



HAL
open science

Actes de la 1^{re} conférence sur les Applications Pratiques de l'Intelligence Artificielle

Domitile Lourdeaux

► **To cite this version:**

Domitile Lourdeaux. Actes de la 1^{re} conférence sur les Applications Pratiques de l'Intelligence Artificielle: APIA 2015. Plate-Forme Intelligence Artificielle, Association Française pour l'Intelligence Artificielle, 2015. hal-04596458

HAL Id: hal-04596458

<https://ut3-toulouseinp.hal.science/hal-04596458>

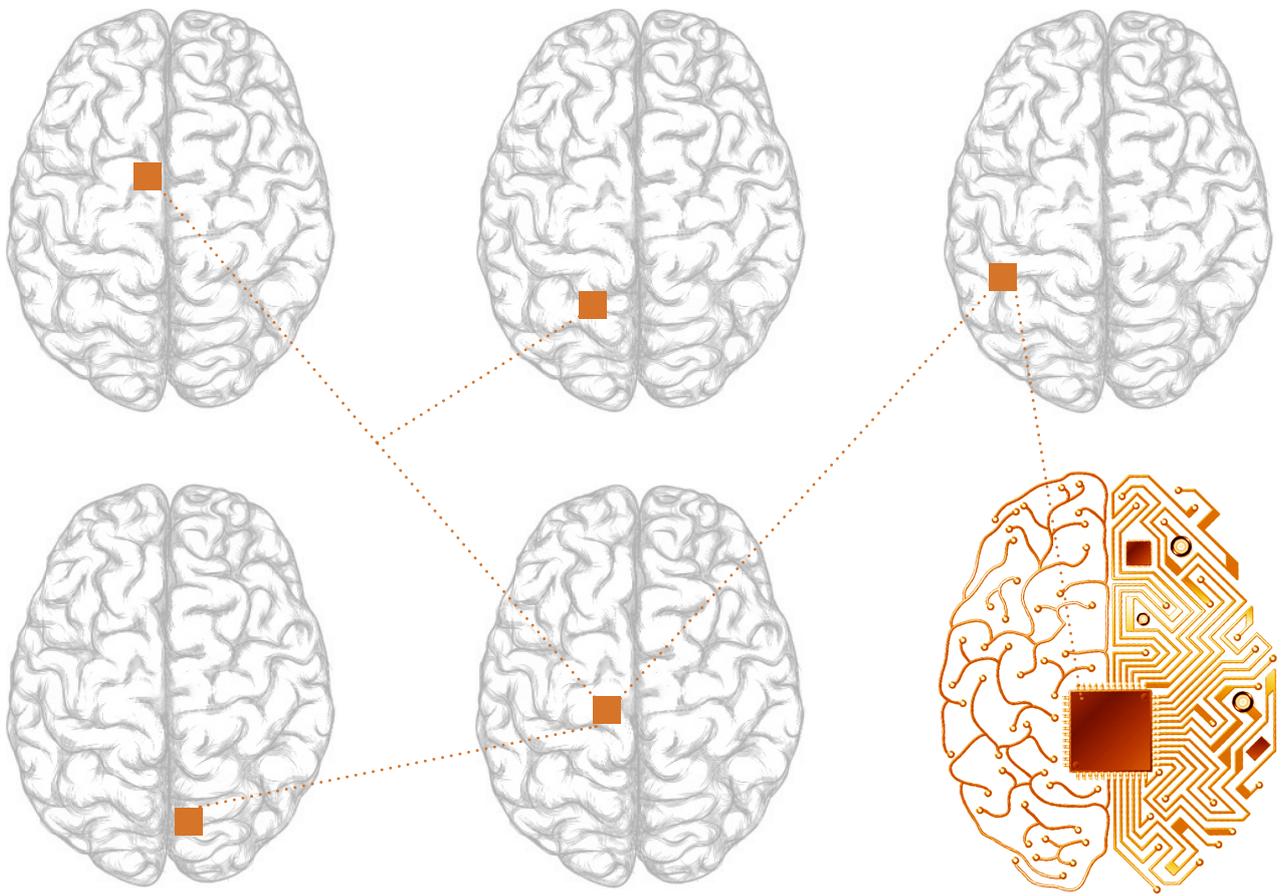
Submitted on 31 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License



PFIA 2015

Plate-forme Intelligence Artificielle
Rennes

Actes APIA

Président CP : Domitile Lourdeaux



AFIA

Association française
pour l'Intelligence Artificielle

1^{ère} Conférence sur les Applications Pratiques de l'Intelligence Artificielle

Présentation de la conférence

La plate-forme accueille la première conférence sur les Applications Pratiques de l'Intelligence Artificielle (APIA).

