The role of Punctuation in Discourse Structure*. In Working Notes of the AAAI Fall Symposium on Discourse Structure in Natural Language Understanding and Generation, Asilomar, pages : 13-14.
- Harris Z. (1954). *Distributional Structure*, in Word, pages. 146-162.
- Kouloughli D.E. (1994). *Grammaire de l'arabe d'aujourd'hui*. Edition Pocket Collection Langues pour tous.
- Muller H., Amerl V., Natalis G. (1980). *Worterkennungsverfahren als Grundlage einer Universalmethode zur automatischen Segmentierung von Texten in Satze*. In Sprache und Datenverarbeitung, 1. Ein Verfahren zur maschinellen Satzgrenzen - bestimmung im Englischen.
- Nunberg G (1990). *The linguistics of Punctuation*. CSLI Lecture Notes 18, Stanford, CA.
- Palmer D., Hearst M (1997). *Adaptive Multilingual Sentence Boundary Disambiguation*.

Computational Linguistics, pages:
241-267(2).

Ramshaw L.A. and Marcus M.P. (1995). *Text Chunking Using Transformation-Based Learning*. In Proceedings of the ACL third Workshop on Very Large Corpora. Cambridge, pages 82-94.

Reynar J. Ratnaparkhi A (1997). *A maximum Entropy Approach to Identifying Sentence Boundaries*. In Proc. of the fifth Applied Natural Language Processing Conference.

Riley M. (1989). *Some Applications of Tree-based Modeling, to Speech and Language Indexing*. In Proc. of the DARPA Speech and Natural Language Workshop, pages: 339-352.

Santos D. (2001). *Punctuation and Multilinguality : some reflections from a language engineering perspective*.
www.oslo.sintef.no/portug/Diana/download/ponctuacao.ps

Siamak R (2001). Tokenizing an Arabic Script Language. In Arabic NLP Workshop at ACL/EACL, France.

Stamatatos E. (2000). Statistical Identification of Genre and Author in Unrestricted Modern Greek Text. Ph.D Thesis, Dept. of Electrical and Computer Engineering, University of Patras, Greece.

Voutilainen A. (1993). *NPTool, a Detector of English Noun Phrases*. In proc.of the Workshop on Very Large Corpora: Academic and Industrial Perspectives, Ohio State University, pages: 48-57.

Chatzivasiliou G. (1995). *Synoptiki Neolliniki Grammatiki kai Syntaxi*⁶. éd. GRIGORIS, Athènes, Grèce.

⁶ Traduit par : Grammaire et Syntaxe Concises du Grec Moderne.

UN POINT SUR LA RELATION ENTRE TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION ET PRODUCTIVITÉ

Marc Bourreau,
Maître de conférence en Sciences Economiques
marc.bourreau@enst.fr, +33 1 45 81 72 46

Chiraz Karamti,
Etudiante Doctorante en Sciences Economiques
karamti@enst.fr, +33 1 45 81 77 76

Adresse professionnelle
Ecole Nationale Supérieure des Télécommunications de Paris
46, rue Barrault 75634 Paris Cedex 13 France

Résumé : La mesure de l'impact des technologies de l'information et de la communication (TIC) sur la productivité a donné lieu à un débat en 2000, entre les tenants d'une vision pessimiste (Gordon, 2000) et ceux d'une vision optimiste (Oliner et Sichel, 2000 ; Jorgenson et Stiroh, 2000). Pour Gordon (2000), le regain de productivité à la fin des années 90 n'est dû qu'à un phénomène d'intensification capitaliste passager. Pour Oliner et Sichel et Jorgenson et Stiroh, au contraire, ce regain s'expliquerait par l'impact profond des TIC sur l'économie. Dans ce papier, nous faisons le point sur ce débat, en nous appuyant sur les dernières études publiées. Nous montrons que la thèse d'un impact important des TIC sur l'économie semble l'emporter aujourd'hui sur celle de Gordon. Nous avançons également qu'avec la généralisation des méthodes de prix hédoniques pour la mesure de l'évolution des prix des services, l'impact estimé des TIC sur la productivité pourrait être plus important.

Mots-clés : technologies de l'information et de la communication ; productivité ; prix hédoniques.

Abstract : The measurement of the impact of information and communication technologies (ITC) on the productivity gave place to a debate in 2000, between researchers holding of a pessimistic vision (Gordon, 2000) and those of an optimistic one (Oliner et Sichel, 2000 ; Jorgenson et Stiroh, 2000). For Gordon (2000), the renewal of productivity at the end of the Nineties is due only to one momentary phenomenon of capital intensification. For Oliner and Sichel and Jorgenson and Stiroh, on the contrary, this renewal would be explained by the major impact of the ITC on the economy. In this paper, we give a progress report on this debate, by relying on the last studies published. We show that the thesis of an important impact of the (ITC) on the economy seems to carry it today on that of Gordon. We also advance that with the generalization of the hedonic methods for the measurement of the trend of services prices, the impact estimated of the ITC on the productivity could be more important.

Key-words : communication and information technologies; productivity ; hedonic prices.

Un point sur la relation entre technologies de l'information et de la communication et productivité

1. INTRODUCTION

Dans les années 1995-2000, la croissance de la productivité aux Etats-Unis s'est accélérée de façon marquée, atteignant presque le double du taux de croissance annuel modeste obtenu pendant les années de stagnation, soit 1,4% de 1973 à 1995 (Triplett, 2001). Dans le même temps, les Etats-Unis ont connu un taux de chômage et un taux d'inflation très faibles.¹ Différents auteurs ont cherché à comprendre les raisons de cette accélération de la croissance de la productivité. Deux thèses se sont affrontées. Pour la plupart des auteurs, et en particulier pour Oliner et Sichel (2000) et Jorgenson et Stiroh (2000), les technologies de l'information et de la communication (TIC) étaient responsables de ce sursaut de la productivité, même si leurs mesures de l'impact des TIC sur la productivité différaient. Pour Gordon (2000), au contraire, ce rebond de la productivité s'expliquait essentiellement par des effets cycliques.

Alors que leurs travaux cités ci-dessus s'appuyaient sur les données de la période 1995-1999, ces auteurs ont pu récemment intégrer l'année 2000 dans leurs mesures. L'actualisation en 2002 de leurs travaux permet de faire le point sur le débat qui les a opposés surtout après l'éclatement de la bulle financière et la crise de mars 2000. L'objet de cette contribution est de discuter, vu la variété des points de vue et la diversité des outils de vérification, les résultats contrastés de ces travaux pour évaluer l'impact probable des TIC sur la productivité.

Nous montrons que les dernières études publiées renforcent la thèse qui soutient que les TIC ont un impact déterminant sur la croissance. En particulier, dans ses derniers travaux, Gordon semble se ranger au moins partiellement à cet avis. Si les TIC ont un impact important sur la croissance, nous pensons que celui-ci est vraisemblablement encore sous-évalué, dans la mesure où les indices de prix utilisés pour certains biens ou services TIC ne sont pas encore satisfaisants.

Le reste du papier est structuré de la façon suivante. Dans un premier temps, nous analysons le regain de croissance de la productivité aux Etats-Unis dans les années 1995-2000, en nous appuyant sur l'actualisation des travaux de Gordon, Oliner et Sichel et Jorgenson et

Stiroh. Dans un second temps, nous discutons de l'importance des indices de prix pour une meilleure mesure de la croissance de la productivité. Enfin, dans un troisième temps, nous apportons quelques éléments de conclusion.

2. TIC ET PRODUCTIVITÉ

L'augmentation des investissements dans les TIC dans les années 90 s'est accompagnée d'une augmentation de la croissance de la productivité. Ceci a poussé les chercheurs à analyser l'effet des TIC sur l'accélération de la productivité aux Etats-Unis à partir de 1995. Avant de discuter si l'émergence du secteur des TIC peut accroître durablement le rythme de la croissance de la productivité, nous commençons par définir le concept de productivité. Nous identifions alors les canaux par lesquels les TIC peuvent influencer la productivité.

2.1. Définition et mesure de la productivité

Généralement, l'étude de la dynamique de croissance passe d'abord par celle de la productivité. En effet, la croissance de la productivité permet d'abaisser les coûts de production, et d'augmenter le salaire réel grâce à une économie de partage². De ce fait, la vigueur de la croissance permet le retour au quasi-plein emploi, d'autant plus que le redressement marqué de la productivité, s'il s'avère permanent, permet de modérer considérablement l'inflation.

Deux mesures de la productivité sont employées. La productivité moyenne du travail ou PMT (Average Labor Productivity - ALP) mesure la croissance de l'output par rapport à l'input travail. L'idée est que le facteur travail est le moteur du progrès technique et donc, que les variations de la productivité des facteurs doivent être mesurées en termes de travail incorporé. Cependant, le processus de production emploie de moins en moins de main d'œuvre et substitue au facteur travail d'autres formes de capital. Le concept de productivité totale des facteurs permet de prendre en compte la contribution des autres facteurs.

La productivité totale des facteurs ou PTF (Total Factor Productivity -TFP- ou multifactor productivity) appelée parfois résidu de Solow ou progrès technique non incorporé (dans les facteurs de production) mesure la croissance de l'output par

¹ En 1999-2000, le taux de chômage aux Etats-Unis est tombé à 4%, sans que cela s'accompagne d'une accélération de l'inflation.

² Il s'agit d'une indexation d'une fraction croissante du revenu des salariés sur des indicateurs de performance financière de la firme.

rapport à une moyenne pondérée de plusieurs inputs (en particulier, travail, capital, parfois d'autres – comme matériels, énergie, importations). Cette mesure peut être restreinte aux seuls facteurs physiques de production (travail, capital et inputs intermédiaires) ou bien être étendue à d'autres facteurs (les effets d'échelle, la qualité du travail, les progrès d'organisation, les facteurs cycliques, les erreurs de mesure, etc.).

Afin de mieux comprendre la contribution des différents facteurs à la croissance, nous introduisons la forme néo-classique de la production. Soit Y_t l'output agrégé produit à partir des inputs capital K_t et travail L_t . Nous ajoutons un autre facteur de production, le niveau de technologie, A_t . Dans cette modélisation, le progrès technologique est exogène (Solow, 1956). En considérant que le progrès technique est neutre au sens de Hicks, on peut écrire :

$$Y_t = A_t \cdot X(K_t, L_t) \quad (1)$$

Sous l'hypothèse de rendements d'échelle constants, la croissance de l'output est égale à la somme pondérée de la croissance de chacun des inputs, auquel s'ajoute un résidu :

$$\Delta \log Y_t = \alpha \Delta \log K_t + \beta \Delta \log L_t + \Delta \log A_t \quad (2)$$

où α est la part du facteur capital, β la part du facteur travail, $\alpha + \beta = 1$, et Δ représente une différence.

L'équation (2) nous permet d'identifier les contributions de chacun des inputs à la croissance économique. Pour présenter les résultats en termes de croissance de la productivité moyenne du travail, nous introduisons les notations suivantes. En posant : H_t les heures travaillées, $y_t = Y_t / H_t$ représente la productivité moyenne du travail et $k_t = K_t / H_t$ le capital par heure travaillée. L'équation (2) devient :

$$\Delta \log y_t = \alpha \Delta \log k_t + \beta (\Delta \log L_t - \Delta \log H_t) + \Delta \log A_t \quad (3)$$

Cette fonction de production s'apparente à celle proposée par Gordon (2000). La croissance de la productivité moyenne du travail ($\Delta \log y_t$) s'explique par trois facteurs :

1. Le premier terme à droite ($\Delta \log k_t$) représente l'accumulation du capital ou intensification capitaliste (capital deepening) ; il s'agit de la croissance du capital par heure travaillée. Du fait de l'accumulation du capital, les travailleurs deviennent plus productifs, parce qu'ils utilisent plus de capital pour chaque heure travaillée. Par conséquent, une

augmentation du capital disponible par travailleur accroît la productivité de travail en proportion de la part α du capital.

2. Le deuxième terme ($\Delta \log L_t - \Delta \log H_t$) représente l'amélioration de la qualité du travail, c'est-à-dire la différence entre la croissance de l'input travail et la croissance des heures effectivement travaillées. L'accroissement de la qualité du travail reflète l'augmentation du poids relatif des travailleurs qui produisent des biens et services à forte marge. Ce terme peut être positif si, par exemple, la qualification de la main d'œuvre s'améliore. L'amélioration de la qualité du travail augmente la productivité du travail en proportion de la part β du travail dans l'équation (3).
3. Enfin, la variation de la productivité moyenne du travail qui n'est pas expliquée par les deux premiers facteurs est attribuée à la productivité totale des facteurs incorporée dans la variable A. La contribution de la PTF à la croissance de la productivité s'obtient donc de façon résiduelle, en retranchant à la croissance de la productivité les contributions des facteurs travail et capital. Normalement, une hausse de la PTF permet d'augmenter la production sans accroître la quantité des facteurs. Il s'agit simplement de déduire la croissance expliquée de la croissance observée.

2.2. L'accélération de la croissance de la productivité dans les années 1995-2000

Gordon (2000, 2002) identifie trois périodes dans l'histoire économique récente des Etats-Unis. Dans la période qu'il qualifie d'âge d'or (golden age) et qui s'étend de 1950 à 1972, la croissance annuelle de la productivité était de 2,66% par an. Dans la période de ralentissement qui a suivi (dismal slowdown), de 1972 à 1995, la croissance de la productivité est tombée à 1,42% par an. Enfin, les années 1995 à 2000 correspondent à un renouveau de l'économie américaine, avec une croissance annuelle de la productivité de 2,86% par an.

Les études macroéconomiques de Oliner-Sichel, de Jorgenson-Stiroh et de Gordon s'accordent au moins sur deux points quant au mécanisme qui aurait conduit à la nette contribution des TIC à l'accélération de la croissance de la productivité dans les années 1995-2000 (aux États-Unis 1,1 point de pourcentage par année contre 0,6 point de pourcentage par année durant la période 1991-1995). Une accélération du progrès technique dans les microprocesseurs, et donc une augmentation massive de la puissance de calcul, a engendré une forte chute des prix des produits TIC (à performance constante),

notamment des ordinateurs³. On a pu distinguer deux effets sur la croissance.

Premièrement, la prise en compte de l'amélioration des performances des produits des TIC dans le secteur producteur, se traduit généralement par une baisse des prix et une augmentation du capital, appelé aussi "capital deepening" ou "progrès technique incorporé". Si les prix n'intègrent pas l'effet qualité, ceci se traduit par une augmentation de la PTF (ou aussi le "progrès technique non incorporé").

Les producteurs de TIC deviennent, dès lors, plus efficaces grâce à l'amélioration des performances des machines au sein du processus de production et produisent beaucoup plus de matériels informatiques et de logiciels à partir des mêmes inputs. La productivité des industries productrices des TIC augmente et elle contribue sensiblement à la croissance de la productivité totale des facteurs en global. De ce fait, la productivité du travail s'en trouve pareillement accrue dans l'industrie et au niveau de l'économie dans son ensemble (Jorgenson, 2002).

Deuxièmement, la loi de Moore s'exporte vers les autres secteurs par le biais d'une baisse continue des prix des biens informatiques. Les entreprises auraient ainsi augmenté leurs volumes d'investissement en TIC. Les TIC peuvent à ce titre influencer la **productivité** des secteurs utilisateurs est la croissance de leur PTF de plusieurs façons. D'abord, les secteurs intensifs en TIC sont devenus plus efficaces parce que le coût d'acquisition et de gestion de l'information a beaucoup baissé, permettant aux firmes d'améliorer l'allocation de leurs ressources. Ensuite, l'équipement de main d'œuvre en TIC s'améliore, ce qui permet de stimuler la croissance de la capacité de production des industries utilisatrices des TIC⁴. Les TIC constituent donc une composante du stock du capital et influencent sensiblement la croissance économique en tant que facteur de production. De ce fait, l'accélération de la croissance de productivité aux Etats-Unis est largement imputable à la croissance importante des investissements dans les TIC (OCDE, 2001). Cet éventuel effet, prêté toujours à controverse.

Pour arriver à ce résultat, les chercheurs ont séparé la croissance des prestations du capital TIC de celle relative à d'autres prestations du capital. On a pu alors constater que la contribution de tout autre élément constitutif du stock de capital -autre que les TIC- à la croissance, bien que positive, est restée constante avant et après 1995 ; la croissance du capital TIC des entreprises aurait alors eu des effets directs et peut-être indirects sur leur productivité.

³ La contribution importante des ordinateurs montre que la loi de Moore est un moteur essentiel de la croissance de la PTF.

⁴ Voir "Digital Economy 2000". Le tableau 3.1, en p. 23, établit la liste des industries productrices des TIC.

2.2.1. L'accélération du progrès technique

L'accélération du changement technique dans les ordinateurs, les périphériques et les semi-conducteurs, expliquerait en grande partie l'accélération de la croissance de la productivité depuis 1995. Selon les mesures du Bureau of Economic Analysis (BEA), la baisse de l'indice des prix des équipements informatiques (ordinateurs et périphériques) s'est accélérée à partir de 1995⁵. Alors que cet indice a baissé de 12% par an en moyenne entre 1987 et 1995, la baisse de l'indice des prix a atteint -29%/an entre 1996 et 1998. L'accélération de la baisse des prix des matériels informatiques serait due, essentiellement, à une accélération de la baisse des prix des semi-conducteurs qui a atteint plus de 40% par an depuis plus de trois décennies (Jorgenson, 2001, 2002).

La chute de l'indice des prix des équipements informatiques a conduit les entreprises à augmenter leurs investissements dans ce type d'équipements et de substituer des facteurs de production en faveur du capital informatique. Mairesse, Cette et Kocoglu (2000) indiquent que dans le cas de la France, le taux de croissance annuel moyen en volume de l'investissement en matériel informatique est de 30% pour les trente dernières années. Par comparaison, le taux de croissance annuel moyen des matériels non informatiques est de 3,6%.

La question est alors de savoir quel impact ces investissements ont pu avoir sur la croissance de la productivité.

2.2.2. Effets direct et indirect des TIC

La littérature économique identifie deux types d'effets des TIC sur la productivité : des effets directs et des effets indirects.

Pendant la période 1995-2000, les TIC ont eu un effet direct sur la productivité. Du fait de la chute des prix des équipements informatiques, dans la plupart des secteurs, les firmes ont investi massivement, ce qui a produit un effet d'intensification capitaliste (capital deepening), c'est-à-dire une croissance du capital par heure travaillée. L'intensification capitaliste augmente la productivité car une augmentation du capital des entreprises rend les travailleurs plus productifs en proportion de leur part de capital.

A ce premier effet direct s'ajoute un second effet direct, dans le secteur producteur des TIC : la baisse des prix des équipements informatiques et la croissance de la demande pour ces équipements ont

⁵ Les méthodes utilisées par le BEA pour déflater les prix des équipements informatiques sont des méthodes hédoniques.

généralisé une augmentation de la production dans le secteur des TIC.

Si les investissements en TIC ont un taux de retour plus élevé que les autres types d'investissement, ils peuvent aussi engendrer des effets de spillover ou effets indirects, c'est-à-dire un progrès technique autonome. Pour Gordon (2002), pour prouver que ces effets indirects existent, il faut montrer que l'accélération de la croissance de la PTF s'observe en dehors des secteurs informatiques (c'est-à-dire dans le reste de l'économie).

Si l'existence d'effets directs n'est presque jamais mise en cause, l'existence d'effets indirects a fait débat. Dans la partie suivante, nous présentons différentes mesures de ces effets directs et indirects.

2.3. La mesure de l'impact des TIC sur la productivité

Afin d'analyser l'impact des TIC sur la productivité, nous redressons la croissance de la productivité du travail de différents facteurs en nous basant sur les travaux de Gordon (2002), d'une part, et d'Oliner et Sichel (2002) et Jorgenson et Stiroh (2002), d'autre part.

Gordon (2000), comme d'autres économistes, n'est pas convaincu que les TIC ont un impact structurel sur la productivité moyenne du travail. Pour soutenir sa thèse, il décompose la PMT de façon à neutraliser dans la croissance de la productivité les effets cycliques, les effets de mesure et les effets d'amélioration de la qualité de la main d'œuvre. Il montre que le résidu, qui représente une mesure de l'accélération structurelle de la PMT, est négligeable.

2.3.1. Les effets cycliques dans la croissance de la productivité

Gordon plaide que la productivité du travail varie de façon pro-cyclique. Son argumentation est que le facteur travail est un facteur "quasi-fixe" qui ne s'ajuste que partiellement lors des oscillations cycliques de l'output autour de la tendance. Par conséquent, une croissance de l'output au-dessus de la tendance s'accompagne d'une augmentation de la productivité ; cependant, cette augmentation n'est que temporaire. Sur une croissance de la PMT de 2,86% pour le secteur privé non agricole (nonfarm private business), Gordon (2002) estime que 0,40% peuvent être attribués à cet effet cyclique. La croissance tendancielle de la productivité du travail serait donc de 2,46%, c'est-à-dire toujours fortement supérieure à la croissance tendancielle de la PMT entre 1972 et 1995 (1,42%).

Cependant, les effets cycliques ne peuvent expliquer que la croissance soutenue de la productivité aux Etats-Unis se soit maintenue sur une période aussi longue.

D'ordinaire, ce processus -effet cyclique- rend compte de l'accélération de la productivité seulement lors des périodes de reprise (d'Arvisenet, 2001). Or, sur la période 1991-2001, la croissance de la productivité reste très forte et elle est caractérisée par un caractère "a-cyclique", à l'encontre de ce qu'affirme Gordon. Lors du cycle des années 1980, les gains de productivité ont suivi un schéma traditionnel : les gains de productivité se concentrent sur les premières années du cycle et diminuent au-delà. La période 1991-2001 n'obéit pas à cette logique. On n'observe pas de tendance au ralentissement, mais plutôt un mouvement irrégulier (cf. figure 1 ci-dessous).

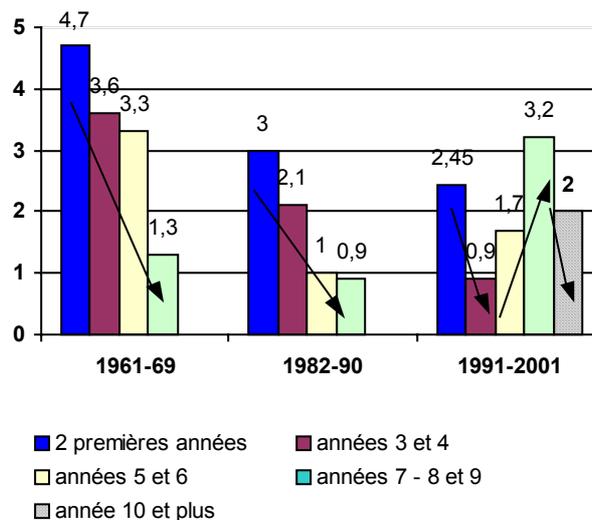


Figure 1 : Croissance de l'output par heure dans le secteur non-agricole durant les périodes d'expansion (croissance annuelle moyenne en %).

Source : Calcul des auteurs, d'après les données du US Bureau of Labor Statistics (BLS)

Des récentes analyses indiquent que les effets structurels observés à la fin des années 90 se maintiennent, malgré le repli économique depuis 2000⁶ comme le montre la figure 1 (McGuckin et Van Ark, 2003).

2.3.2. Les effets de mesure et les effets de qualité de la main d'œuvre

Une partie de l'accélération de la croissance entre 1995 et 2000 par rapport à la période 1972-1995 (+1,04%) s'explique selon Gordon par des biais sur la mesure des prix (0,14%) et sur la composition de la main d'œuvre (0,01%).

Les biais sur la mesure des prix sont dus à l'amélioration des méthodes de mesure de l'indice de prix à la consommation (IPC) entreprises depuis 1995

⁶ La croissance de la productivité aux Etats-Unis a chuté à 0,4%, mais tout de suite suivie d'une forte reprise 2,8% en 2002.

dans le NIPA (National Income and Product Accounts), notamment, l'utilisation des "méthodes hédoniques"; il est nécessaire alors de procéder d'abord à des corrections pour que les données d'avant 1995 soient comparables à celles d'après 1995. Le second point est lié à l'amélioration de la qualité de la main d'œuvre, c'est-à-dire de l'input travail par heure travaillée (une main d'œuvre composée de travailleurs plus qualifiés aura une meilleure productivité).

2.3.3. L'intensification capitaliste et la croissance de la PTF

Gordon (2000) montrait que l'impact des équipements informatiques (plus généralement, des TIC) sur la croissance était important dans les secteurs des TIC mais quasiment nul en dehors des secteurs de biens durables. En effet, Gordon calculait que l'accélération structurelle de la productivité du travail sur la période 1995-1999 (+0,64%) se décomposait en 0,33% d'intensification capitaliste, 0,29% de croissance de la PTF dans les secteurs TIC et de 0,02% seulement de croissance de la PTF dans les autres secteurs. Il en concluait que l'accélération de la croissance de la productivité était due non pas à des effets structurels mais plutôt à des effets transitoires.

Gordon (2002) revient sur cette première analyse. Dans ce travail plus récent qui inclut l'année 2000 dans les mesures, Gordon conclut que le sursaut de la productivité observé est réel et important et qu'une grande partie s'explique par des effets structurels plutôt que cycliques. On l'observe dans la production d'ordinateurs, mais aussi dans l'usage des ordinateurs. En effet, d'après ses calculs, l'accélération structurelle de la productivité du travail de 0,89% se décompose en un effet d'intensification capitaliste pour 0,37%, une croissance de la PTF dans la production d'équipements informatiques de 0,30% et une accélération structurelle de la PTF de 0,22%.

Cependant, pour Gordon, il n'y a toujours pas trace d'une accélération de la croissance de la PTF en dehors de la production d'équipements informatiques et du reste des secteurs manufacturiers de biens durables. Il estime que l'accélération structurelle de la PTF dans le secteur privé hors production de biens durables n'est que de +0,07%⁷. La conclusion de Gordon (2002) est que, du fait de l'accélération de la chute des prix des équipements informatiques, l'économie américaine a vécu une explosion de l'investissement dans ces équipements. Ceci a produit un effet d'intensification capitaliste mais le capital informatique a le même taux de retour sur investissement que les autres types de capital. Le tableau ci-dessous (tableau 1) résume l'évolution des mesures de Gordon.

Etude	Gordon (2000) 1995:4 – 1999:4		Gordon (2002) 1995:4 – 2000:4	
	NFPB (1)	NFPB moins biens durables	NFPB	NFPB moins biens durables
Accélération structurelle de la productivité	0,64	0,04	0,89	0,44
Contribution intensification capitaliste	0,33	0,33	0,37 (dont TIC 0,60)	0,37 (dont TIC 0,60)
Contribution de la croissance de la PTF dans les TIC	0,29	-	0,30	-
Accélération structurelle de la PTF	0,02	-0,29	0,22	0,07

(1) Non farm private business (secteur privé non agricole).

Tableau 1 : Comparaison des mesures de la productivité dans Gordon (2000) et Gordon (2002).

Etude	Gordon (2002)	Gordon (2002)	Jorgenson et al. (2002)	Oliner et Sichel (2002)
Période	1995:4 – 2000:4	Période	1995-2000	1996-2001
Accélération par rapport à	1972:2 – 1995:4	1972:2 – 1995:4	1973-1995	1991-1995
Secteurs inclus	NFPB (1)	NFPB - durables	(2)	NFPB
Accélération structurelle de la productivité	0,89	0,44	0,92	0,89
Contribution intensification capitaliste	0,37 (dont 0,60 TIC ³)	0,37 (dont 0,60 TIC ³)	0,52 (dont 0,44 TIC ³)	0,67 (dont 0,56 TIC ³)
Contribution de la croissance de la PTF dans les TIC	0,30	-	0,27	0,35
Accélération structurelle de la PTF	0,22	0,07	0,20	0,06

(1) Non farm private business (secteur privé non agricole) mesuré comme point de % par an.

(2) La définition de l'output est plus large que celle de Gordon ; Jorgenson et al. (2002) incluent le secteur à but non lucratif (nonprofit) et prennent en compte le logement résidentiel dans le capital.

(3) Pour Jorgenson et Stiroh, les TIC recouvrent les ordinateurs, les logiciels et les équipements de communications. Pour Oliner et Sichel, elles recouvrent les ordinateurs et les semi-conducteurs intégrés aux ordinateurs.

Tableau 2 : Mesures récentes de l'impact des TIC sur la productivité

⁷ Remarquons cependant que Gordon (2000) trouvait un ralentissement de la croissance de la PTF (-0,29%) dans les secteurs de biens non durables.

Les mesures récentes faites par Jorgenson et al. (2002) et Oliner et Sichel (2002) sont comparables aux résultats de Gordon (2002), comme le montre le tableau 2 ci-dessus. Même s'il admet l'existence d'une accélération structurelle de la productivité, Gordon (2002) estime que l'impact des TIC dans le reste de l'économie reste faible. Au contraire, l'évaluation de Jorgenson et al., (2002) montre que l'élévation de la PTF est aussi importante dans les secteurs producteurs des TIC que dans les secteurs utilisateurs entre les deux moitiés des années quatre-vingt-dix. Ce résultat suggère que le progrès technique dans les TIC se diffuse à l'ensemble de l'économie.

Ces différents travaux montrent que l'accélération structurelle de la productivité sur la période 1995-2000 est très importante (environ +0,9 point de pourcentage). Elle s'explique pour moitié par un effet d'intensification capitaliste concentré dans les équipements TIC (effet direct). L'autre moitié de l'accélération de la productivité est due à une accélération structurelle de la PTF (effet indirect). Les travaux récents suggèrent que les TIC ont à la fois un effet direct et un effet indirect responsables des trois quarts de l'accélération de la croissance de la productivité du travail dans la deuxième moitié des années quatre-vingt-dix aux Etats-Unis (Duval, 2000).

Une analyse sectorielle de ces évolutions⁸ nous fournit un autre enseignement essentiel : plus une industrie est intensive en capital TIC, plus la croissance de la productivité dans cette industrie est forte. D'après les estimations de l'Economic and Statistics Administration (ESA) basées sur les données du Bureau of Economic Analysis (BEA), la croissance de la productivité dans les industries les plus intensives en capital TIC (téléphonie, services de transport, assurances, équipements électriques et électroniques...) est de 2,95% par an sur la période 1989-2000, alors qu'elle n'est que de 0,58% dans les industries moins intensives en capital TIC (santé, textile, hôtellerie, construction...). Une grande étude sectorielle effectuée par *McKinsey Global Institute* (2001) a identifié six gros secteurs responsables de l'accélération de la productivité américaine après 1995 où l'utilisation effective des TIC a été importante à savoir : le commerce de gros, le commerce de détail, la fabrication d'équipements informatiques, la fabrication de semi-conducteurs, le courtage de valeurs immobilières et les communications.

La figure 2 montre également que, contrairement à la thèse défendue par Gordon (2000), la croissance de la productivité n'est pas circonscrite aux secteurs producteurs de TIC. Cette figure présente la contribution des industries du secteur non agricole, y compris les industries productrices de TIC, à l'accélération de la croissance de la productivité entre la période 1995-2000 (2,48%) et la période 1989-1995

⁸ Deux études sectorielles majeures ont été effectuées, celle de Stiroh (2001a) et celle de McKinsey Global Institute (2001).

(1,02%) à savoir 1,46% d'accélération⁹. La contribution à la croissance au PIB la plus importante est celle du secteur des assurances et de la finance (37%) suivi du commerce de détail (28,3%), du commerce de gros (20,8%) et des services (15,8%). La Figure 2 représente la contribution de ces différents secteurs à la croissance de productivité.

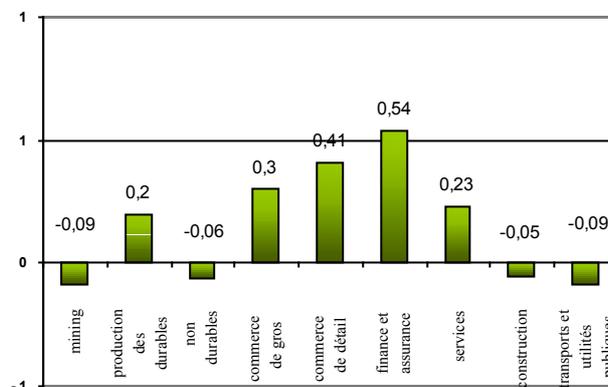


Figure 2 : Contribution à l'accélération de la productivité du secteur non-agricole des différentes industries pour la période 1995-2000 par rapport à la période 1989-1995 (en points de pourcentage).

Source : d'après les estimations de l'ESA (Digital Economy, 2002)

Le secteur des biens durables doit évidemment sa performance à l'industrie informatique. Pour le commerce, la co-évolution de plusieurs facteurs a permis l'essor de ce secteur : l'utilisation de la **Toile**, en premier lieu pour en faire la vitrine de l'entreprise et lui donner une image de modernité, ensuite avec les différentes fonctions (achats, comptabilité, etc.) et outils comme les échanges de données informatisés (EDI) ou les réseaux Intranet, les entreprises, à partir d'une base de données unique, ont accru leurs échanges informatisés avec leurs partenaires extérieurs (avec les sous-traitants, fournisseurs via les logiciels B2B, avec les clients via la publicité et la vente électronique, et avec les organismes et administrations publiques grâce au développement de l'administration électronique) ; la standardisation des codes des produits, la généralisation des lecteurs des codes-barres, la diffusion du juste-à-temps, etc., sont tous des facteurs qui ont permis le développement de ce secteur.

Les TIC ont joué un rôle primordial dans l'évolution du secteur financier (Boyer, 2002). Pour les services, qui utilisent plus de 70% des équipements en TIC (Baily et Lawrence, 2001, p.311), les résultats restent modestes. Ceci s'explique au moins en partie par des problèmes de mesure. Ce secteur s'inscrit d'ailleurs parmi un certain nombre de branches du tertiaire où

⁹ La contribution du secteur des secteurs de biens durables à la croissance de productivité était de 0,56 points de pourcentage sur la période 1989-1995 et de 0,83 points de pourcentage sur la période 1995-2000.

la mesure de la production s'avère être particulièrement délicate¹⁰.

Toutefois, l'adoption des TIC ne suffit pas pour que la productivité des entreprises s'améliore. Une réorganisation des circuits d'informations et de décision des entreprises est nécessaire. En effet, mis à part le secteur producteur des infrastructures matérielles des réseaux, des ordinateurs et des logiciels, c'est la gestion des entreprises que les TIC affectent d'abord et non pas directement la production ou la distribution. Ce n'est qu'ensuite que s'expliquent les mécanismes qui ont procréé l'accroissement de la productivité, notamment aux Etats-Unis (Boyer, 2002).

En effet, l'informatisation des processus de production et de gestion entraîne des effets d'apprentissage et des coûts d'adaptation des organisations au nouveau mode de communication qui peuvent relativement freiner l'accélération de productivité. Mais ces coûts disparaissent sous l'effet de diffusion des TIC qui implique des économies d'échelle due à ces processus de réorganisation des systèmes de production.

Les pays nordiques, notamment, la Finlande et l'Irlande donnent le meilleur exemple de petites économies ouvertes, qui ont pu enregistrer des gains de productivité nettement plus élevés (respectivement 4% et 3,1% à la deuxième moitié des années 1990) par rapport aux autres pays de l'OCDE, essentiellement grâce à leur maîtrise de l'usage et/ou de la production des TIC. Ainsi, il s'avère nécessaire de noter que, les transformations intervenues en matière de management et de concurrence¹¹ sont aussi des facteurs clés qui expliquent l'évolution de la productivité plutôt que les investissements en TIC à eux seuls (Triplett, 2001 ; Boyer, 2002).

Enfin, les TIC permettent incontestablement l'accélération de la croissance de la productivité dans les autres secteurs de l'économie. Prenons l'exemple de la biotechnologie, notamment le problème du décodage du génome humain résolu uniquement grâce à la puissance de calcul des ordinateurs ouvrant ainsi la voie à de nombreuses découvertes. De ce fait, nous pouvons affirmer que les TIC représentent réellement des biens génériques différents des innovations passées et dont l'apport à l'accélération de la productivité ne peut être restreint au seul secteur producteur.

¹⁰ McGuckin et Stiroh (2001a) estiment l'erreur dans la mesure du produit américain d'un demi pourcent de la croissance économique.

¹¹ Le marché extrêmement compétitif et la déréglementation économique et financière ont facilité l'apparition de nouveaux secteurs d'activité ainsi que le financement des jeunes pousses. Ceci est plus visible aux Etats-Unis qu'en Europe, ce qui constitue l'une des raisons qui explique l'écart de performance entre les deux régions. Une étude de l'OCDE a montré l'existence d'une liaison significative, notamment un impact négatif, entre le poids de la réglementation et les performances en matière d'accélération de la productivité totale des facteurs (OCDE, 2000).

3. L'IMPACT DES TIC SUR LA CROISSANCE ÉCONOMIQUE EST-IL TOUJOURS SOUS-ESTIMÉ ?

Pour mieux cerner la contribution des TIC à la croissance de la productivité, il est nécessaire de bien évaluer les investissements TIC réalisés par les entreprises. Ceci n'est possible que par application d'un système de déflateurs spécifique aux équipements TIC permettant de prendre en compte les progrès technologiques fréquents et très rapides de ces produits. Le problème posé est celui du partage prix/volume des dépenses d'investissement et de la prise en compte de l'effet qualité. Pour réaliser ce partage, la meilleure méthode étant celle des indices de prix hédoniques considérée comme la nouveauté la plus importante des années 80 en matière de comptabilité nationale à l'échelle mondiale. Nous commençons par en rappeler le principe et nous analysons l'impact de cette méthode sur la mesure de la productivité.

L'utilisation des indices de prix hédoniques et son impact sur la mesure des dépenses réelles d'informatique, est au cœur du débat sur le processus de diffusion des TIC : dans quelle mesure ces indices de prix sont-ils capables d'expliquer l'accélération de la croissance de la productivité ainsi que de la modération de l'inflation depuis la deuxième moitié des années quatre-vingt-dix ?

Les indices de prix hédoniques ne sont pas encore généralisés pour la mesure de l'évolution des prix. Nous suggérons que leur prise en compte pour mesurer l'évolution des prix de certains services – comme la téléphonie mobile – pourrait conduire à réévaluer la mesure de l'impact des TIC sur la productivité.

3.1. Définition et rôle des indices de prix hédoniques

Le principe de la méthode des prix hédoniques est simple : il s'agit d'expliquer les fluctuations du prix d'un produit ou d'un service en les reliant aux variations des caractéristiques de ce produit ou service¹². Dans le cas des ordinateurs pour lequel la méthode hédonique a été utilisée avec succès¹³, le prix des ordinateurs est relié à différentes caractéristiques quantifiables comme la mémoire, la vitesse, la taille du disque dur, etc., ainsi qu'à des variables qualitatives comme le modèle, la marque, la présence d'un lecteur de disque compact, etc. On procède à des régressions successives du prix du produit i à la période t (P_{ti}) par rapport aux

¹² Afin de mieux comprendre les ajustements qualitatifs par les méthodes hédoniques, voir Berndt (1991).

¹³ Voir Berndt, Dulberger et Rappaport (2000).

caractéristiques du bien notées z_{tki} , de la façon suivante :

$$\ln p_{it} = \beta_0 + \sum_{t=2}^T \delta_t D_t + \sum_{k=1}^K \beta_k z_{tki} + \varepsilon_{it}$$

où D_t est une variable temporelle (dummy), ε_{it} est un terme d'erreur et les β_k et les δ_t sont les coefficients à estimer.

La méthode hédonique permet de produire des séries chronologiques corrigées des variables qualitatives ou des séries d'indices de prix dans lesquelles les corrections qualitatives sont prises en compte de façon implicite. Par exemple, si on considère que le prix d'un ordinateur est à peu près constant dans le temps, mais que la performance des ordinateurs augmente fortement chaque année, la méthode des prix hédoniques capture l'effet qualité en faisant apparaître une baisse de l'indice.

Mal saisir cette baisse des prix, déclenchée essentiellement par une accélération du progrès technique et donc de la qualité des produits, revient à mal estimer la PTF. Comme le note Shreyer (2001), la prise en compte des changements de qualité dans la mesure des prix des produits TIC est essentielle pour bien mesurer la productivité. L'importance des TIC comme facteur de croissance de la productivité aux Etats-Unis résulte justement de l'application dans ce pays des indices de prix hédoniques. Cette méthode est utilisée également dans les comptes nationaux par l'agence statistique canadienne (Statistics of Canada).

Cette baisse des prix du matériel informatique a considérablement augmenté la demande des TIC. En volume, les firmes investissent relativement plus, tout en y consacrant relativement moins d'argent. Il en résulte une augmentation de la productivité du travail. Néanmoins, cet accroissement de productivité n'est permanent que si les prix des TIC poursuivent leur baisse conditionnée elle-même par l'apparition de nouvelles innovations.

La question cruciale qui se pose dès lors aujourd'hui est de savoir si les prix des équipements TIC ajustés sur la qualité vont continuer à baisser à la même vitesse que pendant la deuxième moitié des années 90. D'un point de vue purement technologique, il semble que dans les cinq à sept prochaines années, le rythme des améliorations technologiques dans les circuits intégrés, les microprocesseurs et les circuits de mémoire ne va pas être moins rapide que ce qu'il a été dans les trente dernières années. Cependant, ceci ne sera vrai que si l'offre de technologie rencontre une demande, en développant de nouveaux usages.

Une étude récente du Congressional Budget Office (CBO, 2002) avance néanmoins que la baisse des prix des équipements TIC, si elle va se poursuivre, sera dans l'avenir plus faible qu'elle ne l'a été sur la période

1996-2000, même si elle reste plus rapide que pendant la période 1975-1995. Le caractère exceptionnel des années 1996-2000 s'expliquerait par certains facteurs transitoires : essentiellement par la baisse rapide et continue des prix des ordinateurs elle-même due à la chute des prix des semi-conducteurs résultant du passage en 1995 du cycle de 3 à 2 ans (Jorgenson, 2001); la croissance rapide d'Internet, le problème du bug de l'an 2000 et le besoin des firmes d'établir une présence sur Internet à travers le commerce électronique. Le CBO affirme dans ses projections de juin 2002 que les améliorations de qualité dans les ordinateurs vont ajouter 0,10 points de pourcentage à la croissance de la PTF pour la période 2002-2012.