Les recherches en IA menées ces dernières années ont abouti à des résultats prometteurs et l'IA se trouve au cœur de nombreuses applications très performantes qui révolutionnent notre vie quotidienne ou plus discrètes, d'autres très prometteuses sont en passe de le devenir.

L'objectif de la 1^{ère} Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle (APIA 2015) est de faire un tour d'horizon des applications concrètes de l'IA qui couronnent de succès cette opérationnalisation de l'IA et des travaux de recherche qui le seront bientôt.

Ces applications peuvent mettre en œuvre une ou plusieurs facettes de l'IA qui ont permis ces succès, comme :

- Agents autonomes et systèmes multiagents : décision individuelle, décision collective
- Applications de l'Intelligence Artificielle, méthodologie, évaluation
- Apprentissage numérique et symbolique
- Environnements Informatiques d'apprentissage humain
- Évolution artificielle, systèmes situés, systèmes adaptatifs
- Fouille de données, bases de données avancées, web sémantique
- Ingénierie et partage des connaissances
- Intelligence collective, intelligence sociale, réseaux sociaux
- Interfaces Intelligentes, interaction homme-machine, intelligence ambiante
- Langages de programmation pour l'IA, programmation logique
- Logique formelle et outils pour l'Intelligence Artificielle, sémantique
- Passage à l'échelle, organisation de systèmes, émergence
- Plates-formes et environnements de développement en IA
- Raisonnement à base de modèles, raisonnement à partir de cas
- Raisonnement probabiliste et incertain, logique floue
- Raisonnement spatial et temporel, environnements physiques
- Recherche opérationnelle, programmation par contraintes, ordonnancement
- Représentation des connaissances, extraction et gestion des connaissances
- Réseaux de neurones artificiels, approches neuromimétiques
- Robotique, vision par ordinateur, capteurs intelligents, systèmes physiques
- Sciences cognitives et Intelligence Artificielle, cognition, informatique affective
- Systèmes à base de règles, planification, aide à la décision
- Traitement automatique du langage, terminologie, langage naturel contrôlé, explication
- Traitement du signal et de l'image, traitement de la parole
- Web intelligence, internet du futur, protection de la vie privée

Quelles sont les applications qui s'appuient sur de l'IA ou qui nécessitent de l'IA ? Du système de surveillance militaire au système d'aide au diagnostic médical, du climatiseur à l'assistant personnel, du système d'aide à la conduite à l'analyse de données massives, etc., les applications sont nombreuses. Qu'elles soient industrielles, sociétales, économiques, politiques, environnementales, artistiques ou autre, la conférence est l'occasion de présenter des applications concrètes et des travaux dont l'objet d'étude n'est pas uniquement des cas de laboratoire. Nous attendons des contributions dans des domaines très divers : automobile, robotique, militaire, logistique, télécommunications, domotique, agronomie, réseaux sociaux, risque, big data, médical, aide à la personne, jeux vidéo, réalité virtuelle/mixte, musées, EIAH, serious games, récit interactif, et bien d'autres encore. . .

L'objectif est également de voir comment ces applications concrètes font remonter des verrous scientifiques que l'IA doit résoudre pour démocratiser encore plus son utilisation. Par exemple : comment la prise en compte des réalités concrètes vues dans leur globalité amènent la prise en compte par l'IA de problèmes complexes décuplés qu'il n'est pas possible de rendre compte dans des approches réductionnistes de problèmes de laboratoire ? L'IA est-elle suffisamment expressive et intelligible pour être utilisée ? Est-elle fiable et robuste ? Est-elle capable de passer à l'échelle ? Quels sont les problèmes éthiques liés à son utilisation ? Il ne s'agit pas d'opposer recherche fondamentale/appliquée mais laboratoire/terrain et comment les applications concrètes apportent des problématiques fondamentales ou encore comment des recherches fondamentales apportent des solutions à des problèmes complexes difficiles à résoudre sans IA.

La conférence favorisera l'échange entre chercheurs académiques, industriels et autres pour qu'ils puissent partager leurs expériences, débattre des différents verrous qu'ils rencontrent, les méthodes qu'ils mettent en œuvre pour enrichir le potentiel applicatif des modèles et outils de l'IA et les besoins naissants, en mettant en valeur l'IA de ces applications.