L'étude du CBO réalise aussi une étude de sensibilité qui montre comment l'intensité de la baisse des prix des ordinateurs (ajustés sur la qualité) influence la croissance de la PTF. Le CBO a imaginé trois scénarios (voir aussi le tableau 3) :

- Dans le scénario médian, l'amélioration de la qualité des ordinateurs sur la période 2002-2012 va être plus faible que sur la période 1996-2000 et plus forte que sur la période 1975-1995. L'amélioration de la qualité des ordinateurs contribuerait à augmenter de 0,10% la croissance de la PTF sur la période 2002-2012.
- Dans le deuxième scénario (scénario bas), la baisse annuelle des prix des ordinateurs revient à son niveau de la période 1973-1994 (-14,4% par an). La croissance annuelle de la PTF baisserait alors pour atteindre 1,22% (contre 1,28% dans le scénario médian).
- Dans le troisième scénario (scénario haut), la baisse annuelle des prix des ordinateurs reste la même que sur la période 1995-2001 (-24,2% par an). L'amélioration de la qualité des ordinateurs contribuerait alors à ajouter 0,18% à la croissance de la PTF pour la période 2002-2012 (qui est alors de 1,36%/an).

La méthode de calcul utilisée par le CBO dans ce tableau pour ajuster les prix des ordinateurs à la qualité est une méthode hédonique. Il ressort logiquement que, plus la baisse des prix est forte, du fait des améliorations de la qualité, plus la croissance de la productivité est élevée. Jorgenson et Stiroh (2000) proposent une étude de sensibilité comparable. Ils montrent que la contribution directe des TIC à la croissance est d'autant plus élevée que la baisse des prix des produits TIC est forte.

Contrairement à la plupart des pays de la zone Euro, comme la France, les méthodes hédoniques sont très utilisées aux Etats-Unis (Lequiller, 2001)¹⁴.

¹⁴ Aux Etats-Unis, les méthodes hédoniques sont utilisées pour développer des indices de prix pour les ordinateurs,

L'utilisation des indices hédoniques aux Etats-Unis pourrait expliquer, au moins pour partie, les évolutions divergentes des économies de la zone Euro et des Etats-Unis. Cecchetti (2000) estime ainsi que l'application dans la zone Euro des méthodes hédoniques utilisées aux Etats-Unis permettrait d'accroître la croissance de la productivité mesurée dans la zone au cours des cinq dernières années d'environ 0,25 points de pourcentage. L'écart entre les économies européennes et américaines serait ainsi réduit.

Cependant, même aux Etats-Unis, les méthodes hédoniques ne sont pas généralisées. Il est donc possible que les estimations actuelles ne donnent qu'une borne inférieure de la contribution des TIC à la croissance.

<i>Projections du COB de juin 2002 : scénario médian</i>	
Taux de croissance tendanciel	1,07
Ajustement sur la qualité des ordinateurs	0,10
Autres ajustements	0,09
Total	1,28
<i>Scénario bas incorporant une baisse plus faible des prix des ordinateurs</i>	
Taux de croissance tendanciel	1.07
Ajustement sur la qualité des ordinateurs	
Utilisant la baisse moyenne des prix des ordinateurs de 1973-1994	0.04
Autres ajustements	0,09
Total	1,22
<i>Scénario haut incorporant une baisse plus rapide des prix des ordinateurs</i>	
Taux de croissance tendanciel	1.07
Ajustement sur la qualité des ordinateurs	
Utilisant la baisse moyenne des prix des ordinateurs de 1995-2001	0.18
Autres ajustements	0,09
Total	1,36

Tableau 3 : l'effet de différents sentiers de prix des ordinateurs sur la productivité totale des facteurs sur la période 2002-2012 (en % par an)

Source : Congressional Budget Office.

3.2. Réévaluer la productivité dans les secteurs tiers de l'économie

Alors qu'on attribue aux TIC l'accélération de la croissance de la productivité, dans certains secteurs de services, gros utilisateurs des TIC (notamment, les services financiers, la distribution, les assurances), on observe peu ou pas de gains de productivité. Ceci conduit Gordon à conclure que l'effet des TIC dans les secteurs de biens non durables reste limité. Une

les périphériques, les logiciels prêts à l'emploi et les composants électroniques. En France, les indices hédoniques ne sont utilisés que pour les ordinateurs et les imprimantes légères. Dans plusieurs pays de la zone Euro, comme la Belgique, les méthodes hédoniques ne sont pas du tout appliquées.

explication alternative à celle de Gordon serait que, dans certaines industries de services, les TIC créent de nouvelles formes d'output, très difficiles à capturer par les statistiques.

Dans le cas des services financiers, par exemple, l'introduction des opérations bancaires électroniques et des guichets électroniques a contribué à une baisse sensible du nombre de chèques émis. Dans la mesure où le nombre de chèques émis entre dans la composition de l'output et que les opérations électroniques ne sont, elles, pas encore comptabilisées, la baisse du nombre de chèques émis se traduit dans les statistiques par une baisse de la productivité du secteur financier. De plus, alors que leur volume n'est pas mesuré, le coût de ces nouveaux services l'est. Par conséquent, cette amélioration qualitative due à l'automatisation des produits bancaire se traduit dans les statistiques par une baisse de productivité.

Si le paradoxe de productivité de Solow a été résolu mais n'a pas disparu complètement, des paradoxes de productivité sectoriels subsistent donc. Ils s'expliquent vraisemblablement par les mêmes raisons qui ont été avancées pour expliquer le paradoxe de Solow. Tout d'abord, les gains de productivité des investissements TIC n'apparaissent pas immédiatement, mais à l'issue de processus d'apprentissage et de réorganisation au sein des entreprises. Par ailleurs, les nouvelles formes d'output issues de ces investissements sont mal mesurées au départ. Comme l'indique Caron (1997), leur mesure constitue un défi statistique majeur.

3.3. L'emploi des méthodes hédoniques pour la téléphonie mobile !

La téléphonie mobile dont l'essor a été spectaculaire représente le produit emblématique des NTIC (Magnien, 2003) et pose ainsi le même problème de mesure. En France, ce secteur n'a été représenté dans l'indice du prix à la consommation des ménages que très récemment¹⁵, alors que 64% des français sont équipés d'un téléphone mobile¹⁶. En effet, la construction d'un indice de prix pour la téléphonie mobile est très délicate, du fait de la complexité des offres tarifaires (forfaits, prix de l'appareil compris ou non) et de l'évolution rapide des offres.

L'utilisation d'un indice classique consisterait à décomposer la consommation globale en micro-consommations (abonnement, appel de type 1, ...

¹⁵ L'indice " sans friction " pour la téléphonie mobile a été introduit dans l'indice des prix à la consommation en janvier 2003.

¹⁶ Taux de pénétration au 31 décembre 2002 (source : ART).

appel de type n) dont les prix et les poids seraient agrégés par un indice de Laspeyres. Cet indice est incapable de saisir certains aspects complexes de la tarification ou de la consommation dans les mobiles : tarification plus élevée pour les minutes hors forfait, effets de substitution suite à des changements tarifaires, introduction de nouveaux produits, etc. De plus que l'apparition même de la téléphonie mobile en tant que nouveau produit à côté de la téléphonie fixe accroît le bien-être des consommateurs par l'élargissement des choix de consommation qui leurs sont offerts. Evaluer cette augmentation du bien-être due à l'arrivée de la téléphonie mobile sur le marché est une tâche extrêmement délicate et compliquée dans les calculs des indices des prix.

Une contribution importante dans ce cadre est celle d'Hausman (1999b). Il propose d'introduire la téléphonie mobile dans l'indice de prix à la consommation (IPC) aux Etats-Unis, en tant que nouveau produit ceci contribuerait à baisser fortement l'IPC. Autre aspect important de ce travail est que Hausman utilise dans ses estimations un indice qui renvoie à la notion d'utilité constante ce qu'on appelle "*l'indice à utilité constante*" (IUC) ou aussi "*l'indice de coût de la vie*" (COLI)¹⁷ que le BLS considère aujourd'hui comme cadre théorique pour l'IPC (BLS, 1997).

Le COLI représente le ratio ou l'évolution minimale des dépenses des consommateurs entre une période de base et une période courante, nécessaire pour maintenir à la période courante le niveau d'utilité de la période de base face à la variation différenciée des prix des produits offerts. Il permet aussi de mesurer l'augmentation du bien-être des consommateurs, consécutive à l'introduction d'un nouveau produit comme la téléphonie mobile. Hausman montre qu'en omettant la téléphonie mobile, l'indice des prix des services de télécommunications est biaisé de l'ordre de 0,8% à 1,9% par an sur la période 1983-1998 (période entre l'apparition des mobiles sur le marché américain et leur introduction dans le CPI). Toutefois, le COLI a un aspect plus conceptuel que pratique et pose, par conséquent, des problèmes importants lorsqu'il doit être mis en place au niveau opérationnel (Lacroix et Magnien, 2001).

Lacroix et Magnien (2001) proposent une approche différente pour la construction d'un indice de prix pour la téléphonie mobile qui renvoie plutôt à la notion d'"usage constant" beaucoup plus facile à mettre en œuvre. La méthode consiste, dans un premier temps, à réduire la liste des produits en définissant des classes de produits substituables permettant de satisfaire des

modes de consommations voisins (par exemple : "petits" consommateurs, "gros" consommateurs, etc.). Dans un second temps, pour chaque classe de produit, on procède à une classification des consommateurs en fonction de leur profil de consommation en utilisant des valeurs moyennes de certaines variables "*comportementales*" : nombre et durée mensuelle des appels, répartition entre plages horaires, nombre de SMS envoyés, etc. (Magnien, 2003). On retient alors le produit le moins cher parmi l'ensemble de produits offerts par les opérateurs et satisfaisant ce besoin spécifique. Enfin un indice de valeur unitaire est calculé pour chaque profil de consommation. La dernière étape consiste à affecter des pondérations aux différents profils de consommations et à calculer un indice de Laspeyres à partir des indices de valeur unitaire de chacun des profils¹⁸.

L'avantage principal de cette méthode est qu'elle permet grâce à la segmentation des consommateurs en profils, la prise en compte de la mobilité des consommateurs entre produits et donc d'éviter le biais de l'indice dû aux effets de substitutions. Cependant, il s'avère nécessaire de noter que les indices de valeur unitaire retracent l'évolution de la valeur moyenne des biens ; il ne s'agit donc pas de véritables indices de prix. Le problème est qu'un indice de valeur unitaire peut varier sans que les prix et les quantités totales aient bougé. Or, la finalité d'un indice de prix est de suivre le mouvement des prix dans le temps pour construire un déflateur comptable.

Alors qu'on reproche aux indices traditionnels de ne pas prendre en compte les effets de substitution, c'est le contraire avec la méthode d'indice de prix moyen ou de valeur unitaire : les effets de substitution ont trop d'impact. Une autre limite de cette méthode est que, pour deux modes de consommation différents – par exemple, une distribution différente des appels entre jour d'une part et soir et week-end d'autre part – peuvent procurer le même degré de satisfaction, avec des coûts différents ! (Magnien, 2003).

La construction d'un indice de prix pour la téléphonie mobile reste donc un problème d'actualité. Une piste intéressante à explorer serait de construire un indice de prix hédonique, pour prendre en compte de façon satisfaisante les améliorations de qualité des services de la téléphonie mobile jusqu'ici ignorées dans le calcul de l'IPC par toutes les méthodes proposées. Procéder de la sorte pour un produit TIC aujourd'hui mal mesuré (la téléphonie mobile) pourrait conduire, dans un second temps, à réévaluer à la hausse l'impact des TIC sur la productivité.

¹⁷ Le terme "indice de coût de la vie" est utilisée de manières différentes en France et aux Etats-Unis. Dans ce pays, ce terme fait référence à la théorie économique du consommateur et a un aspect plus conceptuel que pratique. Alors qu'en France, l'indice de coût de la vie renvoie à la notion de "budget type minimum".

¹⁸ L'étude repose sur la statistique des minutes consommées, et donc, sur le prix moyen de la minute consommée, qui est égal au chiffre d'affaires divisé par le nombre total de minutes.

4. CONCLUSION GÉNÉRALE

Dans cet article, nous avons montré qu'au regard des dernières études disponibles et malgré la crise que traverse l'industrie des TIC, les TIC ont toujours un impact profond sur l'économie. La contribution des TIC à la croissance de la productivité reste importante. Nous avons suggéré que cet impact pourrait être non pas sur-estimé mais sous-estimé. Car, contrairement à ce qu'affirment beaucoup de chercheurs, l'informatisation de la production dans l'industrie manufacturière n'est pas un phénomène nouveau puisqu'elle a commencé, il y a 40 ans déjà. La nouveauté est assignée donc principalement aux services et aux entreprises opérant dans les secteurs de haute technologie (Boyer, 2002). C'est justement le problème, car, dans certaines industries, en particulier les industries de services, les TIC semblent ne pas avoir d'effet sur la productivité ; Cette constatation pourrait s'expliquer essentiellement par des problèmes de mesure. Rares encore sont les recherches qui ont étudié l'éventuelle accélération de la PTF dans les branches du tertiaire grandes utilisatrices de TIC.

La généralisation des indices hédoniques – par exemple, pour la construction d'un indice de prix pour la téléphonie mobile– pourrait conduire à réévaluer l'impact des TIC sur la productivité.

Par ailleurs, nous nous sommes concentrés sur le cas américain. Si les TIC contribuent fortement à la croissance économique, leur impact n'est pas le même dans les différents pays industrialisés. Mairesse, Cette et Kocoglu (2000) montrent ainsi que la contribution des TIC à la croissance à la fin des années 90 serait de l'ordre du simple au triple entre les Etats-Unis et la France.¹⁹ Comme le note ces auteurs, cet écart est préoccupant, car il influera sur les performances à moyen-long terme des économies de ces deux pays.

BIBLIOGRAPHIE

- Arvisenet, P.d'. (2001), " Que reste-il de la nouvelle économie? ", *Conjoncture*, PNB Paribas, 5, mai, p.2-11.
- Baily, M.N and Lawrence, R.Z. (2001), "Do We Have a New Economy? ", *American Economic Review*, Papers and Proceedings, 91 (2), may, p.308-312.
- Baker, D. (1998), "The Computer Driven Productivity Boom, Challenge, November-December.
- Baudchon, H. et O. Brossard (2001), " Définitions et mesures des technologies de l'information et de la communication : enjeux et paradoxes ", 9^{ème} colloque de comptabilité nationale.

- Berndt, E. (1991), " The Practice of Econometrics: Classic and Contemporary ", Addison-Wesley Longman.
- Berndt, E., Dulberger, E. and Rappaport, N. (2000), " Price and Quality of Desktop and Mobile Personal Computers: A Quarter Century of History ", National Bureau of Economic Research.
- Caron, F. (1997), "Les deux Révolutions industrielles du XXème siècle ", Albin Michel.
- Cecchetti, S.G. (2000), " Early warning signs of the US productivity pickup: implications for Europe ", unpublished working paper, Ohio State University and NBER, 14 August.
- Duval, R. (2000), " Quel crédit faut-il accorder à la Nouvelle Economie" américaine ? ", Mimeograph Direction de la Prévision, Paris.
- Gordon, R.J. (2000), " Does the "New Economy" Measure up to the Great Inventions of the Past ? ", May 1, 2000 draft of a paper for the *Journal of Economic Perspectives* , vol. 14, pp. 49-74.
- Gordon, R.J. (2002), " Chapter 3: The United States", *Technological Innovation and Economic Performance*, in Richard Nelson, Benn Steil et David Victor (éd.), Princeton: Princeton University Press, pp. 49-73.
- Gorko, J. et Bordé, F. (1999), "Wired Telecommunication Carriers Price Indexes
- Hausman, J. (1999), " Cellular Telephone, New Products and the CPI", *Journal of Business & Economic Statistics*, April 1999, vol. 17, n°2 pp. 188-194.
- Jorgenson, D.W. (2001), " Information Technology and the U.S. Economy", *American Economic Review*, vol.91, n°1, pp.1-32
- Jorgenson, D.W. (2002), " La croissance économique américaine à l'ère de l'informatisation ", *isuma*, vol.3, n°1.
- http://www.isuma.net/v03n01/jorgenson/jorgenson_f.pdf
- Jorgenson, D.W. and K.J. Stiroh (2000), " Raising the Speed Limit: US economic growth in the Information Age ", *Brookings Papers on Economic Activity*, n°1, 125-211.
- Jorgenson, D.W., Ho, M.S. and K.J. Stiroh (2003), " Projecting Productivity: Lessons from the U.S. Growth Resurgence".
- Lacroix, T. et Magnien, F. (2001), " Methodology for constructing a price index for mobile-telephony services", INSEE, paper presented at the Sixth Meeting of the International Working Group on Price Indices Canberra, Australia 2-6 April.

¹⁹ Ceci est dû en particulier à un écart important dans la diffusion des TIC entre les deux pays.

- Lequiller, F. (2001), “ La nouvelle économie et la mesure de la croissance du PIB”, *INSEE*, Série des documents de travail de la Direction des Etudes et Synthèses Economiques
- Mairesse, J., Cette, G. et Y. Kocoglu (2000), “ Les technologies de l’information et de la communication en France : diffusion et contribution à la croissance ”, *Economie et Statistique*, N°339-340, pp. 117-146.
- Magnien, F. (2003), “ Mesurer l’évolution des prix des services de la téléphonie mobile : une entreprise difficile”, *Economie et Statistiques* n°362.
- McKinsey Global Institute United States (2001), “Productivity Growth 1995-2000”. October.
- McGuckin, R. and Stiroh, K. (2001), “Do Computers Make Output Harder to Measure ? ”, *Journal of Technology Transfer*, 26, pp.295-321.
- McGuckin, R.H. et Ark, V.B. (2003), “ Performance 2002, Productivity, Employment, and Income in the World’s Economies ”, *The Confence Board*.
- <http://www.ai2.as/docs/CEDE/Performance%202002.pdf>
- OCDE, Nicoletti, G., Scarpetta, S. et Boylaud, O. (2000), “ Summary indicators of product market regulation with an extension to employment protection legislation”, Economic Department, Working Paper, n°226, OCDE, avril.
- [http://www.oilis.oecd.org/olis/1999doc.nsf/63c71d2d4054d0fd0125685d0053aee4/5ef586bbe13dd52ac125684a003a8da0/\\$FILE/00075836.PDF](http://www.oilis.oecd.org/olis/1999doc.nsf/63c71d2d4054d0fd0125685d0053aee4/5ef586bbe13dd52ac125684a003a8da0/$FILE/00075836.PDF)
- OCDE : “ La nouvelle économie: mythe ou réalité? ”, Rapport de l’OCDE sur la croissance, Paris (2001).
- Oliner, S.D. et D.E. Sichel (2000), “ The resurgence of growth in the late 1990s: is information technology the story?”, *Journal of Economic Perspectives*, 14, pp. 3-22.
- Oliner, S.D. et D.E. Sichel (2002), “ Information Technology and Productivity: Where Are We Now and Where Are We Going? ”, mimeo.
- Schreyer, P (2001), “ Computer price indices and international growth comparisons”, Growth Project Background papers, OCDE, Paris. www.oecd.org/subject/growth/products/index.htm.
- Solow, R.M. (1956), “ A Contribution to the Theory of Economic Growth ”, *Quarterly Journal of Economics*, 70, pp. 65-94.
- Triplett, J.e. (2001), “ Les Etats-Unis ont-ils connu une Nouvelle Economie ”, Intervention lors de la 9^{ème} Conférence sur la comptabilité nationale de l’Association de Comptabilité Nationale ‘Comment mesurer la nouvelle économie ?’. Paris novembre.
- The Congress of the United States Congressional Budget Office (CBO) (2002), “ The Role of Computer Technology in the Growth of Productivity ”.
- Economics and Statistics Administration (2000), “ *Digital Economy 2000* ”, Washington, DC, U.S. Department of Commerce, juin.
- Economics and Statistics Administration: (2002), “ *Digital Economy 2002* ”, Washington, DC, U.S. Department of Commerce. February.

Cyril Pachon,

Laboratoire Vérimag, Centre Equation, 2, Avenue de Vignate 38610 Gières
Cyril.pachon@imag.fr

Résumé: La production et l'exécution de séquences de tests restent une approche incontournable pour la validation d'un logiciel. Pour cela, et afin de réduire les coûts, il est intéressant de générer automatiquement les séquences de tests. La plupart des techniques et outils existants sont axés sur le test de conformité d'un logiciel par rapport à sa spécification. Nous proposons une approche axée sur le test de robustesse. Elle a pour but de garantir un comportement acceptable du logiciel en dehors de ses conditions nominales (spécification) vis à vis d'une propriété donnée. C'est à dire sa capacité à résister à différents aléas provenant son de environnement ou de défaillances internes.

Mots Clés: Spécification mutée, Modèle de fautes, Test de robustesse, Vérification et validation de modèles

Summary: The production and the execution of tests suites are a crucial activity for software validation. For that, and to reduce the costs, it is interesting to produce automatically the tests suites. The particular problem of testing if an implementation is "correct" with respect to its specification is referred to as conformance testing. We propose here an alternative approach focussing on robustness testing. It consists in checking the ability of an implementation to satisfy some given properties in spite of internal failures or within a non nominal environment.

Key words: Mutated Specification, faults Model , Robustness testing, Checking and validation of models.

Une approche pour la génération automatique de tests de robustesse

1. INTRODUCTION

La complexité des systèmes réactifs ¹ nécessite l'évolution et le développement des méthodes et des outils adéquats pour leur validation. Un dysfonctionnement de programmes critiques, même temporaire, peut avoir de lourdes conséquences économiques, voire attenter à la sécurité des individus. Leur validation avant leur mise en place est essentielle et doit tenter de garantir la correction de l'ensemble des comportements possibles du programme.

La production et l'exécution de séquences de tests restent une approche incontournable dans le processus de validation d'un logiciel [Tretmans 1992, Phalippou 1994]. Pour améliorer la pertinence de la phase de test, et réduire ainsi son coût, il est intéressant en pratique de pouvoir générer automatiquement les séquences de tests à partir d'une description initiale d'un programme (soit le programme source, soit une spécification de plus haut niveau). Divers critères sont alors envisageables pour sélectionner ces séquences de tests : syntaxiques (la couverture du flot de données, ou du flot de contrôle) ou sémantique (le test d'une fonctionnalité précise). Cette dernière approche a donné lieu à de nombreux outils comme Lurette [Raymond 1998], TestComposer de ObjectGEODE [Groz 1999], TGV (Test Generation using Verification technology) [Fernandez 1997], TorX [Berlinfante 1999], TVeda [Phalippou 1994bis], etc.

Les techniques existantes de génération automatique de tests sont essentiellement axées sur la *conformité* [Phalippou 1994bis, Tretmans 1992, Morel 2000] : le test a pour

objectif de trouver des séquences d'exécution incorrectes par rapport à un comportement nominal ² du programme. Toutefois, il est souvent nécessaire de considérer également le comportement du logiciel en dehors de ses conditions nominales. Nous parlons alors de la *robustesse* d'un système, c'est à dire sa capacité à résister à différents aléas dus aussi bien à son environnement d'exécution (conditions aux limites, malveillances, etc) qu'à des défaillances internes de certains de ses composants. Si le test de robustesse est assez bien étudié et mis en œuvre dans le domaine du matériel, très peu de travaux existent à notre connaissance dans le cas du logiciel [Arlat 2002, Koopman 1999]. Dans ce papier, nous proposons une approche pour la génération de tests de robustesse en nous inspirant des techniques issues du test de conformité.

Dans un premier temps, en section 2., nous présentons les différents modèles utilisés dans la suite. Puis, en section 3., nous proposons une définition formelle du test de robustesse. Nous décrivons ensuite, en section 4., comment s'effectue la génération automatique des cas de tests. Et enfin avant de conclure, nous présentons, en section 5., un exemple.

2. MODELES

IOLTS : Systèmes de transitions étiquetées avec entrées/sorties : Un IOLTS (Input-Output Labelled Transition Systems) est un des modèles utilisés pour décrire le comportement d'un système réactif. C'est un raffinement des LTS (Labelled Transition Systems) dans lesquels les entrées et les

¹Par systèmes réactifs, nous définissons tout composant logiciel en interaction permanente avec son environnement

²Nous entendons par nominal toutes les conditions idéales d'utilisation décrites par une spécification initiale

sorties sont distinguées. Cette distinction est nécessaire du fait de la nature asymétrique de l'activité du test. Ainsi, nous considérons un alphabet fini d'actions A , partitionné en deux ensembles: *les actions d'entrées* A_I et *les actions de sorties* A_O . Notons que les actions internes sont étiquetées par un symbole τ , ($\tau \in A$). τ représente des actions non observables par rapport à l'environnement du système (le testeur) tandis que les autres actions A_I et A_O sont visibles par rapport ce même environnement.

Définition 2.1 Nous considérons un IOLTS $M = (Q_M, A_M, T_M, q^{init}_M)$ où Q_M est l'ensemble fini des états, q^{init}_M l'état initial, A_M l'ensemble fini des actions avec $A_M \subseteq A$, et $T_M \subseteq Q_M \times A_M \cup \{\tau\} \times Q_M$ la relation de transition.

Notations : Pour un ensemble fini X , nous notons X^* (resp $X^\omega = X \rightarrow N$) l'ensemble des séquences finies (resp infinies) sur X . Pour un IOLTS M , si $\rho \in Q_M^\omega$, nous notons $\text{inf}(\rho) = \{q \mid \forall n \exists m, m \geq n \text{ et } \rho(m) = q\}$, $\text{inf}(\rho)$ dénote l'ensemble des états de Q_M qui se produit infiniment souvent dans ρ . Soit $\sigma \in A_M^*$, $\alpha, \alpha_i \in A_M, p, q, p_i \in Q_M. p \xrightarrow{\tau^* \alpha} q$ si et seulement si $\exists p_0, p_1 \dots p_n. P = p_0 \xrightarrow{\tau} p_1 \xrightarrow{\tau} \dots p_n \xrightarrow{\tau} q, p \xrightarrow{\sigma} q$ si et seulement si $\exists \alpha_0, \alpha_1 \dots \alpha_n \in A_M, p_0, \dots, p_n \in Q_M, \sigma = \alpha_1, \alpha_2 \dots \alpha_{n-1}, p_0 = p, p_i \xrightarrow{\tau^* \alpha_i} p_{i+1}, i < n$ et $q_n = q. A(q) = \{\alpha \mid \exists q' \text{ et } q \xrightarrow{\tau^* \alpha} q'\}$ est l'ensemble des actions immédiates après q , $I(q) = A(q) \cap A_I$ est l'ensemble des entrées après q , et $O(q) = A(q) \cap A_O$ est l'ensemble des sorties après q . Le langage fini accepté par M , noté $L(M)$ est : $L(M) = \{\sigma \mid \exists q \in Q_M \text{ et } q_M^{init} \xrightarrow{\sigma} q\}$. Notons par $p - / \alpha$, si une action α est non présente dans le domaine de sortie d'un état p .

Modèle de spécification : La description des comportements d'un système complexe directement sous la forme d'un IOLTS n'est pas très lisible. En pratique, nous utilisons plutôt des langages spécialisés comme SDL [ITU-T, 1992], Lotos [Logrippo, 1991, Salona, 1993], etc. Ces langages ont une sémantique opérationnelle en termes de IOLTS.

Modèle d'implémentation :

L'implémentation réelle sous test (implementation under test : *IUT*) est une boîte noire. Seules les interactions avec l'environnement sont observables. Le comportement de l'implémentation est supposé modélisé par un IOLTS où les entrées (resp. les sorties) sont les sorties de l'environnement (resp. les entrées).

Architecture de tests : L'architecture de tests détermine une seconde partition de l'ensemble d'action A en actions contrôlables A_c (délivrées par le testeur) et observables A_u (observées par le testeur): $A = A_u \cup A_c$.

Observabilité et Contrôlabilité : Un IOLTS est *déterministe* si et seulement si il existe une seule transition sortante étiquetée par une même action : $\forall \alpha. p \xrightarrow{\alpha} p' \wedge p \xrightarrow{\alpha} p'' \Rightarrow p' = p''$. Nous supposons dans la suite que le IOLTS associé à la spécification est déterministe (il suffit de le déterminer dans le cas contraire) et sans action interne.

Un IOLTS satisfait la condition de contrôlabilité si et seulement si pour chaque état, une sortie est possible, si et seulement si il y a exactement une transition sortante. Formellement, si $|X|$ dénote la cardinalité de l'ensemble X , alors : $\forall p. |I(p)| = 0 \vee (|I(p)| = 1 \wedge I(p) = A(p))$.

Blocage et Divergence: Un blocage (ou absence d'actions) n'est pas forcément une erreur. Il peut être de type deadlock (plus d'évolution du système), livelock (divergence

d'actions internes) ou exister dans la spécification initiale. Les blocages sont explicités par une action δ sur le IOLTS de la spécification. Pour ce faire, nous remplaçons chaque transition de blocage $(p, -/\delta)$ ou (p, τ^*, p) par (p, δ, p) [Tretmans 1996].

Cas de tests : Le type de test auquel nous nous intéressons ici est de type boîte noire à travers des interfaces appelées *PCO* (point de contrôle et d'observation). Un cas de test *CT* est un IOLTS contrôlable et déterministe. Chaque trace est étiquetée par un verdict pour détecter, à l'exécution, la conformité ou la non-conformité de l'IUT vis-à-vis de la spécification [Tretmans, 1996].

3. TEST DE ROBUSTESSE

L'IEEE définit la robustesse comme "le degré selon lequel un système, ou un composant, peut fonctionner en présence d'entrées invalides ou de conditions environnementales stressantes". Nous adaptons ici cette définition et proposons la définition informelle suivante:

3.1 Présentation

Définition informelle 3.1

Un système/composant logiciel est dit robuste vis à vis d'une propriété si et seulement si celle-ci est préservée lors de toutes exécutions, y compris en dehors des conditions nominales. Plus précisément, le système garde un fonctionnement acceptable en présence d'entrées invalides, de dysfonctionnements internes, de conditions de stress, de pannes, etc.

A partir de cette définition informelle, nous proposons une définition du test de robustesse basée sur trois éléments:

- **S** est une spécification exprimant les comportements de l'IUT dans l'environnement nominal. S est décrit par un IOLTS déterministe.

- **MF** est un modèle de fautes décrivant l'ensemble des fautes et événements inopportuns pouvant survenir en entrée et en sortie de l'IUT pendant une exécution dans un environnement "réel" (non nominal). Cet ensemble peut inclure les pannes de l'IUT, les pertes lors de communication entre composants, les entrées non prévues et survenues de façon spontanées lors d'une exécution, etc. En particulier MF permet d'étendre ou de restreindre certains comportements décrits par S. Une représentation exacte de cet ensemble dépend du langage de spécification considéré. Il peut par exemple être aussi donné sous forme d'annotations ou de mutations de S. Le formalisme d'un tel modèle reste indépendant du logiciel. Il contient toutes les caractéristiques de l'environnement d'implémentation. Si le modèle de fautes est donné seules les références utiles au logiciel seront considérées. Si ce modèle est induit automatiquement à partir des références du logiciel il sera alors gardé dans sa totalité.

- **P** est une propriété dite de *robustesse*. Cette propriété doit être vérifiée par l'IUT lorsque celle-ci évolue dans un environnement "réel", spécifié par MF. Par exemple une propriété peut exprimer : " l'IUT peut toujours revenir dans un état de fonctionnement acceptable lorsqu'il y a eu une erreur ", ou "l'IUT peut toujours préserver un invariant", etc. Il existe de nombreux formalismes pour exprimer une propriété *P*, sur les séquences finies ou infinies. Ces formalismes se rangent dans deux catégories : les logiques temporelles et les automates. Nous supposons que la propriété *P* peut être par exemple exprimée en une logique temporelle linéaire.

Une différence essentielle avec le test de conformité est donc que nous cherchons à établir une relation entre l'IUT et une propriété externe *P*. La spécification *S* et le modèle de fautes *MF* ne servant ici que de guide pour la génération du test. Par ailleurs, le test généré peut avoir des comportements

qui n'apparaissent pas nécessairement dans la spécification.

Informellement, tester la robustesse de l'IUT consiste à exécuter des traces permettant de valider ou d'invalidier la propriété P donnée. Ces traces d'exécution sont extraites de la spécification S, en tenant compte du modèle de fautes MF. Nous cherchons donc à extraire des séquences ne vérifiant pas P, et à tester leur exécutabilité sur l'IUT.

Finalement, nous supposons qu'il y a cohérence entre ces différents éléments et l'architecture de test. La propriété de robustesse doit être vérifiable par un testeur extérieur. La description de l'environnement par le modèle de fautes, doit demeurer "générable" pendant le temps d'exécution du test.

3.2 Propriété de robustesse

Pour trouver les séquences d'exécution d'une spécification en respectant la propriété P, il est souvent plus facile de construire un automate observateur reconnaissant les séquences de $\neg P$ (et donc de caractériser la non robustesse).

Pour envisager les cas où l'IUT ne respecte pas la spécification, l'observateur (OBS) doit être complet afin d'accepter les événements observables non spécifiés de l'IUT. Selon la propriété P considérée, la non robustesse peut être caractérisée soit par des séquences d'exécution finies soit par des séquences d'exécution infinies, par exemple:

- Montrer qu'une implémentation ne viole pas un invariant est exprimé par une séquence finie.

- Montrer qu'une implémentation ne retourne jamais dans une situation d'exécution normale après une situation d'erreur est exprimée par une séquence infinie.

Pour cette dernière description nous utilisons les automates de Rabin [Rabin 1969] reconnaissant des séquences infinies.

Définition 3.2 Un automate de Rabin R_a est un couple (S, P) ou $S = (Q_s, A_s, T_s, q^{init}_s)$ est un IOLTS et $P = \langle (L_1, U_1), (L_2, U_2), \dots, (L_k, U_k) \rangle$ est un ensemble de couples $(L_i, U_i) \subseteq \{Q_s \times Q_s\}$

pour $i \in \{1, 2, \dots, k\}$.

Le langage accepté par R_a , noté $L(R_a)$ est :

$L(R_a) = \{ \sigma \mid \sigma \in A_s^\omega \exists k \exists \rho \in Q_s^\omega \forall i \text{ et}$

$\rho_i \sigma_i \rightarrow \rho_{i+1} \rho_0 = q^{init}_s \mathbf{inf}(\rho) \cap L_k \neq \emptyset \text{ et } \mathbf{inf}(\rho) \cap U_k = \emptyset \}$.

Dans le cadre du test, il est nécessaire de se restreindre à des séquences finies. Pour cela, nous paramétrons la longueur des séquences d'exécution en associant à chaque ensemble L_i (resp U_i) un compteur cl_i (resp cu_i). Nous introduisons ainsi les automates de Rabin Paramétrés bornés.

Définition 3.3 Un automate de Rabin Paramétré borné par compteurs AR_a est un couple (R_a, C) où R_a est un automate de Rabin et $C = \{(cl_1, cu_1), \dots, (cl_p, cu_p)\}$ avec $(cl_i, cu_i) \in \mathbb{N} \times \mathbb{N}$ un ensemble de couple de compteur. Une séquence d'exécution $\sigma \in A_s^*$ est acceptée par AR_a si et seulement si $\exists k' \mid |\sigma \cap [L]_{k'}| \geq cl_{k'}$ et $|\sigma \cap U_{k'}| \leq cu_{k'}$.

Nous donnons par exemple la négation d'une propriété informelle "Le système peut toujours revenir dans un mode nominal après être entré dans un mode dégradé" par l'automate de Rabin de la figure 1:

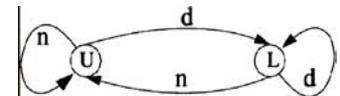


FIG. 1: Automate de Rabin d'une séquence infinie

L'action "d" (respectivement "n") représente l'accès au mode dégradé (respectivement l'accès au mode nominal). L'ensemble des séquences non robustes est exprimé par $(n+d)^*d^\omega$. Cet ensemble correspond aux séquences reconnues par l'automate (figure

1). Pour respecter la définition, il faut enfin associer aux états de L et de U les compteurs respectifs avec $cu < cl$.

Nous décrivons une propriété de sécurité "Toutes actions doivent être différentes de l'action d" par un automate de Rabin Paramétré borné et paramétré par compteur. Le fait de rester sur un état sans action possible est un blocage prévu par la spécification (noté par le symbole δ sur la figure 2). Nous complétons notre observateur par la possibilité d'accepter toutes les actions de l'IUT non prévues par la spécification (noté par le symbole * sur la figure 2). Cette propriété s'écrit enfin:

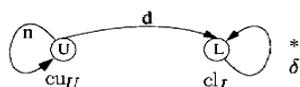


FIG. 2: Automate de Rabin Paramétré borné d'une propriété de sécurité

3.3 Test de robustesse

Dans notre approche le critère de robustesse est exprimé par une propriété P : une implémentation à tester (IUT) est considérée robuste si et seulement si elle satisfait P en présence de fautes suivant un comportement de la spécification. $\neg P$ est donc décrit comme un observateur (OBS).

Formellement, un cas de test de robustesse est un IOLTS CT, contrôlable, déterministe. Le verdict associé à une séquence σ (trace de CT) est :

- NONROBUST si σ est une séquence de l'observateur (OBS) exécutable sur l'IUT : $\sigma \in L(IUT) \cap L(OBS) = \text{NONROBUST}$.
- ROBUST si σ est une séquence guidée par la spécification mutée mais non inclus par l'observateur : $\sigma \in L(IUT) \cap L(S) - L(OBS) = \text{ROBUST}$.

4. GENERATION DE CAS DE TESTS

4.1 Principe

La technique de génération de cas de tests se décrit de la façon suivante: Premièrement,

nous calculons le comportement de la spécification S , enrichie du modèle de fautes MF. Cette technique est intéressante dans le cas où l'IUT ne satisfait pas la propriété. Si le comportement enrichi contient des séquences d'exécution incorrectes, alors ces séquences sont candidates pour devenir des cas de tests. Pour effectuer cette génération, nous proposons l'architecture suivante, figure 3:

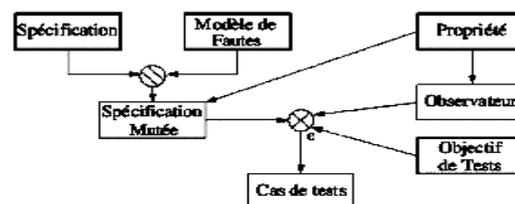


FIG. 3: Architecture de la génération de cas de test

Génération d'une spécification mutée S_m constituée par composition entre la spécification nominale S et le modèle de fautes MF.

Génération d'un observateur OBS, exprimant la propriété de non robustesse $\neg P$. L'objectif de l'observateur a deux rôles, d'abord de vérifier la non robustesse de S_m puis de marquer les séquences d'exécution incorrectes respectant $\neg P$.

Synthèse des cas de tests (à partir du produit synchrone entre S_m et OBS): Les séquences d'exécution incorrectes sont extraites de S_m . Elles deviennent des cas de tests exécutables produisant un verdict de robustesse.

Le processus ci-dessus est entièrement automatisable. La spécification mutée est obtenue par transformation syntaxique. La génération de cas de tests à partir de la spécification mutée, de l'observateur et éventuellement de l'objectif de tests est basée sur l'utilisation d'algorithmes en profondeur d'abord comme dans l'outil TGV.

4.2 Composition entre Modèle de fautes et spécification

Les annotations du modèle de fautes vont pouvoir faire muter le code de la spécification. Le résultat est un IOLTS (S_m) augmenté ou diminué en terme de comportements par rapport à la spécification nominale.

4.3 Produit synchrone étendu

L'objectif de ce produit synchrone est de marquer les séquences d'exécution de S . Cependant, le produit a besoin d'être étendu. D'une part, pour ne pas bloquer les exécutions de l'IUT, tant qu'il y a vérification de la robustesse. Et d'autre part, pour prendre en compte les séquences d'exécution non définies de l'IUT par la spécification mutée. En d'autres termes, le résultat de ce produit définira un observateur pour l'IUT. Le produit synchrone étendu est défini de la manière suivante: $S_{\otimes_e} = S_{sm} \otimes_e S_{obs} = (Q_{\otimes_e}, A_{\otimes_e}, R_{\otimes_e}, q_{\otimes_e}^0)$ où $q_{\otimes_e}^0 = (q_{sm}^0, q_{obs}^0)$ est l'état initial, Q_{\otimes_e} est le sous ensemble de $Q_{sm} \times Q_{obs}$ des états accessibles depuis $q_{\otimes_e}^0$ par la relation de transition R_{\otimes_e} définie comme suit:

$(p_{sm}, p_{obs}) \xrightarrow{a} (q_{sm}, q_{obs})$ si et seulement si une des trois conditions suivantes est vérifiée :

$$p_{sm} \xrightarrow{a} q_{sm} \text{ et } p_{obs} \xrightarrow{a} q_{obs} \quad (1)$$

$$p_{sm} \xrightarrow{a} q_{sm} \text{ et } p_{obs} \xrightarrow{-/a} \text{ et } a \in A_I \quad (2)$$

$$p_{sm} \xrightarrow{-/a} \text{ et } p_{obs} \xrightarrow{a} q_{obs} \text{ et } a \in A_o \quad (3)$$

4.4 Construction d'un graphe de test à partir du produit synchrone

Principe : Pour calculer les séquences d'exécution vérifiant P , nous effectuons une recherche des composantes fortement connexes sur le produit synchrone (CFC algorithme [Tarjan 1972]). Nous enrichissons cette recherche par l'étiquetage des racines

des CFC trouvées. Ce marquage distingue les composantes contenant des états de U et/ou de L ou aucun des deux. Ce marquage se fait de la manière suivante : La CFC est marquée L si elle comporte au moins un état de L. Elle est marquée U si elle ne contient pas d'état de L et contient au moins un état de U. Elle est marquée rien s'il n'y a ni état de L ni état de U.

Nettoyage du graphe : Seules les composantes terminales menant à des CFC terminales étiquetées L sont acceptées par l'observateur. Notre algorithme supprime itérativement toutes les feuilles non étiquetées L ainsi que les transitions entrantes dans ces feuilles jusqu'à n'avoir que des séquences qui vérifient $\neg P$.

4.5 Sélection des cas de tests et verdicts

La sélection est faite parmi les séquences restantes. Dès l'extraction d'une séquence, les verdicts sont mis en fonction des états de L et U. A l'exécution les compteurs vérifient des valeurs fixées aux départs. Le verdict est donc ROBUST si un compteur de U ne dépasse pas une valeur alors que celui de L dépasse une valeur. S'il y a non respect de ces compteurs le verdict est NONROBUST.

5. EXEMPLES

Présentation : L'exemple présenté ici est une modélisation d'ouverture de porte. L'automate est réduit à ouvrir puis refermer la porte dès réception d'un ordre d'ouverture. L'IOLTS de la spécification comporte trois états et quatre actions (réception d'un ordre *REQOPEN* ou *OTHER* émission de commande *OPEN* ou *CLOSE*). Le modèle de fautes est réduit à la perte éventuelle des messages *other* transmis et à la possibilité de recevoir plusieurs fois le message ReqOpen. La propriété P est : Après toute ouverture, il y a une fermeture. La propriété $\neg P$ est décrite par un automate de Rabin Paramétré réduit à deux états (un état de L et un d'état U), les

compteurs associés aux états respectant $cu_1 \ll cl_2$.

Mutation de la spécification : Les annotations du modèles de fautes vont ici supprimer certains comportements de la spécification nominale. En effet, là où nous pensions que le message other arriverait toujours, il n'est plus retenu. La mutation a eu comme effet de supprimer cette action sur la spécification. De plus nous rajoutons la possibilité d'avoir un message ReqOpen sans produire le message Open. Concrètement la mutation augmente le comportement de la spécification initiale.

Produit synchrone : Le produit donne tous les comportements possibles.

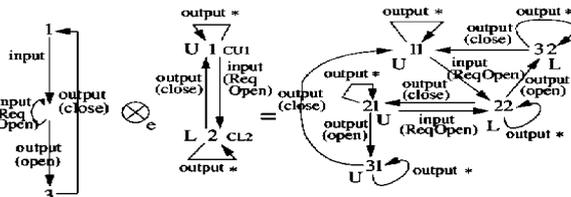


FIG. 4: produit synchrone étendu

Grphe et nettoyage du graphe : Du produit, nous appliquons l'algorithme de recherche des composantes fortement connexes, nous marquons les racines puis nous supprimons toutes parties ne concernant pas la propriété par l'algorithme de nettoyage. Sur cet exemple toutes les séquences sont conservées car le graphe est fortement connexe et comporte des états de L.

Sélection des séquences de tests : Du graphe restant, les séquences sélectionnées sont celles comportant des états de L. Un verdict est associé à chaque état. Par exemple une séquence possible est en débutant de 11 : $(\text{output}^{\circ} \wedge (\text{input ReqOpen})) \wedge (\text{output}^{\circ} \vee ((\text{output close}) \wedge (\text{output}^{\circ} \vee (\text{output open}))) \wedge (\text{output}^{\circ} \vee (\text{output close})))$

6. CONCLUSION

Nous avons proposé une approche formelle pour générer automatiquement des tests de robustesse. Elle est basée sur les notions de mutation de code pour exprimer la dégradation (par modèle de fautes) que peut subir une implémentation dans un environnement non nominal. Un test est caractérisé par une propriété de robustesse qui doit être préservée afin de garantir un comportement acceptable lors de toutes exécutions de l'implémentation. Enfin, un prototype est en cours de réalisation.

Remerciement : Je tiens à remercier toutes les personnes des laboratoires IRISA, LAAS, LaBri, LRI et Vérimag participant du groupe de travail de l'Action Spécifique sur le test de robustesse.

Bibliographie

- [Arlat, 2002] Arlat J., Fabre J.-C., Rodriguez M. et Salles F.. – Dependability of COTS microkernel-based systems. *IEEE Trans. on Computers*, vol. 51 (2002)
- [Belinfante, 1999] Belinfante A., Feenstra J., de Vries R., Tretmans J., Goga N., Feijs L., Mauw S. et Heerink L.. – Formal Test Automation : a Simple Experiment. *In : 12th International Workshop on Testing of Communicating Systems.* – G. Csopaki et S. Dibuz et K. Tarnay (1999).
- [Fernandez, 1997] Fernandez J.-C., Jard C., Jéron T. et Viho C. – An Experiment in Automatic Generation of Test Suites for Protocols with Verification Technology. *Science of Computer Programming*, vol. 29, pp. 123–146 (1997).
- [Groz, 1999] Groz R., Jéron T. et Kerbrat A. – Automated test generation from SDL specifications. *In : SDL'99 The Next Millenium, 9th SDL Forum, Montréal, Québec*, éd. par Dssouli R., von Bochmann G. et Lahav Y., pp. 135–152. – Elsevier (1999).

- [ITU-T, 1992] ITU-T. – ITU-T: Specification and Description Language (SDL), ITU-T Recommendation Z.100, International Telecommunication Union, Geneva (1992).
- [Koopman, 1999]Koopman P. et DeVale J.. – Comparing the robustness of posix operating systems. *In: 29th International Symposium on Fault-Tolerant Computing Systems.* – Madison (WI) (1999).
- [Logrippo, 1991]Logrippo L., Faci M. et Haj-Hussein M. – An introduction to LOTOS: learning by examples. *Computer Networks and ISDN Systems*, vol. 23, n-25em.2ex_5, pp. 325–342 (1991).
- [Morel, 2000]Morel P. – *Une algorithmique efficace pour la génération automatique de tests de conformité.* – Thèse de PhD, UFR IFSIC/ lab IRISA (2000).
- [Phalippou, 1994]Phalippou M. – *Relations d'Implantation et Hypothèses de Test sur les Automates à Entrées et Sorties.* – Thèse de PhD, Bordeaux, (1994).
- [Phalippou, 1994bis]Phalippou (M.). – Test Sequence Generation using Estelle or SDL Structure Information. *In: Proceedings of FORTE/PSTV(Berne ,Switzerland)*, éd. par et S. Leue (D. H.), pp. 405–420 (1994).
- [Rabin, 1969]Rabin M. – Decidability of second-order theories and automata on infinite trees. *Trans. of Amer. Math. Soc.*, vol. 141, pp. 1–35 (1969).
- [Raymond, 1998]Raymond P., Weber D), Nicollin X. et Halbwachs N. – Automatic testing of reactive systems. *In: 19th IEEE Real-Time Systems Symposium.* – Madrid, Spain (1998).
- [Salona, 1993]Salona A., Vives J. et Gomez S. – An introduction to LOTOS (1993).
- [Tarjan]Tarjan R. – Depth first search and linear graph algorithms. *SIAM Journal on computing*, pp. 146–160, (1972).
- [Tretmans, 1992]Tretmans J. – *A Formal Approach to Conformance Testing.* – Thèse de PhD, Univ of Twente, Enschede, Th Netherlands (1992).
- [Tretmans, 1996]Tretmans J. – Test generation with inputs, outputs, and repetitive quiescence. *Software–Concepts and Tools*, vol. 17, pp. 103–120 (1996).

***UNE SOLUTION OPTIMALE POUR UN PROBLEME DE ROUTAGE DANS UN CANAL
DENSE AVEC CONTRAINTES DE CAPACITE***

Maria Zrikem

Equipe Combinatoire - Laboratoire CEDRIC,
CNAM 292 rue St-martin 75141 Paris cedex
zrikem@cnam.fr + 33 1 45 18 12 73

Résumé : Le routage est un problème de base dans la conception des circuits VLSI. Une instance du problème dans un canal dense consiste en une grille rectangulaire où tous les points des extrémités sont occupés par des terminaux. Ces derniers sont séparés en paires appelés « liaisons ». Les terminaux de chaque liaison appartiennent à deux extrémités différentes et doivent être reliés entre eux. Une solution (routage) est donnée par un ensemble de chemins connectant les différents terminaux en respectant la capacité donnée de chaque ligne et de chaque colonne.

Dans ce papier nous nous intéressons au problème de routage dans un canal dense avec des capacités paires sur les lignes et unitaires sur les colonnes. Nous l'appelons « un problème de multichemins ». Nous proposons un algorithme polynomial qui calcule des solutions optimales pour le problème et nous montrons que ces solutions sont toutes composées de plus courts chemins.

Summary : Channel routing is a basic problem in the design VLSI-circuits. An instance the problem in a dense channel consists of a rectangular grid-graph where all points of the lower and upper boundary are occupied by terminals. These are partitioned into pairs, called nets with one terminal on each boundary. These nets have to be connected. A solution (layout) is given by paths that connect corresponding grid points with respect of capacities given on each line and column of the grid.

In this note we deal with the routing problem in a dense channel with even capacities on lines and unit capacities on columns. We called it a « multipaths problem ». We propose a polynomial algorithm computing optimal solutions of the problem and prove that these solutions are all composed of shortest paths.

Mots clés : Optimisation Combinatoire, Multichemins, Grille, Routage.

Une solution optimale pour un problème de routage dans un canal dense avec contraintes de capacité

1 - INTRODUCTION

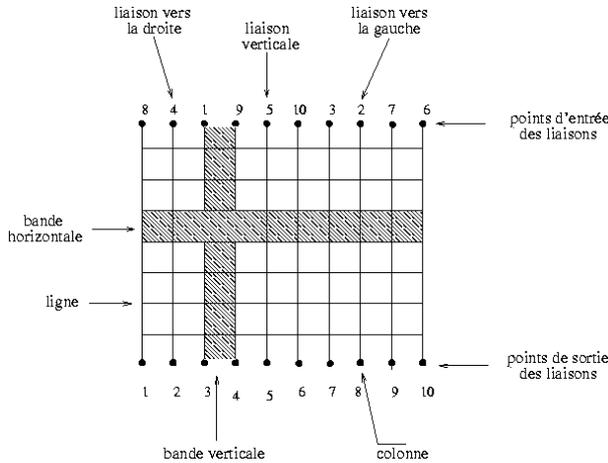


Figure 1 : Problème de multichemins dans une grille

Le problème de multichemins que nous traitons consiste en un ensemble de paires de sommets à relier sur une grille rectangulaire. Nous étudions en particulier sa résolutions dans une grille appelée par Forman, Wagner et Wagner (1993) « canal dense ». Chaque paire de sommets représente une liaison. Tous les tenants des liaisons sont situés sur l'une des deux extrémités de la grille et tous les aboutissants sur l'autre extrémité. De plus, les sommets des extrémités supérieure et inférieure sont tous occupés par un tenant ou un aboutissant (voir figure 1).

Chaque arête de la grille est munie d'une capacité fixe. Une solution du problème est donnée par un ensemble de chemins connectant les différents tenants et aboutissants, de longueur totale minimale et respectant les contraintes de capacité sur les arêtes de la grille.

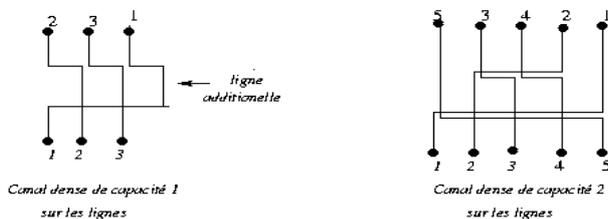


Figure 2 : Exemples de grilles avec différentes capacités

Le cas connu en littérature est celui d'une grille avec des capacités unitaires sur les arêtes. Dans ce cas, le problème devient un problème de multichemins arête-disjoints (voir figure 2). C'est un problème de base dans la conception des circuits VLSI. La plupart des travaux ont porté sur le problème de routage sans tenir compte de la longueur de la solution (Frank (1982), Gao, Kaufmann (1994), Kuchemn, Wagner, Wagner (1989),

Sarrafzadeh, Preparata (1985), Sarrafzadeh (1987a)). Peu de travaux ont pris en considération la longueur de la solution proposée : Sarrafzadeh (1987b) présente une borne de cette longueur. Kaufmann et Tollis (1988) proposent un algorithme calculant une solution de longueur totale minimale. Cependant les longueurs des solutions calculées par cet algorithme sur des exemples proposés par Sarrafzadeh (1987b), ont pu être améliorées plus tard par Forman, Dorothea Wagner et Frank Wagner (1993).

2 - DESCRIPTION DU PROBLEME

Nous présentons dans cette section certaines notions que nous utilisons dans le reste de ce papier. Le schéma représenté dans la figure 1 illustre toutes ces notions.

Grille

Une grille rectangulaire est déterminée par H lignes horizontales et V colonnes verticales. H est aussi dite la largeur de la grille.

On appelle bande verticale l'espace entre deux colonnes j et $j+1$. Elle est notée \vec{j} ou encore $\overleftarrow{j+1}$. De la même façon, l'espace limité par deux lignes est dit bande horizontale.

Liaison

Une liaison à router est définie par un tenant qu'on appellera *point d'entrée* et un aboutissant désigné par *point de sortie*. Une liaison l est donc donnée par la paire (E_l, S_l) , $1 \leq E_l \leq V$, $1 \leq S_l \leq V$ où E_l est la position du point d'entrée de la liaison sur la ligne supérieure et S_l est la position du point de sortie sur la ligne inférieure.

Une liaison est dite « vers la droite » (resp. « vers la gauche ») si $E_l < S_l$ (resp. $E_l > S_l$). Si $E_l = S_l$, on dit que la liaison l est « verticale ».

Une colonne j est simultanément le point d'entrée d'une liaison et le point de sortie d'une autre. Suivant que ces liaisons sont « vers la droite » ou « vers la gauche », j est de l'un des cinq types suivants (voir figure 3. La partie supérieure correspond au point d'entrée et la partie inférieure au point de sortie) :

- \lrcorner : si j est occupée par le point d'entrée d'une liaison
 \lceil
 «vers la gauche» et par le point de sortie d'une liaison
 «vers la gauche».
- \llcorner : si j est occupée par le point d'entrée d'une liaison
 \rfloor
 «vers la droite» et par le point de sortie d'une liaison
 «vers la droite».

- \lrcorner : si j est occupée par le point d'entrée d'une liaison «vers la gauche» et par le point de sortie d'une liaison «vers la droite».
- \llcorner : si j est occupée par le point d'entrée d'une liaison «vers la droite» et par le point de sortie d'une liaison «vers la gauche».
- \lrcorner : si j est occupée par les deux points d'entrée et de

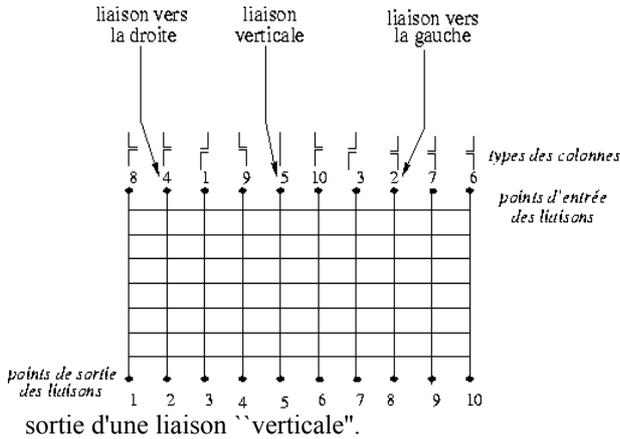


Figure 3 : Description d'un canal dense

Si on considère l'exemple de la figure 3, la colonne 2 est du type \llcorner car elle est occupée par le point d'entrée de la liaison 4 qui est «vers la droite», et le point de sortie de la liaison 2 qui est «vers la gauche».

Densité

On définit la densité locale d'une bande verticale \vec{j} comme étant le nombre de liaisons qui doivent traverser cette bande, on la note $d(\vec{j})$. La densité du problème est le maximum des densités locales $\max_j d(\vec{j})$ (pour des exemples voir figure 4). Une grille où toutes les liaisons sont verticales a donc une densité nulle.

La densité est au plus égale au nombre total des liaisons. Si les capacités sur les lignes sont unitaires, la densité d représente une borne inférieure de la largeur de la grille (nombre de lignes) nécessaire pour réaliser le routage, car la bande verticale de densité locale égale à d sera traversée par d liaisons.

Signalons aussi que dans le cas des capacités unitaires, il a été montré par Frank (1982) que le problème admet une solution si la largeur de la grille est au moins égale à la densité et si la grille a au moins une colonne additionnelle à droite ou à gauche.

Diagramme de densité

On désigne par un «segment unitaire horizontal» un segment horizontal de longueur égale à la largeur d'une bande. Le diagramme de densité est un diagramme obtenu en représentant les densités locales des bandes

verticales de la façon suivante : à chaque bande verticale de densité $2k$ sont associés k segments unitaires horizontaux superposés verticalement (voir figure 4). Nous dirons que ces segments sont «de flux 2», ils seront saturés dans la solution proposée dans la suite.

Le diagramme obtenu est composé de différents niveaux de densité où chaque niveau est constitué d'un ou plusieurs segments horizontaux. Un segment horizontal est la concaténation de plusieurs «segments unitaires horizontaux» (Voir figure 4).

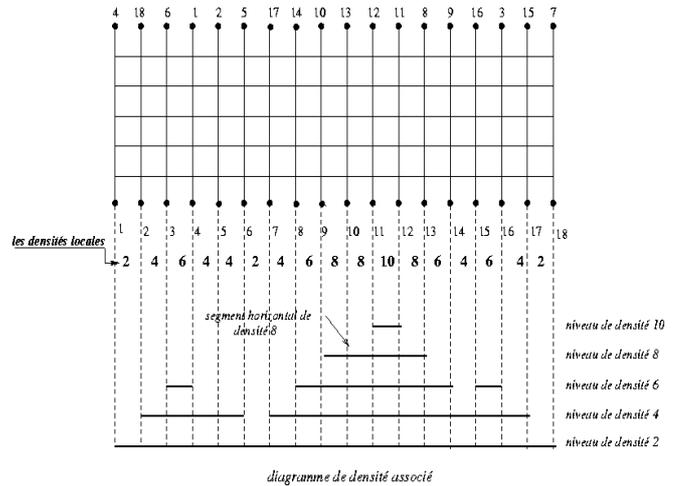


Figure 4 : Diagramme de densité associé à un problème de densité 10)

3 - GRILLE AVEC CAPACITES EGALES A 2 SUR LES LIGNES ET A 1 SUR LES COLONNES

Nous considérons dans ce paragraphe le cas d'une grille où les capacités sont égales à 2 sur toutes les lignes et à 1 sur les colonnes. Nous proposons un algorithme qui construit une solution optimale où chaque liaison est routée suivant l'un de ses plus courts chemins. La solution proposée sature toutes les arêtes horizontales de la grille correspondant au diagramme de densité. Les autres arêtes horizontales ne sont pas utilisées. Les arêtes verticales de la grille sont toutes utilisées entièrement. Autrement dit, le diagramme de densité nous donne l'allure de la solution qu'on va obtenir puisqu'il représente les arêtes horizontales qui vont être utilisées avec le flux sur chaque arête.

Nous allons voir dans la section 4 où nous traitons le cas général avec une capacité paire $c = 2p$ sur les lignes que les segments du diagramme de densité n'ont pas tous un flux égal $2p$ (ils ne sont pas tous saturés dans la solution proposée).

3.1 - Préliminaires

Les preuves de toutes les propriétés fournies dans cette section sont détaillées dans la thèse de Zrikem (2001).

Propriété 1 : La densité est toujours paire.

Dans la suite, si nous considérons une grille de densité $d = 2k$, nous utiliserons le nombre minimale de lignes nécessaires et qui est égal à $d/2 = k$. Nous supposons aussi sans aucune perte de généralité, qu'aucune liaison n'est «verticale».

Avant d'énoncer notre algorithme, nous avons besoin d'établir les deux propriétés suivantes concernant les segments du diagramme de densité.

Propriété 2 : Si \vec{j} est une bande verticale de densité $d(\vec{j})$, alors les bandes verticales adjacentes (à droite et à gauche) sont de densité $d(\vec{j})$ ou $d(\vec{j}) - 2$ ou $d(\vec{j}) + 2$ (voir figure 5).

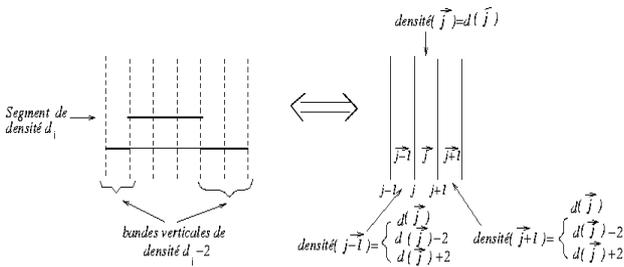


Figure 5 : Les segments de densité

Propriété 3 : Sur un segment de densité, la bande verticale de l'extrémité droite (resp. gauche) est du type \lrcorner (resp. \llcorner), les autres sont du type \lrcorner ou du type \llcorner (voir figure 6).

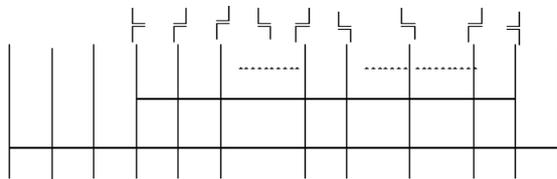


Figure 6 : Types des bandes verticales

3.2 - Un algorithme pour une solution optimale saturant le diagramme de densité

L'algorithme que nous proposons calcule une solution optimale où chaque liaison est routée suivant l'un de ses plus courts chemins. Les arêtes verticales ont une capacité égale à 1 et elles sont toutes utilisées dans la solution proposée. Si un canal est composé de H lignes, tous les plus courts chemins d'une liaison donnée comportent exactement $H+1$ arêtes verticales de la grille. Cela vient du fait que pour descendre de l'extrémité supérieure vers l'extrémité inférieure de la grille, il faut absolument traverser les H lignes de la grille (voir figure 7).

En ce qui concerne les arêtes horizontales dont la capacité est égale à 2, l'algorithme propose une solution qui sans créer de détours, sature les segments horizontaux du diagramme de densité. Les autres arêtes horizontales de la grille ne sont pas utilisées. La figure 7 schématise des chemins avec détours.

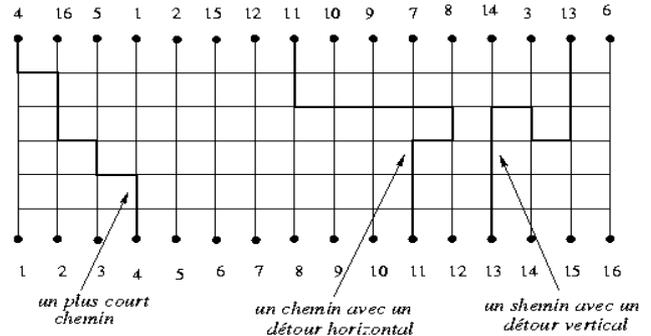


Figure 7 : Exemples de chemins sans ou avec détours

L'algorithme est basé sur le principe suivant :

On sature les niveaux de densité du diagramme un par un, du plus haut au plus bas.

– On sature un par un les segments de chaque niveau de la droite vers la gauche.

– Pour chaque segment $[a,b]$ du niveau courant on procède à :

1. Une phase de remplissage de droite à gauche qui consiste à «tirer» vers la gauche certaines liaisons du type « liaisons vers la gauche ». A la fin de cette phase, une unité de capacité du segment $[a,b]$ (la moitié) sera utilisée (figure 8).

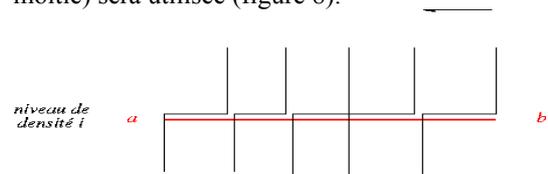


Figure 8 : Demi saturation du segment de densité $[a,b]$ à la fin d'une phase de remplissage de droite à gauche

2. Une phase de remplissage de gauche à droite qui est la symétrique de la première. Elle consiste à «tirer» vers la droite certaines liaisons du type « liaisons vers la droite » et permet de saturer entièrement le segment courant (figure 9).

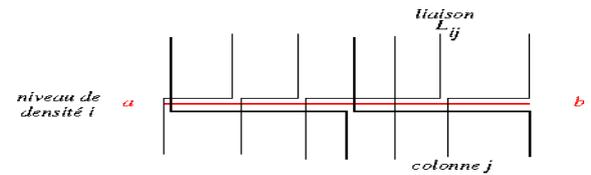


Figure 9 : Saturation entière à la fin d'une phase de remplissage de gauche à droite

Ainsi on peut dire que dans cet algorithme, à chaque étape i nous saturons le niveau de densité d_i , et nous obtenons un problème résiduel de densité $d_i - 2$.

En chaque point (colonne) j d'un segment de densité $[a,b]$ sur un niveau i , on a une liaison qui « arrive » verticalement du niveau précédent ; on note le numéro de cette liaison $L_{i,j}$ (voir figure 10).

Étant donné un canal dense de densité $d=2k$ et de n liaisons, les étapes de cet algorithme sont décrites par le pseudo-code qui suit :

Procédure min-cap-multichemins

Router toutes les liaisons vers le bas d'un segment unitaire vertical;

Pour j de 1 à n **Faire**

$L_{k,j} \leftarrow j$;

/ au niveau initial k , chaque liaison j « arrive » verticalement sur la colonne j */*

FinFaire

*/*descendre d'une arête verticale*/*

Pour i de k à 1 avec un pas -1 **Faire**

*/*Pour chaque niveau de densité du haut en bas*/*

Pour tout segment $[a,b]$ du niveau i **Faire**

Remplissage_droite_a_gauche($[a,b],i$);

*/*saturer le segment à moitié */*

Remplissage_gauche_a_droite($[a,b],i$);

/ saturer le segment entièrement */*

FinFaire

Router toutes les liaisons vers le bas d'un segment unitaire vertical;

Pour tout point intermédiaire du niveau i **Faire**

/ point où la liaison qui "arrive" verticalement n'a été tirée ni à droite ni à gauche*/*

$L_{i-1,j} \leftarrow L_{i,j}$;

FinFaire

Finfaire

FinProcédure

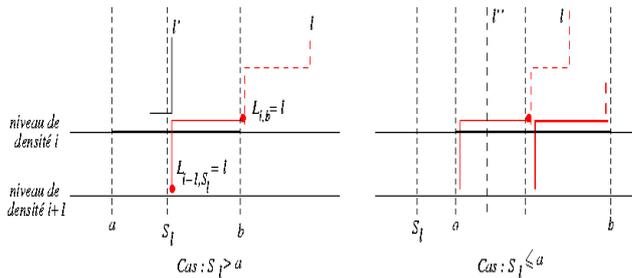


Figure 10 : Routage des liaisons dans une phase de remplissage de droite à gauche

La procédure `remplissage_gauche_a_droite()` est la symétrique de la procédure `remplissage_droite_a_gauche()`. Nous présentons uniquement le pseudo code de la deuxième procédure qui est illustrée par la figure 10.

Procédure `Remplissage_droite_a_gauche($[a,b],i$)`

$l \leftarrow L_{i,b}$; */* liaison qui "arrive" verticalement en b */*

$a_est_atteint \leftarrow Faux$;

Tant que `non($a_est_atteint$)` **Faire**

Si $S_i > a$ **Alors** */* on n'a pas atteint a (cas1)*/*

/ S_i point de sortie de l est rencontré avant a */*

Router la liaison l horizontalement vers la gauche jusqu'à S_i ;

$L_{i-1,S_i} \leftarrow l$;

/ l « arrive » verticalement au niveau $i-1$ en colonne S_i */*

$l \leftarrow L_{i,S_i}$;

*/*On considère la nouvelle liaison qui « arrive » horizontalement au niveau i au point S_i */*

Sinon */* a est atteint : cas2 */*

Tirer la liaison l horizontalement vers la gauche jusqu'en a ;

$L_{i-1,a} \leftarrow l$;

/ l "arrive" verticalement au niveau $i-1$ en colonne a */*

$a_est_atteint \leftarrow Vrai$;

Finsi

Finfaire

FinProcédure

La procédure `remplissage_gauche_a_droite()` est obtenue en échangeant partout a et b ainsi que droite et gauche, le test «si $S_i > a$ » étant remplacé par «si $S_i < b$ ».

Un exemple

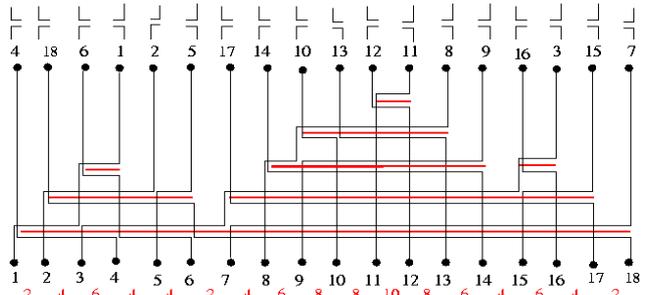


Figure 11 : Le routage obtenu pour un problème de densité 10

Nous présentons dans la figure 11 la solution obtenue par notre algorithme pour la grille schématisée dans la figure 4. La densité étant égale à 10, seulement 5 lignes sont utilisées

3.3 - Preuve de l'algorithme

Nous discutons dans cette section la validité de l'algorithme que nous avons proposé dans la section précédente. Cette discussion s'intéresse uniquement à la procédure de `remplissage_gauche_a_droite()`. Nous montrons qu'à chaque étape de cette procédure il est possible de router les liaisons considérées sans créer de détours ni verticaux ni horizontaux. Ensuite, nous établissons qu'à la fin de cette procédure, le segment de densité concerné est saturé.

Le routage est toujours possible

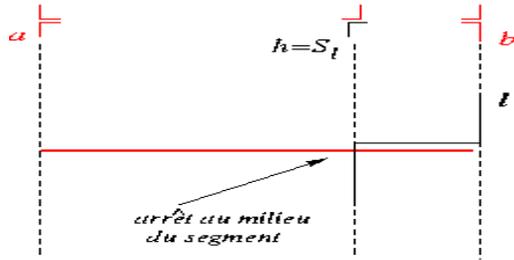


Figure 12 : Routage toujours possible sur un segment de densité

L'algorithme tel qu'il est décrit ci-dessus permet que le routage des liaisons soit toujours effectué. En effet :

– On a vu précédemment que l'extrémité droite de chaque segment de densité est du type \downarrow . Ainsi, il est

toujours possible de tirer vers la gauche la première liaison « vers la gauche » située sur la colonne b . Quand l'algorithme arrête la phase de remplissage avant l'extrémité du segment courant (cas 1) à la colonne h , pour que le routage ne se bloque pas, la colonne h doit être du type \downarrow . Lorsque l'algorithme

arrête la phase remplissage avant l'extrémité du segment, la liaison courante l , qui est « vers la gauche », a donc son point de sortie sur la colonne $h = S_l$. Par la suite cette colonne est du type \downarrow (voir

figure12). Or, toute liaison au milieu d'un segment de densité ne peut être que du type \downarrow ou du type \uparrow .

Donc la colonne h est nécessairement du type \downarrow , ce

qui permet de continuer le routage là où la liaison l s'est arrêtée.

Saturation de chaque segment de densité

Dans une phase de remplissage de droite à gauche, on commence à partir de l'extrémité droite et on tire continûment vers la gauche jusqu'à atteindre l'extrémité gauche du segment courant. Chaque liaison continue là où la précédente s'est arrêtée. De ce fait, à la fin de cette phase le segment est saturé à moitié.

Le même principe permet que le segment de densité soit entièrement saturé à la fin d'une phase de remplissage de gauche à droite.

Pas de détours

L'algorithme proposé ne crée pas de détours dans les chemins calculés. En effet :

– les liaisons ne sont jamais tirées vers le haut. Ceci permet d'éviter les détours verticaux.

– toute liaison du type « liaison vers la droite » est toujours tirée vers la droite et toute liaison du type « liaison vers la gauche » est toujours tirée vers la gauche. Ceci permet d'éviter des détours horizontaux.

Donc tous les chemins sont de longueur minimale.

Propriété de persistance

A chaque itération i de la procédure *min-cap-multichemins*, après la saturation du niveau i de densité

$d_i = 2i$, on obtient un problème résiduel de multichemins de longueur totale minimale dans une grille de densité $d_i - 2$ (voir figure 13)

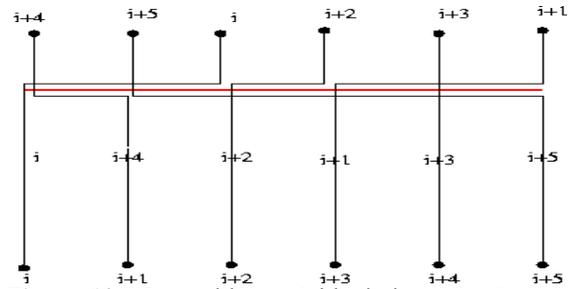


Figure 13 : Le problème résiduel obtenu après une itération

Après l'itération i , les segments du niveau i sont saturés. Chaque bande verticale de ces segments a été traversée par deux liaisons (une de droite à gauche et l'autre de gauche à droite) qui ne vont plus la traverser dans les étapes suivantes du fait que l'algorithme ne génère pas de détours. Les bandes verticales qui ne sont pas couvertes par les segment du niveau i , ont des densités locales inférieures ou égale à $d_i - 2$ (propriété2). Ces densités ne changent pas après l'itération i puisque les liaisons de chaque segment du niveau i n'avancent pas au delà de ce segment. Donc, le problème résiduel obtenu a une densité égale à $d_i - 2$.

De plus, il y a un point d'entrée et un point de sortie sur chaque colonne et la position des points de sortie est invariante. Les valeurs $L_{i-1,j}$, fournissent les nouveaux points d'entrée. On a correctement calculé la valeur $L_{i-1,j}$ pour chaque $j=1, \dots, n$. En effet, dans les deux procédures de remplissage, le résultat de ce calcul consiste à permuter entre elles les colonnes j où les liaisons sont tirées à droite ou à gauche. Dans la procédure principale, ce calcul consiste à garder la même position pour les colonnes intermédiaires (la permutation identité).

A l'étape finale, le problème résiduel est de densité $0 = d - \underbrace{2 - 2 - \dots - 2}_{d/2 \text{ fois}}$ et toutes les liaisons sont

verticales. Alors chaque point d'entrée a atteint le point de sortie qui lui correspond.

3.4 - Complexité de l'algorithme

L'algorithme que nous proposons a un temps d'exécution proportionnel à la superficie de la section du routage : soit à $(n - 1) * d/2$, où n est le nombre des liaisons et d la densité de la grille (i.e. $d/2$ est le nombre de lignes).

A chaque itération, dans les procédures *Remplissage_droite_à_gauche([a,b],i)* et *Remplissage_gauche_à_droite([a,b],i)*, la boucle « Tant que » est itérée au maximum $b-a$ fois (longueur du segment $[a,b]$), puisque partant de la colonne b en se déplace au moins d'une

arête à chaque passage jusqu'à atteindre la colonne a . Or, $(b - a) \leq n$, donc la boucle « *Tant que* » est en $O(n)$. Comme la procédure principale est effectuée $k = d/2$ fois, on peut conclure que l'algorithme que nous proposons est en $O(nd)$.

4 - GENERALISATION AU CAS D'UNE GRILLE AVEC DES CAPACITES PAIRES ≥ 2 SUR LES LIGNES ET EGALE A 1 SUR LES COLONNES

Les résultats et l'algorithme proposés dans le paragraphe précédent peuvent être généralisés à une grille avec des capacités unitaires sur les colonnes et des capacités paires sur les lignes d'une valeur $c = 2p \geq 2$.

Notons que dans ce cas, si la densité est d alors le nombre de lignes nécessaires H vaut d/c si $\text{mod}(d/c) = 0$ ($\text{mod}()$: reste de la division entière), sinon, il vaut $d/c + 1$.

Le cas général admet aussi une solution optimale formée des plus courts chemins. Pour obtenir cette solution, nous procédons de la façon suivante :

4.1 - Du cas général au cas simple

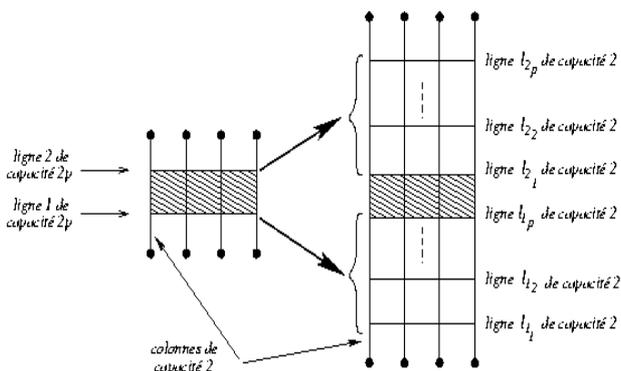


Figure 14 : Du cas général au cas simple

Soit une grille de densité d composé de H lignes, où la capacité sur chaque ligne est égale à $2p$ et est unitaire sur chaque colonne. Nous transformons cette grille en éclatant chaque lignes i (de capacité $2p$) en p lignes i_1, i_2, \dots, i_p , chaque ligne étant de capacité 2. La figure 14 illustre cette transformation.

De cette façon, nous obtenons une grille de même densité d et ayant $H.p$ lignes de capacité 2. La capacité des colonnes est unitaire. Nous pouvons donc appliquer l'algorithme *min-cap-multichemins* pour obtenir une solution optimale qui route chaque liaison sur un plus court chemin.

4.2 - Vers une solution du cas général

Pour déduire une solution pour le cas général (grille initiale), nous proposons de regrouper dans la grille obtenue chaque paquet de lignes $\{i_1, i_2, \dots, i_p\}$

($1 \leq i \leq H$) et de le remplacer par une seule ligne i .

Ceci revient à annuler dans la solution du cas simple toutes les transitions verticales effectuées par les liaisons au niveau des paquets $\{i_1, i_2, \dots, i_p\}$, $1 \leq i \leq H$ (voir figure 15).

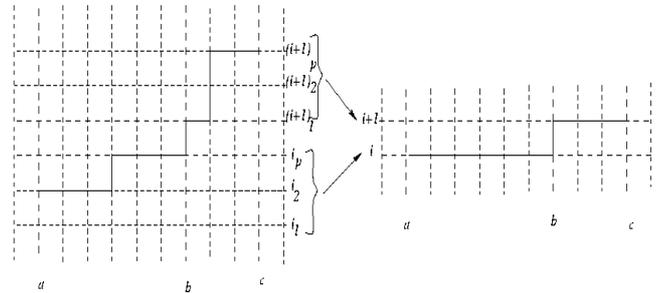


Figure 15 : Déduction d'une solution du cas général

Nous avons montré dans Zrikem (2001) que la solution ainsi obtenue est une solution admissible pour le cas général et qu'elle est sans détours.

5 - CAS DES CAPACITES IMPAIRES SUR LES LIGNES

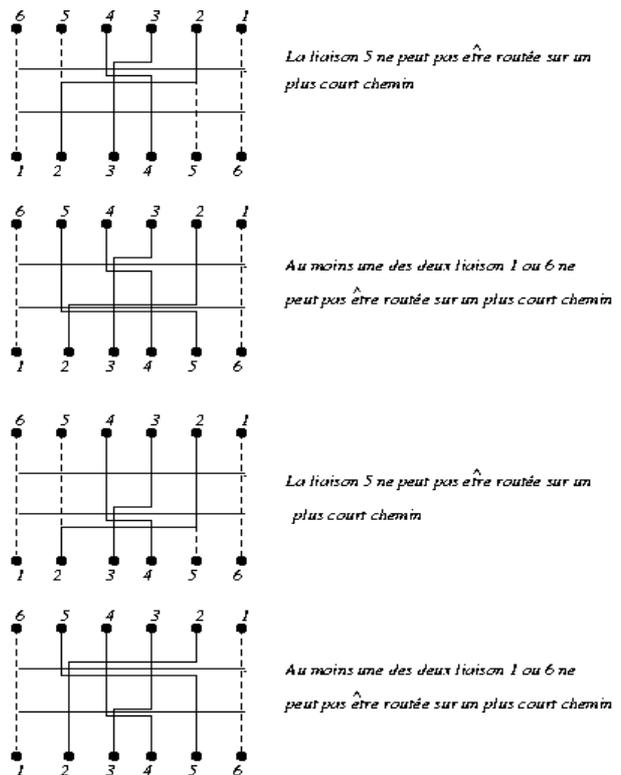


Figure 16 : Exemple n'admettant aucune solution des plus courts chemins

Contrairement au cas d'une grille avec des capacités paires sur les lignes, le problème de multichemins avec longueur totale minimale dans une grille avec des capacités impaires sur les lignes et unitaires sur les colonnes, n'admet pas toujours une solution optimale des plus courts chemins.

Le cas d'un canal dense avec des capacités unitaires sur les lignes et les colonnes admet une solution si le canal a au moins une colonne additionnelle tout à fait à droite ou tout à fait à gauche (d'après un résultat de Frank (1982)). Cette solution peut contenir des détours. En ce qui concerne le cas général avec des capacités impaires >1 sur les lignes, nous donnons dans la figure 16 un exemple de capacité égale à 3 sur les lignes pour lequel toutes les solutions qui routent les liaisons sur des plus courts chemins sont non admissibles : dans cet exemple, pour respecter les contraintes de capacité égales à 1 sur les colonnes, la liaison 3 et la liaison 4 doivent se croiser horizontalement sur la même ligne, et également la liaison 2 avec 5 et 1 avec 6. Ainsi, on peut énumérer quatre cas de figures. Dans chaque cas, au moins une liaison ne peut pas être routée sur un plus court chemin.

6 - CONCLUSION

Nous avons considéré dans ce travail un problème de multichemins dans une grille sous contraintes de capacité et de longueur totale minimale.

Nous avons proposé un algorithme linéaire en la taille de la grille qui résout le problème si les capacités sont unitaires sur les colonnes et paires sur lignes. Cet algorithme route chaque liaison sur un plus court chemin.