Conférencier Invité : Rodney Brooks (CSAIL/MIT - Rethink Robotics)

Biographie

Rodney Brooks a été le responsable du laboratoire d'IA du MIT pendant plus de 20 ans où il y a développé ses recherches en vision, en intelligence artificielle et en robotique. Pendant des décennies, il a conçu des robots pour effectuer des tâches très spécifiques – soudage, rivetage, livrant le courrier interne. Avec le développement des techniques de robotique, son objectif est maintenant de concevoir un robot qui peut comprendre par lui-même, comme les humains le font, en particulier par l'interaction et en contexte. Il est le co-fondateur de l'entreprise iRobot et dirige maintenant Rethink Robotics, une entreprise développant une nouvelle génération de robots d'entreprise adaptés à un environnement de travail mixte humain-robot.

Adding a Little Intelligence to Factory Robots

Traditional industrial robots show up without any software besides a compiler or interpreter for a robot motion language, and no external sensors or safety systems. Systems integrators and consultants are used to define how the robot will operate in a production line, sensors and interfaces to other machines are procured, and a specific new program is written for the application. In this talk I will describe a new way of thinking about industrial robots, where they come with sensors, machine interfaces, a User Interface, and in intelligence software stack that together provide most of the control that is needed to get the robot to do useful work. Factory workers then show the robot what it is to do, in task coordinates rather than robot coordinates, and the robot is able to carry out the task and notice anomalous situations. The robots are not very intelligent by the standards of AI researchers, but they are robust and reliable, and just a little intelligence makes them so much more powerful than traditional industrial robots.

Comité de Programme

Présidente du comité de programme

Domitile Lourdeaux, Université de Technologie de Compiègne, UMR CNRS 7253 Heudiasyc

Membres du comité de programme

- Camille Barot, North Carolina State University – Liquid Narrative Group, USA
- Jean-Paul Barthès, UTC - Heudiasyc UMR CNRS 7253, France
- Didier Bazalgette, DGA, France
- Jean-Sébastien Bedo, Orange Labs Recherche, France
- Fabien Belmonte, RATP, France
- Alain Berger, Ardans, France
- Come Bissuel, EDF R&D Dpt. STEP, France
- Antoine Bordes, Facebook, USA
- Abdelmadjid Bouabdallah, UTC - Heudiasyc UMR CNRS 7253, France
- Mathieu Boussard, MASA Group, France
- Bertrand Braunschweig, INRIA, France
- Valérie Camps, Université Paul Sabatier - IRIT UMR CNRS 5505, France
- Kevin Carpentier, UTC - Heudiasyc UMR CNRS 7253, France
- Marc Cavazza, Teeside University, UK
- François Cheriaux, EDF R&D Dpt. STEP, France
- Pierre Chevallier, ENIB - Lab-STICC, UMR CNRS 6285, France
- Max Chevalier, Université Paul Sabatier - IRIT UMR CNRS 5505, France
- Emmanuel Chiva, SILKAN et Marine Nationale, France
- Caroline Chopinaud, MASA Group, France
- Jan Ciger, Reviattech, France
- Laurent Da Dalto, Mimbus, France
- Philippe David, SNCF, France
- Etienne de Sévin, MASA Group, France
- Florent Delomier, Synakène
- Yves Demazeau, LIG CNRS, France
- Thierry Denoeux, UTC - Heudiasyc UMR CNRS 7253, France
- Sylvie Despres, Université Paris 13 Nord – LIMB, France