Nous avons montré également sur un exemple que le cas des capacités impaires sur les lignes n'admet pas toujours une solution des plus courts chemins. Nous essayons actuellement de voir si l'algorithme proposé dans [Forman, 1993] pour le cas des capacités unitaires sur les lignes et sur les colonnes peut être généralisé au cas des capacités impaires quelconques sur les lignes.

7 - BIBLIOGRAPHIE

- Forman M., Wagner D. Wagner F. (1993), "Routing through a dense channel with minimum total wire length". *Journal of Algorithms*, Vol. 15, p. 267-283 (1993).
- Frank A. (1982), "Disjoint paths in a rectilinear grid". *Combinatorica*, Vol. 2, p. 361-371 (1982).
- Frank A. (1990), "Paths, flows, and VLSI-layout, chap. Packing paths, circuit, and cuts: a survey". Springer Verlag ED, p. 47-1000 (1990).
- Gao S., Kaufmann M. (1994), "Channel routing of multiterminal nets". *Journal of the ACM (JACM)*, Vol. 41, N. 4, p. 791-818 (1994).
- Kaufmann M., Tollis I. G. (1988), "Channel routing with short wires". *Proceedings Aegean Workshop on computing (AWOC'88)*, Lecture notes in Computer Science, Springer -Verlag ed. Vol. 319, p. 226-236, New York/Berlin (1988).
- Kuchem R., Wagner D., Wagner F. (1989), "Atrea optimal three-layer channel routing". *Proceeding of 30th Annual symposium foundations of computer science (FOCS'89)*, p. 506-5011 (1989).
- Sarrafzadeh M., Preparata F. P. (1985), "Compact channel routing of multiterminal nets". *Ann. Discrete Math* Vol. 25, p. 255-280 (1985).
- Sarrafzadeh M. (1987a), "On the complexity of the general channel routing problem in the Knock-knee mode". *IEEE Transactions, Computer-Aided Design Integrated Circuits Systems*, Vol. 4, p. 503-506 (1987).
- Sarrafzadeh M. (1987b), "Channel routing with provably short wires", *IEEE Transactions, Circuits systems*, Vol. 34, p. 1133-1135 (1987).
- Zrikem M. (2001), "Optimisation du routage de câbles dans les installations de productions d'électricité". Thèse de doctorat, Département d'informatique, CNAM, Paris (2001).

***VERS UNE ARCHITECTURE MULTI-AGENTS POUR DU CLUSTERING DYNAMIQUE D'UNE
POPULATION D'AGENTS***

Roland Coma,

Doctorant en informatique spécialité intelligence artificielle
roland.coma@univ-lehavre.fr, +33 2 32

Adresse professionnelle

Laboratoire d'Informatique du Havre (LIH)
25 rue Philippe Lebon BP 540 ★ 76058 Le Havre

Résumé : Les travaux présentés dans cet article concernent l'analyse de populations d'agents par clustering dynamique. Les agents sont des entités dynamiques, et leurs propriétés évoluent sans cesse. La principale difficulté est de trouver une technique de clustering applicable à des agents et capable de réaliser un clustering au fur et à mesure de l'arrivée des agents (aspect incrémental), mais également capable de prendre en compte les modifications des données déjà clusterisées (aspect dynamique des clusters).

Summary : The works presented in this article concern the analysis of populations of agents by dynamic clustering. The agents are dynamic entities, and thus their properties evolve ceaselessly. The main difficulty is to find a technique of clustering applicable to agents and capable of realizing a clustering according to the arrival of the agents (incremental aspect), but also able to take into account the modifications of the data already clustered (dynamic aspect of clusters).

Mots clé : clustering dynamique, système multi-agents.

Vers une architecture multi-agents pour du clustering dynamique d'une population d'agents

L'objectif est de mettre en oeuvre des techniques de clustering dynamiques afin de mettre en évidence l'organisation d'une population d'agents par des groupes d'agents jugés similaires. Une des spécificités du problème posé est la dynamique d'un système multi-agents. Cette dynamique se traduit par deux points importants: l'évolution des agents et l'arrivée de nouveaux agents. Les algorithmes de clustering classiques ne prennent en compte que des données statiques, même si certains (K-means séquentiel, FC) intègrent l'aspect incrémental. Nous cherchons donc des algorithmes de clustering classiques qui peuvent être utilisés ou adaptés pour résoudre ce problème. Ce problème est apparu dans le cadre du développement d'un outil de veille préventive basé sur un système multi-agents Boukachour et al. (2002). Après avoir développé une couche fondée sur le paradigme des agents aspectuels permettant de représenter les différents aspects de la situation, décrit par **Error! Reference source not found.** Durant (1999), nous cherchons à classer les aspects de la situation courante afin d'analyser et de prédire la suite des événements. Pour cela, nous avons besoin d'une technique de clustering, mais le problème est que les données fournies par la couche inférieure sont dynamiques. Dans un premier temps, nous allons généraliser ce problème en considérant une population d'agents quelconque. Puis, nous décrirons plus en détail l'aspect dynamique du problème. Pour finir, nous proposerons une architecture multi-agents pour résoudre ce problème, ainsi qu'un protocole de test permettant de valider cette architecture.

1 - GENERALISATION

Dans le domaine du clustering classique, on manipule des objets et des attributs. Les objets sont des vecteurs d'attributs et les attributs sont les données du problème. L'objectif est d'obtenir des regroupements d'objets en clusters. Un cluster est un groupe d'objets jugés similaires. La quasi totalité des algorithmes de clustering classiques utilisent un calcul de distance pour évaluer la proximité ou la dissimilarité entre les différents objets qui deviennent alors des points d'un espace à n dimensions, n étant le nombre d'attributs des objets. Dans cet article, nous nous intéressons au clustering d'agents. Quels vont donc être dans ce contexte les données de l'algorithme de clustering?

On suppose que chaque agent de la population d'agents à catégoriser met à disposition un ensemble de caractéristiques permettant de décrire son état courant. Par conséquent, par rapport au clustering classique, les objets sont les ensembles de caractéristiques des agents, leurs attributs sont les différentes caractéristiques de chaque agent, ce qui permet d'utiliser les mêmes calculs de distance que dans le clustering classique. Au sein d'une population d'agents, les caractéristiques de chaque agent évoluent sans cesse : c'est le problème de l'évolution des données. De plus de nouveaux agents peuvent apparaître avec leurs propres caractéristiques : c'est le problème de l'arrivée de nouvelles données. Le but est que le regroupement des agents en clusters soit pertinent à chaque instant. Les caractéristiques de ces agents sont les données du clustering, l'évolution et l'arrivée de ces nouveaux agents va donc modifier les regroupements effectués précédemment.

1.1 - Arrivée de nouvelles données

Lorsque les données à clusteriser arrivent au fur et à mesure on peut parler de clustering dynamique, dans le sens où les données arrivent dynamiquement : il s'agit d'un clustering incrémental. Il est à noter que certains algorithmes comme celui proposé par Barbara (2000) gèrent des flux de données, le problème de l'arrivée de nouvelles données est le même pour nous. Dans le clustering classique le cardinal de l'ensemble d'objets est fixe, mais dans notre cas de nouveaux agents peuvent apparaître ou disparaître. En d'autres termes, $card(A)$ n'est pas fixe et évolue au cours du temps. On peut donc considérer que $card(A)=f(t)$.

1.2 - Évolution des données

Afin de mieux comprendre ce problème, prenons l'exemple d'une personne qui veut trier un tas de papiers administratifs, sans savoir à l'avance quelles sont les différents types de document. Cette personne doit alors examiner les documents et faire des tas. Supposons maintenant que les documents se modifient même une fois qu'ils sont déposés dans un des tas. Ce type de problème nous montre bien un aspect dynamique au sens où les données elles mêmes sont dynamiques.

Dans le clustering classique, les objets sont représentés par des vecteurs d'attributs (valeurs), mais les valeurs de ces attributs sont statiques. Par contre,

dans notre cas, un objet est l'ensemble des caractéristiques d'un agent et ses attributs sont les différentes caractéristiques de cet agent. De plus les valeurs de ces attributs peuvent varier au cours du temps. On a donc un ensemble d'agents A et un ensemble de caractéristiques C où :

-Chaque agent a le même nombre de caractéristiques n donc $card(C)$ est fixe,

-Chaque agent est représenté par un vecteur de caractéristiques donc $\forall a \in A$ on a $a \in C^n$ ou encore $a = (c_1, c_2, \dots, c_n), \forall c_i \in C$,

On peut donc de représenter les valeurs des attributs comme des fonctions du temps ainsi :

$$a = (c_1(t), c_2(t), \dots, c_n(t)) \quad (1)$$

Remarque : Le terme de "clustering dynamique" est ambiguë car il peut signifier que les données arrivent dynamiquement et nous sommes alors dans le cas d'un clustering d'un flux de données, ou encore que les données à clusteriser peuvent être modifiées au cours du temps même si elles sont déjà clusterisées. Pour un clustering d'agent, nous avons les deux problèmes en même temps.

2 - METHODES DE CLUSTERING CLASSIQUES

Le clustering Kamber et Han (2001, chap 8) est un mécanisme permettant de classer **intelligemment** des données dans le but de prédire des phénomènes, d'analyser des données expérimentales ou encore de travailler sur des classes de données plutôt que sur des données en très grand nombre. L'analyse par classes est une méthode importante dans l'apprentissage humain. Les enfants commencent par reconnaître les objets en les assimilant à des groupes. Le clustering permet de repérer les régions les plus denses ou encore de reconnaître des formes et de faire des liens pertinents entre les différents objets en fonction de leur attributs. Certains mécanismes font appel à des techniques de classification comme l'étude de marché des consommateurs ou en biologie avec la classification des espèces animales ou végétales.

Les algorithmes utilisent des mesures de distance pour évaluer la similarité entre les objets. La plus connue est la distance euclidienne.

2.1 - Les catégories de clustering

Il existe de nombreux algorithmes de clustering qui peuvent être classés dans les catégories suivantes :

- Méthodes par partitions,
- Méthodes hiérarchiques,
- Méthodes basées sur la densité,
- Méthodes basées sur un modèle,
- Méthodes basées sur les fourmis.

Certains algorithmes intègrent les idées de plusieurs catégories, il n'est donc pas simple de les classer parmi une de ces catégories.

Bilan : Il existe de nombreuses catégories de clustering. Malheureusement les méthodes plutôt prévues pour des données statiques ne sont pas bien adaptées ou difficilement adaptables aux problèmes posés par les aspects dynamiques de l'analyse d'une population d'agents. Seul les algorithmes fournis semblent prometteurs. Nous nous sommes donc plus particulièrement intéressés à ce type d'algorithmes, et plus particulièrement à **ANTCLASS** proposé par Monmarché, Sliman et Venturini (1999) présenté dans la section suivante.

3 - ANTCLASS

Jusqu'ici, nous avons cherché des algorithmes de clustering classique qui pourraient répondre au problème posé par l'aspect dynamique des données. Nous nous sommes rendus compte que l'on pouvait examiner d'autres approches comme l'hybridation de plusieurs algorithmes. Nous nous sommes donc intéressés à **AntClass** qui mélange deux algorithmes dont un algorithme de type fourmis. Ici, nous allons uniquement présenter le principe de fonctionnement de cet algorithme.

3.1 - Principe

Cet algorithme se décompose en quatre étapes :

- un algorithme de type fourmis classique pour constituer les classes initiales,
- un algorithme ACM (Algorithme des Centres Mobils) pour corriger les erreurs du premier,
- un algorithme fourmis sur les classes (clusters) proposés par l'ACM,
- un algorithme ACM pour corriger les erreurs subsistantes après l'étape précédente.

La première étape fournit une classification pertinente mais avec un certain nombre d'erreurs et d'**objets libres** dû au effet du problème de convergence des algorithmes de fourmis de classification. La deuxième permet de corriger ces erreurs mais laisse un trop grand nombre de classes. La troisième qui est la même que la première mais appliquée sur les classes permet de réduire le nombre de classes en les regroupant. Enfin, la quatrième est là pour corriger une nouvelle fois les erreurs de la précédente.

3.2 - Discussion

Rappelons que dans notre problème, les données (les caractéristiques des agents) évoluent et de nouvelles données peuvent apparaître. Ce qui n'est à priori pas le cas pour AntClass.

Avantages :

–Cela paraît intéressant de combiner des algorithmes de clustering pour atténuer les inconvénients rencontrés,
–l'utilisation de l'algorithme des fourmis paraît répondre aux contraintes dynamiques de nos données. Les fourmis se déplaçant en permanence, il est possible de prendre en compte l'arrivée de nouveaux objets. Il suffit de les placer aléatoirement sur la grille. On peut penser également que le problème de l'évolution des données peut être directement pris en compte par cet algorithme. Il suffit qu'une fourmi détecte que l'objet qui a évolué est devenu trop éloigné du centre du tas auquel il appartient (trop dissimilaire). Un autre algorithme qui interviendrait à intervalle de temps régulier corrigerait les erreurs et se chargerait des nouvelles données.

Inconvénients :

Comment déterminer les critères d'arrêt et de reprise des différents algorithmes ?

Quand appliquer le second algorithme (non fourmis) ?

Les expérimentations montrent que les algorithmes fourmis fournissent une base pour répondre aux contraintes dynamiques. Les problèmes de convergence de ce type d'algorithmes obligent à envisager une combinaison avec un autre algorithme. Les auteurs d'«AntClass» ont choisi l'ACM qui présente l'inconvénient d'être statique. Nous nous orientons plutôt vers une seconde couche plus dynamique reposant sur des agents présentée dans la section suivante.

4 - ARCHITECTURE PROPOSEE

Le but de l'architecture proposée est de mettre en oeuvre un clustering dynamique tenant compte des modifications des caractéristiques des agents au cours du temps et de l'arrivée de nouveaux agents. Cette architecture repose d'une part sur une version modifiée de l'algorithme fourmis de **AntClass** et d'autre part sur une seconde couche d'agents interagissant avec les fourmis. Le rôle de cette couche d'agents est d'améliorer la convergence des fourmis.

Nous présentons le rôle et le fonctionnement des différents composants de l'architecture. Cette architecture multi-agents permettant de gérer de la même façon les nouveaux arrivants et les modifications des anciens. Nous allons donc décrire précisément les différents composants schématisés dans la figure suivante (cf. fig. 1) nous allons également décrire un protocole de tests pour valider notre architecture.

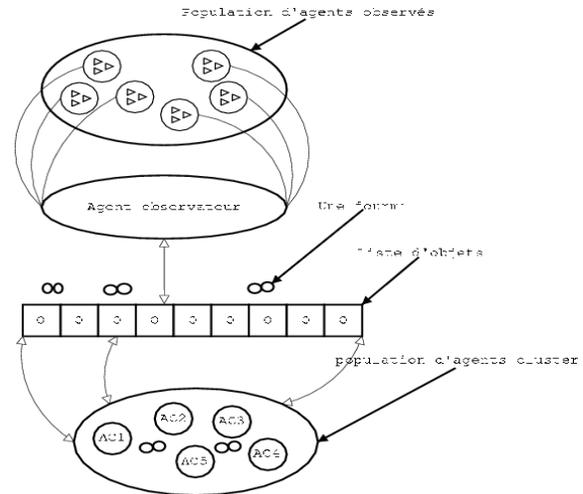


Figure 1: Le principe de notre architecture

Les différents composants de cette architecture sont :

- l'agent observateur,
- les fourmis,
- les agents cluster,
- la liste d'objets,

Une des contraintes ayant guidé la conception de cette architecture est de rendre indépendant le fonctionnement de l'organisation d'agents de bas niveau analysée et celui de la couche de clustering.

4.1 - L'agent Observateur

L'agent observateur scrute en permanence la couche d'agents de bas niveau à analyser. Lorsqu'il détecte un nouvel agent, il crée un objet contenant uniquement ses caractéristiques observables et le place dans une liste. Si les caractéristiques de cet agent changent, l'agent observateur est capable de le détecter et de modifier en conséquence l'objet qu'il a créé. C'est la passerelle d'information entre la couche d'agents à analyser et la couche qui réalise cette analyse par clustering.

Dans notre plate-forme, il gère un fichier scénario et se charge de mettre à jour les objets ou de créer de nouveaux objets. Lorsque l'utilisateur demande une classification, il génère un fichier ou une base de données contenant l'état des objets au moment où l'utilisateur a demandé sa classification. Cette étape est nécessaire pour notre protocole de tests que nous détaillerons plus tard.

4.2 - Les fourmis

Leur rôle est capital. Elles font des va-et-viens entre la liste des objets et les agents cluster. Pendant ces allers et retours, elles peuvent déposer des objets dans les agents cluster et créer de nouveaux agents cluster.

Toutefois, il est à noter que ces fourmis ont surtout un rôle transport afin de soulager le travail des agents cluster.

Ce sont elles qui réalisent le clustering. Leur comportement s'inspire de celui proposé dans **AntClass**. La principale différence réside dans le fait qu'elles ne se déplacent plus sur une grille, mais sur une liste d'objets.

Dans **AntClass**, les fourmis se déplacent aléatoirement, il est donc possible qu'elles ne passent pas sur les bons tas au bon moment. Pour limiter ce phénomène, nous obligeons les fourmis à parcourir tous les clusters avant de créer un nouveau cluster. Cependant, pour éviter que les fourmis se dirigent au même endroit en même temps, ce parcours est aléatoire : à chaque visite d'un cluster, la fourmi le marque comme visité. De plus, en procédant ainsi le nombre de clusters à un instant donné sera borné par le nombre de cluster réel. Autrement dit, nous cherchons à limiter le nombre de tas pouvant être fusionnés.

Déplacement d'une fourmi de la liste vers les agents cluster

Lorsqu'une fourmi arrive sur la liste, elle ramasse aléatoirement un objet dans la liste. Puis elle se dirige vers les agents cluster pour essayer de déposer l'objet dans un des clusters. Une fourmi qui va sur la liste ne doit pas porter d'objet.

Déplacement d'une fourmi des agents cluster vers la liste

Lorsqu'une fourmi arrive sur la liste des agents cluster, elle choisit au hasard un agent cluster et vérifie si elle peut y déposer l'objet qu'elle transporte. Si elle ne peut pas, elle en choisit un autre, et ainsi de suite. Si elle n'a pas déposé son objet dans un des agents cluster, elle crée un nouvel agent cluster contenant cet objet avant de retourner sur la liste.

4.3 - Les agents cluster

Un agent cluster représente un cluster. Son rôle est de vérifier la pertinence de son cluster. En particulier, il doit détecter les modifications des données appartenant à son cluster. Si un objet devient trop éloigné du centre du cluster, l'agent le rejette immédiatement dans la liste des objets, où les fourmis peuvent le ramasser comme s'il s'agissait d'un nouvel objet. Le but est de détecter les modifications de données aussitôt, c'est pourquoi nous faisons appel à des fourmis pour transporter les objets et ne pas surcharger le travail des agents cluster. On peut considérer qu'un agent cluster a un comportement régi par un automate simple à deux états.

—état 1 : il vérifie que tous les objets qu'il contient sont similaires,

—état 2 : il recalcule son centre, après avoir rejeté ou reçu un objet.

Pour juger si un objet est dissimilaire, on utilise une distance et un seuil de similarité, tout comme dans Monmarché,sliman et Venturini (1999). Le but des agents clusters est de maintenir en permanence un **cluster sain**, c'est-à-dire qui ne contient pas d'objet trop dissimilaire. Pour cela, nous définissons un paramètre D_{min} qui est la distance minimum entre le centre du cluster et un objet du cluster. La principale action de cet agent est donc de vérifier en permanence si :

$$\forall i, d(o_i, o_c) < D_{min} \text{ avec}$$

o_i : le i ème objet du cluster

o_c : le centre du cluster

L'agent cluster est aussi capable de recalculer son centre s'il reçoit un nouvel objet ou s'il rejette un objet. Dans le cas où il rejette un objet, il le place dans la liste des objets. Pour recevoir un objet, il doit d'abord accueillir une fourmi puis la laisser vérifier si elle peut déposer l'objet qu'elle transporte.

Naissance d'un agent cluster

Lorsqu'une fourmi n'a pas déposé un objet après avoir parcouru toute la liste des agents cluster, elle doit créer un nouvel agent cluster pour y déposer son objet. Cet agent cluster n'a alors qu'une durée de vie limitée en nombre d'itérations. S'il reçoit au moins deux autres objets dans cet intervalle de temps, l'agent cluster peut alors calculer le nouveau centre de gravité de son cluster. Ainsi nous évitons le problème des objets isolés.

Mort d'un agent cluster

Un agent cluster de moins de trois objets à une durée de vie limitée. Si pendant une durée fixée en paramètre, il ne reçoit aucun autre objet, il disparaît en rejetant avant les objets de son cluster dans la liste.

4.4 - La liste

La liste des objets permet de stocker les objets non encore classés. Les fourmis peuvent se déplacer dessus et prendre un objet. Les agents cluster rejettent également dans cette liste les objets jugés trop dissimilaires. Elle est mise à jour par l'agent observateur en cas d'arrivée de nouveaux agents de bas niveau. Cette structure permet de gérer à la fois l'arrivée de nouvelles données et la modification de données. Lorsque l'agent observateur détecte un nouvel agent, il crée un objet qu'il place dans la liste d'objet. Lorsqu'un agent cluster rejette un objet, il le place également dans la liste d'objets. De plus cette

liste est visité par les fourmis. Il est donc clair, qu'elle doit être accessible par ces trois agents.

4.5 - Le protocole de tests

Pour tester notre architecture nous proposons un protocole permettant de comparer les résultats obtenus par notre architecture à des résultats obtenus par des méthodes de clustering classiques. Les données traitées par notre architecture sont dynamiques, il est donc nécessaire de les figer à l'instant où l'utilisateur demande un résultat.

Le fichier généré à cet instant par l'agent observateur va alors servir de base pour appliquer un clustering classique. Ce résultat sera ensuite comparé à celui fourni par notre architecture. Ainsi, nous pourrions déterminer si notre architecture donne des résultats cohérents par rapport au clustering classique sur des données statiques.

4.6 - Bilan

L'arrivée de nouvelles données est prise en charge par cette architecture. Lorsqu'un nouvel agent apparaît au sein de la population d'agents de bas niveau, l'agent observateur copie ses caractéristiques dans un objet et envoie cet objet dans la liste où il sera traité comme les autres objets. De plus, lorsqu'un objet subit une modification qui le rend trop dissimilaire au sein de son cluster, l'agent cluster repère instantanément cet objet et le rejette immédiatement dans la liste où là encore, il sera traité comme les autres objets.

5 - CONCLUSION ET PERSPECTIVES

Nous avons réalisé ce travail, afin de développer un système multi-agents représentant une situation courante grâce au paradigme des agents aspectuels. Nous cherchons un moyen de regrouper les agents en fonction de leurs caractéristiques, pour pouvoir analyser cette situation et éventuellement prédire les événements suivants. Il s'agit donc d'un clustering mais avec des données dynamiques d'où le terme de «clustering dynamique». Nous voulons également étendre cet notion de clustering dynamique à d'autres systèmes multi-agents.

Le premier intérêt de ce travail est donc de définir la notion de clustering dynamique pour un système multi-agents : de nouveaux agents peuvent apparaître ou disparaître et les agents peuvent évoluer. Dans un premier temps, nous avons donc fourni une formalisation des données de ce problème. Cette formalisation nous a permis de mieux cerner les contraintes que nous avons à prendre en compte.

Dans le but de voir s'il était possible d'utiliser ou d'adapter un algorithme existant nous avons classé différents algorithmes de clustering classiques en fonction des contraintes de ce problème. Cependant, nous nous sommes aperçu que la plupart de ces algorithmes ne sont pas adaptés aux contraintes de notre problème. Seul les algorithmes fournis, semblent pouvoir résoudre en partie notre problème car ils peuvent s'adapter à l'arrivée des nouvelles données mais aussi à l'évolution des données.

Nous avons donc étudié «AntClass», un algorithme de clustering reposant en partie sur un algorithme de type fournis. Puis nous avons expérimenté cet algorithme sur des données dynamiques. Les résultats nous ont montrés que cet algorithme contient la base pour résoudre en partie notre problème.

Nous avons donc choisi de combiner un algorithme de type fourni avec un système multi-agents pour exploiter le caractère dynamique de l'algorithme fourni, tout en assurant une bonne convergence. Nous proposons donc une architecture permettant de mettre en oeuvre un clustering dynamique d'une population d'agents. Nous donnons également les spécifications de l'architecture et d'un protocole de tests permettant une validation future par rapport des techniques de clustering classiques.

Les tests sont actuellement en cours et les premiers résultats montrent que les clusters sont purs, mais souvent trop nombreux. Le comportement des agents clusters doit donc être complété afin d'autoriser par exemple des fusions permettant de limiter le nombre de clusters produits.

Les travaux de D. Servat porte sur le regroupement de goûtes d'eau. Ces travaux sont similaires aux notes. Néanmoins, il s'agit d'un domaine bien précis le ruissellement des goûtes d'eau, Servat, Perrier et Treuil (1998a, 1998b). De plus, les regroupements sont effectués par les agents de bas niveau, alors que nous avons fait le choix de ne pas influencer le travail de ces agents car nous avons voulu que notre architecture puisse s'appliquer à tous les systèmes multi-agents. Cependant, les performances de cet algorithme pourraient être intéressantes à comparer avec les performances de notre architecture.

Remerciements

L'implémentation est réalisé par M. coletta et G. Simon.

Je tiens à remercier tous les membres de l'équipe de recherche **Agents et Risques Majeurs** du **Laboratoire d'Informatique du Havre**, dans laquelle ont été effectués ces travaux.

Je salue également **G. Simon**, **M. Coletta** et **B. Mermet** pour l'aide qu'ils m'ont apporté à la rédaction de cet article.

RÉFÉRENCES

Barbará, D. "Using the fractal dimension to cluster datasets". Proc. of the International Conference in Knowledge Discovery and Data Mining (ACM-SIGKDD) (2000).

H. Boukachour, G. Simon, M. Coletta, T. Galinho, P. Person, F. Serin, "Système de veille préventive : modélisation par organisations d'agents", IC'2002, 13eme journées fran-cophones d'Ingénierie des Connaissances, Rouen, 28-30 mai 2002, p 187-195.

Durant, S. "Représentation de points de vues multiples dans une situation d'urgence: modélisation par organisation d'agents". PhD thesis, Université du Havre (1999).

Kamber M., Han J., "Data mining concepts and techniques", Morgan Kaufmann publishers, chapitre 8, (2001).

McKenna, S., Gong, S., "Modelling facial colour and identity with gaussian mixtures". Proc. of the Pattern Recognition, volume 31, pages 1883–1892 (1998).

Monmarché, N., Slimane, M., G. Venturini, "On improving clustering in numerical databases with artificial ants". Proc. of the Floreano, D., Nicoud, J., editors, 5th European Conference on Artificial Life (ECAL'99), Lecture Notes in Artificial Intelligence, volume 1674, pages 626–635, Swiss Federal Institute of Technology, Lausanne, Switzerland. Springer-Verla.

R. Coma, G. Simon, M. Coletta, "A multi-agent architecture for agents clustering", ABS'2003, 4th Workshop on Agent-Based Simulation, p 15-20, Montpellier, April 28-30 2003.

D. Servat, E. Perrier, J.-P. Treuil, A. Drogoul, "Towards virtual experiment laboratories: How multi-agent simulations can cope with multiple scales of analysis and viewpoints". Lecture Notes in Computer Science, Vol. 1434, p. 205–? ? (1998a).

D. Servat, E. Perrier, J.-P. Treuil, A. Drogoul, "When agents emerge from agents: Introducing multi-scale viewpoints in multi-agent simulations". Proc. of the MABS, pages 183–198 (1998b).

ADAPTATION DES APPLICATIONS REPARTIES A BASE DE COMPOSANTS AUX TERMINAUX MOBILES EN ENVIRONNEMENT SANS FIL

Nabil Kouici, Denis Conan et Guy Bernard

GET / INT, CNRS Samovar
9 rue Charles Fourier, 91011 Évry, France
{Nabil.Kouici, Denis.Conan, Guy.Bernard}@int-evry.fr
Télé : 33 1 60 76 44 15, 45 34, 45 67
Fax : 33 1 60 76 47 80

Résumé : les terminaux mobiles tels que les assistants personnels numériques ou les téléphones portables sont de plus en plus utilisés. Cependant, l'accès aux applications réparties à partir de ces terminaux soulève le problème de la disponibilité de ces services en présence des déconnexions. Le système et les applications distribuées fonctionnant sur ces terminaux doivent s'adapter aux changements de l'environnement mobile. Le travail présenté dans cet article est la continuation de Domint, une plate-forme pour la gestion des déconnexions pour les applications à base d'objets CORBA. Cet article propose une approche pour la gestion des déconnexions pour des applications à base de composants CCM (*CORBA Component Model*). Nous proposons un patron de conception pour le partitionnement des entités de l'application. Ce partitionnement est basé sur deux méta-données : « déconnectabilité » et « nécessité ». Ensuite, nous présentons D*MINT, une architecture qui permet l'adaptation des applications réparties à base de composants CCM aux déconnexions dans les environnements mobiles.

Summary : The popularity of mobile terminals such as personal digital assistants and mobile phones raises the problem of the service availability in the presence of disconnections. The system and the distributed applications running on these terminals must adapt to mobile environment changes. The work presented in this article is the continuation of Domint, a platform to cope with disconnections in mobile environments for CORBA-objects-based applications. In this article, we propose a promising approach for the disconnections management of CCM-based applications (*CORBA Component Model*). We propose a design pattern for the partitioning of application entities. This partitioning is based on two metadata: "disconnectability" and "necessity". Finally, we present D*MINT, a platform to cope with the disconnections in mobile environments for CCM-based applications.

Mots-clés : Mobilité, intergiciel, déconnexion, composant, méta-donnée.

Adaptation des applications réparties à base de composants aux terminaux mobiles en environnement sans fil

1 - INTRODUCTION

Ces dernières années ont été marquées par une forte évolution des équipements utilisés dans les environnements répartis. Nous sommes successivement passés de réseaux locaux à des réseaux à grande échelle (Internet), puis à des réseaux sans fil interconnectant des machines mobiles comme des téléphones portables ou des assistants personnels numériques (PDA). Cette évolution a abouti à la définition d'une nouvelle technologie qui se base sur l'infrastructure des réseaux mobiles. Cette technologie est l'informatique mobile dans laquelle l'utilisateur peut continuer d'accéder à l'information fournie par une infrastructure distribuée, sans tenir compte de son emplacement.

La construction d'applications réparties converge de plus en plus vers l'utilisation des intergiciels (*middleware*) à composants, en particulier les modèles de composants CORBA [Object Management Group, 2002] de l'OMG, EJB [DeMichiel, 2002] de SUN et .NET [Microsoft, 2002] de Microsoft. Le modèle de composants offre une séparation entre les fonctionnalités métier du composant et les fonctionnalités système (sécurité, transaction, cycle de vie,...) de l'intergiciel. Cette séparation est réalisée suivant le paradigme composant/conteneur. En effet, le composant encapsule le code métier et le conteneur gère les aspects extra-fonctionnels (fonctionnalité système) [Szyperski *et al.*, 2002].

Le travail présenté dans cet article est la continuation de Domint [Conan *et al.*, 2002a, Conan *et al.*, 2002b], une plate-forme à base d'objets CORBA qui offre aux applications mobiles la continuité de services en mode déconnecté dans des environnements mobiles. L'idée principale de Domint est de créer sur le terminal mobile des entités mandataires appelées « objets déconnectés ». Dans Domint, la gestion de la connectivité utilise un mécanisme d'hystérésis qui donne les différents niveaux de connectivité (connecté, partiellement connecté et déconnecté).

La contribution principale des travaux présentés dans cet article est la proposition d'une architecture à base de composants nommée D*MINT pour la gestion de la déconnexion. Dans cette architecture, nous proposons un patron de conception pour la classification des entités d'une application répartie, pour distinguer les entités qui peuvent avoir une entité déconnectée dans le terminal mobile et sélectionner les entités nécessaires dans le terminal

mobile pour que l'application continue à fonctionner en mode déconnecté. Nous proposons aussi un mécanisme pour la création et la gestion des entités déconnectées et un mécanisme pour opérer le transfert d'états entre les entités déconnectées et les entités distantes.

La suite de l'article est divisée comme suit. La section 2 identifie nos motivations et objectifs. Ensuite, dans la section 3, nous présentons les métadonnées que nous utilisons dans la section 4 dans le contexte des intergiciels orientés composants. La section 5 décrit brièvement l'architecture D*MINT. La section 6 établit le lien avec les travaux existants. Finalement, la section 7 conclut l'article et donne quelques perspectives.

2 - MOTIVATIONS ET OBJECTIFS

Les terminaux mobiles devenant extrêmement courants, leur utilisation ne cesse de se généraliser dans tous les domaines. Leur utilisation doit donc devenir aussi naturelle que possible. En particulier, la variation de la connectivité ne doit pas être vue comme une faute dans les environnements mobiles. Pour des applications distribuées, la variation de connectivité peut se traduire par des déconnexions.

Nous considérons deux types de déconnexions : les déconnexions volontaires et les déconnexions involontaires. Les premières, décidées par l'utilisateur depuis son terminal mobile, sont justifiées par les bénéfices attendus sur le coût, l'énergie et la minimisation des désagréments induits par les déconnexions inopinées. Les secondes sont le résultat de coupures intempestives des déconnexions physiques du réseau, par exemple, lors de passage de l'utilisateur dans une zone d'ombre radio.

Les environnements d'exécution d'applications réparties qui se basent sur les intergiciels actuels reposent sur des hypothèses (utilisateurs fixes, terminaux puissants, connexions réseau de bonne qualité et peu coûteuses) incompatibles avec les caractéristiques de l'informatique mobile, qui se distingue en particulier par le nomadisme des utilisateurs accédant depuis n'importe quelle localisation géographique à leurs applications. En dépit de tous les problèmes mentionnés ci-dessus, un utilisateur mobile souhaite se déplacer librement et continuer à travailler le plus normalement possible avec son terminal mobile. Il est donc souhaitable de

fournir une continuité de service malgré les déconnexions et les perturbations du réseau sans fil. Le besoin de continuer à travailler dans un environnement mobile soulève de nombreux problèmes. Tous d'abord, l'approche visant à résoudre la disponibilité passe par une réplication des données et du code sur le terminal mobile. En outre, d'après [Satyanarayanan, 1996a], l'adaptation aux caractéristiques de l'informatique mobile est exprimable en trois stratégies : « laissez-faire » où l'adaptation est entièrement de la responsabilité de l'application, donc sans aucun support du système ; « transparence » où aucun changement de l'application n'est nécessaire puisque c'est le système qui est responsable de l'adaptation ; « collaboration » où l'adaptation est faite en collaboration entre l'application et le système. De nombreux travaux synthétisés dans [Jing *et al.*, 1999] montrent que les approches « laissez-faire » et « transparence » ne sont pas adéquates. Ainsi, dans nos travaux, nous nous basons sur la stratégie d'adaptation « collaboration ».

La réplication des données sur le terminal mobile consiste à instancier le même type d'entité sur le terminal mobile que sur le serveur fixe où elle se situe normalement. Ces entités déconnectées vont être utilisées pour assurer la continuité de service lors de déconnexions. Les opérations effectuées pendant les phases de déconnexion sont journalisées localement pour être envoyées lors de la reconnexion aux serveurs pour effectuer la réconciliation. Ce processus de traitement des déconnexions encapsule le déploiement des entités distantes sur le terminal mobile. La plupart des algorithmes de déploiement existant dans la littérature traitent ce problème selon un aspect quantitatif : nombre d'entités en mémoire et l'élection des entités que le système supprime en cas de dépassement de la taille mémoire. Cependant, peu de travaux traitent l'aspect qualitatif de l'entité susceptible d'être déployée sur le terminal mobile. C'est sur ce dernier aspect que se penche notre travail dans cet article où nous proposons deux métadonnées : « nécessité » et « déconnectabilité ». Ces deux métadonnées répondent à deux questions. Premièrement, avoir des mandataires dans le terminal mobile augmente la disponibilité du service, mais est-ce que toutes les entités distantes peuvent avoir une entité déconnectée (déconnectabilité) ? Deuxièmement, est-ce que ces entités déconnectables sont nécessaires pour le fonctionnement de l'application en mode déconnecté ?

Dans le méta-modèle que nous présentons dans la section 3, nous utilisons le mot « entité » dans l'acception « granularité de l'entité manipulée par le système » (par exemple, « fichier » dans les systèmes orientés fichiers, « enregistrement » dans les systèmes de bases de données, « objet » dans les systèmes orientés objets et « composant » dans les systèmes orientés composants). Dans la section 5,

nous présentons l'architecture D*MINT avec des composants CORBA. Nous avons choisi CCM pour son interopérabilité, son utilisation dans des domaines variés et sa portabilité sur des assistants personnels numériques. Les résultats peuvent cependant être appliqués à d'autres modèles de composants.

3 - MÉTA-MODÈLE POUR LA GESTION DE LA CONNECTIVITÉ

La gestion de la connectivité dépend largement de la sémantique de l'application. Cette sémantique peut être modélisée par des données appelées métadonnées. Dans cette section, nous utilisons les métadonnées, d'une part, pour spécifier si une entité distante peut avoir une entité déconnectée sur le terminal mobile (cf. Section 3.1), d'autre part, pour spécifier si cette entité déconnectée est nécessaire pour l'exécution de l'application en mode déconnecté (cf. Section 3.2). Pour mieux comprendre les métadonnées que nous définissons dans cette section, nous illustrons leur utilisation avec une application de messagerie électronique que nous avons implantée. Cette application comporte trois types d'entités distribuées : l'entité `MailBoxManager` responsable de la création, de la suppression et de la localisation des entités `MailBox` ; ces dernières sont responsables de l'envoi, de la réception et de la suppression des messages ; enfin, l'entité `AddressBook` fournit un outil de consultation, d'ajout et de suppression d'adresses des utilisateurs.

3.1 - Méta-donnée déconnectabilité

La méta-donnée « déconnectabilité » indique si cette entité présente dans le terminal fixe accepte une entité déconnectée dans le terminal mobile. Ainsi, les entités distribuées de l'application sont partitionnées en deux groupes. La figure 1 présente un patron de conception pour le partitionnement des entités de l'application, plus l'attribution de la méta-donnée « nécessité » présentée dans la section 3.2. Le partitionnement en entités déconnectables et entités non déconnectables est réalisé par le développeur¹ de l'application qui doit concevoir son application suivant ces contraintes. Ce partitionnement dépend de la sémantique de l'application. Par exemple, pour des raisons de sécurité, le développeur de l'application

¹ Dans le processus de développement d'applications à base de composants CORBA, le développeur de l'application peut être décomposé en concepteur du composant, concepteur de composition de composants, implanteur du composant, empaqueteur du composant et déployeur du composant.

Entité	Déconnectabilité	Nécessité		
		Choix du programmeur	Choix de l'utilisateur	Finale
MailBox	Oui	EN	n/a	EN
MailBoxManager	Non	n/a	n/a	n/a
AddressBook	Oui	ENN	EN	EN

déconnectable. Par contre, les entités MailBox et AddressBook peuvent

peut décider de déployer des entités de l'application dans des serveurs sécurisés. Ces entités ne peuvent

TAB. 1. Partitionnement des entités de l'application de messagerie électronique : n/a signifiée « non applicable ». création des entités déconnectées pour ces entités n'est pas autorisée. Les entités déconnectées doivent être conçues pour faire face à une future réconciliation avec les entités distantes. Dans cet article, le problème de la réconciliation ne sera pas abordé. Ce problème est abordé dans [Chateigner *et al.*, 2003].

La seconde colonne de la table 1 présente l'affectation de la méta-donnée pour les entités de l'application de messagerie électronique. Dans cette application, supposons que le client administrateur de l'application crée dans son terminal mobile un mandataire pour l'entité MailBoxManager.

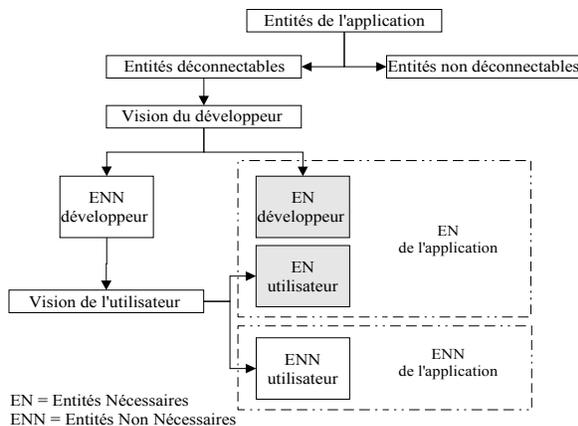


FIG. 1. Méta-modèle pour la partitionnement des entités de l'application.

Dans le scénario où l'administrateur peut manipuler cet objet en mode déconnecté et créer une nouvelle boîte aux lettres pour un autre utilisateur, ce dernier utilisateur ne pourra pas utiliser sa boîte aux lettres tant que l'administrateur est déconnecté et que la réconciliation n'a pas eu lieu. Donc, l'entité MailBoxManager peut être classée par le développeur de l'application comme étant une entité non

avoir une entité déconnectée dans le terminal

veut lire, envoyer ou supprimer des messages. Les messages manipulés par l'utilisateur dans l'entité déconnectée sont les messages téléchargés à partir de l'entité dans le serveur fixe avant la déconnexion. Ce mécanisme permet quand même à l'entité restée dans le réseau fixe de recevoir les messages à destination de l'utilisateur. Cet utilisateur verra ses messages lors de la reconnexion et les messages qu'il a envoyés à partir de l'entité déconnectée seront alors acheminés vers les destinataires. L'entité AddressBook est personnelle à un utilisateur, c'est une entité passive. Ainsi, si l'utilisateur ajoute ou supprime des entrées dans son carnet d'adresses, ces changements n'auront pas un impact sur d'autres entités partagées de l'application. Cette entité est donc déconnectable.

3.2 - Méta-donnée nécessité

La méta-donnée que nous avons présentée dans la section précédente ne traite pas le problème de la quantité d'entités déconnectées devant co-exister dans le faible espace mémoire du terminal mobile. Pour résoudre ce problème, nous introduisons une autre méta-donnée : « nécessité », qui permet de donner un « poids » aux entités de l'application quant à leur présence dans le terminal mobile. Ce concept nous amène à classer les entités déconnectables en entités nécessaires (EN) et entités non nécessaires (ENN). Une entité nécessaire est une entité dont la présence dans le terminal mobile est obligatoire pour le fonctionnement en mode déconnecté. Une entité non nécessaire est une entité dont l'absence dans le terminal mobile ne peut pas empêcher le fonctionnement de l'application en mode déconnecté. L'application doit être construite de telle façon qu'une telle entité absente pendant une déconnexion n'empêche pas l'application de continuer à fonctionner. Comme présentée dans la figure 1, la classification des entités de l'application en EN et ENN est effectuée d'abord par le développeur et ensuite par l'utilisateur. Le développeur de l'application fournit la première classification en « EN développeur » et « ENN développeur ». Ensuite, à chaque lancement de

l'application ou dans la configuration de cette dernière à l'installation, l'utilisateur peut forcer un « ENN développeur » à devenir nécessaire, une telle entité étant alors appelée « EN utilisateur ». Dans la figure 1, les rectangles grisés représentent les EN de l'application, c'est-à-dire l'union des « EN développeur » et des « EN utilisateur ». En outre, l'utilisateur ne peut pas forcer un « EN développeur » à devenir un « ENN utilisateur ». Nous considérons que le développeur connaît mieux que l'utilisateur quelles sont les entités déconnectées qui doivent impérativement exister dans le terminal mobile pour le fonctionnement en mode déconnecté.

La troisième colonne de la table 1 présente l'affectation de la méta-donnée « nécessité » pour les entités de l'application de messagerie électronique. Il est clair que la présence d'un mandataire pour MailBox est nécessaire pour le fonctionnement de l'application. Par contre, AddressBook est classée comme ENN par le développeur puisque l'utilisateur peut se passer de son carnet d'adresses et n'envoyer des messages qu'aux destinataires dont il connaît l'adresse. Lorsque l'utilisateur lance son application de messagerie électronique sur son terminal mobile ; il s'aperçoit qu'il a le droit de modifier la nécessité de AddressBook. S'il pense qu'il va envoyer plusieurs messages à des destinataires différents et qu'il ne connaît pas par cœur leur adresse, il peut forcer la nécessité de AddressBook.

4 - LE MODÈLE DE COMPOSANT CCM

L'utilisation de modèles et langages orientés objets pour la construction des applications réparties existe depuis longtemps. L'objectif de ces recherches est d'améliorer la modélisation d'une application, d'optimiser la réutilisation du code métier et d'adresser l'ensemble du cycle de développement. Cependant, la conception orientée objet n'est pas adaptée à la description de schémas de coordination et de communication complexes. Pour pallier les limites de l'approche orientée objets et adresser des notions non prévues initialement, l'approche composant est apparue. Dans cette section, nous présentons d'abord le modèle de composant CCM (cf. Section 4.1), ensuite nous discutons l'utilisation des méta-données présentées dans la section 3 dans le contexte CCM (cf. Section 4.2).

4.1 - Vue d'ensemble de CCM

Il n'existe pas, aujourd'hui, une définition unique du « composant logiciel ». Toutefois, plusieurs définitions ou caractéristiques sont acceptées. Szyperski [Szyperski *et al.*, 2002] définit un composant comme « une unité de composition avec des interfaces

contractuellement spécifiées et des dépendances explicites sur son contexte. Un composant peut être déployé indépendamment et il est sujet à des compositions par tiers. »

Dans la suite de cet article, notre raisonnement est illustré dans le cadre de CCM dont des implantations existent déjà : par exemple, OpenCCM [Marvie et Merle, 2001] et microCCM [Pilhofer, 2002]. Le choix de CCM est justifié par plusieurs points. Premièrement, par rapport aux autres modèles tels que EJB [DeMichiel, 2002], COM [Rogerson, 1997] et .NET [Microsoft, 2002], CCM peut être vu comme l'union de ces modèles. Comme dans EJB, un composant CORBA est créé et géré par une maison et exécuté dans un conteneur. Comme dans COM, un composant CORBA offre et utilise des interfaces et il permet la navigation et l'introspection. Comme dans .NET, un composant CORBA peut être écrit dans plusieurs langages de programmation et peut être empaqueté pour être distribué. Deuxièmement, CCM est multilinguages, multiOS, multiORB, multivendeur, contrairement à EJB qui est purement JAVA, et COM et .NET qui sont purement Windows.

Un composant CCM est caractérisé par des ports classés en synchrones (facettes et réceptacles) et asynchrones (sources et puits d'événements), et des attributs pour la configuration (cf. figure 2). Un autre avantage avec CCM est que le composant peut être segmenté (cf. figure 3).

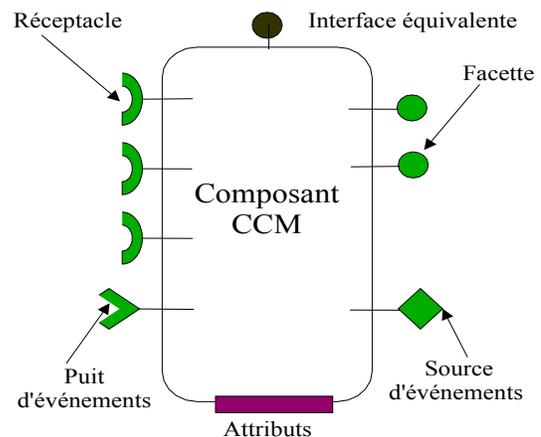


FIG. 2. Vision externe d'un composant CCM.

Pour chaque segment, CCM génère un squelette. Les segments sont activés indépendamment et possèdent un état. Enfin, chaque segment est séparément identifié dans le système.

4.2 - Déconnectabilité et nécessité dans CCM

Dans cette section, nous présentons juste une brève discussion sur l'utilisation des méta-données « déconnectabilité » et « nécessité » dans le contexte CCM. Ce point a été développé dans [Kouici *et al.*, 2003]. Un composant déconnectable est un composant qui peut avoir un composant déconnecté dans le terminal mobile utilisé en mode déconnecté. Dans le cas d'une implantation monolithique du composant, tous les services offerts par le composant sont implantés dans une seule classe. Si le composant est déconnectable, le segment qui correspond à la totalité du composant est aussi déconnectable.

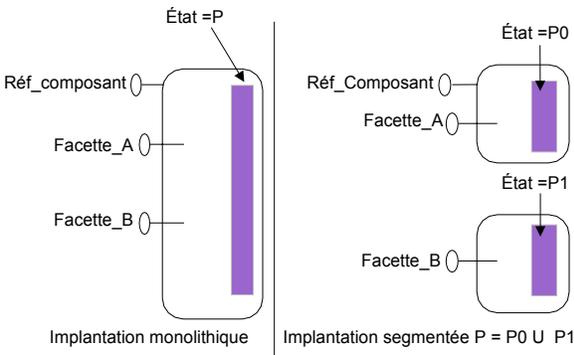


FIG. 3. Composants monolithique et segmenté.

Dans le cas d'une implantation segmentée, ce mandataire ne contient que des segments déconnectables. En outre, un composant déconnectable contient au moins un segment déconnectable et s'il existe un segment déconnectable alors ce composant est déconnectable.

CCM définit la notion de segment principal (*main segment*). Ce segment contient tous les services et attributs non affectés à d'autres segments. Ce segment possède aussi l'exclusivité de manipulation du contexte par réflexion. En particulier, il offre la possibilité de localiser les autres segments par l'interface `ExecutorLocator`. Ainsi, il est clair que la déconnectabilité du composant implique la déconnectabilité de son segment principal.

Nous définissons aussi la notion de composant et de segment nécessaires par analogie avec la déconnectabilité. Un composant nécessaire doit contenir au moins un segment nécessaire et si un segment est déclaré nécessaire alors son composant est aussi nécessaire. La localisation des services et des segments étant nécessaire en mode déconnecté, la nécessité d'un composant implique la nécessité de son segment principal.

5 - ARCHITECTURE DE D*MINT

La première partie de cette section décrit l'intégration des méta-données présentées dans la section 4.2 dans la plate-forme OpenCCM. Ensuite, nous présentons brièvement l'architecture D*MINT.

OpenCCM est une plate-forme écrite en Java. Elle fournit une première implantation partielle de CCM. CCM étend le langage OMG IDL pour permettre d'exprimer les caractéristiques du composant décrites dans la section 4.1. En plus, il définit un nouveau langage, *Component Implementation Definition language* (CIDL), pour spécifier l'architecture interne des composants. Nous avons ajouté dans le CIDL de nouvelles entrées grammaticales pour spécifier la déconnectabilité et la nécessité d'abord pour le composant, c-à-d. pour le segment principal, ensuite pour les autres segments du composant. Nous avons aussi ajouté des étapes dans la chaîne de compilation d'OpenCCM [Flissi, 2003] pour inclure dans chaque squelette du segment le code nécessaire pour spécifier la déconnectabilité et la nécessité.

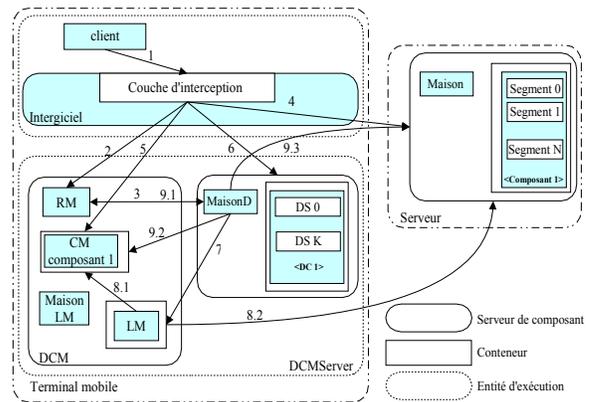


FIG. 4. Architecture de D*MINT.

La figure 4 représente l'architecture de D*MINT. D*MINT crée pour chaque terminal mobile un service de gestion des déconnexions DCMServer qui se trouve dans une entité d'exécution indépendante de celles des applications dans le terminal mobile. Le DCMServer est accessible par l'intergiciel de l'application via une couche d'interception propre à chaque instance de l'intergiciel. Dans CORBA, cette couche d'interception est pilotée par des objets CORBA locaux appelés les intercepteurs portables [Object Management Group, 2001a]. Cette couche intercepte toutes les requêtes du client vers les serveurs. Le gestionnaire des composants déconnectés (DCM) comporte une fabrique pour les composants/segments déconnectés (RM) et un

gestionnaire du journal d'opérations (LM). À chaque composant déconnecté est associé un gestionnaire de connectivité (CM).

La couche d'interception détient une table qui fait le lien entre une référence de segment et un gestionnaire de connectivité. Imaginons le scénario où D*MINT doit créer un segment/composant déconnecté pour chaque segment/composant à la première invocation. Lorsqu'un client invoque un service du composant (via le segment associé) (1), la couche d'interception intercepte cette requête et contrôle la déconnectabilité de ce segment. Dans le cas où ce segment accepte un mandataire, la couche d'interception contrôle si le composant déconnecté associé à ce segment existe déjà dans DCMServer. La requête étant la première pour ce composant, la couche d'interception obtient la référence de RM (2). RM crée d'abord une maison déconnectée (3) utilisée pour la création du composant déconnecté (DC). Ensuite, RM crée le gestionnaire de connectivité (CM) associé et le segment déconnecté (DS) pour le segment invoqué. Au moment de la création de DC/DS, l'intergiciel transmet la requête du client vers le composant distant (4).

Les requêtes du client sont interceptées par la couche d'interception. Cette dernière contrôle la connectivité via le CM associé au segment invoqué (5). Si la connectivité est bonne, la requête est dirigée vers le composant distant (4). Sinon, la requête est dirigée vers le DS/DC associé (6). Les opérations effectuées sur DC/DS vont être sauvegardées dans LM (7). Périodiquement, LM teste la connectivité (8.1), et si possible, envoie les requêtes au composant distant (8.2). La maison déconnectée comporte du code nécessaire pour chercher la référence de CM (9.1), tester la connectivité (9.2) et faire un transfert d'état entre DC et le composant distant (9.3). Le mécanisme de réconciliation entre l'entité distante et les entités déconnectées est étudié dans [Chateigner *et al.*, 2003].

6 - TRAVAUX CONNEXES

La problématique traitée dans cet article aborde les domaines de recherche autour de la mobilité des applications (gestion de la déconnexion), des méta-données (configuration par réflexion) et des modèles de composants.

Un panorama est donné dans [Jing *et al.*, 1999] sur l'adaptation à la mobilité des applications client/serveur, qui se concentre de plus en plus sur la gestion et le déploiement² des entités déconnectées dans le terminal mobile. Coda [Satyanarayanan,

1996b, Satyanarayanan, 1996a] est un système de gestion de fichiers qui définit la notion de données implicites et de données explicites. Les données implicites représentent l'historique d'utilisation du client. Les données explicites prennent la forme d'une base de données construite par le client de l'application. Si l'utilisateur fait un mauvais choix pour ses données explicites suite à une mauvaise compréhension de l'application, cette dernière peut ne pas fonctionner en mode déconnecté. Odyssey [Mummert, 1996, Noble et Satyanarayanan, 1999] ajoute pour chaque objet distant de l'application deux méta-données : « fidélité » et « fenêtre de tolérance ». Ces méta-données sont bien adaptées pour la gestion de la faible connectivité, mais pas pour les déconnexions. Rover [Joseph *et al.*, 1997] définit les notions d'objet dynamique relogeable (*Relocatable Dynamic Object, RDO*) et d'appel de procédure à distance non-bloquant (*Queued Remote Procedure Call, QRPC*). Rover traite tous les objets de l'application de la même façon et ne tient pas compte de la sémantique de l'application. En outre, le développeur doit concevoir son application en terme de RDO.

L'usage des méta-données a été aussi introduit dans les intergiciels réflexifs [Blair *et al.*, 1998, Parlavantzis *et al.*, 2000]. XMIDDLE [Capra *et al.*, 2001] définit la méta-donnée « profil » qui décrit ce que l'intergiciel doit faire lorsqu'il se trouve dans un contexte particulier. Cette méta-donnée contient des informations sur les ressources externes (bande passante, batterie...) et ne prend pas en compte la sémantique de l'application. Ce type de méta-donnée a été aussi introduit dans CARISMA [Capra *et al.*, 2003], qui associe à chaque application un fichier de description de comportements.

Dans le contexte CORBA, les projets Π^2 [Ruggaber *et al.*, 2000] et Alice [Lynch, 1999] traitent le problème des courtes déconnexions provoquées par le changement de cellule (*handover*) dans *wireless* CORBA [Object Management Group, 2001b]. Π^2 ne traite que le problème des déconnexions involontaires et utilise deux objets déconnectés, un sur le terminal mobile et l'autre sur le terminal fixe. Alice fournit des mécanismes pour traiter les déconnexions volontaires et involontaires en utilisant les exceptions produites par l'intergiciel lors de changements de cellules. Dans Alice, le code nécessaire pour commuter vers le mode déconnecté doit être inclus dans le code de l'application. Dans notre approche, l'utilisation d'une couche d'interception au niveau de l'intergiciel permet de limiter l'impact sur le code de l'application. CASCADE [Chockler *et al.*, 2000, Atzmon *et al.*, 2002] définit un service de gestion des mandataires générique et hiérarchique pour des objets CORBA. Il définit un service de désignation pour les serveurs de mandataires et les clients utilisent ce

² Ici, le mot déploiement signifie le téléchargement ou la création des entités déconnectées sur le terminal mobile.

service pour trouver des serveurs où créer des mandataires. Dans CASCADE, si le client se déconnecte avec tous les serveurs comportant les mandataires, il ne peut pas continuer à travailler si sa machine n'est pas elle-même un serveur de mandataires.

Enfin, dans le domaine des composants, CESURE [Marangozova et Hagimont, 2002] traite le problème de la réplication des composants d'une manière extra-fonctionnelle. Ce projet introduit la notion de composant de déconnexion qui représente une réplique du composant distant, et qui contient en plus le code nécessaire pour la réconciliation. L'approche utilisée dans CESURE ne traite pas le problème des déconnexions involontaires puisque les mandataires sont créés sur des hôtes autres que le terminal mobile.

7 - CONCLUSION

Dans cet article, nous avons présenté l'état actuel de nos travaux sur l'adaptation des applications réparties en environnements mobiles. Nous avons proposé un patron de conception pour des applications réparties fonctionnant dans des environnements mobiles. Ce patron de conception se base sur l'utilisation des méta-données « déconnectabilité » et « nécessité ». La méta-donnée « déconnectabilité » permet de spécifier si une entité dans un serveur fixe peut avoir un mandataire dans le terminal mobile. La méta-donnée « nécessité » permet de donner un « poids » à l'entité déconnectée quant à sa présence dans le terminal mobile. Ces méta-données permettent de créer un lien de collaboration entre le développeur de l'application et l'utilisateur de cette dernière.

En utilisant ces méta-données et en s'appuyant sur la plate-forme Domint, nous avons présenté D*MINT. Nous montrons avec cette architecture que la collaboration pour le traitement de la mobilité ne s'arrête pas à l'utilisation des méta-données, mais demande aussi l'utilisation d'autres ressources systèmes tels que les intercepteurs au niveau de l'intergiciel.

Les méta-données que nous avons présentées dans cet article ne traitent pas le problème de la priorité entre les différentes entités de l'application. Cette priorité doit être spécifiée dans une autre méta-donnée : « criticité ». Les autres perspectives de nos travaux sont les suivantes. Tout d'abord, nous allons terminer l'implantation de D*MINT, porter cette architecture sur des terminaux mobiles (PDA) et faire des tests de performances. L'architecture actuelle de D*MINT utilise des composants métier. Cependant, la gestion de la mobilité doit être vue comme un aspect extra-fonctionnel géré par le conteneur lui-même. Donc, une perspective supplémentaire est l'intégration de D*MINT dans les conteneurs CCM en utilisant des

composants système. Une approche prometteuse est celle des conteneurs ouverts [Vadet et Merle, 2001]. Enfin, D*MINT ne prend en compte que les invocations synchrones. Or, les communications asynchrones deviennent de plus en plus utilisées dans les intergiciels via des services d'événements ou de notification. Le modèle de composant CCM permet les deux types de communication. Nous travaillons donc à étendre D*MINT dans ce sens.

BIBLIOGRAPHIE

- [Atzmon *et al.*, 2002] Atzmon H., Friedman R. et Vitenberg R. (2002). Replacement Policies for a Distributed Object Caching Service. Dans *Proceedings of the International Symposium on Distributed Objects and Applications*, pages 661–674, California, Irvine, USA.
- [Blair *et al.*, 1998] Blair G. S., Coulson G., Robin P. et Papathomas, M. (1998). An Architecture for Next Generation Middleware. Dans *Proceedings of the IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing*, London, England
- [Capra *et al.*, 2001] Capra L., Emmerich W. et Mascolo C. (2001). Exploiting Reflection and Metadata to built Mobile Computing Middleware. Dans *Proceedings of the Workshop on Middleware Mobile Computing, IFIP/ACM*, Heidelberg, Germany.
- [Capra *et al.*, 2003] Capra L., Emmerich W. et Mascolo C. (2003). CARISMA : Context-Aware Reflective Middleware System for Mobile Applications. *IEEE Transaction on Software Engineering*.
- [Chateigner *et al.*, 2003] Chateigner L., Chabridon S. et Bernard G. (2003). Intergiciel pour l'informatique nomade : réplication optimiste et réconciliation. Dans *Actes de la Manifestation des Jeunes Chercheurs STIC, (MAJECSTIC)*, Marseille, France.
- [Chockler *et al.*, 2000] Chockler G., Dolev D., Friedman R. et Vitenberg R. (2000). Implementing a caching service for distributed CORBA objects. Dans *Proceedings of the 2nd FIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing*, pages 1–23.

- [Conan *et al.*, 2002a] Conan D., Chabridon S., Villin O. et Bernard G. (April 2002a). Disconnected Operations in Mobile Environments. Dans *Proceedings of the 2nd IPDPS Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, Ft. Lauderdale, USA.
- [Conan *et al.*, 2002b] Conan D., Chabridon S., Villin O., Bernard G., Kotchanov A. et Saridakis T. (2002b). Handling Network Roaming and Long Disconnections at Middleware Level. Dans *Proceedings of the Workshop on Software Infrastructures for Component-Based applications on Consumer Devices*, Lausanne, Switzerland.
- [DeMichiel, 2002] DeMichiel L. (2002). *Enterprise JavaBeans Specifications, version 2.1, proposed final draft*. Sun Microsystems, <http://java.sun.com/products/ejb/docs.html>.
- [Flissi, 2003] Flissi A. (2003). Inside OpenCCM. Rapport technique, ObjectWeb.
- [Jing *et al.*, 1999] Jing J., Helal A. et Elmagarmid A. (June 1999). Client-Server Computing in Mobile Environments. *ACM Computing Surveys*, 31(2).
- [Joseph *et al.*, 1997] Joseph A., Tauber J. et Kaashoek M. (1997). Mobile computing with the Rover toolkit. *ACM Transactions on Computers*, 46(3).
- [Kouici *et al.*, 2003] Kouici N., Conan D. et Bernard G. (2003). Disconnected Metadata for Distributed Applications In Mobile Environments. Dans *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications*, Las Vegas, Nevada, USA.
- [Lynch, 1999] Lynch N. (1999). Supporting Disconnected Operation in Mobile CORBA. Thèse de doctorat, Trinity College Dublin.
- [Marangozova et Hagimont, 2002] Marangozova V. et Hagimont D. (2002). An Infrastructure for CORBA Component Replication. Dans *Proceedings of the First International IFIP/ACM Working Conference on Component Deployment*, Berlin (Germany).
- [Marvie et Merle, 2001] Marvie R. et Merle P. (2001). CORBA Component Model: Discussion and Use with OpenCCM. Rapport technique, Laboratoire d'Informatique Fondamentale de Lille, France.
- [Microsoft, 2002] Microsoft (2002). Microsoft Developer Network. <http://www.msdn.microsoft.com>.
- [Mummert, 1996] Mummert L. (September 1996). Exploiting Weak Connectivity in a Distributed File System. Thèse de doctorat. Université de Carnegie Mellon.
- [Noble et Satyanarayanan, 1999] Noble B. D. et Satyanarayanan M. (1999). Experience with Adaptive Mobile Applications in Odyssey. *Mobile Networks and Applications*, 4(4) page 245–254.
- [Object Management Group, 2001b] Object Management Group (June 2001b). Wireless Access and Terminal Mobility in CORBA Specification. OMG Document dtc/01-06-02.
- [Object Management Group, 2002] Object Management Group (June 2002). CORBA Components. OMG Document formal/02-06-65, Version 3.0.
- [Object Management Group, 2001a] Object Management Group (September 2001a). Portable Interceptors. Interceptors Finalization Task Force. Published draft.
- [Parlavantzas *et al.*, 2000] Parlavantzas N., Coulson G., Clarke M. et Blair G. (2000). Towards a Reflective Component-based Middleware Architecture. Dans *Proceedings of the Workshop on Reflection and Metalevel Architectures*, Sophia Antipolis et Cannes, France.
- [Pilhofer, 2002] Pilhofer F. (2002). Writing and Using CORBA Component. Rapport technique, ALCATEL. <http://www.fpx.de/MicoCCM/>.
- [Rogerson, 1997] Rogerson D. (1997). *Inside COM*. Microsoft Press.
- [Ruggaber *et al.*, 2000] Ruggaber R., Seitz J. Et Knapp M. (July 2000). Π^2 A Generic Proxy Platform for Wireless Access and Mobility. Dans *Proceedings of the 19th ACM*

Symposium on Principles of Distributed Computing, Portland, Oregon.

[Satyanarayanan, 1996a] Satyanarayanan M. (1996a). Fundamental Challenges in Mobile Computing. Dans *Proceedings of the 15th Symposium on Principles of Distributed Computing*, pages 1–7.

[Satyanarayanan, 1996b] Satyanarayanan M. (1996b). Mobile Information Access. *IEEE Personal Communications*, 3(1).

[Szyperski et al., 2002] Szyperski C., Gruntz D. et Murer S. (2002). *Component Software, Beyond Object-Oriented Programming*. Addison-Wesley.

[Vadet et Merle, 2001] Vadet M. et Merle P. (2001). Les conteneurs ouverts dans les plates-formes à composants. Dans *Actes de la Journée Thème Émergent Composants*, Besançon, France.

SEGMENTATION D'IMAGE PAR MINIMISATION DE LA COMPLEXITE STOCHASTIQUE

Frédéric Galland,

Doctorant en traitement d'images
frederic.galland@fresnel.fr + 33 4 91 28 80 78

Nicolas Bertaux,

Maître de conférences en traitement d'images
nicolas.bertaux@fresnel.fr + 33 4 91 28 84 98

Philippe Réfrégier,

Professeur en traitement d'images
Philippe.refregier@fresnel.fr +33 4 91 28 88 09

Adresse professionnelle

Equipe Physique et Traitement d'Images, Institut Fresnel, UMR CNRS 6133, ENSPM,
D.U. St Jérôme, 13397 Marseille cedex 20, France.

Résumé : Dans ce papier, nous proposons un algorithme de segmentation en zones homogènes d'images fortement bruitées. Cette segmentation est obtenue à l'aide d'une grille active polygonale et est basée sur la minimisation de la complexité stochastique de l'image. Cette approche permet d'obtenir un critère MDL (Longueur de Description Minimum) sans paramètre et généralisable au cas des images multi-composantes. Nous présentons ensuite des résultats sur des images réelles et synthétiques, de type scalaires, multi-dates ou multi-capteurs perturbées par des bruits Gamma, Poisson ou Bernoulli, qui démontrent la capacité de cet algorithme à estimer le nombre de régions et la position des contours. Nous proposons enfin une approche originale permettant simultanément de segmenter et de recalibrer deux images.

Summary : We propose in this paper an algorithm for noisy image segmentation into homogeneous regions. This segmentation is performed thanks to an active polygonal grid and is based on the minimization of the stochastic complexity of the image. This approach leads to a Minimum Description Length (MDL) criterion without free parameter which can be extended to deal with multicomponent images. Some segmentation results are then presented on scalar, multi-date or multi-sensor images (real and synthetic) which are corrupted with Gamma, Poisson or Bernoulli noise and demonstrate the ability of the algorithm to estimate the number of regions in the image and the position of the contours. We finally present a new approach to simultaneously segment and register two images.

Mots clés : Traitement d'images, Segmentation, Images multi-composantes, Principe MDL.

Segmentation d'image par minimisation de la complexité stochastique

1 - INTRODUCTION

La segmentation d'images est à la base de nombreux systèmes de traitement d'images tels que le filtrage, la classification ou la reconnaissance de formes.

Un des principaux enjeux de la segmentation d'image est l'obtention d'un algorithme automatique, c'est-à-dire sans paramètre à ajuster de la part de l'utilisateur. Dans cette optique, le principe de Description de Longueur Minimale (MDL) présenté par Rissanen (1978, 1989) a été généralisé à la segmentation d'images [Leclerc 1989, Zhu 1996, Figueiredo 2000]. Cette approche, basée sur la théorie de l'information, permet notamment d'estimer les paramètres libres d'un modèle.

Un autre enjeu majeur est le traitement d'images bruitées. Récemment, modèles déformables et approches statistiques ont été réunies [Staib 1992, Kervrann 1994, Storvik 1994, Nguyen 1992, Zhu 1996, Figueiredo 2000]. Un algorithme rapide de segmentation par contour actif statistique polygonal adapté au cas des images bruitées a ainsi été proposé par Germain (1996), permettant la segmentation d'un objet unique présent dans une image. Cette approche a ensuite pu être généralisée à une approche multi-régions grâce à une grille active statistique [Germain 2001]. La principale limitation de cette grille active est qu'elle repose sur une pré-segmentation. D'un autre côté, grâce à un critère MDL analogue à celui introduit auparavant par Figueiredo (2000), Ruch (2001) a régularisé le contour actif statistique polygonal en estimant le nombre de nœuds du polygone même en présence d'objets fortement non convexes.

Enfin, avec le développement des systèmes d'imagerie, il devient possible d'obtenir différentes images issues d'une même scène mais acquises par exemple par différents capteurs ou à différentes dates. Il est alors important de pouvoir combiner les informations contenues dans ces différentes images lors du traitement. L'apport de telles images multi-composantes, par rapport aux images mono-composantes classiques, a notamment été prouvé dans le cas d'images multi-dates perturbées par un bruit de speckle [Germain 1999].

L'objectif de cet article est de proposer un algorithme de segmentation d'images en zones homogènes multi-régions et adapté aux images bruitées ainsi que son extension au cas des images

multi-composantes. Guidés par tous les travaux mentionnés précédemment, nous combinerons approche statistique, contours actifs et principe MDL. Ce papier reprend ainsi les principaux résultats que nous venons de publier [Galland, 2003] et fait le point sur les différentes avancées qui ont été faites depuis, notamment en ce qui concerne les images multi-composantes et le recalage d'images.

2 - CRITERE MDL ET ALGORITHME DE SEGMENTATION

2.1 - Détermination du critère MDL

Le principe du MDL appliqué à la segmentation d'images consiste à choisir la segmentation qui permet de décrire l'image entière avec le nombre minimum de bits.

Nous avons proposé récemment un algorithme de segmentation d'images mono-composantes en zones homogènes qui repose sur un critère MDL et qui est adapté au cas des images perturbées par un bruit de speckle [Galland, 2003]. La partition de l'image se fait à l'aide d'une grille polygonale, c'est-à-dire un ensemble de nœuds reliés par des segments afin de définir des régions homogènes. L'image $s = \{s(x, y) | (x, y) \in [1, N_x] \times [1, N_y]\}$ contient $N = N_x \times N_y$ pixels et est composée de R régions homogènes Ω_r numérotées de 1 à R et contenant chacune N_r pixels. Dans chaque région, les niveaux de gris des pixels sont supposés être distribués selon une densité de probabilité (pdf) de vecteur de paramètres θ_r . La segmentation d'une image consiste à estimer le nombre de régions ainsi que le nombre et la position des nœuds de la grille. Elle est obtenue en minimisant un critère MDL qui correspond à une approximation de la complexité stochastique de l'image et est mesuré en nat ($1 \text{ nat} = (\log 2) \text{ bits}$). Ce critère peut être divisé en trois termes: le premier Δ_G représente le nombre de nats nécessaires pour coder la grille, le deuxième Δ_p sert à coder le vecteur de paramètres θ_r dans chaque région et le dernier Δ_L permet de coder les niveaux de gris des pixels dans chaque région et peut être approché par l'opposé de la log-vraisemblance de l'image. On obtient ainsi le

nombre total de nats Δ nécessaires pour coder l'image:

$$\begin{aligned} \Delta &= \Delta_G + \Delta_p + \Delta_L \\ &= n(\log N + \log p) + p(\log(2\hat{m}_x) + \log(2\hat{m}_y)) \\ &\quad + \log p + \sum_{r=1}^R \left(\frac{\alpha}{2} \log N_r - l_e[\Omega_r | \theta_r] \right) \end{aligned} \quad (1)$$

avec n le nombre minimum de sous-graphes Eulériens nécessaires pour décrire la grille, p le nombre de segments de la grille, \hat{m}_x (resp. \hat{m}_y) la longueur moyenne des projections horizontales (resp. verticales) des segments de la grille, α le nombre de paramètres de la pdf des niveaux de gris et $l_e[\Omega_r | \theta_r]$ la log-vraisemblance dans la région Ω_r .

Nous allons maintenant détailler l'expression de la log-vraisemblance apparaissant dans l'éq.(1) pour des niveaux de gris distribués selon différentes lois appartenant à la famille exponentielle. Ces expressions sont présentées dans le tab.1 et sont obtenues en remplaçant les paramètres inconnus des différentes lois par leurs estimées au sens du Maximum de Vraisemblance (MV).

2.2 - Algorithme de segmentation

Une fois déterminée la longueur de code associée à chaque segmentation, le but est de déterminer la segmentation permettant de minimiser cette longueur de code, c'est-à-dire le critère MDL de l'éq.(1). Ce critère dépend de trois inconnus: le nombre de régions R ainsi que le nombre et la position des nœuds. Le but est donc de trouver leurs valeurs qui permettent de minimiser le critère MDL. Cette optimisation est effectuée en trois étapes, comme cela est illustré sur l'image synthétique de la fig.1 composée de 256×256 pixels et perturbée par un bruit gamma d'ordre 1. Le contraste (défini comme le rapport des moyennes entre deux régions) est égal à 4. L'algorithme de segmentation correspond aux étapes qui suivent:

1. Tout d'abord, on génère une grille initiale fine et régulière (cf. (b)).
2. Ensuite, afin de diminuer rapidement le nombre de régions et de nœuds, on fusionne les régions à l'aide d'un critère de type GLRT¹ (Generalized Likelihood Ratio Test) avec un seuil fixe \tilde{S} . Dans toutes les expériences présentées dans cet article, nous avons pris $\tilde{S} = 3$, permettant de réduire le nombre de régions tout en évitant une sur-fusion² (cf. (c)).
3. Nous déplaçons alors les nœuds afin d'estimer leur position en minimisant le critère MDL (cf. (d)).

¹ Nous avons dérivé ce test du critère MDL.

² C'est-à-dire une fusion conduisant à une sous-estimation du nombre de régions.

4. Puis nous supprimons les nœuds inutiles jusqu'à ce que plus aucune suppression de nœud ne fasse décroître le critère MDL (cf. (e)).

5. Enfin nous effectuons à nouveau une étape de fusion de régions, mais basée sur le critère MDL cette fois-ci et qui consiste à fusionner les régions voisines deux à deux tant que cela entraîne une baisse du critère MDL (cf. (f)).

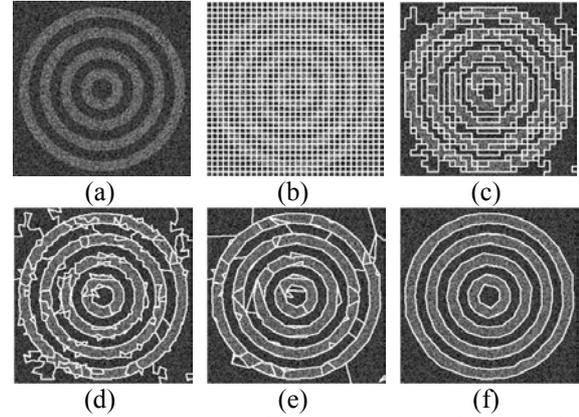


Fig.1: Segmentation d'une image synthétique de 256×256 pixels perturbée par un bruit gamma d'ordre 1. a: image - b: grille initiale - c: fusion de régions - d: déplacement des nœuds - e: suppression des nœuds inutiles - f: fusion de régions. Les niveaux de gris de l'image ont été modifiés pour une meilleure visualisations.

Afin d'assurer la convergence, il est en général nécessaire de recommencer un ou deux cycle d'optimisation (c'est-à-dire recommencer les étapes 3, 4 et 5). Il est intéressant de souligner que les étapes 3, 4 et 5 sont basées sur la minimisation d'un critère MDL unique et sans paramètre. De plus, cet algorithme utilise une technique détaillée dans [Chesnaud 1999, Germain 2001, Galland 2003] qui permet de remplacer les sommations sur des surfaces par des sommations sur des contours, ce qui conduit à un algorithme rapide (moins de 3 secondes pour l'image de la fig.1 sur un PC à 2.8 GHz).

Dans tous les exemples de segmentation présentés dans cet article, le processus de segmentation sera similaire à celui qui vient d'être décrit: seule la grille initiale pourra changer. Il est intéressant de noter que la taille de la grille initiale influence assez peu le résultat de la segmentation. De plus, il est possible de choisir automatiquement la meilleure grille au sens du MDL: il suffit pour cela de segmenter l'image avec plusieurs grilles initiales différentes et de sélectionner celle qui conduit au critère MDL minimum. Dans toute la suite, les résultats de segmentation seront présentés en utilisant cette meilleure grille initiale au sens du MDL.

pdf	α	log-vraisemblance	$\widehat{\theta}_r$
Gaussienne	2	$-\frac{1}{2} \sum_{r=1}^R N_r \log \left(\widehat{\theta}_r^{(2)} \right)$	$\widehat{\theta}_r^{(1)} = \frac{1}{N_r} \sum_{(x,y) \in \Omega_r} s(x,y)$ $\widehat{\theta}_r^{(2)} = \frac{1}{N_r} \sum_{(x,y) \in \Omega_r} s(x,y)^2 - \left(\widehat{\theta}_r^{(1)} \right)^2$
Gamma d'ordre L	1	$-L \sum_{r=1}^R N_r \log \left(\widehat{\theta}_r \right)$	$\widehat{\theta}_r = \frac{1}{N_r} \sum_{(x,y) \in \Omega_r} s(x,y)$
Rayleigh	1	$-\sum_{r=1}^R N_r \log \left(\widehat{\theta}_r \right)$	$\widehat{\theta}_r = \frac{1}{N_r} \sum_{(x,y) \in \Omega_r} s(x,y)^2$
Poisson	1	$\sum_{r=1}^R N_r \widehat{\theta}_r \log \left(\widehat{\theta}_r \right)$	$\widehat{\theta}_r = \frac{1}{N_r} \sum_{(x,y) \in \Omega_r} s(x,y)$
Bernoulli	1	$\sum_{r=1}^R N_r \left\{ \widehat{\theta}_r \log \widehat{\theta}_r + (1 - \widehat{\theta}_r) \log(1 - \widehat{\theta}_r) \right\}$	$\widehat{\theta}_r = \frac{1}{N_r} \sum_{(x,y) \in \Omega_r} s(x,y)$

Tab.1: Log-vraisemblance et nombre de paramètres α pour différentes lois de probabilités. Le vecteur de paramètres θ_r , inconnu a été remplacé par son estimée $\widehat{\theta}_r$, au sens du Maximum de Vraisemblance (MV). Les termes constants indépendants de la segmentation (mais qui dépendent de l'image et de l'ordre dans le cas Gamma) n'ont pas été pris en compte dans l'expression des log-vraisemblances.

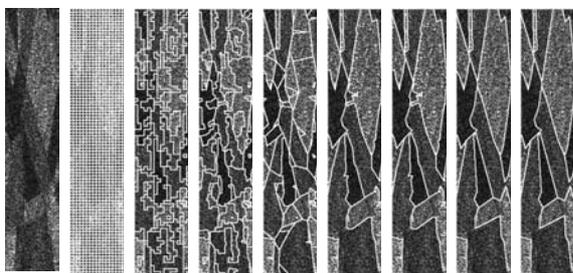


Fig.2: Segmentation d'une image SAR simple-vue (105x522 pixels) d'une région agricole près de Bourges (France) obtenue par le satellite ERS-1 (fournie par le CNES et distribuée par l'ESA). (a) image SAR - (b) grille initiale - (c) fusion de régions - (d) déplacement des nœuds - (e) suppression des nœuds inutiles - (f) fusion des régions - (g) déplacement des nœuds - (h) suppression des nœuds inutiles - (i) fusion de régions.

3 - SEGMENTATION D'IMAGES MONO-COMPOSANTES

3.1 - Résultats

Nous allons maintenant présenter, à titre d'exemple, la segmentation d'une image mono-composante réelle. La fig.2-a est une image Radar à Ouverture Synthétique (SAR) simple vue et mono-composante d'une région agricole près de Bourges (France), obtenue par le satellite ERS-1 (image fournie par le CNES et distribuée par l'ESA). Comme nous pouvons le voir, cette image est fortement perturbée par un phénomène de speckle qui peut être modélisé par une loi gamma. De plus, comme il s'agit d'une image simple-vue (c'est-à-dire qu'il n'y a pas eu de moyennage spatial ou temporel de l'image), l'ordre de cette loi gamma peut être pris égal à 1. Nous verrons dans la partie suivante qu'il est de toute façon possible d'estimer la valeur de cet

ordre. En utilisant le critère MDL adapté pour une loi gamma (cf. éq.(1) et tab.1) et en utilisant l'algorithme présenté précédemment, nous obtenons les résultats de la fig.2.

3.2 - Estimation de l'ordre du speckle

Nous allons maintenant voir que lorsque l'ordre de la loi gamma utilisé pour modéliser le speckle n'est pas connu, il est possible de l'estimer à l'aide de la complexité stochastique. Pour cela, notons tout d'abord que l'ordre intervient uniquement comme un facteur multiplicatif dans la log-vraisemblance présentée dans le tab.1. Cela signifie que plus l'ordre est élevé, plus le terme de log-vraisemblance

Δ_L est prépondérant par rapport aux autres termes Δ_P et Δ_G : une diminution de l'ordre dans le critère MDL entraîne donc une simplification de la grille. L'idée est donc la suivante. Après avoir généré une grille initiale, l'utilisateur fixe l'ordre à une valeur élevée dans le critère MDL, par exemple $L=10$ puis lance une segmentation MDL. Une fois l'algorithme terminé, la grille obtenue sert alors de grille initiale pour une nouvelle segmentation, mais en imposant cette fois un ordre $L=L-1$, etc, jusqu'à avoir $L=1$. On obtient ainsi un résultat de segmentation et une valeur du critère MDL pour chacun des 10 ordres $L=\{1,2,\dots,10\}$. Il suffit alors de comparer ces 10 valeurs du critère MDL: le minimum de ce critère fournit simultanément une estimation de l'ordre et une segmentation de l'image. Bien sûr, il est important de remarquer qu'il faut ajouter à la log-vraisemblance fournie dans le tab.1, les termes indépendants de la segmentation mais dépendant de l'ordre L . Dans le cas d'une loi gamma d'ordre L , ces termes sont les suivants:

$$N(L \log L - \log \Gamma(L) - L) + (L-1) \sum_{\text{image}} \log(s(x,y))$$

Cette approche est illustrée sur la fig.3.