— Sébastien Destercke, UTC - Heudiasyc UMR CNRS 7253, France
— Jean-Yves Donnart, Thalès Training & Simulation, France
— Bernard Dubuisson, UTC - Heudiasyc UMR CNRS 7253, France
— Yves Duthen, Université Toulouse Capitole – IRIT UMR CNRS 5505, France
— Lydie Edward, CRICM UPMC / INSERM S975, France
— Dominique Ernadote, Airbus Defence & Space, France
— Bernard Espinasse, Université Aix-Marseille – LSIS, France
— Vincent Feuillard, Airbus GROUP Innovations, France
— Fabien Flacher, Thalès, France
— Béatrice Fuchs, Université Jean Moulin Lyon 3 - LIRIS, France
— Jean-Gabriel Ganascia, Pierre et Marie Curie – LIP6, France
— Catherine Garbay, Université Joseph Fourier Grenoble - LIG CNRS, France
— Samir Garbaya, ENSAM ARTS ET METIERS PARISTECH, France
— Domenico Gattuso, Mediterranea University of Reggio Calabria, Italie
— Jérôme Gensel, Université Pierre Mendès France Grenoble 2 – LIG CNRS, France
— Abdelkader Gouaich, Université Montpellier - LIRMM, France
— Michel Grabisch, Université Paris I Panthéon-Sorbonne, France
— Zahia Guessoum, LIP6 et CReSTIC, France
— Christophe Guettier, SAGEM, France
— Olivier Gutknecht, ACM, France
— Céline Hudelot, Ecole Centrale Paris - MAS, France
— Jean-Marc Labat, Université Pierre et Marie Curie – LIP6, France
— Mathieu Lafourcade, Université Montpellier II - LIRMM, France
— Pascale Le Gall, Ecole Centrale Paris - MAS, France
— Dominique Lenne, UTC - Heudiasyc UMR CNRS 7253, France
— Jean Lieber, Université de Lorraine - LORIA, France
— Vanda Luengo, Université Joseph Fourier Grenoble - LIG CNRS, France
— Sylvain Mahé, EDF R&D Dpt. STEP, France
— Alamin Mansouri, Université de Bourgogne - L2ei UMR CNRS 6306, France
— Jacques Marc, INRS, France
— Jean-Charles Marty, Université de Savoie - LIRIS Lyon, France
— Nada Matta, UTT - TechCICO, France
— Philippe Mathieu, University Lille 1 – LIFL, France
— Juliette Mattioli, Thalès, France
— Eunika Mercier, Bull, France
— Alain Mille, Université Claude Bernard Lyon 1 - LIRIS, France
— Philippe Morignot, LIFEWARE, INRIA, France
— Pierre Morizet Mahoudeaux, UTC - Heudiasyc UMR CNRS 7253, France
— Mustapha Mouloua, University of Central Florida, USA
— Amedeo Napoli, Université de Lorraine – LORIA, France
— Pascal Negros, Arch4IE, France
— Thuy N'GUYEN, EDF R&D Dpt. STEP, France
— Selmin Nurcan, Université Paris I, France
— Jean-Marc Ogier, Université de la Rochelle – L3i, France
— François Pachet, Sony CSL, France
— Pavol Barger, Société Monsieur Pavol Barger, France
— Romaric Redon, Airbus Group Innovations, France
— Jean-Christophe Riat, PSA Peugeot-Citroën, France
— Benoit Ricard, EDF R&D Dpt. STEP, France
— Jean Rohmer, ESILV, France
— Dominique Romary, France
— Francis Rousseaux, Université Reims – CReSTIC et IRCAM, France
— Nicolas Sabouret, Université Paris Sud – LIMSI, France
— Jean Sass, Dassault Aviation, France
— Nicolas Schneider, Airbus GROUP Innovations, France
— Françoise Soulie Folgeman, KXEN, France
— Nicolas Szilas, Université de Genève - TECFA-FPSE, Suisse
— Catherine Tessier, ONERA-DCSD, France
— Nicolas Usunier, UTC - Heudiasyc UMR CNRS 7253, France

Table des matières

Carole Adam, Elise Beck, Julie Dugdale. Modélisation du comportement tactique de la population australienne dans un feu de brousse	6
Lamia Belouaer, Matthieu Boussard, Patrick Bot. Strategic Decision Making In Yacht Match Racing : Stochastic Game Approach	15
Alain Berger. Evolution dans l'industrie du métier de l'ingénieur cognitif ou l'ingénieur de la connaissance entre 1985 et 2015	23
Stephan Brunessaux, Patrick Giroux. Dix ans : L'âge de la maturité pour la plate-forme WebLab	35
Caroline Chopinaud, Etienne de Sevin. Conception de « Scènes de Vie » pour le Peuplement d'Environnements Virtuels	43
Andrei Doncescu, Pierre Siegel. Deux applications des logiques non monotones : Décision à bord d'un sous marin, Représentation et découverte pour la cellule cancéreuse	49
Elias Egho, Dominique Gay, Marc Boullé, Nicolas Voisine, Fabrice Clérot. Une approche Bayésienne pour l'apprentissage de règles séquentielles de classification	57
Emmanuelle Gaillard, Jean Lieber, Emmanuel Nauer. TAAABLE : un système de raisonnement à partir de cas qui adapte des recettes de cuisine	65
Christophe Guettier, Willy Lamal, Israel Mayk, Jacques Yelloz. Modélisation, Résolution et Expérimentation de méthodes de Planification et d'Ordonnancement Collaboratif pour le Commandement de Brigade	71
Thomas Hureau, Nicolas Sabouret, Yvon Haradji, François Sempé. Simulations multi-agents de l'activité humaine : application dans le contexte énergétique résidentiel français	79
Florence Le Ber, Laura Martin, Alain Mille, Marc Benoît. Un système de RÀPC pour prévoir l'allocation de parcelles agricoles	87
Philippe Morignot, Oyunchimeg Shagdar, Fawzi Nashashibi. Densité de trafic émergente pour des véhicules intelligents communiquants guidés par heuristique	95
Florian Richoux. Problèmes d'optimisation dans les jeux avec GHOST	103
Erwan Tranvouez, Ali Oulhaci, Bernard Espinasse, Sébastien Fournier. Simulation et Evaluation Multi-Agents dans les Jeux Sérieux collaboratifs : le projet SIMFOR	109
Aravind Venkatesan, Nordine El Hassouni, Florian Phillipe, Cyril Pommier, Hadi Quesneville, Manuel Ruiz, Pierre Larmande. Towards efficient data integration and knowledge management in the Agronomic domain	117

Modélisation du comportement tactique de la population australienne dans un feu de brousse

Carole Adam¹, Elise Beck², Julie Dugdale^{1,3}

¹ UNIV. GRENOBLE ALPES, LIG, F-38000 GRENOBLE, FRANCE
carole.adam@imag.fr

² UNIV. GRENOBLE ALPES, PACTE, F-38000 GRENOBLE, FRANCE
elise.beck@ujf-grenoble.fr

³ UNIVERSITY OF AGDER, NORWAY
julie.dugdale@imag.fr

Résumé : Dans cet article nous proposons un modèle à base d'agents du comportement de la population pendant un feu de brousse dans l'état du Victoria en Australie. Ce modèle a pour but de servir ultérieurement de base à des simulations d'aide à la décision des gestionnaires du risque. Pour que les résultats d'une telle simulation soient valides, notre modèle doit donc être aussi réaliste que possible, en particulier en ce qui concerne l'architecture des agents. Après avoir introduit le contexte au sujet du comportement de la population du Victoria dans les incendies récents, nous justifions le besoin d'une architecture BDI pour les agents du modèle. Suite au manque d'outils supportant l'intégration de tels agents dans les simulations, nous montrons comment une méthodologie existante pour la modélisation des tactiques militaires peut être adaptée dans ce contexte. La contribution de cet article est double : nous fournissons un modèle à base d'agents du comportement de la population dans les feux de brousse, ainsi qu'une méthodologie générale qui peut être utilisée par les modélisateurs dans le domaine de la gestion de crise.

Mots-clés : Modélisation et simulation à base d'agents, modèles de comportement d'agents, méthodologies multi-agent, architecture BDI (Belief-Desire-Intention).

1 Introduction

Les techniques d'IA ont trouvé récemment un nouveau domaine d'application particulièrement intéressant d'un point de vue sociétal : la gestion de crise, domaine où les gestionnaires du risque ont besoin d'outils d'aide à la décision. En particulier, la modélisation et la simulation offrent un outil puissant pour évaluer l'efficacité des politiques de gestion de crise et pour les améliorer avant la survenue d'une crise réelle où des vies humaines sont en jeu. Ces techniques donnent un grand degré de contrôle sur les variables contextuelles, permettant de reproduire des expérimentations avec exactement les mêmes paramètres, ou au contraire de tester de multiples scénarios. Comparés aux modèles mathématiques, les modèles multi-agents offrent une description du comportement humain à un plus haut niveau d'abstraction, plus fidèle à la réalité. Mais pour que les résultats de telles simulations soient valides et donc utiles, le modèle sous-jacent doit être très réaliste (van Ruijven, 2011), à la fois en ce qui concerne l'environnement (simulation spatialisée basée sur des données réelles issues de Systèmes d'Information Géographique) et le comportement des individus (architecture agent d'une complexité appropriée, autorisant un comportement proactif, émotionnel). Malheureusement ce n'est pas toujours le cas, et de nombreuses simulations utilisent des environnements simplistes de type grilles et/ou des agents réactifs trop simplifiés, notamment à cause du manque d'outils pour soutenir l'intégration d'agents plus complexes (Adam *et al.*, 2011). Il existe par contre déjà de bons modèles du comportement du feu (par exemple Phoenix, cf (Scerri *et al.*, 2012)). Des outils industriels existent aussi pour développer des simulations réalistes, tels que SYNERGY (MASA Group,