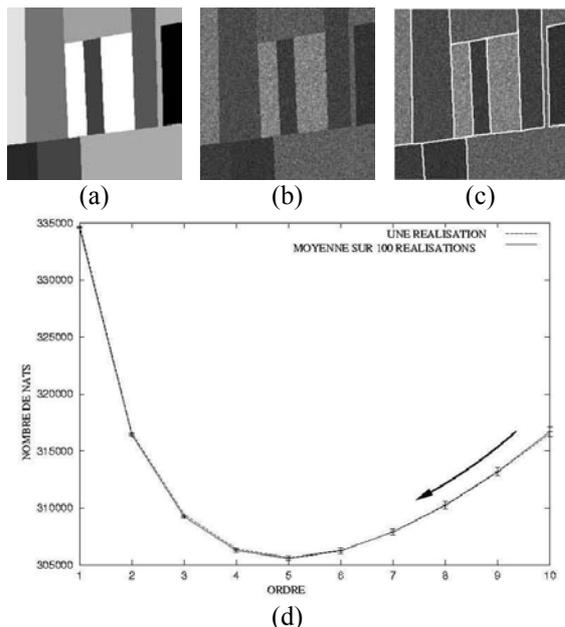


Fig.3: Segmentation avec estimation de l'ordre du speckle. On génère à partir de l'image synthétique de 256×256 pixels (a) une image perturbée par un bruit de speckle d'ordre 5 (b). L'algorithme de segmentation est le suivant: après avoir généré une grille initiale, on segmente l'image en ayant fixé un ordre $L=10$. On pose alors $L=L-1$ et on se sert de la grille issue de la segmentation précédente pour segmenter à nouveau l'image, etc, jusqu'à avoir $L=1$. La courbe (d) montre l'évolution du critère MDL lorsque L diminue (trait plein: évolution pour une réalisation - trait pointillé: moyenne sur 100 réalisations). Nous obtenons bien un minimum pour $L=5$, qui correspond à la valeur exacte de l'ordre. La segmentation correspondante est affichée sur la figure (c).

4 - SEGMENTATION D'IMAGES MULTI-COMPOSANTES

4.1 - Généralisation du critère MDL au images multi-composantes

Nous allons maintenant généraliser l'approche présentée dans les parties 2 et 3 au cas des images multi-composantes. Une image multi-composantes (c'est-à-dire vectorielle) \mathbf{s} est composée de M images mono-composantes (c'est-à-dire scalaires) s_i ($i \in \{1, 2, \dots, M\}$):

$$\mathbf{s}(x, y) = (s_1(x, y), s_2(x, y), \dots, s_M(x, y))$$

Le but de la segmentation est, comme dans le cas scalaire, de diviser l'image \mathbf{s} en R régions homogènes Ω_r , ce qui signifie que la partition de l'image est la même pour toutes les composantes. Dans chaque région, les vecteurs des niveaux de gris de chaque pixel $\mathbf{s}(x, y)$ suivent une densité de

probabilité P_{θ_r} de vecteur de paramètres θ_r . De plus, chacune des composantes s_i est supposée indépendante, ce qui signifie que $P_{\theta_r} = \prod_{i=1}^M P_{\theta_r^i}^i$ où $P_{\theta_r^i}^i$ est la pdf des niveaux de gris $s_i(x, y)$, $(x, y) \in \Omega_r$ de la $i^{\text{ème}}$ composante et θ_r^i son vecteur de paramètres.

Réécrivons maintenant le critère MDL obtenu en (1) pour les images multi-composantes. Il est facile de voir que le terme de codage de la grille Δ_G reste le même dans les cas mono et multi-composantes, la grille étant commune à toutes les composantes de l'image. Ainsi, dans l'éq.(1), seuls le nombre de paramètres de la densité de probabilité et l'expression de la log-vraisemblance changent. Grâce à l'hypothèse d'indépendance de chaque composante, on peut écrire:

$$l_e[\Omega_r | \theta_r] = \sum_{i=1}^M l_e[\Omega_r^i | \theta_r^i] \quad (2)$$

où $\Omega_r^i = \{s_i(x, y), (x, y) \in \Omega_r\}$ et où $l_e[\Omega_r^i | \theta_r^i]$ est la log-vraisemblance de la région Ω_r pour la $i^{\text{ème}}$ composante. On obtient également:

$$\alpha = \sum_{i=1}^M \alpha_i \quad (3)$$

où α_i est le nombre de paramètres de la pdf $P_{\theta_r^i}^i$ de la $i^{\text{ème}}$ composante.

En remplaçant ces équations (2) et (3) dans le critère MDL pour les images scalaires de l'éq.(1), on obtient le critère MDL multi-composantes suivant:

$$\Delta = n(\log N + \log p) + p(\log(2\hat{m}_x) + \log(2\hat{m}_y)) \quad (4)$$

$$+ \log p + \sum_{r=1}^R \left[\left(\sum_{i=1}^M \frac{\alpha_i}{2} \right) \log N_r - \sum_{i=1}^M l_e[\Omega_r^i | \theta_r^i] \right]$$

Dans cette équation, comme α_i et $l_e[\Omega_r^i | \theta_r^i]$ sont le nombre de paramètres et la log-vraisemblance de la $i^{\text{ème}}$ composante, c'est-à-dire une image scalaire, on peut utiliser leurs expressions données dans le tab.1.

L'optimisation de ce critère se fait de la même manière que dans le cas des images mono-composantes. Nous allons maintenant présenter quelques résultats de segmentation d'images vectorielles.

4.2 - Images de type multi-dates

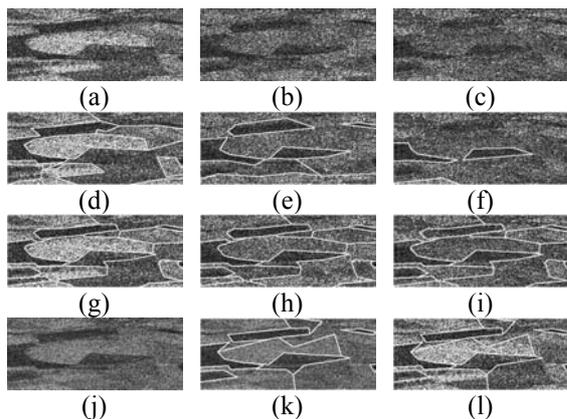


Fig.4: Segmentation d'une image SAR 3-dates obtenue par le satellite ERS-1 (fournie par le CNES et distribuée par l'ESA). Les 3 dates (a),(b),(c) sont des images SAR simple vue (150×350 pixels) de zones agricoles près de Bourges (France). Les résultats de segmentation avec une approche scalaire sont affichés sous chacune des dates (cf. (d),(e),(f)). La segmentation obtenue avec une approche vectorielle a été superposée sur les 3 dates (cf. (g),(h),(i)). De plus, une segmentation scalaire sur la somme des trois dates a été effectuée: l'image de la somme est présentée en (j) et le résultat de la segmentation a été superposé sur la somme (cf. (k)) et sur la première date (cf.(l)).

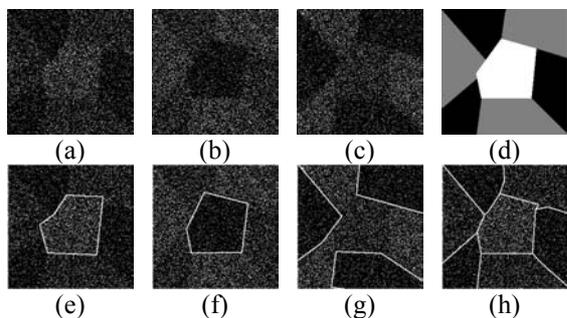


Fig.5: Segmentation d'une image synthétique 3-dates (256×256 pixels) perturbée par un bruit de Bernoulli. (a),(b),(c): images des 3 dates; la moyenne dans chaque région diffère d'une date à l'autre; de plus la somme des trois dates est un bruit uniforme sur toute l'image et ne contient donc aucune information exploitable. (d): image non bruitée. (e),(f),(g): résultats de segmentation avec une approche scalaire sur chacune des 3 dates. (h): segmentation avec une approche vectorielle sur les 3 dates; le résultat a été affiché sur la date (a).

Dans cette partie, nous allons nous intéresser aux images de type multi-dates. Dans le cas des images SAR, elles sont obtenues en observant la même zone à plusieurs dates différentes. La végétation ayant changé entre-temps, il en résulte que les moyennes varient entre chacune des différentes dates. On obtient ainsi une image où la famille de la densité de probabilité des niveaux de gris est la

même pour toutes les composantes (à savoir une loi gamma d'ordre 1), mais où le paramètre de cette loi (la moyenne) au sein d'une même région varie selon la composante considérée. Nous présentons la segmentation d'une image SAR vectorielle composée de 3 dates sur la fig.4. La segmentation d'une image synthétique de type 3-dates perturbée par un bruit de Bernoulli est présentée sur la fig.5. Ces deux exemples montrent l'apport d'une approche vectorielle par rapport à une approche scalaire sur une seule composante ou même sur la somme des différentes composantes.

4.3 - Images de type multi-capteurs sans recalage

Nous allons maintenant nous intéresser à un autre type d'images multi-composantes. Dans la partie précédente, nous avons étudié le cas d'images prises par le même capteur mais à des dates différentes. Il est également possible d'imaginer l'acquisition simultanée d'images à l'aide de plusieurs capteurs. Cela suppose bien sûr que les différents capteurs soient pointés rigoureusement sur la même scène, ou sinon que les différentes images soient recalées entre elles. Pour l'instant nous supposons que les images ont été préalablement recalées: nous verrons dans la partie suivante que le problème du recalage peut être abordé en même temps que la segmentation.

Le bruit présent dans une image dépend fortement du capteur. Dans la suite, nous traiterons le cas d'une image synthétique pouvant être obtenue en combinant un système d'imagerie active et une caméra à faible flux (cf. fig.6). L'image fournie par ces deux capteurs sera alors perturbée par un bruit de speckle pour la première (b) alors que ce sera un bruit de Poisson pour la deuxième (c). De plus, comme nous supposons que ces deux images sont obtenues simultanément, la réflectivité (c'est-à-dire la moyenne) dans chaque région est la même quelle que soit l'image considérée. Nous obtenons ainsi une image à deux composantes ne différant que par la distribution de leurs niveaux de gris (et non par la moyenne de ces niveaux de gris).

Sur la fig.6, nous pouvons voir le résultat de la segmentation lorsque l'on prend en compte seulement la première image (cf. (e)): seule la partie basse a été correctement segmentée, alors que nous obtenons le résultat inverse en travaillant sur la deuxième image (cf. (f)). Cela résulte du fait que le contraste dépend du type de bruit présent dans l'image. En effet, bruitée par du speckle, l'image (a) aura un meilleur contraste sur le bas de l'image que sur le haut, alors que ce sera le contraire si cette même image est bruitée par un bruit de Poisson. Une approche vectorielle permet alors de combiner les avantages des deux capteurs:

la segmentation obtenue est en effet correcte aussi bien dans la partie haute que dans la partie basse de l'image (g-h).

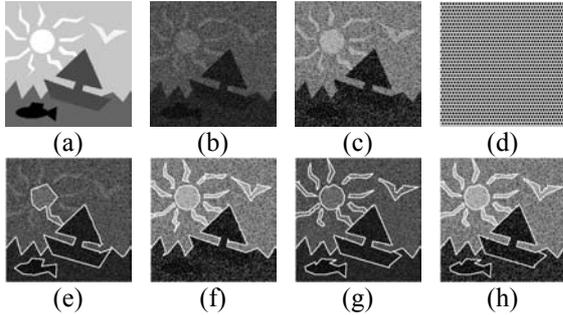


Fig.6: Segmentation d'une image de type multi-capturer de 256×256 pixels (a) et perturbée par un bruit de speckle d'ordre 1 (b) et par un bruit de Poisson (c). La grille initiale est affichée en (d). La fig.(e) (resp. (f)) a été obtenue en effectuant la segmentation uniquement sur l'image (b) (resp. (c)). Avec une approche vectorielle sur les deux images (b) et (c) simultanément, nous obtenons les fig.(g) et (h) où la grille a été superposée sur chacune des deux images (b) et (c). Pour des questions de visualisation, les niveaux de gris ont été modifiés à l'affichage.

4.4 - Segmentation avec recalage simultané

Dans la partie précédente, nous avons supposé que les images issues de capteurs différents avaient été préalablement recalées. Nous allons maintenant montrer qu'il est possible d'effectuer simultanément le recalage et la segmentation d'une image à deux composantes.

Pour cela nous proposons d'estimer le décalage entre les deux images à l'aide du critère MDL de l'éq.(4): le meilleur recalage au sens du MDL peut être défini comme le recalage qui permet de minimiser la longueur de description des deux images. Pour recalcer ces deux images, il suffit donc de décaler une des deux images puis de segmenter les deux images avec une approche vectorielle. On obtient ainsi la segmentation et la valeur du critère MDL pour un décalage donné. On compare ensuite les valeurs du critère MDL obtenues pour différents décalages: le minimum de ce critère nous donne accès simultanément à la valeur du décalage à effectuer et à la segmentation associée.

Illustrons maintenant cette approche sur l'exemple de la fig.7. Pour cela, nous avons généré à partir d'une image synthétique deux images avec des moyennes différentes et bruitées par un bruit de speckle d'ordre 1 pour la première (fig.(a), première ligne) et un bruit de Poisson pour la deuxième (fig.(a), deuxième ligne). Un décalage de 5 pixels horizontalement et verticalement a été généré entre ces deux images. Nous procédons alors de la façon

suivante: nous translatons la deuxième image d'un vecteur $(d_x, d_y) \in [-20, +20]^2$ puis nous segmentons l'image bi-composante ainsi obtenue. La valeur minimum du critère MDL est obtenue pour $d_x = 5$ et $d_y = 5$, ce qui correspond au vrai décalage. Sur la colonne (b) de la fig.7, nous avons présenté les résultats de segmentation obtenus lorsque $d_x = d_y = 0$ et sur la colonne (c) lorsque $d_x = d_y = 5$. La grille a été superposée sur la première composante (ligne 1) et la deuxième (ligne 2). La colonne (c) représente le résultat de la segmentation vectorielle effectuée sur les images recalées. Cet exemple permet de voir que lorsque les images ne sont pas recalées (cf. colonne (b)), cela entraîne une surestimation de la complexité de la grille Δ_G et l'apparition de régions inhomogènes, c'est-à-dire une surestimation de la complexité des niveaux de gris Δ_L .

L'intérêt principal de cette méthode de recalage est d'être applicable même dans les cas où moyennes et lois de probabilités diffèrent entre les deux images à recalcer.

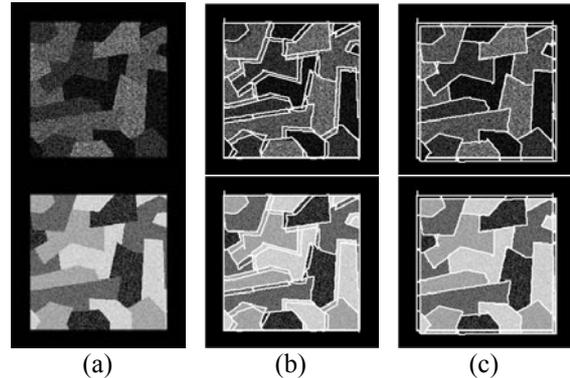


Fig.7: Segmentation avec recalage simultané d'une image synthétique bi-composante (256×256 pixels). Colonne (a): image bi-composante. Colonne (b): segmentation vectorielle sans recalage. Colonne (c): segmentation vectorielle après un décalage de la composante 2. La ligne 1 est relative à la première composante (bruit de speckle d'ordre 1) et la ligne 2 à la deuxième (bruit de Poisson). Les niveaux de gris des images ont été modifiés pour une meilleure visualisation.

5 - CONCLUSION

Nous avons proposé un nouvel algorithme de segmentation d'images en zones homogènes qui repose sur la minimisation d'un critère MDL, ainsi que son extension au cas des images multi-composantes. Les résultats présentés montrent l'efficacité de cette approche pour traiter des images fortement bruitées. De plus, nous avons mis en

évidence l'apport d'une approche vectorielle par rapport à une approche scalaire, que ce soit pour traiter des images de type multi-dates ou multi-capteurs. Par ailleurs, nous avons développé une nouvelle approche permettant d'estimer au sens du MDL différents paramètres intervenant lors de l'optimisation ou dans le modèle d'image: nous avons ainsi vu qu'il était possible de segmenter une image tout en estimant simultanément la taille de la grille initiale, l'ordre du bruit de speckle ou même le décalage entre deux composantes d'une image vectorielle.

6 - REMERCIEMENTS

Nous tenons à remercier le CNES qui nous a fourni les images SAR ainsi que F.Goudail et P.Martin pour leurs précieux conseils.

BIBLIOGRAPHIE

- Rissanen, J. (1978), *Modeling by shortest data description*, Automatica, vol.14, p.465-471.
- Rissanen, J. (1989), *Stochastic Complexity in Statistical Inquiry*, World Scientific, Singapore.
- Leclerc, Y.G. (1989), *Constructing simple stable descriptions for image partitioning*, Computer Vision, vol.3, p.73-102.
- Zhu, Z.C., Yuille, A. (1996), *Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation*, IEEE Trans. Pattern Analysis and Machine Intelligence, no.9, p.884-900.
- Figueiredo, M., Leitão, J., Jain, A.K. (2000), *Unsupervised contour representation and estimation using B-splines and a minimum description length criterion*, IEEE Trans. Image Processing, vol.9, p.1075-1087.
- Staib, L.H., Duncan, J.S. (1992), *Boundary finding with parametrically deformable contour models*, IEEE Trans. Pattern Anal. and Machine Intell., vol.14, no.11, p.1061-1075.
- Kervrann, C., Heitz, F. (1994), *A hierarchical statistical framework for the segmentation of deformable objects in image sequences*, IEEE Conf. Comp. Vision Pattern Recognition, Seattle, p.724-728.
- Storvik, G. (1994), *A Bayesian approach to dynamic contours through stochastic sampling and simulated annealing*, IEEE Trans. Pattern Anal. Machine Intell., vol.16, no.10, p.976-986.
- Nguyen, C., Herlin, I., Graffigne, C. (1992), *A deformable region model using stochastic processes applied to echadiographic images*, Conf. on Computer Vision and Pattern Recognition, Urbana Chapagn, USA, IEEE.
- Germain, O., Réfrégier, Ph., (1996), *Optimal snake-based segmentation of a random luminance target on a spatially disjoint background*, Opt. Lett., vol.21, no.22, p.1845-1847.
- Germain, O., Réfrégier, Ph. (2001), *Statistical active grid for segmentation refinement*, Pattern Recognition Letters, vol.22, no.3, p.1125 -1132.
- Ruch, O., Réfrégier, Ph., (2001), *Minimal complexity segmentation with a polygonal snake adapted to different noise models*, Optical Letter, vol.22, p.1125-1132.
- Germain, O., Réfrégier, Ph. (1999), *Snake-based method for the segmentation of objects in multichannel images degraded by speckle*, Optics Letters, vol.24, no.12, p.814-816.
- Galland, F., Bertaux, N., Réfrégier, Ph. (2003), *Minimum description length synthetic aperture radar image segmentation*, IEEE Trans. on Image Processing, vol.12, no.9, p.995-1006.
- Chesnaud, C., Réfrégier, Ph., Boulet, V. (1999), *Statistical region snake-based segmentation adapted to different physical noise models*, IEEE Trans. Pattern Analysis and Machine Intelligence, vol.21, no.11, p.1145-1157.

**DESCRIPTION ET CONCEPTION D'UNE PLATE-FORME ROBUSTE
COMBINANT DES ANALYSEURS D'ENONCE**

Brunet-Manquat Francis,

Doctorant en Informatique

Francis.Brunet-Manquat@imag.fr , + 33 4 76 51 43 59

Adresse professionnelle

Laboratoire CLIPS-IMAG, équipe GETA ★ BP 53 ★ 38041 Grenoble Cedex 9

Résumé : L'objectif de cet article est de présenter nos travaux sur la combinaison d'analyseurs pour produire une analyse robuste de la langue. L'idée est de créer une plate-forme robuste d'analyse capable d'intégrer des analyseurs linguistiques existant (analyseurs syntaxiques ou/et de dépendances), et de fusionner leurs résultats dans le but d'obtenir une analyse de dépendances pour des énoncés quelconques.

Summary : The article's goal is to present our work about the combination of parsers to produce a robust parsing. The idea is to create a robust analysis tool capable of integrating existing linguistic parsers (syntactic parsers or dependency parsers), to merge their results and to calculate the best information to obtain a dependency parse of the input sentence.

Mots clés : Traitement automatique de la langue naturelle, analyseur de dépendances, analyseur syntaxique.

Description et conception d'une plate-forme robuste combinant des analyseurs d'énoncé

L'objectif de cet article est de présenter nos travaux sur la combinaison d'analyseurs pour produire une analyse robuste de la langue. L'idée est de créer une plate-forme robuste d'analyse capable d'intégrer des analyseurs linguistiques existant (analyseurs syntaxiques ou/et de dépendances), et de fusionner leurs résultats dans le but d'obtenir une analyse de dépendances pour des énoncés quelconques.

1 - INTRODUCTION

Notre laboratoire est impliqué dans deux projets internationaux importants : CSTAR et son projet européen associé NESPOLE ! (<http://nespole.itc.it>) pour la traduction simultanée de l'oral et Universal Networking Language, UNL (<http://www.unl.ias.unu.edu>), pour la traduction de l'écrit. Ces projets se caractérisent notamment par la présence d'une représentation pivot des énoncés et par le fait que l'énoncé à traduire est susceptible d'être *bruité*, c'est-à-dire pas forcément conforme à la grammaire académique de la langue française. Dans un système à pivot, l'énoncé d'une langue source donnée est *analysé* dans la représentation pivot avant d'être *généré* vers une ou plusieurs langues cibles. La nécessité de pouvoir traiter des entrées bruitées implique des outils robustes d'analyse de la langue, capable de fournir une analyse même partielle de la phrase.

L'idée est de concevoir une telle plate-forme en prenant le « meilleur » d'un ensemble d'analyseurs. Le principe étant de combiner différents résultats d'analyse pour une même phrase, puis de calculer les meilleures informations pour obtenir la ou les meilleures analyses possibles. Notre idée s'inspire du système ROVER, Recognizer Output Voting Error Reduction [Fis97], qui combine des résultats de systèmes de reconnaissance de parole pour obtenir la meilleure reconnaissance possible.

Notre approche se base principalement sur la méthode dite du « vote à la majorité », plus une information sera commune aux différents analyseurs, plus le poids de cette information

sera fort, mais également sur un apprentissage permettant d'adapter les poids associés aux informations fournis en fonction de l'énoncé (par exemple, entrée bruitée ou non). À partir d'une telle approche, nous comptons construire un analyseur de dépendances robuste car regroupant un maximum d'informations issues d'une combinaison d'analyseurs.

Nous décidons de fonder notre plate-forme sur une représentation par dépendances, décrivant les relations entre mots (les relations peuvent être de tous types : syntaxique, logique ou sémantique). Fort des différents projets réalisés, ce type de représentation nous semble plus adapté pour une analyse robuste. Elle nous permet, par exemple, de représenter clairement et simplement l'analyse partielle d'une phrase mal formée.

2 – SPECIFICATIONS DE LA PLATE-FORME D'ANALYSE

La plate-forme d'analyse ne doit pas intégrer les analyseurs, mais elle doit être capable d'extraire les informations linguistiques des résultats qu'ils produisent, de les interpréter, de les fusionner et enfin de produire un arbre de dépendances (ou plusieurs) combinant le meilleur des informations extraites.

2.1 - Les différentes étapes du traitement

La plate-forme comporte deux étapes : la *normalisation* des résultats d'analyse et la *construction* des analyses de dépendances.

L'étape de normalisation se compose de deux phases :

- La phase d'*extraction* des informations linguistiques des résultats obtenus avec chaque analyseur linguistique ;
- La phase de *projection* qui traite les informations extraites pour obtenir un ensemble de structures de dépendances dite normalisée, conforme à notre norme de description linguistique (voir 3.2.3 – Vers une norme de description linguistique). Chaque information normalisée se verra associé un indice de confiance donné en

fonction de l'analyseur l'ayant fournie et de l'énoncé traité (Cet indice est pré-calculé par un processus d'apprentissage sur un corpus de références).

L'étape de construction se compose de deux phases :

- La phase de *fusion* qui permet, dans un premier temps, de faire *correspondre* les différentes structures normalisées fournies par l'étape précédente, et, dans un second temps, de *fusionner* les informations de ces structures, à l'aide des correspondances établies précédemment, pour donner une unique représentation de dépendances contenant toutes les informations linguistiques possibles, même les informations contradictoires. Toutes ces informations ainsi fusionnées verront leurs indices de confiance recalculés ;
- La phase de *production* qui permet de construire une ou plusieurs structures de dépendances en fonction des informations fusionnées, des indices de confiance recalculés et de contraintes linguistiques et structurelles de production.

2.2 – Architecture fonctionnelle

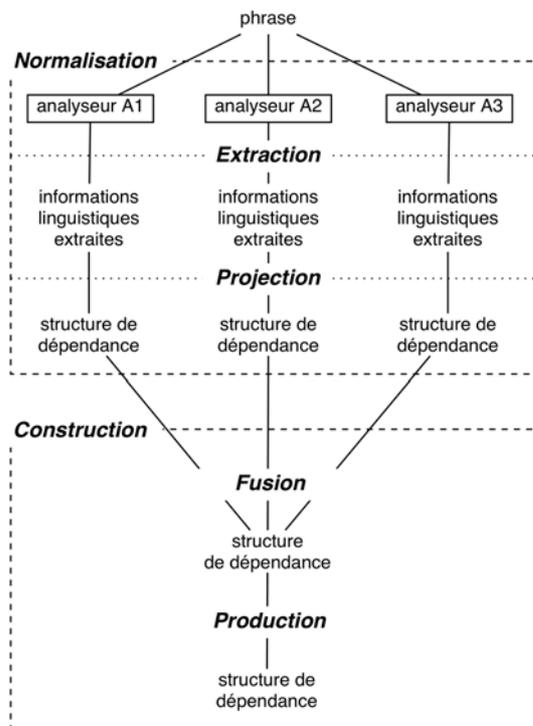


figure 1 : Architecture fonctionnelle de la plate-forme d'analyse

3 – NORMALISATION DES RESULTATS D'ANALYSE

La première étape consiste à normaliser chaque résultat d'analyse, c'est-à-dire transformer les différents résultats d'analyse obtenus (constituants et/ou dépendances) en un ensemble de structures de dépendances.

3.1 - Structure de dépendances

3.1.1 - Définition

Une structure de dépendances permet de décrire les relations entre les mots d'un énoncé. Les relations peuvent être de tous types : syntaxique, logique ou sémantique. Une structure de dépendances est composée d'un ensemble de nœuds et d'un ensemble d'arcs :

- Un *nœud* est associé à un mot d'un énoncé. Il est composé d'un ensemble d'informations linguistiques correspondant à ce mot (catégorie morphosyntaxique, variables grammaticales, variables sémantiques et variables logiques) ;
- Un *arc* entre deux nœuds correspond à une relation linguistique entre deux mots d'un énoncé (relation syntaxique, sémantique ou logique).

Exemple de structure de dépendances syntaxiques représentant la phrase (a) « le vieux chat attrape la souris. »

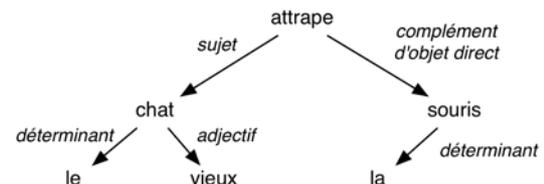


figure 2 : arbre de dépendances syntaxiques

Nous pouvons également ajouter d'autres types d'informations linguistiques à cette structure comme des informations sémantiques :

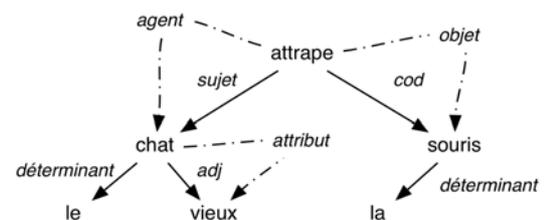


figure 3 : graphe de dépendances syntaxiques et sémantiques

3.1.2 – Matrice de dépendances (MD)

Les structures de dépendances sont décrits dans notre plate-forme par une représentation matricielle. Notre représentation, nommée *matrice de dépendances MD*, est un couple $\langle L, M \rangle$ composée :

- D'une *liste de nœuds (L)*, un nœud étant composé d'informations linguistiques relatives aux mots qu'il décrit ;
- D'une *matrice carrée (M)* permettant de décrire les arcs entre nœuds. La case (i, j) contient l'ensemble des arcs (relations) entre le nœud i et le nœud j de la liste de nœuds.

La représentation matricielle correspondant au graphe de dépendances multi-niveau (figure 3) de la phrase (a) sera :

L =

le	::	cat=déterminant
vieux	::	cat=adjectif
chat	::	cat=nom
attrape	::	cat=verbe, temps=présent
la	::	cat=déterminant
souris	::	cat=nom

M =

	le	vieux	chat	attrape	la	souris
le						
vieux						
chat	det	adjectif attribut				
attrape			sujet agent			cod objet
la						
souris					det	

Une représentation matricielle des données présente de nombreux avantages pour le traitement informatique :

- *Maniabilité* : De nombreux outils mathématiques sont associés aux matrices : ajout, suppression, comparaison, etc. Tous ces outils permettent un traitement simple de l'information contenue dans une matrice ;
- *Efficacité* : Les méthodes utilisant les matrices comme structure de données, telle que la reconnaissance de motifs ou la fusion de matrice, se montre très efficaces et très simple à mettre en place.

Notre représentation matricielle permet également de représenter différents contenus linguistiques simplement :

- *Multi-niveau* : elle peut regrouper plusieurs niveaux linguistiques (syntaxique, sémantique, logique, etc.) comme dans l'exemple précédent ;
- *Ambiguïtés* : elle peut regrouper des informations contradictoires avant la phase de production des arbres de dépendances. Par exemple, suite à la phase de fusion, deux dépendances syntaxiques, *sujet* et *objet*, entre les mots *vieux* et *chat* ont été combinés dans la MD. Ces informations seront traitées lors de la phase suivante, la phase de production (voir 4.3 - Production des arbres de dépendances).

3.2.3 – Vers une norme de description linguistique

Pour construire notre plate-forme combinant différents analyseurs, il est nécessaire de connaître les caractéristiques de chaque analyseur et particulièrement les caractéristiques de leur résultat. Pour ce faire, une première étude a été réalisée permettant de classer les différentes informations linguistiques produites par chaque analyseur. Cette classification nous permet d'établir une norme de description linguistique à utiliser lors de l'étape de normalisation des informations.

Les analyseurs syntaxiques

Pour cette première étude, nous avons utilisé 3 analyseurs du Français : IFSP (Incremental Finite-State Parser) développé par XEROX, l'analyseur du GREYC développé par Jacques Vergne et l'analyseur du projet LIDIA du GETA.

Nous pouvons distinguer deux types d'analyseurs : les analyseurs linguistiques, fondés sur des formalismes grammaticaux, et les analyseurs probabilistes, fondés sur l'apprentissage à partir de corpus. Pour notre étude et la réalisation de notre plate-forme, nous avons choisi d'utiliser dans un premier temps des analyseurs linguistiques. La plupart de ces analyseurs se répartissent en trois catégories en fonction des résultats qu'ils fournissent [Mon02] :

- *Les analyseurs fondés sur les constituants* qui retournent une segmentation en groupes.

- Les *analyseurs fondés sur les dépendances* qui retournent les dépendances entre mots d'une phrase.
- Les *analyseurs fondés sur les constituants et les dépendances* qui retournent une segmentation en groupes et des relations de dépendances entre ces groupes et/ou entre les mots.

Les 3 analyseurs utilisés pour notre étude se situent dans cette dernière catégorie.

Exemples de résultats à traiter :

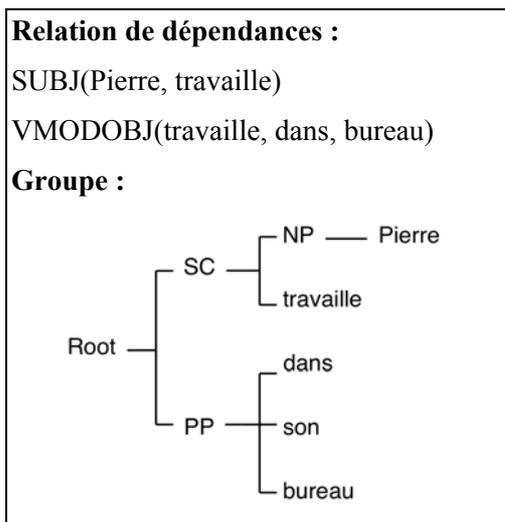


figure 4 : Résultat filtré de l'analyseur IFSP pour la phrase "Pierre travaille dans son bureau."

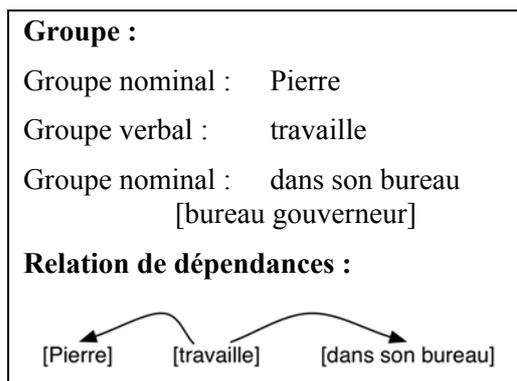


figure 5 : Résultat filtré de l'analyseur développé par J. Vergne pour la phrase "Pierre travaille dans son bureau."

La norme

Cette étude nous a permis de répertorier les différentes informations produites par les analyseurs (relations entre mots, groupements de mots, etc.) et de comparer les différentes particularités de chaque analyseur afin de

vérifier la présence d'informations utiles pour réaliser une analyse de dépendances. Cette étude nous permet surtout d'évaluer la complémentarité des analyseurs et de définir certains des critères à utiliser lors de la fusion des résultats des différents analyseurs. Nous avons classé toutes ces informations linguistiques en 6 catégories. Notre norme est donc constituée de :

- *Syntagmes* : groupe nominal, groupe verbal, groupe adjectival, groupe adverbial, groupe prépositionnel, proposition relative, proposition subordonnée, etc.
- *Relations syntaxiques* : sujet, objet, gouverneur, complément d'agent, complément circonstanciel, coordination, apposition, inclusion, etc.
- *Relations logiques* (relations prédicatives) : numérotation des arguments, etc.
- *Relations sémantiques* : contexte, but, cause, conséquence, instrument, bénéficiaire, conséquence, manière, matière, lieu, qualification, etc.
- *Catégories morpho-syntaxiques* : nom, verbe, adjectif, adverbe, subordonnant, coordonnant, ponctuation, mot inconnu, pronom, préposition, déterminant, participe, etc.
- *Variables grammaticales* : genre, nombre, temps, mode, personne, etc.

	Analyseur Vergne	Analyseur Xelda	Analyseur Lidia
Syntagmes			
Groupe nominal	x	x	x
Groupe verbal	x	x	x
Groupe prépositionnel	x	x	x
Groupe adjectival		x	x
Groupe adverbial			x
Proposition relative	x		x
Proposition infinitive			x
Proposition participiale	x	x	x
Proposition subordonnée	x		x
Relations syntaxiques			
Sujet	x	x	x
Objet	x	x	x
Complément d'agent			x
Complément circonstanciel			x
Coordination	x		x
⋮			

figure 6 : Extrait d'un tableau de classement

Nous avons cherché à détailler chacune de ces six catégories puis de faire correspondre, si possible, à chaque information détaillée les informations fournies par les analyseurs (voir figure 6). Ceci nous permet de voir plus clairement quelles informations sont disponibles, quelles informations pourront être contradictoires et surtout quelles informations seront complémentaires.

3.3 – Extraction des informations contenues dans les résultats d’analyse

La première phase consiste à extraire les informations des résultats fournis par chaque analyseur. Dans un premier temps, il faut être capable de décrire le format de représentation de ces résultats, dans notre cas nous décrivons ce format sous la forme d’une grammaire JAVACC (un générateur d’analyseurs syntaxiques pour JAVA), nommé grammaire de description. Puis à l’intérieur de cette grammaire, nous introduisons des méthodes d’extraction en fonction de la catégorie du résultat (constituant et/ou dépendances) :

- AjoutRelation(TypeRelation, Mot1, Mot2)
- AjoutSyntagme(TypeSyntagme, Ensemble de Mots)
- AjoutCategorie(Mot, Information Linguistique).

3.3 – Projection des informations extraites

Après avoir extrait toutes les informations linguistiques des résultats, la seconde phase consiste à *normaliser* toutes ces données, c’est-à-dire les faire correspondre à une structure de dépendances (un ensemble de relations entre nœuds) normalisée, correspondant à la norme établie précédemment.

3.3.1 – Règles de normalisation

Pour normaliser toutes ces informations extraites de chaque analyseur, nous nous aidons du tableau de classement décrit précédemment (voir figure 6). Ce tableau fait correspondre à chaque donnée extraite son information normalisée (catégorie, variable grammaticale ou relations entre deux nœuds). Par exemple, pour le résultat fourni par IFSP, la relation SUBJ sera normalisée pour correspondre à la relation SUJET et ainsi de suite pour toutes les relations et les catégories simples à normaliser. Mais ce tableau devra également contenir, pour certaines informations, des renseignements

supplémentaires concernant la normalisation de celle-ci. Par exemple, la relation SUBPASS (sujet passif) extraite de l’analyse de IFSP pourra être transformée en une relation sujet grâce à la règle de normalisation qui permet d’inverser les nœuds de la relation :

```
SUBPASS($var11, $var2) := SUJET($var2, $var1)
```

D’autres relations posent des problèmes plus difficiles par exemple la relation VMODOBJ extraite de l’analyse de IFSP pourrait avoir plusieurs significations syntaxiques : complément circonstanciel, complément d’objet direct, etc. L’idée est de fournir toutes les projections possibles vers notre norme et d’associer à chacune de ces informations normalisées un indice de confiance²:

```
VMODOBJ($var1, $prep, $var2) :=  
  COMPLEMENT_CIRCONSTANCIEL($var1, $var2)  
  indice_de_confiance = 0,8  
  
VMODOBJ($var1, $prep, $var2) :=  
  COMPLEMENT_OBJET_DIRECT($var1, $var2)  
  indice_de_confiance = 0,2
```

Cet indice sera pré-calculé à l’aide d’une phase d’apprentissage réalisée à partir d’un corpus de référence³. Cet indice exprime la confiance relative à l’information en fonction de l’analyseur et de la typologie de l’énoncé. Cet indice sera recalculé par la suite lors de la phase de fusion (voir 4.2 – Fusion des informations linguistiques).

3.3.2 - Première expérimentation

Cette expérimentation nous a surtout permis de définir clairement la spécification et les différents processus à utiliser pour concevoir notre plate-forme.

Nous avons réalisé une première maquette de notre outil ne comportant que les deux premiers phases : extraction et projection en utilisant une représentation classique en arbre pour représenter nos données. Cette maquette

¹ La variable \$var représente soit un mot soit un syntagme.

² L’indice varie de 0 à 1, 1 étant l’indice de confiance maximum.

³ Le module d’apprentissage est en cours de réalisation et sera basé sur une méthode de correction d’indices utilisant des techniques de rétro-propagation de données. Pour le moment, les indices de confiance sont fixés manuellement.

ne charge que deux analyseurs : l'analyseur de la plateforme Xelda et l'analyseur développé par Jacques Vergne. Cette première réalisation avait pour but de vérifier nos hypothèses de départ, c'est-à-dire : la possibilité d'extraire facilement de l'information et de la normaliser, ainsi que la possibilité d'intégrer facilement de nouveaux analyseurs à une plateforme d'analyse [Bru03].

Le corpus de phrases utilisé provient d'un corpus UNL, Universal Networking Language, contenant une phrase et son hypergraphe UNL associé. Ce corpus était constitué de quarante phrases courtes, neuf mots par phrase en moyenne (minimum 3 mots, maximum 56 mots), décrivant de nombreux phénomènes linguistiques : coordination, négation, relative, etc. Par exemple :

- Une tulipe est plus belle qu'une rose.
- Il sait que tu ne viendras pas et il ne le regrette pas.

Dans un premier temps, nous avons donc traité ces quarante phrases avec l'analyseur de Xelda et avec l'analyseur développé par J. Vergne. Ce premier traitement, appliqué à toutes les phrases du corpus, nous a permis de déterminer quelques règles de normalisation à associer aux informations fournies par les analyseurs. Pour cette expérimentation, nous avons 15 règles de normalisation associées à chaque analyseur concernant essentiellement les relations syntaxiques.

Pour intégrer les deux analyseurs à notre plateforme, nous avons donc réalisé leurs grammaires de description puis associé les méthodes d'extraction. Les règles de normalisation pour chaque analyseur ont ensuite été développées. Notre outil a extrait toutes les informations linguistiques fournies par les résultats d'analyse. Avec seulement 15 règles de normalisation, notre outil a déterminé 78% des liaisons entre mots (relations non étiquetées) et 67% de ces relations ont pu être étiquetées. Les premiers résultats normalisés fournis pour notre outil sont encourageants, même s'ils restent limités à quelques phrases et quelques règles de normalisation, mais surtout les hypothèses de départ ont pu être vérifiées.

Mais plusieurs problèmes se sont présentés, tout d'abord, la lourdeur de la représentation arborescente utilisée. En effet, nous nous sommes vite rendu compte de l'utilité de

changer de représentation pour les deux premières phases mais surtout pour la future phase de fusion et d'adopter la représentation de type matricielle présentée précédemment, plus maniable et plus adaptée à nos besoins (voir 3.1.2 – Matrice de dépendances (MD)). Le second problème concerne le besoin d'informations linguistiques supplémentaires dans le cas des analyseurs fondés sur les constituants. Comment trouver les relations entre mots à partir des syntagmes si aucune information n'est pas fournie concernant le gouverneur du syntagme. Dans le cas de l'analyseur développé par J. Vergne, l'information concernant le gouverneur du syntagme est fournie mais pas dans le cas de l'analyseur de Xelda. Nous aurons donc besoin d'une base de connaissances linguistiques simple permettant de résoudre ce genre de problèmes.

4 – CONSTRUCTION D'UNE ANALYSE DE DÉPENDANCES

À la fin de l'étape de normalisation, un ensemble d'arbres de dépendances est associé à chaque phrase. La phase suivante consiste à fusionner toutes ces structures arborescentes pour obtenir une unique représentation de dépendances contenant toutes les informations linguistiques présentes dans ces structures (catégories, variables grammaticales, relations). Pour réaliser cette fusion, il faut, dans un premier temps, mettre en *correspondance* ces structures arborescentes. En effet, l'un des problèmes rencontrés lors de la fusion est le problème des différentes segmentations possibles fournies par chaque analyseur pour une phrase donnée.

4.1 - Correspondance de structures

4.1.1 - Arbre de dépendance étendu

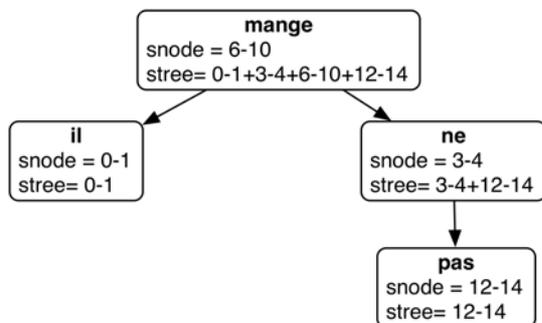
Pour réaliser la correspondance entre plusieurs arbres de dépendance, il faut être capable de déterminer si deux nœuds de structures différentes représentent la même segmentation. Pour se faire, nous étendons la définition de nos arbres de dépendance en nous basant sur la proposition de [Boi88], *Structured String-Tree Correspondences (SSTC)*. Cette proposition nous permet de réaliser la correspondance entre la phrase et l'arbre de dépendance associé et, par extension, nous permet de déterminer facilement si deux nœuds

correspondent à la même segmentation dans la phrase (même chaîne).

Nous ajoutons donc à la structure de dépendance des informations relatives à la correspondance entre la phrase et la structure arborescente associée. Nous attachons à chaque nœud N de l'arbre deux séquences d'intervalles :

- $SNODE(N)$: séquence d'intervalles représentant la sous-chaîne correspondante au nœud N.
- $STREE(N)$: séquence d'intervalles représentant les sous-chaînes correspondantes aux nœuds contenu dans le sous-arbre ayant comme tête le nœud N.

Pour la phrase « Il ne mange pas. », un arbre de dépendance étendu possible sera :



[il] [] [ne] [] [mange] [] [pas]
0-1 3-4 6 — 10 12—14

La plupart du temps, les analyseurs fournissent assez d'informations relatives aux mots pour que la correspondance chaîne-nœud se réalise sans difficulté : unité lexicale (chaîne), lemme et ses variables grammaticales associées, position dans la phrase.

Les informations SNODE et STREE nous permettrons également de vérifier certaine propriété relative aux arbres de dépendance lors de la dernière phase du traitement, la production (voir 4.3 - Production des arbres de dépendances) :

L'arbre est *non-recouvrant* si :

- $STREE(N1)$ et $STREE(N2)$ ont une intersection vide si N1 et N2 sont indépendants.
- $SNODE(N1)$ et $STREE(N2)$ ont une intersection vide si N2 est un fils de N1.

L'arbre est *projectif* si :

- L'arbre est *non-recouvrant*.
- Pour tous nœuds frères N1 et N2, N1 étant à gauche de N2, $STREE(N1)$ est complètement à gauche de $STREE(N2)$. Cela signifie :

Si $STREE(N1) = w(i1_j1)...w(ip_jp)$ ou \emptyset
et $STREE(N2) = w(k1_l1)...w(kq_lq)$ ou \emptyset
alors $jp \leq k1$

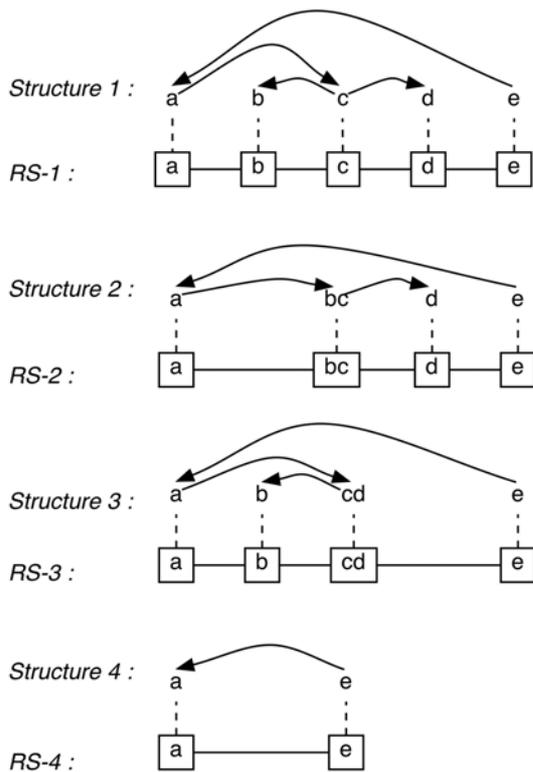
4.1.2 - Réseau de segmentation

La correspondance de structures consiste à regrouper les nœuds représentant la même segmentation dans la phrase. Mais elle consiste également à représenter les discordances issues des différentes segmentations.

Pour ce faire, nous créons une structure, appelée *réseau de segmentation (RS)*, représentant les différentes segmentations de la phrase et permettant de lier les nœuds des structures normalisées. Ce réseau peut être vu comme un « *pivot de liaison* » entre ces structures. Ce réseau est un treillis, chaque nœud du réseau représentant une segmentation possible d'un mot et servant de liaison entre les nœuds des structures de dépendances. Concrètement, un nœud Nrs d'un RS contient deux informations :

- $SNODE(Nrs)$: séquence d'intervalles représentant la segmentation dans la phrase, basée sur la proposition de [Boi88] *Structured String-Tree Correspondences (SSTC)*.
- L : un ensemble contenant les nœuds des structures normalisées liés au nœud Nrs.

La première étape consiste à créer un réseau de segmentation initial pour chaque arbre de dépendances. Chaque nœud Nrs du RS initial est créé en fonction d'un nœud N1 de l'arbre A1 : $SNODE(Nrs) = SNODE(A1.N1)$ et $L(Nrs) = \{A1.N1\}$. Les nœuds du RS seront insérés dans le treillis selon l'ordre d'apparition dans la phrase (en fonction du SNODE). Dans la suite du traitement, nous prendrons comme exemple les quatre arbres de dépendances et leurs RS initiaux suivants :



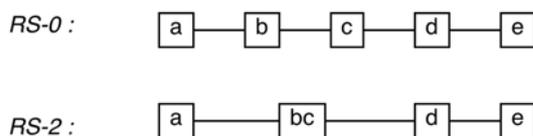
Le premier réseau initial RS-1 est désigné comme le réseau de base, RS-0, qui servira tout au long du traitement. La suite consiste à introduire les particularités des autres RS initiaux dans le réseau de base. Pour ce faire, nous utilisons deux règles de construction :

- **Règle 1) Correspondance** : Si le nœud Ni de RS-i est équivalent à l'un des nœuds N0 de RS-0 (l'équivalence est vraie si $SNODE(Ni) == SNODE(N0)$), N0 sera lié aux nœuds de la structure que représente Ni :

$$L(N0) = L(N0) \cup L(Ni).$$

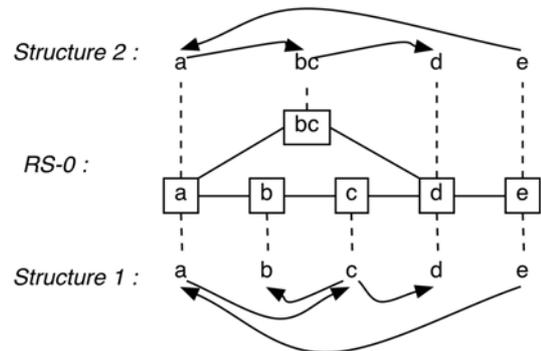
- **Règle 2) Insertion** : Si le nœud Ni de RS-i n'est équivalent à aucun nœud de RS-0, le nœud Ni est inséré dans RS-0 en fonction de son SNODE (ordre d'apparition dans la phrase).

Faisons la correspondance entre RS-0 et RS-2 :

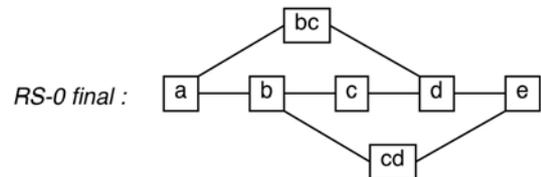


Le premier nœud a de RS-2 vérifie la première règle, il correspond au nœud a de RS-0. Le second nœud bc de RS-2 vérifie la seconde règle, il est donc inséré dans RS-0. Le reste des

nœuds de RS-2 vérifie la première règle. Nous obtenons donc le réseau suivant :



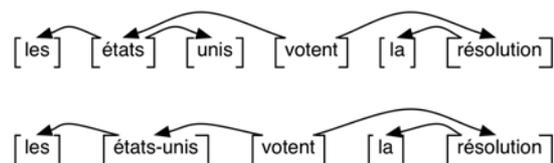
Le traitement se répète en ajoutant RS-3 et RS-4. Le réseau de correspondance final sera donc :



Le réseau de segmentation final obtenu représente les segmentations possibles et lie les nœuds des structures entre eux. Maintenant que la correspondance entre les nœuds des structures est établie, nous pouvons fusionner ces structures pour fournir une unique représentation de dépendances combinant toutes les informations linguistiques relatives aux structures : catégories, variables grammaticales et relations (voir 4.2 – Fusion des informations linguistiques).

4.1.3 - Traitement des mots composés

Une segmentation discordante entre structures peut provenir d'un mot composé. Un mot composé est assimilé à une différence de segmentation si, comme dans l'exemple ci-dessous, un autre analyseur ne reconnaît pas la composition du mot :



Nous voulons que notre plate-forme soit capable de déterminer la correspondance entre le mot composé et le sous-arbre fourni par un autre analyseur. La solution envisagée est de définir une règle de correspondance (RC) qui décrira des contraintes linguistiques et

structurelles permettant de déterminer une telle correspondance.

Une règle de correspondance est constituée :

- D'un modèle de correspondance permettant de déterminer les nœuds des structures à traiter (en rapport avec les nœuds du RS).
- De contraintes linguistiques sur les nœuds et les arcs des structures.
- De contraintes structurelles

Exemple simple de règle de correspondance permettant de vérifier l'exemple précédent :

```
// description du modèle
RC[Nmc ⇔ (N1,N2)] =

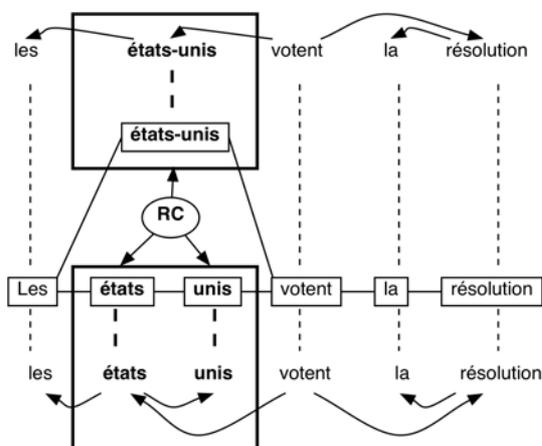
// contraintes linguistiques
Nmc.categorie == "nom" ;
N1.categorie == "nom" ;
N2.categorie == "adjectif" ;

// N2 dépendant de N1
N1 → N2 ;

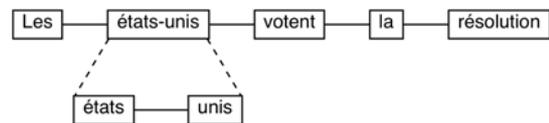
// N2 dépendant d'aucun autre noeud
N* → N2, N*/N1 = ∅ ;

// N2 n'a pas de dépendants
N2 → N*, N* = ∅ ;
```

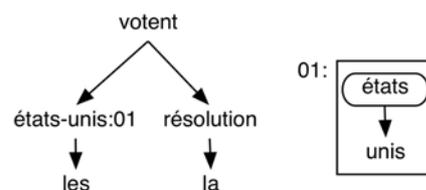
Les règles de correspondance s'appliquent sur les nœuds discordants du réseau de segmentation. Ci-dessous, un exemple d'application de la règle décrite précédemment :



Si une règle est vérifiée, la correspondance est établie et le réseau de segmentation sera modifié pour prendre en compte cette nouvelle information. Nous ajouterons une dimension supplémentaire au réseau établissant ainsi une correspondance particulière entre ces nœuds :



Ce type d'information nous permettra, lors de la phase de fusion, de combiner les informations relatives aux deux sous-structures mis en correspondance, mais également, lors de la phase de production, de fournir une information structurelle sur le mot composé. Par exemple, pour l'exemple vu précédemment, la plate-forme d'analyse fournira comme résultat : (01 étant la référence du sous-arbre de tête le nœud *état*)



4.2 – Fusion des informations linguistiques

Les correspondances entre les différentes structures étant établies, la fusion des informations. La méthode utilisée lors de cette phase est basée sur la méthode dite de « vote à la majorité » : *plus une information sera commune aux différents analyseurs, plus son poids augmentera*, dans notre cas plus son indice de confiance augmentera. Chaque indice pourra être vu comme le *vote pondéré* de l'analyseur pour l'information, ce vote étant *adapté* aux différentes possibilités de l'analyseur en fonction de l'énoncé (par exemple, entrée bruitée ou non) lors de la phase d'apprentissage.

À la fin de la phase de correspondance, à chaque phrase est associée un ensemble de matrices de dépendances et un réseau de segmentation permettant de les lier. La suite du traitement consiste à créer pour chaque réseau de segmentation une matrice de dépendances, nommée *matrice de fusion*, (les nœuds du RS serviront de nœuds pour cette représentation), puis à la compléter en fusionnant toutes les informations linguistiques contenues dans les matrices de dépendances associées. Certaines

informations linguistiques seront équivalentes, d'autres contradictoires. Il ne s'agit pas simplement de regrouper toutes ces informations, il faut également calculer de nouveaux indices de confiance, *les indices de fusion*, pour chaque information en fonction des indices de confiance fournis par l'étape de normalisation.

Calcul simple⁴

L'indice de fusion de l'information I pour un nœud ou un arc donné est égal à la somme des indices de confiance des informations I pour ce nœud ou cette arc :

$$\text{indice(I)}_{\text{fusion}} = \frac{\sum_{i = \text{analyseur fournissant l'information I}} (I_i)}{n}$$

Par exemple, calculons l'indice de fusion à associer à l'information catégorie ADJ pour le mot x, fournie à la fois par l'analyseur A1 et par l'analyseur A2. L'indice de fusion associé à ADJ(x) est égal à la somme des deux indices de confiance $\text{indice(ADJ::A1)}=0,6$ et $\text{indice(ADJ::A2)}=0,8$. L'indice de fusion associé à ADJ(x) est égal à $(0,6+0,8) = 1,4$.

Calcul normalisé

L'indice de fusion de l'information I est égal à la somme des indices de confiance divisé par le nombre d'analyseurs pouvant fournir cette information :

$$\text{indice(I)}_{\text{fusion}} = \frac{\left(\sum_{i = \text{analyseur fournissant l'information I}} (I_i) \right)}{n}$$

Par exemple, calculons l'indice de fusion à associer à l'information relation OBJ entre les mots x et y, fournie à la fois par l'analyseur A1 et par l'analyseur A2. L'indice de fusion associé à OBJ(x,y) est égal à la somme des deux indices de confiance $\text{indice(OBJ::A1)}=0,5$ et $\text{indice(OBJ::A2)}=0,7$ divisée par le nombre d'analyseurs pouvant fournir ce type d'information (ici trois pour l'exemple), $(0,5+0,7+0)/3 = 0,4$. Si le troisième analyseur fournit une information de type SUBJ entre les mots x et y, et que l'indice de confiance relatif à cette information est de 0,8.

L'indice de fusion associé à SUBJ(x,y) est égal à $(0+0+0,8)/3 = 0,26$.

Calcul corrigé⁵ :

L'indice de fusion de l'information I est égal à la somme des indices de confiance des informations I moins la somme (multipliée par un coefficient de correction) des indices de confiance des informations contradictoires à l'information I, le tout divisé par le nombre d'analyseurs pouvant fournir l'information I :

$$\text{indice(I)}_{\text{fusion}} = \frac{\left(\sum_{i = \text{analyseur fournissant l'information I}} (I_i) - \beta \times \sum_{p = \text{analyseur fournissant une information contradictoire à I}} (I_p) \right)}{n}$$

Pour l'exemple précédent, les relations syntaxiques entre les mots x et y sont contradictoires : soit OBJ(x,y), soit SUBJ(x,y). L'indice de fusion associé à OBJ(x,y) est égal à $((0,5+0,7) - (0,4 * 0,8))/3 = 0,29$. (en prenant comme coefficient de correction 0,4) et l'indice de fusion associé à SUBJ(x,y) est égal à $(0,8 - 0,4 * (0,5+0,7))/3 = 0,1$

On peut constater que ces calculs favorisent les informations fournies par le plus grand nombre d'analyseurs. Ces nouveaux indices de fusion ainsi calculés serviront lors de la phase de production.

4.3 - Production des arbres de dépendances

Cette dernière phase permet d'obtenir une ou plusieurs structures de dépendances grâce à toutes les informations recueillies. Les structures de dépendances résultats sont produites à partir des indices de confiance associés aux informations, de contraintes linguistiques et de contraintes structurelles.

Nous pouvons voir cette phase comme une phase de satisfaction de contraintes comportant 4 règles :

- Seuls les indices de confiance élevés seront conservés (au-dessus d'un certain seuil).
- Aucune discordance de segmentation entre les nœuds.
- Respect des contraintes linguistiques imposées par l'utilisateur pour éviter les contradictions.

⁴ n : nombre d'analyseurs pouvant fournir l'information I

⁵ β : coefficient de correction

- Respect des contraintes structurelles, comme la projectivité de la structure résultat et le fait de n'avoir qu'une et une seule relation de même type (syntaxique, sémantique, logique) entre deux nœuds.

Notre plate-forme produira donc plusieurs structures résultats pour un énoncé, à chaque structure résultat sera associé un indice de pertinence, moyenne pondérée des indices de fusion présents dans la structure.

5 - PERSPECTIVES

Une seconde maquette est en cours de réalisation et sera évaluée sur 5 analyseurs : l'analyseur de la plateforme Xelda, l'analyseur Xip développé par Xerox, l'analyseur développé par J. Vergne, l'analyseur du projet Lidia, l'analyseur développé par J. Chauché. Cette nouvelle maquette intégrera tous les modules et fournira en résultat une ou plusieurs structures de dépendances.

À plus long terme, nous comptons utiliser notre plate-forme d'analyse de dépendances, pour générer des hypergraphes UNL (Universal Networking Language). Un hypergraphe décrit le *sens* de l'énoncé dans un contexte donné. Il est composé d'arcs représentant des relations sémantiques (tels qu'agent, objet, but, etc.) et de nœuds représentant les UW (« Universal Word », ou acceptions interlingues) auxquelles sont associés des attributs sémantiques [Sér00]. Pour générer ce type de graphe, la structure de dépendances fournie par notre plate-forme d'analyse nous semble particulièrement adéquate, vu qu'elle représente également un ensemble de relations entre mots. Une telle structure associée à un dictionnaire spécifique nous permettra dans un premier temps de générer des hypergraphes simples, composés d'informations syntaxiques et sémantiques de surface.

REMERCIEMENTS

Je tiens à remercier XEROX et Jacques Vergne pour m'avoir permis d'utiliser leurs analyseurs.

BIBLIOGRAPHIE

- [Boi88] Boitet Ch. And Zaharin Y. (1988), "Representation trees and string-tree correspondences", published in COLING-88, pp 59-64
- [Bru03] Brunet-Manquat F. (2003), "Fusionner pour mieux analyser: quelques idées et une première expérience", published in RECITAL'03
- [Fis97] Fiscus J.G. (1997), "A post-processing system to yield reduced error word rates: Recognizer output voting error reduction (ROVER)", published in IEEE Workshop on Automatic Speech Recognizer and Understanding, pp 347-354
- [Ill99] Illouz G. (1999), "Méta-étiqueteur adaptatif: vers une utilisation pragmatique des ressources linguistiques", published in TALN'99
- [Mon02] Monceaux L., Isabelle Robba I. (2002), "Les analyseurs syntaxiques : atouts pour une analyse des questions dans un système de question-réponse ? ", Actes de TALN'2003, pp.195-204.
- [Sch00] Schwenk H. and Gauvain J.L. (2000), "Combining multiple speech recognizers using voting and language model information", published in IEEE International Conference on Speech and Language Processing (ICSLP), pp. II:915-918
- [Sér00] Sérasset G., Boitet Ch. (2000), "On UNL as the future "html of the linguistic content" & the reuse of existing NLP components in UNL-related applications with the example of a UNL-French deconverter", in COLING 2000, Saarebruecken, Germany.

RESEAUX CELLULAIRES DE VORONOÏ

Baert Anne-Elisabeth,

Docteur en Informatique

Baert@laria.u-picardie.fr , +33 3 22 82 54 03

Adresse professionnelle

Laboratoire de Recherche en Informatique,

Université de Picardie Jules Verne

33 rue Saint Leu

80039 Amiens Cedex 1

Résumé : Classiquement, le modèle utilisé pour les réseaux mobiles cellulaires est le maillage hexagonal. Un modèle plus réaliste, présenté ici, est basé sur le diagramme de Voronoï et la triangulation de Delaunay. Dans le réseau mobile cellulaire de Voronoï, la taille des cellules est déterminée par deux facteurs de base : la position géographique et la zone de couverture des stations de base. Dans les réseaux cellulaires, les algorithmes de diffusion simple réalisent un arbre couvrant du graphe et la complexité de ces algorithmes est le diamètre du réseau. Nous proposons dans cet article des résultats sur l'estimation du diamètre et de l'excentricité de la triangulation de Delaunay.

Summary: The classical model used for mobile cellular networks is the hexagonal mesh. We present here a more realistic model based on Voronoï diagram and Delaunay triangulation. In Voronoï cellular networks, the feature of cell is determined by two basic factors: the geographical position and the geographical reach of the base stations. In cellular networks, one-to-all broadcasting algorithm realizes a minimum spanning tree of the network, the complexity of those algorithms is the diameter of network. In this paper, we present some results on the diameter and the eccentricity of the Delaunay triangulation.

Mots clés : Réseau cellulaire, Graphe planaire, Diagramme de Voronoï, Triangulation de Delaunay.

Keywords: Cellular networks, Planar graphs, Voronoï diagram, Delaunay triangulation

Réseaux cellulaires de Voronoï

Classiquement, le modèle utilisé pour les réseaux mobiles cellulaires est le maillage hexagonal. Un modèle plus réaliste, présenté ici, est basé sur le diagramme de Voronoï et la triangulation de Delaunay. Dans le réseau mobile cellulaire de Voronoï, la taille des cellules est déterminée par deux facteurs de base : la position géographique et la zone de couverture des stations de base. Ce réseau mobile cellulaire peut être vu comme un graphe planaire triangulaire dont les triangles ne sont pas réguliers. Dans les réseaux cellulaires, les algorithmes de diffusion simple réalisent un arbre couvrant du graphe. Pour les réseaux cellulaires de Voronoï, la complexité de ces algorithmes est alors en $O(\text{Diam}(D))$, où $\text{Diam}(D)$ représente le diamètre de la triangulation de Delaunay. Nous proposons dans cet article des résultats sur l'estimation du diamètre et de l'excentricité de la triangulation de Delaunay.

1- INTRODUCTION

Les nouveaux systèmes de télécommunications basés sur les réseaux sans fil cellulaire se sont développés rapidement, Black (1996), Lee (1996), Mac (1979). Ils ont apporté de nouveaux problèmes pour l'évaluation de la performance de tels systèmes. Le principe de la radiocommunication mobile cellulaire consiste à diviser l'espace géographique couvert par le réseau en petites zones appelées *cellules*. Afin de lutter contre la pénurie de fréquences disponibles les cellules partagent entre elles, d'une façon dynamique, les fréquences utiles entre le maximum d'utilisateurs. Ce concept repose sur une propriété essentielle des ondes électriques, celle de s'atténuer en fonction de la distance. Les *stations de base* (BTS : Base Transceiver Station) s'occupent de la transmission radio sur les cellules. Par contraste avec les réseaux stationnaires, les émetteurs d'appels sont mobiles et ne sont pas connectés par un lien physique (câble) au poste émetteur correspondant. Les réseaux de communication cellulaires sont modélisés par des graphes. Les sommets du graphe représentent les stations de base, et les arêtes sont les liens de communications.

Le phénomène principal caractérisant les *ondes radioélectriques* est son affaiblissement progressif : la puissance du signal transmis décroît proportionnellement à la distance de la source. Pour couvrir la plus grande aire possible avec un minimum de stations (sans perdre d'espaces, et minimiser les problèmes d'interférences entre les

cellules), on a recours à un pavage du plan. Si on suppose que toutes les stations émettrices ont la même puissance d'émission, on peut alors utiliser un pavage du plan régulier.

Un modèle basé sur un pavage du plan hexagonal a été introduit par Chen, Shin, et Kandlur (1990). Dans ce modèle de réseau cellulaire, il faut relier les stations émettrices voisines afin d'obtenir un pavage triangulaire du plan. On peut se reporter aux articles de Carle et Myoupo (2001), Katzela et Naghshineh (1996) ou Rappaport (1999).

Les stations de base dans un réseau mobile cellulaire de grandes tailles sont réparties de façon irrégulière. La taille des cellules doit dépendre de la densité de population et de la demande dans une région donnée. Nous présentons ici, un modèle de réseau cellulaire qui tient compte de ces deux facteurs. Ce modèle cellulaire est basé, dans un premier temps, sur le diagramme de Voronoï et la triangulation de Delaunay et tient ainsi compte de l'irrégularité de la position des stations Baccelli, Klein, Lebourges, et Zuyez (1997), Tchoumatenko (1999). Dans un second temps, nous pondérons chaque site pour tenir compte de la couverture maximale de chaque station de bases.

Les algorithmes de diffusion dans les réseaux cellulaires réalisent un arbre couvrant du graphe représentant le réseau. Dans le cas où l'algorithme est optimal, cet arbre couvrant est minimal en nombre de liens. Le diamètre du graphe correspond au nombre d'étapes nécessaires, dans le pire des cas, pour qu'un sommet diffuse un message à tous les autres sommets du réseau. La complexité de l'algorithme est alors en $O(\text{Diam}(G))$. Dans les réseaux cellulaires de Voronoï la diffusion de messages s'effectue dans la triangulation de Delaunay. En utilisant des résultats combinatoires sur la dualité du diagramme de Voronoï, nous exprimons le diamètre moyen de la triangulation de Delaunay en fonction de l'excentricité moyenne ainsi que du diamètre moyen du diagramme de Voronoï.

Le plan de l'article est le suivant. Dans la première partie, nous présentons le réseau cellulaire de Voronoï. Pour cela, nous introduisons tout d'abord le diagramme de Voronoï de puissance, puis nous présentons son dual la triangulation de Delaunay. Dans la seconde partie, nous nous intéressons au problème du diamètre de la triangulation de Delaunay. La conclusion termine l'article.

2- DESCRIPTION DES RESEAUX CELLULAIRES DE VORONOÏ

2.1- Le diagramme de Voronoï

La modélisation des réseaux mobiles cellulaires est rarement possible par le modèle classique basé sur un maillage hexagonal. Dans ce modèle, le plan est divisé en maille hexagonale de même taille et les stations de base se trouvent au centre des cellules hexagonales. Dans le réseau hexagonal chaque station possède la même puissance. La zone de couverture est représentée par le même motif : des cellules en « nid d'abeilles ».

Dans les réseaux cellulaires de grandes tailles, l'emplacement des stations de base est très irrégulier. A chaque station de base est associée une zone de couverture d'ondes. Dans les réseaux réels, les zones de couverture doivent tenir compte de nombreux facteurs comme par exemple la densité de la population, la demande des consommateurs et la topographie de cette zone. Le modèle classique basé sur les cellules hexagonales n'est pas, en ce sens, un modèle réaliste pour les réseaux cellulaires. La taille des cellules doit être proportionnelle à deux facteurs de bases : la position géographique des stations de base dans le réseau cellulaire et la couverture géographique de ces stations.

Les stations et les souscripteurs de service sont représentés par leurs coordonnées dans le plan \mathbb{R}^2 . Nous supposons que les n stations de base sont positionnées arbitrairement dans le plan. A chaque station v_i du réseau cellulaire correspond une puissance maximale notée r_i .

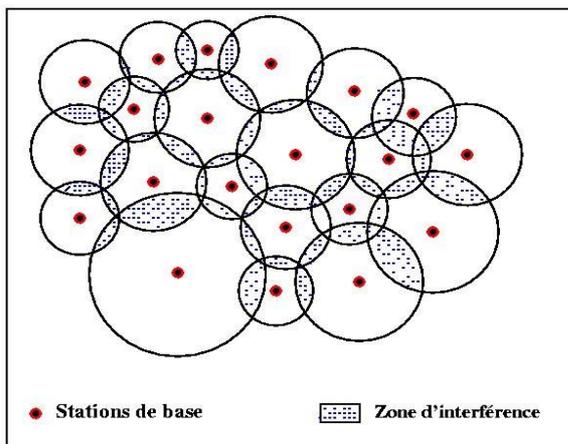


Figure 1 : Zone pavée par des cellules circulaires irrégulières

Le plan est, dans un premier temps, divisé en cellule circulaire P_i (voir Figure 1). Chacune de ces n cellules circulaires a pour centre la station émettrice v_i et pour rayon la puissance maximale r_i lui correspondant. La répartition des cellules est effectuée de telle sorte qu'il n'existe pas de zone non couverte par une station. La zone de service sera appelée *cellule de Voronoï*. Le diagramme de Voronoï généralisé peut être informellement défini comme étant un partitionnement du plan en régions, où chacune de ces régions est le lieu de points de \mathbb{R}^2 le plus proche d'un objet donné $P \in S$ que de tout autre objet $Q \in S$, pour $P \in Q$. Considérons un ensemble d'objets de S dans \mathbb{R}^2 tels que $|S| = n$.

Définition 1 : Nous définissons la médiatrice généralisée comme le lieu géométrique de points équidistants à deux objets distincts P_i et P_j d'un ensemble S d'objets :

$$B(P_i, P_j) = \{x \mid d(x, P_i) = d(x, P_j)\}$$

Cette médiatrice divise l'espace en deux demi-plans ; nous notons $H(P_i, P_j) = \{x \mid d(x, P_i) < d(x, P_j)\}$ le demi-plan contenant P_i .

Le diagramme de Voronoï généralisé pour un ensemble d'objets S est alors la collection de régions de Voronoï généralisé $V(P_i)$, $P_i \in S$ où :

$$V(P_i) = \bigcap_{P_j \in S \setminus \{P_i\}} H(P_i, P_j)$$

Nous utilisons la notation suivante : $V(S)$ est le diagramme de Voronoï formé par la réunion des régions de Voronoï de S . Une région de Voronoï d'un diagramme de Voronoï de n sites est donc obtenue par l'intersection de $n-1$ demi-plans. Cette région est représentée par un polygone convexe qui peut être non-fini.

Le diagramme de Voronoï est un *graphe abstrait*, avec pour sommets ceux du polygones de Voronoï, et comme arêtes celles des lignes polygonales (suites finies de segments consécutifs). On peut se reporter aux livres de : Preparata et Shamos (1985), Okabe, Boots et Sugihara (1992) ou Aurenhammer et Klein (2000) pour plus de détails sur les diagrammes de Voronoï.

L'une des généralisations des diagrammes de Voronoï consiste à modifier les règles de distance selon lesquelles le diagramme de Voronoï est calculé. Nous pouvons dès lors introduire le concept de *pondération* d'un site P_i qui revient à associer à chaque site un nombre réel, $w(P_i)$, appelé *poids*. La

distance depuis P_i à un point $x \in \mathcal{R}^2$ est alors mesurée comme une fonction du poids $w(P_i)$ et de la distance $d(x, p_i)$, où p_i est un *point de référence* fixe appartenant au site P_i . Lorsque P_i est un cercle, il est habituel de prendre le centre du cercle pour point de référence.

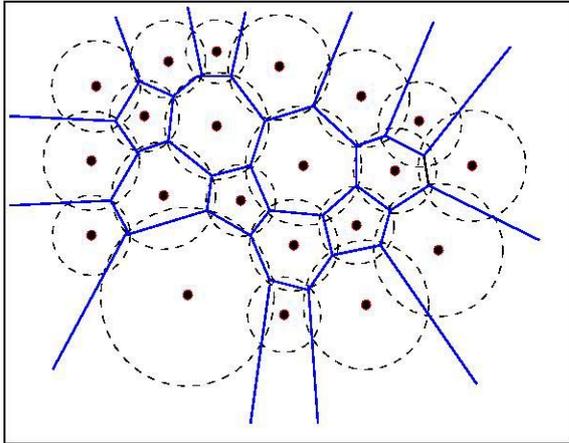


Figure 2 : Réseau cellulaire de Voronoï de puissance

La distance de puissance entre un cercle de centre p et de rayon r_p et un point x est alors défini comme :

$$d_p(x, P)^2 = \text{pow}(x, P)^2 = d(x, p)^2 - r_p^2,$$

où $d(x, p)$ est la distance Euclidienne entre x et p . L'interprétation géométrique de la distance de puissance est que la racine carrée de la distance entre x et un cercle P peut être calculée le long d'une droite tangente au cercle P . Un point x se trouve à l'intérieur du cercle P où appartient à sa fermeture si et seulement si la distance de puissance $d_p(x, P)$ est négative ou nulle.

A chaque station de base v_i , nous associons une variable r_i correspondant à la puissance maximale de cette station. Cette caractéristique peut être introduite dans le diagramme de Voronoï en considérant le *diagramme de Voronoï de puissance* ou appelé *diagramme de Laguerre*. Il peut être interprété comme le diagramme de Voronoï des cercles dans \mathcal{R}^2 , Aurenhammer (1987). Les n cellules circulaires P_i sont servis respectivement par les n stations v_i (qui se situent au centre des cellules circulaires). Le poids associé à chaque cellule P_i est alors le rayon de la cellule considérée. Les cellules P_i sont alors des *cellules de Voronoï de puissance*, elles forment une partition du plan en un ensemble de n régions polygonales que nous appelons *réseau de Voronoï de puissance* (Figure 2).

Ce réseau contient, au sens de la couverture des stations, toutes les informations de proximité d'un ensemble de stations données. Dans ce nouveau modèle de réseau cellulaire, nous devons alors relier entre elles chaque station émettrice pour obtenir un pavage triangulaire.

2.2- La triangulation de Delaunay

Une autre structure fondamentale en géométrie algorithmique est la *triangulation de Delaunay*. En 1934, Delaunay a prouvé le théorème suivant :

Théorème 1 :

Le dual d'un diagramme de Voronoï d'un ensemble S de points est une triangulation de S .

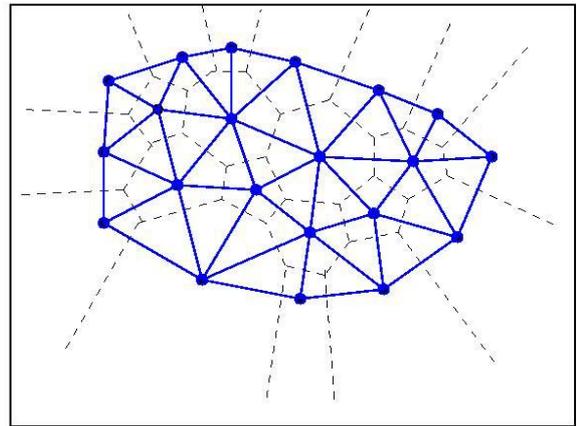


Figure 3 : Triangulation de Delaunay des stations

Le *graphe de Delaunay*, noté D , est par construction un graphe planaire triangulaire (Figure 3). Il est obtenu en joignant tous les couples de points de S dont les polygones de Voronoï partagent une arête commune. C'est un cas particulier du dual d'un graphe planaire. Si G est un graphe planaire, nous obtenons le graphe *dual* de G en considérant chaque couple de faces partageant une même arête : nous choisissons un point à l'intérieur de chacune des deux faces et nous joignons ces deux points par un *arc de Jordan*. Le graphe triangulaire de Delaunay s'appelle le dual par ligne droite du diagramme de Voronoï : on joint par un segment deux points de S dont une arête de Voronoï est la médiatrice du segment. Trois arêtes de Voronoï concourent en x de S en tant que médiatrices des trois cotés d'un triangle de Delaunay dont x est le centre du cercle

circonscrit, appelé *cercle de Delaunay* de x .

3- RESULTATS SUR LE RESEAU CELLULAIRE DE VORONOÏ

Dans cette section nous proposons des résultats sur l'estimation du diamètre et de l'excentricité de la triangulation de Delaunay. Le diamètre et le degré moyen d'un graphe sont des paramètres très liés dans la théorie des graphes. Le diagramme de Voronoï *fini* est un graphe 3-régulier. En utilisant cette propriété, nous allons déterminer le diamètre du diagramme de Voronoï en utilisant les propriétés de dualité entre la triangulation de Delaunay et les diagrammes de Voronoï.

3.1- Arbre couvrant de Voronoï

Nous considérons une courbe fermée simple, notée Γ autour de la partie « intéressante » du diagramme de Voronoï, de telle sorte que cette courbe n'intersecte que les arêtes infinies du diagramme de Voronoï. Si nous enlevons les demi-droites en dehors de Γ , nous obtenons un graphe connexe planaire avec $n_D + 1$ faces, où n_D est le nombre d'arêtes de la triangulation de Delaunay. Ces faces sont les n_D régions de Voronoï auxquelles nous ajoutons la face infinie (celle en dehors de Γ). Ce graphe est appelé le *diagramme de Voronoï fini*, Fortune (1992).

Proposition 1 :

Soient e_V et n_V le nombre d'arêtes et de sommets dans le diagramme de Voronoï. Soient $Cont(V)$ le nombre d'arêtes dans Γ , et $n_{V_{fini}}$ le nombre de sommets dans le diagramme de Voronoï fini. Nous avons les relations suivantes :

$$e_V = \frac{3}{2}n_V,$$

$$n_V = Cont(V) + n_{V_{fini}},$$

$$n_D = \frac{n_{V_{fini}}}{2} + 1.$$

Preuve :

Comme le diagramme fini de Voronoï est un graphe planaire, nous pouvons utiliser la formule d'Euler :

$$n_{V_{fini}} + F_{V_{fini}} = e_{V_{fini}} + 1,$$

où $e_{V_{fini}}$ et $F_{V_{fini}}$ sont le nombre d'arêtes et de faces du diagramme de Voronoï fini.

Par dualité le nombre d'arêtes dans une triangulation de Delaunay est le même que dans un diagramme de Voronoï.

□

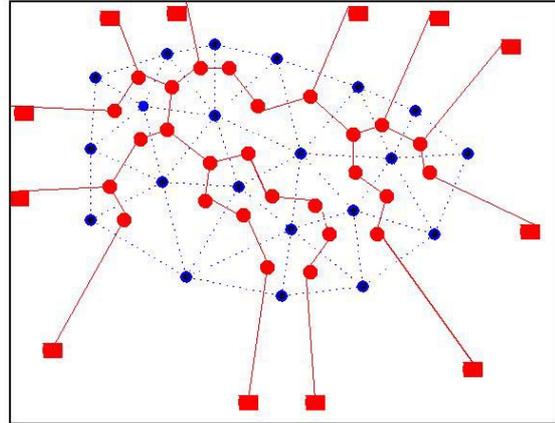


Figure 4 : Arbre couvrant de Voronoï

Les diagrammes de Voronoï finis sont des graphes 3-réguliers; tous les sommets dans le diagramme de Voronoï sont de degré trois. Si nous construisons un arbre couvrant dans ce graphe, les sommets de cet arbre ne peuvent pas avoir plus de trois successeurs. Cet arbre couvrant est appelé un *arbre unaire-binaire* (Figure 4). Chaque sommet de cet arbre couvrant se situe à l'intérieur d'un triangle de la triangulation Delaunay. Pour chaque triangle deux de ses cotés sont « coupés » par deux arêtes de l'arbre couvrant.

3.2- - k-chemin de triangles

Si nous considérons un chemin (ou une branche) dans cet arbre couvrant de Voronoï (Figure 5), ce chemin est connu sous le nom de *chemin triangulaire* ou *t-chemin*, voir Aichholzer (1999).

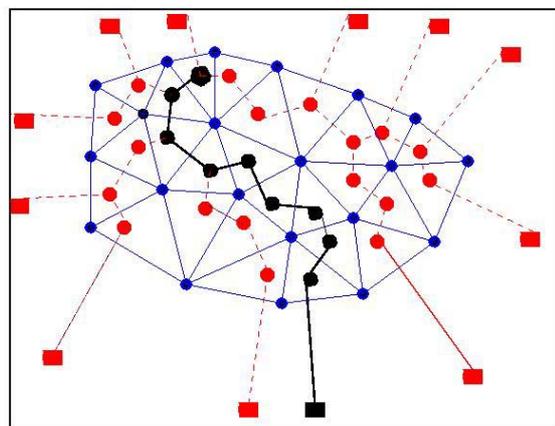


Figure 5 : t-chemin dans un arbre couvrant de Voronoï

Définition 1 :

Soit un t -chemin de longueur $k-1$ Nous appelons k -chemin de triangles, l'ensemble des triangles coupés par ce chemin, et nous notons Φ_k l'ensemble des triangles formés par cette construction.

Remarque :

Il existe une dualité entre un k -chemin de triangles (Figure 6) et un t -chemin de longueur $k-1$.

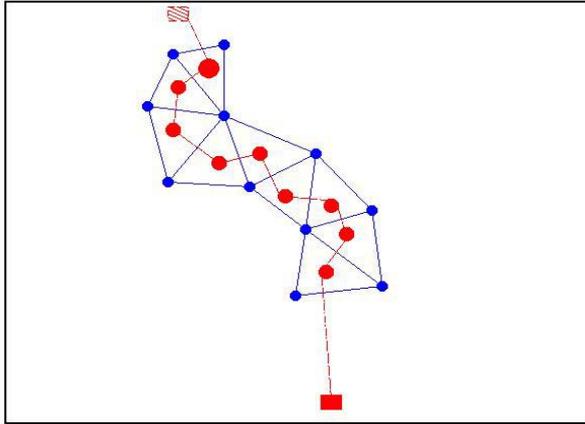


Figure 6 : k -chemin de triangles

Proposition 2 :

Soient F_{Φ_k} , e_{Φ_k} , v_{Φ_k} le nombre de faces, d'arêtes et de sommets dans un k -chemin de triangles. Nous avons les relations suivantes :

$$\begin{aligned} F_{\Phi_k} &= k, \\ e_{\Phi_k} &= 2k + 1, \\ v_{\Phi_k} &= k + 2. \end{aligned}$$

Preuve :

Un k -chemin de triangles, Φ_k , contient, par définition, k triangles et donc k faces (si on ne compte pas la face infinie). Chacune de ces faces est un triangle avec 3 arêtes, et dans chaque triangle une de ses arêtes est comptée deux fois (sauf le premier triangle du k -chemin de triangles). Nous avons donc

$$3k - (k-1) = 2k + 1$$

arêtes.

Par induction sur k , on montre que le nombre de sommets est :

$$2k + 1 - (k-1) = k + 2.$$

□

3.3.–Excentricité et diamètre de la triangulation de Delaunay

L'excentricité d'un sommet dans un graphe de Voronoï peut être calculée comme étant la hauteur de l'arbre couvrant de Voronoï enraciné en ce sommet. La dualité établit entre un k -chemin de triangles et un t -chemin de longueur $k-1$ va nous permettre de calculer la longueur d'une branche d'un arbre couvrant de Voronoï.

Lemme 1 :

Soient i et j deux sommets dans une triangulation de Delaunay., Il existe un t -chemin de longueur $k-1$, noté α_k , tel que i et j appartiennent au k -chemin de triangles associé à α_k et $d(i, j) = \left\lfloor \frac{k}{2} \right\rfloor + 1$.

$$d(i, j) = \left\lfloor \frac{k}{2} \right\rfloor + 1.$$

Preuve :

Comme un k -chemin de triangles contient $k+2$ sommets, le nombre d'arêtes externes est aussi $k+2$. Il est connu que le diamètre d'un anneau de k sommets est

$$\left\lfloor \frac{k}{2} \right\rfloor.$$

□

Nous notons $\overline{ecc}(G)$ et $\overline{diam}(G)$ l'excentricité moyenne et le diamètre respectif d'un graphe G .

Théorème 1 :

$$\overline{diam}(D) \approx \frac{\overline{diam}(V)}{2},$$

$$\frac{\overline{ecc}(V)}{2} \leq \overline{diam}(D) \leq \overline{ecc}(V).$$

Preuve :

Le diamètre d'un graphe est par définition le plus long des plus courts chemins d'un graphe. La relation établie dans le théorème 1 nous permet d'estimer le diamètre de Delaunay en fonction du diamètre de Voronoï

Les relations entre l'excentricité et le diamètre moyen, nous permettent d'écrire, Harary (1994) :

$$\overline{ecc}(V) \leq \overline{diam}(V) \leq 2\overline{ecc}(V).$$

□

4- CONCLUSION

Dans cet article, nous nous sommes intéressés à un nouveau modèle pour les réseaux mobiles cellulaires : le modèle de Voronoï de puissance. Les algorithmes de diffusion simple dans un réseau cellulaire réalisent un arbre couvrant de taille minimal dans le réseau. L'optimalité de la diffusion est effective lorsque l'algorithme a une complexité en diamètre du réseau.

Nous avons présenté les premiers résultats théoriques sur l'estimation de l'excentricité et du diamètre de la triangulation de Delaunay. Des résultats expérimentaux ont déjà été obtenus pour le diamètre et l'excentricité des réseaux de Voronoï dans Baert, Semé (2003).

BIBLIOGRAPHIE

- Aichholzer, O., (1999) « The path of triangulations », Dans 15th Ann. ACM Symp. Computational Geometry, p. 14-23.
- Aurenhammer F. et Klein R., (2000), « Handbook of Computational Geometry: Voronoi diagrams, chapitre 5, p. 201--290. Elsevier Science Publishing. J. Sack and G. Urrutia,
- Aurenhammer, F., (1987), "Power diagrams: properties, algorithms, and applications." SIAM Journal on Computing, 16: p. 78--96.
- Baccelli, F., Klein, M., Lebourges, M. et Zuyev, S., (1997); "Stochastic geometry and architecture of communication networks", Journal of Tel. Systems, 7, p. 209--227.
- Black, U. (1996), "Mobile and Wireless Networks", Prentice-Hall PTR, Upper Saddle River, NJ.
- Baert A.E., et D. Semé, (2003) "One-to-all broadcasting in Voronoi cellular networks", Dans "The 2003 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'03), Las Vegas (USA), p. 844-850.
- Baccelli F., et Zuyev, S. (1997) "Stochastic Geometry Models of Mobile Communication Networks", Dans Frontiers in queueing, p. 227--243.
- Carle, J., Myoupo, J.F., et Semé, D., (2001), "A Basis for 3-D Cellular Networks" Dans Proc. of the 15th International Conference on Information Networking.
- Chen, M.S., Shin, K.G. et Kandlur, D.D, (1990) "Addressing, Routing and Broadcasting in Hexagonal Mesh Multiprocessors", IEEE Trans. on Computers, 39(1): p. 10--18.
- Delaunay, D., (1934), « Sur la sphère vide. A la mémoire de Georges Voronoï », Izvestia Akademia Nauk SSSR Otdelenie Matematicheskii i Estestvennyka Nauk, 7:p. 793—800.
- Fortune, S., (1992), « Voronoi diagrams and Delaunay triangulations. » Dans Ding-Zhu Du et Frank Hwang, éditeurs, Computing in Euclidean Geometry, volume1. World Scientific, Lecture Notes Series on Computing.
- Harary, F., (1994), « Graph Theory », Addison Wesley, Reading, MA.
- Katzela I. et Naghshineh., M., (1996), "Channel Assignment Schemes for Cellular Mobile Telecommunications", Systems: A Comprehensive Survey, IEEE Personal Communications Magazine, 3 :p. 10--31.
- Lee, W., (1996), "Mobile Cellular Telecommunications Systems", McGraw-Hill, New York, 2^{ème} Edition.
- MacDonald., V., (1979) "Advance Mobile Phone Service: The cellular concept", Bell System Technical Journal, 58(1): p.15--41 .
- Okabe, A., Boots, B., et Sugihara, K., (1992), "Spatial tessellations: Concepts and Applications of Voronoi Diagrams", John Wiley & Sons, Chichester, UK, 1992.

Preparata, F.P., Shamos, M.I., (1985),
“Computational Geometry: an
introduction”, Springer Verlag, New-
York.

Rappaport, T., (1999), “Wireless Communications:
Principles and Practice”, Prentice-Hall
PTR, Upper Saddle River, NJ,
Reprinted.

Tchoumatchenko, K., (1999), “Modeling of
Communication Networks Using
Stochastic Geometry”, Thèse de
doctorat, Univeristé de Nice-Sophia
Antipolis, France.

SUIVI DE L'ACTIVITE DE PILOTAGE PAR PREDICTION ET RECALAGE

Charles Lesire,

Doctorant du laboratoire de l'ONERA-DCSD
Charles.Lesire@cert.fr , + 33 5 62 25 22 12

Adresse professionnelle

ONERA-CERT, DCSD ★ 2 av. Edouard Belin ★ 31055 Toulouse Cedex

Résumé : La revue d'accidents aéronautiques tant civils que militaires a montré que l'apparition de conflits dans la gestion du vol est un facteur de dégradation de l'activité et est à l'origine de nombreux accidents. Ces conflits rendent souvent le pilote non réceptif aux alarmes classiques. Un nouveau système d'assistance au pilotage, GHOST, est en cours d'étude et permet de résoudre certains conflits par l'envoi de contre-mesures pertinentes à l'équipage. Ce système nécessite, pour être efficace, un bon suivi de l'activité de pilotage. Après une présentation du système GHOST et des différentes méthodes de suivi d'activités, nous présentons le modèle de représentation des procédures, fondé sur le formalisme des réseaux de Petri à objets, puis les principes de *prédiction* et *recalage* qui assurent le suivi de l'activité de pilotage.

Summary : The review of civilian and military accident reports has shown that a conflictual situation is a precursor to the loss of aircrew's situation awareness and is a major cause of air accidents. Conflicts often prevent pilots from being receptive to classic warnings. A new system for pilot assistance, GHOST, is under study to solve some of the conflicts by sending the crew relevant countermeasures. This system is grounded on the monitoring of the pilot's activity. GHOST and different methods for activity tracking are first presented then the Petri net based model of aeronautical procedures is described and finally the prediction and correction principles designed to track the pilot's activity are explained.

Mots clés : Réseaux de Petri, suivi d'activité, aide au pilotage.

Suivi de l'activité de pilotage par prédiction et recalage

1 - INTRODUCTION

L'étude des rapports d'accidents aéronautiques (BEA, 1996) confirmée par des expérimentations en simulateur (Dehais, 2002) ont permis de mettre en avant des cas de *conflits* menant le pilote à persévérer dans de mauvais choix. Le phénomène de persévération est dû à un stress important (Cohen, 1993) et il est caractérisé par une perte de facultés cognitives du sujet : celui-ci se focalise sur son conflit et ne peut plus en retirer son attention. Dans de nombreux cas les alarmes classiques (visuelles ou auditives) sont inefficaces du fait de la focalisation du pilote sur un instrument ou une situation.

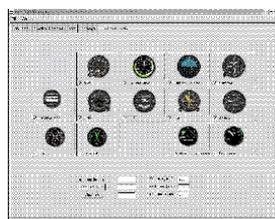
L'idée du système GHOST (Dehais, Tessier, Chaudron, 2003) est de mettre en place des contre-mesures afin d'enrayer le mécanisme de persévération et sortir le pilote de son conflit. Le principe des contre-mesures est fondé sur le retrait d'information : puisque le pilote ne peut détourner son attention volontairement du conflit, GHOST permet de faire disparaître l'objet sur lequel le pilote se focalise, de sorte qu'il puisse prendre conscience de la situation extérieure (alarmes, paramètres, contrôle aérien, etc.)

GHOST est constitué :

- d'un simulateur de vol¹ installé sur un portable muni d'un joystick ; le simulateur a été modifié de façon à permettre la disparition ou le clignotement d'instruments et l'affichage de messages ;
- d'un logiciel de suivi de l'avion sur une carte aéronautique² ;
- d'une interface de magicien d'Oz, permettant de modifier à distance les paramètres de vol et la météo, et d'envoyer des contre-mesures.



Simulateur FlightGear



Interface de magicien d'Oz

FIG. 1: Environnement GHOST

Après une série d'expérimentations qui ont montré la pertinence de l'envoi de contre-mesures (Dehais,

¹ FlightGear, open source <http://www.flightgear.org>

² Atlas, freeware <http://atlas.sourceforge.net>

Tessier, Chaudron, 2003), il s'agit maintenant de modifier GHOST en vue de l'envoi automatique de contre-mesures en fonction de l'activité de pilotage courante et prédite.

Le travail présenté ici concerne donc les principes et la mise en œuvre du suivi de l'activité de pilotage.

2 - SUIVI DE SITUATIONS

Cette section décrit différentes méthodes de reconnaissance d'actions ou de situations, principes sur lesquels est fondé le suivi de l'activité du pilote. Les recherches de Brémond (1997) et Rota et Thonnat (2000) portent sur la reconnaissance d'actions dans des séquences d'images et sont fondées sur la mise en correspondance d'événements observés avec des scénarios composés de listes d'événements et de contraintes portant sur les événements (logiques) ou leurs dates d'occurrence (numériques). L'approche proposée par Dousson (1991) et Ghallab (2002) utilise les *chroniques* dans la reconnaissance d'événements : elle est fondée sur un graphe temporel représentant les contraintes entre les différents événements (relations de précédence temporelle) et la phase de prédiction consiste en une propagation de contraintes sur ce graphe. Largouët et Cordier (2000) confrontent une suite d'observations (images satellites) avec des états références et mettent en œuvre un principe de prédiction et de postdiction pour réduire le nombre d'états retenus. Le projet CATS³ (Callantine, 2002) est destiné au suivi de l'activité de pilotage et à l'analyse en ligne des performances de l'équipage. Un modèle d'activité est représenté par un arbre dont la racine est l'activité et ses fils les actions à réaliser. Lorsqu'une action est reconnue, le système repère les activités susceptibles d'être entreprises par le pilote et attend de celui-ci l'action suivante de façon à affiner la reconnaissance. La mesure des temps d'attente entre deux activités et l'écart par rapport à des temps de référence permet de donner une allure de la performance du pilote.

Nos travaux sont directement issus de l'estimateur symbolique Kalmansymbo - appelé ainsi par analogie avec le filtre de Kalman, (Cossart-Jaupitre, 1999 ; Tessier, 2003) - dont le but est d'élaborer et de mettre à jour une description de l'état d'un environnement (appelée situation), en fonction d'observations qui sont effectuées sur cet environnement grâce à des capteurs dits

³ Crew Activity Tracking System

«symboliques» (c'est-à-dire qui délivrent des informations de même nature que celles que délivrerait un observateur humain). Une situation est un ensemble d'états possibles de l'environnement, ordonnés selon une relation de préférence : les états préférés sont ceux qui correspondent le plus vraisemblablement à l'état (réel) de l'environnement, les états moins préférés sont ceux qui correspondent moins à cet état. L'estimation est effectuée à partir de modèles d'évolution de l'environnement (plans) représentés sous forme de réseaux de Petri. L'estimation est un processus en deux étapes (voir figure 2) : (1) la prédiction permet, à partir d'une situation (donc une relation de préférence sur des états) et des modèles d'évolution (réseaux de Petri), de calculer la situation prédite, c'est-à-dire une relation de préférence sur les états qu'on est susceptible d'observer par la suite ; (2) le recalage permet, à partir d'une situation prédite et d'une observation, de calculer une nouvelle situation, plus conforme à l'observation que ne l'était la situation prédite.

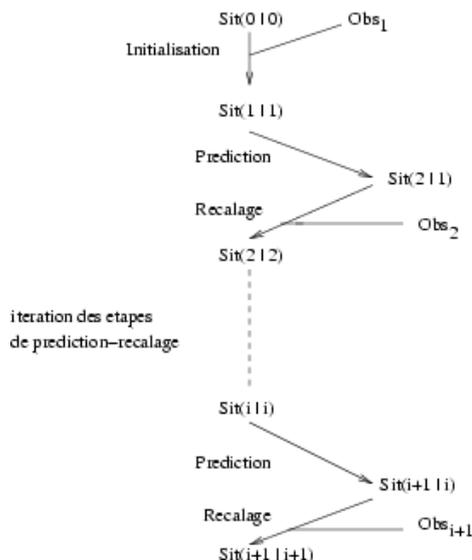


FIG. 2: Processus d'estimation de Kalmansymbo

3 - KALMANSYMBO-AÉRO

L'idée est d'appliquer les principes de Kalmansymbo à GHOST pour suivre et prédire l'activité de pilotage (Kalmansymbo-aéro), et aider ainsi à la prévention des conflits. Les particularités des procédures aéronautiques requièrent des adaptations de ces principes, qui prendront notamment en compte la présence de variables continues et de boucles dans les réseaux de Petri.

3.1 - Exemple la mission Refuel

La mission servant à expérimenter les hypothèses faites lors de l'élaboration de Kalmansymbo-aéro sera la mission *Refuel* (figure 3).

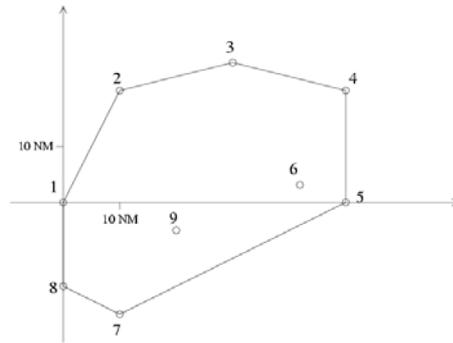


FIG. 3: Mission Refuel

Le pilote doit suivre le plan de vol suivant : décoller (la piste se trouve entre 1 et 8 et orientée vers 1), passer, dans l'ordre, par les points tournants 1, 2, 3, 4, 5, 7 et 8, et atterrir en 1. Le pilote a la possibilité de se ravitailler en passant à la verticale des points 6 et 9. Il peut se diriger vers 6 s'il est entre 4 et 5 ou entre 5 et 7. Il peut se diriger vers 9 s'il est entre 5 et 7.

3.2 - Modélisation

Le formalisme que nous avons choisi est celui des réseaux de Petri à objets, dont la définition est issue de Valette (2000). Les paramètres de vol (latitude, longitude, vitesse, fuel, consommation) sont encapsulés dans le *jeton-mission*, unique instance de la classe *Jeton*.

Places et transitions

Comme introduit dans Tessier (2003), une activité est représentée par un *cube* (Chaudron, Maille, Boyer, 2003), conjonction de prédicats, associé à une place du réseau. Dans le cas de la mission *Refuel*, toutes les activités seront représentées par le prédicat d'arité 2 : *waypoint*. Ce prédicat indique la direction de l'avion (vers quel point tournant il vole) et si l'avion est considéré comme étant à la verticale de ce point. Ainsi *waypoint(2,unreached)* indique que l'avion se dirige vers le point tournant 2 et qu'il ne l'a pas atteint, alors que *waypoint(1,reached)* indique que l'avion a atteint le point tournant 1 (voir figure 4).

Les transitions des réseaux à objets possèdent une condition et une action de franchissement. On note graphiquement ces attributs en étiquetant les transitions sous la forme « *c / a* » où *c* est un cube de conditions et *a* une conjonction d'actions. La condition de passage entre activités représente la capacité de l'avion à se rendre au point suivant. Ce paramètre est dans notre cas modélisé par la quantité de carburant disponible et est représenté par le prédicat *fuel* : *fuel(2,OK)* indique que la quantité de carburant est suffisante pour atteindre le point tournant 2. Dans le cas inverse, la condition de passage est *fuel(2,BAD)*. L'action est représentée par le prédicat *goToWp*. Celui-ci agit sur le *jeton-mission* et modifie ses attributs pour

simuler un déplacement (dans le cas de $\text{goToWp}(2)$), le déplacement de 1 à 2). La figure 4 illustre ce modèle :

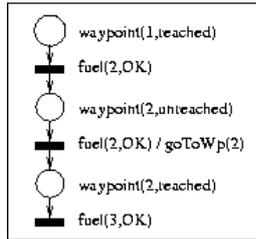


FIG. 4: *Modèle partiel de la mission Refuel*

Les transitions de changement de but

Certaines transitions sont particulières car elles représentent un changement de but (et donc un conflit cognitif potentiel). Ces transitions correspondent à la sélection d'un nouveau but : un point tournant à atteindre différent de celui qui était initialement visé. Pour les différencier des transitions classiques représentées par un rectangle plein, on représente les transitions de changement de but par un rectangle vide. Dans la figure 5, la transition de changement de but est présente pour modéliser l'abandon par le pilote du but 5 ($\text{waypoint}(5, \text{unreached})$) pour se diriger vers le point tournant 6 ($\text{waypoint}(6, \text{unreached})$). La particularité de ces transitions est expliquée dans le cadre de la phase de prédiction (voir section 4).

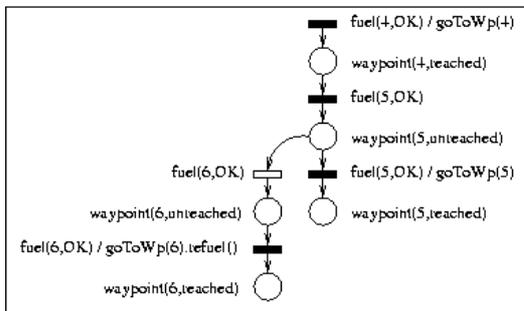


FIG. 5: *Transition de changement de but*

Les boucles

Lors d'un vol, un pilote peut être amené à accomplir plusieurs fois la même activité. C'est pourquoi il est indispensable de gérer les boucles dans les réseaux de Petri. Les boucles posent cependant des problèmes dans la phase de prédiction au cours de laquelle elles peuvent théoriquement être parcourues une infinité de fois. On décide donc, dans le cadre de cette étude, de limiter le nombre de parcours des boucles à 1. On peut donc «casser» les boucles en dupliquant certaines places.

La figure 6 est un exemple de sous-réseau comportant une boucle, qui correspond au détournement de l'avion pour se ravitailler en 6 : le pilote se dirige vers 5 ($\text{waypoint}(5, \text{unreached})$), puis change de but pour aller en 6 ($\text{waypoint}(6, \text{unreached})$), se ravitaill

($\text{waypoint}(6, \text{reached})$), et finalement reprend sa route vers 5 ($\text{waypoint}(5, \text{unreached})$). On remplace ce modèle par celui de la figure 7, dans lequel l'activité $\text{waypoint}(5, \text{unreached})$ a été dupliquée pour distinguer les phases avant et après ravitaillement.

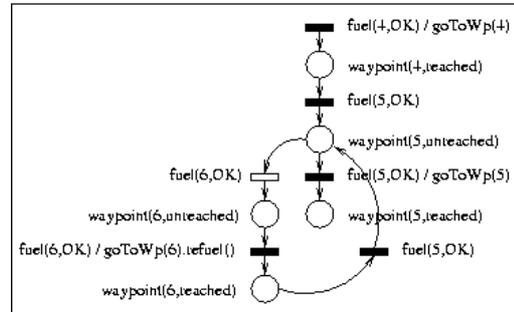


FIG. 6: *Boucle*

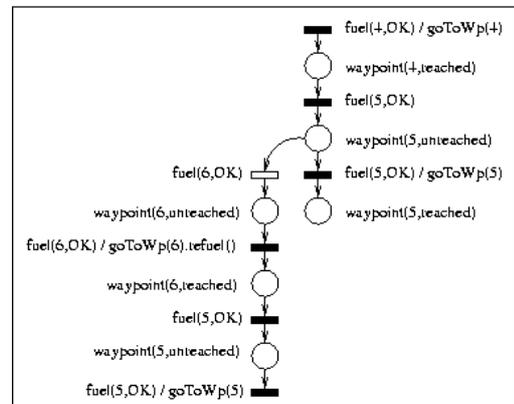


FIG. 7: *Boucle dépliée*

Le temps

Afin de traiter des problèmes réels, qui font intervenir des paramètres continus, on introduit ici une notion temporelle. On assigne au *jeton-mission* une horloge, qui est modifiée par les actions des transitions pour simuler un déplacement de l'avion. L'utilisation de cette horloge sera exposée dans la section traitant de la phase de prédiction (section 4).

4 - PRÉDICTION

La prédiction permet, à partir d'une situation reconnue (modélisée sous forme de réseau de Petri marqué), de calculer la situation prédite, c'est-à-dire une relation de préférence sur les états qu'on est susceptible d'observer par la suite. Ce calcul se fait en construisant l'arbre des marquages accessibles du réseau.

Dans le cas de la mission *Refuel*, on peut constater que seul le *jeton-mission* est présent dans le réseau. Le marquage se limite donc à un seul jeton. On notera par \bullet la place marquée et par \times les places prédites.

4.1 - Les transitions de changement de but

Dans le cas idéal, un pilote ne change pas de but. Il prévoit sa route lors de la préparation du vol. Ainsi, lors de la phase de prédiction, on suit ce modèle «idéal» et on ne prédit pas le franchissement des transitions de changement de but, sauf dans le cas où la place immédiatement précédente est *marquée* (avec un jeton ●).

Dans le cas de la mission *Refuel*, on laisse au pilote la possibilité d'aller se ravitailler (en 6 ou 9). La première transition de changement de but rencontrée est donc celle qui permet au pilote de se dérouter pour aller se ravitailler en 6. Ainsi, lorsque l'activité *waypoint(5,unreached)* est prédite, la transition de changement de but est non franchissable pour la prédiction (figure 8). Elle le devient lorsque cette activité est marquée (figure 9).

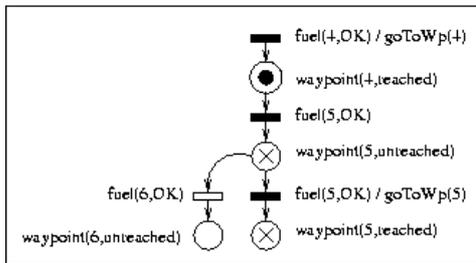


FIG. 8: Transition de changement de but non franchissable en prédiction

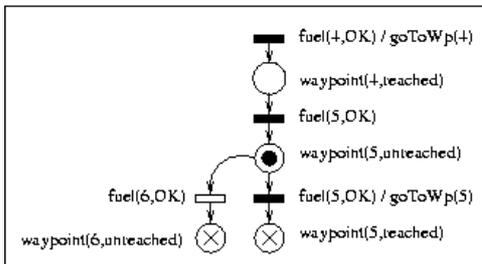


FIG. 9: Transition de changement de but franchissable en prédiction

4.2 - Intervalles de prédiction

L'horloge du *jeton-mission* est modifiée par les actions des transitions. Par exemple, lors du franchissement de la transition qui porte l'action *goToWp(3)* (mission *Refuel*), le *jeton-mission* subit les actions suivantes :

- modification de ses coordonnées (déplacement géographique)
 $position := coordonnées(3);$
- modification de son horloge (déplacement temporel)
 $horloge := horloge + tempsPourAllerA(3);$
- consommation de carburant (déplacement temporel)
 $niveau := niveau - conso * tempsPourAllerA(3).$

L'horloge du *jeton-mission* indique l'heure à laquelle l'avion est prévu dans une activité. Cette heure est communiquée à la place prédite. On

obtient donc un intervalle construit à partir de l'ensemble des heures où le jeton est prévu dans la place. On nommera ces intervalles les *intervalles de prédiction*. La figure 10 illustre ce principe :

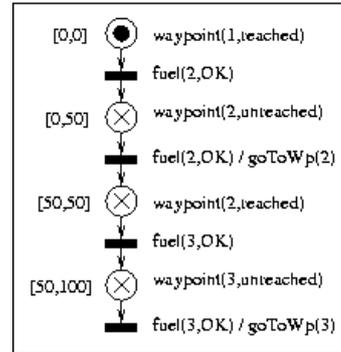


FIG. 10: Intervalles de prédiction

L'état initial correspond à l'activité *waypoint(1,reached)* à $t=0$. La première transition ne modifie pas la valeur de l'horloge du jeton car elle n'a pas d'action. L'intervalle de la première place est donc $[0,0]$. Le jeton entre dans la deuxième place à $t=0$, puis franchit la deuxième transition : son horloge est modifiée, $t=50$, temps nécessaire pour aller de 1 à 2, étant donné la vitesse de l'avion. On considère donc que le jeton a quitté sa place à $t=50$, et on construit l'intervalle $[0,50]$. On construit de la même façon les intervalles suivants.

Lors des phases de prédiction successives, on agrandit ces intervalles en y ajoutant les valeurs de l'horloge. Ainsi, une place possédant l'intervalle $[0,50]$ et à laquelle la prédiction indiquerait un intervalle de $[20,60]$ se verrait attribuer l'intervalle de prédiction $[0,60]$. L'intervalle de prédiction est toujours un intervalle (pas d'union d'intervalles et/ou de singletons disjoints).

4.3 - Horizon de prédiction

La présence d'une horloge lors de la phase de prédiction permet de définir une nouvelle notion : l'*horizon de prédiction*. C'est une valeur temporelle qui permet de limiter la construction de l'ensemble des places accessibles : on arrête la construction lorsque l'horloge du *jeton-mission* va au-delà de l'horizon.

Dans le cas de la mission *Refuel*, on fixe la valeur de l'horizon à 3 mn. Cette valeur est pertinente car elle permet de prévoir les actions du pilote sur 4 à 5 activités, ce qui correspond environ à deux sélections de points tournants. On a ainsi un bon aperçu des états susceptibles d'être atteints sans surcharger l'ensemble des places accessibles.

4.4 - Principe de prédiction

Si dans un premier temps, le fait de cocher sur un réseau l'ensemble des places prédites était suffisant, il faut maintenant revoir ce principe. Les réseaux n'étant pas linéaires, une place peut être marquée

plusieurs fois et donc posséder plusieurs intervalles de prédiction. On pourrait, comme on le fait lors de phases de prédiction successives, faire l'union de ces intervalles.

Cependant, une place souvent marquée aurait un intervalle de prédiction beaucoup trop grand, ce qui enlèverait tout l'intérêt de l'intervalle. De plus, les intervalles de prédiction seront utilisés lors de la mise en correspondance et leur construction a de l'importance.

On décide, plutôt que d'agrandir l'intervalle de prédiction, de dupliquer le réseau. Lorsqu'on est confronté à un choix (conflit au sens des réseaux de Petri), on crée une copie du réseau pour chaque transition franchissable.

Le problème qui subsiste alors est le suivant : lors de la phase de recalage suivante, il est possible que l'observation se mette en correspondance avec une place située en amont du conflit. On est alors amené à réitérer le processus, créant ainsi encore plus de réseaux (figure 11).

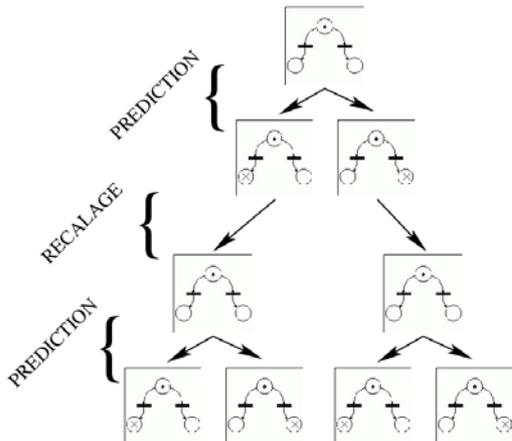


FIG. 11: Multiplication des réseaux en présence d'un conflit

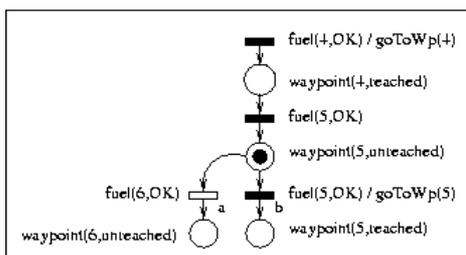


FIG. 12: Prédiction multiple : état initial

On remédie à ce problème en supprimant de chaque réseau les transitions conflictuelles, excepté celle qui est franchie. On obtient ainsi plusieurs réseaux sans interférence des intervalles de prédiction. Les figures 12, 13 et 14 illustrent ce principe : à l'état initial, on est confronté à un conflit mettant en cause deux transitions (notées *a* et *b*). On va donc construire deux réseaux correspondant chacun à une de ces deux transitions.

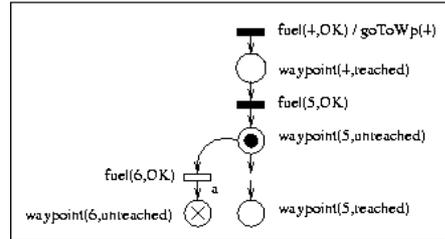


FIG. 13: Prédiction multiple : premier chemin

Le réseau de la figure 13 a été privé de la transition *b*. On continue donc la prédiction en franchissant la transition *a*.

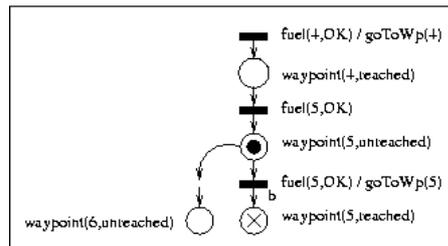


FIG. 14: Prédiction multiple : second chemin

On procède inversement pour le réseau de la figure 14.

5 - RECALAGE

La phase de recalage dispose des plans prédits et d'une nouvelle observation. La première étape est la *mise en correspondance* qui permet de reconnaître les activités correspondant à l'observation.

5.1 - Observation

Une observation est représentée par un objet dont la classe hérite de la classe *Jeton*. L'horloge indique l'heure de l'observation. L'observation possède de plus un cube de propriétés représentant l'état observé.

5.2 - Initialisation

Au lancement du système, celui-ci dispose d'une banque de plans non marqués représentant les plans de vol que le pilote peut suivre. Lors de l'arrivée de la première observation, les plans pouvant correspondre à l'activité observée sont alors sélectionnés.

Chaque place représente une activité et à ce titre porte un cube de propriétés. La mise en correspondance consiste en la comparaison du cube représentant l'observation avec les cubes associés aux places.

La mise en correspondance initiale est identique à celle de la phase de recalage, à ceci près qu'aucune correspondance temporelle n'est faite. Nous distinguerons donc mise en correspondance d'activités, mise en correspondance temporelle et mise en correspondance généralisée.

5.3 - Mise en correspondance d'activités

La mise en correspondance d'activités confronte l'observation aux cubes des activités prédites.

On notera \mathcal{A} l'ensemble des mises en correspondance d'activités. La définition de chacun des éléments de \mathcal{A} , tirée de Tessier (2003), est :

Définition 1 (Mise en correspondance d'activités). Soit A le cube des propriétés de l'activité et O celui de l'observation. On dit que l'activité est en correspondance :

$$PARFAITE \Leftrightarrow A = O$$

$$PARTIELLE\ 1 \Leftrightarrow (A \subset O) \vee (O \subset A)$$

$$PARTIELLE\ 2 \Leftrightarrow (A \not\subset O) \wedge (O \not\subset A) \wedge (A \cap O \neq \emptyset)$$

$$IMPOSSIBLE \Leftrightarrow A \cap O = \emptyset.$$

Exemple (Mise en correspondance d'activités).

Prenons comme cube d'observation

$$O = \{\text{waypoint}(3, \text{reached}), \text{vitesse}(200)\}$$

que l'on doit confronter à deux activités prédites de cubes respectifs

$$A1 = \{\text{waypoint}(3, \text{reached})\}$$

$$A2 = \{\text{waypoint}(3, \text{reached}), \text{vitesse}(150)\}.$$

La mise en correspondance de O et $A1$ est

$$PARTIELLE\ 1 : A1 \subset O.$$

En revanche, la mise en correspondance de O et $A2$

$$\text{est } PARTIELLE\ 2 \quad (A2 \not\subset O, \quad O \not\subset A2 \quad \text{et} \\ A2 \cap O = \{\text{waypoint}(3, \text{reached})\}).$$

Définition 2 (Relation de préférence).

La relation de préférence entre ces différents éléments est :

$$PARFAITE \leq PARTIELLE1 \leq PARTIELLE2 \leq IMPOSSIBLE$$

où $A \leq B \Leftrightarrow A$ est préférée à B .

5.4 - Mise en correspondance temporelle

L'ajout d'un intervalle de prédiction aux places fait intervenir un nouveau critère, celui de la mise en correspondance temporelle.

L'ensemble des mises en correspondance temporelles est \mathcal{T} dont les éléments sont décrits par :

Définition 3 (Mise en correspondance temporelle).

Soit t l'instant de l'observation et I l'intervalle de prédiction de l'activité. On dit que l'activité est en correspondance :

$$IN \Leftrightarrow t \in I$$

$$POST \Leftrightarrow t > \text{sup}(I)$$

$$ANTE \Leftrightarrow t < \text{inf}(I)$$

Contrairement au cas de la mise en correspondance d'activités, on ne peut ici classer les différents cas selon un ordre total. On choisira un ordre partiel dans lequel $IN \leq ANTE$ et $IN \leq POST$.

5.5 - Mise en correspondance généralisée

Définition 4 (Mise en correspondance généralisée).

La réunion des deux mises en correspondance ci-

dessus construit une relation d'ordre partiel. La mise en correspondance d'une observation et d'une activité est affectée de deux attributs (une paire $(a, t) \in \mathcal{A} \times \mathcal{T}$).

On obtient ainsi 12 ($\#\mathcal{A} \times \#\mathcal{T}$) mises en correspondance possibles que l'on peut ordonner partiellement d'après les relations d'ordre précédentes.

Les cas particuliers et délicats sont ceux obtenus lorsqu'il faut classer, par exemple, une mise en correspondance (PARFAITE, ANTE) et (PARTIELLE 2, IN). En d'autres termes, est-il préférable de croire à une erreur de manipulation (mauvaise indication de la fréquence de la prochaine balise \rightarrow PARTIELLE 2) ou à une erreur de prédiction (le pilote a volé plus vite que prévu \rightarrow ANTE) ?

Dans le contexte de ce travail, on privilégiera la mise en correspondance d'activités sur la mise en correspondance temporelle. On définit donc une relation de préférence partielle dont la règle est :

Définition 5 (Relation de préférence).

Soit $m=(a, t)$ et $m'=(a', t')$ deux mises en correspondance. Les cas possibles sont :

$$a \leq a' \Rightarrow m \leq m'$$

$$(a = a') \wedge (t \leq t') \Rightarrow m \leq m'$$

$$(a = a') \wedge (t = t') \Rightarrow m = m'$$

$$(a = a') \wedge (t \text{ et } t' \text{ incomparables}) \Rightarrow m \text{ et } m' \text{ incomparables.}$$

5.6 - Principe de recalage

Lors de l'arrivée d'une observation, celle-ci est mise en correspondance avec les activités prédites. Les mises en correspondance IMPOSSIBLE sont écartées, tandis que les activités restantes (dites *recalées*) sont classées selon l'ordre de préférence exposé dans la définition. Ce nouvel ensemble de plans reconnus est ensuite livré à la phase de prédiction en attendant l'arrivée d'une nouvelle observation.

6 - RÉSULTATS

Afin de valider le modèle et les principes définis, nous avons mis au point un logiciel (figure 15) qui met en œuvre les différents algorithmes et qui permet de visualiser l'ensemble des plans manipulés lors des différentes phases (alternativement prédiction et recalage).

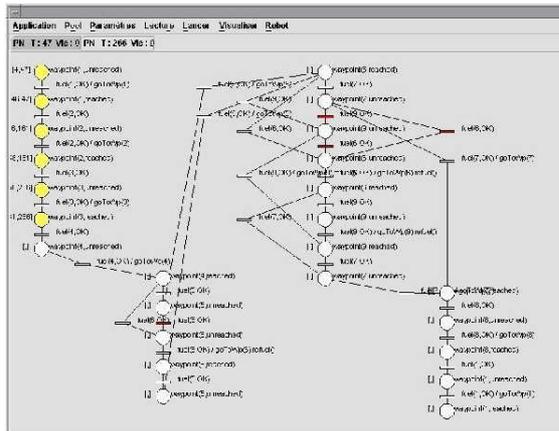


FIG. 15: *Logiciel de suivi d'activité*

À partir des paramètres de vol provenant d'expérimentations menées sur la mission *Refuel* (Dehais, 2003), on a pu suivre et prédire l'activité de plusieurs pilotes expérimentés confrontés à des conflits, et déterminer *a posteriori* les dates de ces conflits et les moments pertinents pour l'envoi de contre-mesures.

7 - CONCLUSIONS ET PERSPECTIVES

Dans l'état actuel, le modèle défini et les principes exposés de Kalmansymbo-aéro permettent le suivi et la prédiction d'activité sur le cas de la mission *Refuel*. La modélisation de différentes procédures (par exemple approche sur l'aéroport de Toulouse-Blagnac) permettra d'affiner ces principes. La poursuite de ces travaux fait l'objet d'une thèse au cours de laquelle les principes de prédiction seront approfondis, un lien entre filtres de Kalman numérique et symbolique sera étudié, et la connexion sera établie avec la détection de conflits et l'envoi des contre-mesures.

8 - BIBLIOGRAPHIE

B.E.A. (1996), *Statistiques générales 1994-1996*, <http://www.bea-fr.org>

Brémond, F. (1997), *Environnement de résolution de problèmes pour l'interprétation de séquences d'images*, Thèse, Sophia-Antipolis, France.

Callantine, T.J. (2002), *Activity tracking for pilot error detection from flight data*, NASA, USA.

Chaudron, L., Maille, N., Boyer, M. (2003), «The CUBE lattice model and its applications», *Applied Artificial Intelligence*, vol 17, n° 3, pp 201-242.

Cohen, H. (1993), *Neuropsychologie expérimentale et clinique*, Gaetan Morin.

Cossart-Jaupitre, C. (1999), *Un estimateur symbolique pour le suivi de situation*, Thèse, Onera-Supaéro, France.

Dehais, F. (2002), «Modelling cognitive conflict in pilot's activity», in *STAIRS*, Lyon, France.

Dehais, F., Tessier, C., Chaudron, L. (2003), «GHOST : experimenting conflicts countermeasures in the pilot's activity», in *IJCAI*, Acapulco, Mexico.

Dousson, C. (1994), *Suivi d'évolutions et Reconnaissance de chroniques*, Thèse, LAAS-CNRS, France.

Ghallab, M. (2002), *Planification et Raisonnement Temporel*, Notes de cours, LAAS-CNRS.

Largouët, C., Cordier, M.O. (2000), «Timed automata model to improve the classification of a sequence of images», in *ECAI'00*, Berlin, Allemagne.

Rota, N.A., Thonnat, M. (2000), «Activity recognition from video sequence using declarative models», in *ECAI'00*, Berlin, Allemagne.

Tessier, C. (2003), «Towards a commonsense estimator for activity tracking», in *AAAI*, Palo Alto CA, USA.

Valette, R. (2000), *Les Réseaux de Petri*, notes de cours, ENAC.