2015), mais le modèle de comportement sous-jacent n'est pas publié ; au contraire nous souhaitons privilégier une conception participative du modèle de comportement, à l'aide d'outils de visualisation permettant aux acteurs impliqués de réagir précocément et de le valider. Dans cet article nous nous intéressons à la prise de décision de la population face à un feu de brousse, dans l'état du Victoria en Australie. Nous nous basons sur notre expérience antérieure dans la création de modèles et simulations spatialisés pour la gestion de crise (Beck *et al.*, 2014; Truong *et al.*, 2013; Minh *et al.*, 2011). Nous montrons que des agents complexes (par exemple des agents BDI – *Belief, Desire, Intention*) sont nécessaires pour modéliser le comportement humain dans ce contexte. Afin de faire face à l'absence d'outils pour intégrer des agents complexes dans notre simulation, nous utilisons une nouvelle méthodologie créée récemment pour modéliser les tactiques militaires (TDF, (Evertsz *et al.*, 2015)), et montrons comment elle peut être adaptée pour modéliser le comportement tactique des civils.

2 Contexte

2.1 Les feux de brousse dans l'état du Victoria

L'état du Victoria en Australie subit des feux de brousse plus ou moins graves tous les étés. Pendant le "Black Saturday", le 7 février 2009, des incendies particulièrement violents ont fait 173 victimes, détruit de nombreuses maisons et brûlé des hectares de forêt. Malgré des campagnes de sensibilisation régulières, la politique de l'état intitulée "Préparez-vous, Restez et défendez, ou Partez tôt" n'a pas été suivie par la population. Des rapports de retour d'expérience de la Commission Royale (Teague *et al.*, 2009) ont établi que la plupart des habitants sont trop attentistes et restent jusqu'à ce que le feu soit trop proche, attendant un ordre d'évacuation personnel alors que les services d'urgence ne diffusent que des alertes générales. La plupart des victimes sont ainsi des personnes ayant évacué au dernier moment. Les chercheurs ont tenté d'expliquer ce comportement attentiste inattendu, et ont trouvé que la surcharge d'information pouvait mener à une "indécision paralysante" (McNeill *et al.*, 2014). Ces résultats suggèrent que des informations et messages d'alertes mieux ciblés pourraient aider à réduire le nombre de victimes, mais il est difficile de prédire l'impact réel de telles mesures à l'avance.

D'autres études (Fire Services Commissioner, 2013; Rhodes, 2014) confirment la divergence entre le comportement de la population attendu par les gestionnaires du risque (sur lequel ils basent leurs décisions d'envoi d'informations) et le comportement réel de la population pendant les incendies (qui détermine leur besoin d'informations). Par conséquent beaucoup d'habitants ignorent les messages, les considérant non pertinents, sans intérêt ou absurdes. Les auteurs de l'étude identifient sept archétypes de comportement, chacun avec leurs propres motivations et besoins d'informations, et recommandent que les gestionnaires du risque les prennent tous en compte dans la rédaction de leurs messages.

2.2 Profils de comportement

Les sept archétypes de comportement de la population identifiés par (Fire Services Commissioner, 2013; Rhodes, 2014) sont les suivants :

- **Défenseurs compétents** (can-do defenders) : déterminés à protéger leur maison, connaissant bien la région, expérimentés et compétents, orientés vers l'action, autonomes, confiants ;

- **Défenseurs réfléchis** (considered defenders) : fortement engagés à rester et défendre leur maison, conscients des risques, efforts délibérés de préparation et d’entraînement ;
- **Défenseurs économiques** (livelihood defenders) : engagés à rester et défendre ce qu’ils considèrent comme leur moyen de subsistance (ferme, hôtel...) et donc bien préparés ;
- **Observateurs de menace** (threat monitors) : n’ont pas l’intention de rester face à une menace sérieuse, ni de partir avant qu’ils le jugent nécessaire, attendent de voir ;
- **Eviteurs de menace** (threat avoiders) : conscients du risque et de leur vulnérabilité, planifient de partir tôt avant toute menace réelle, mais pas de plan s’ils sont surpris ;
- **Réactifs inconscients** (unaware reactors) : inconscients du risque, ne se sentent pas concernés, aucune connaissance des réactions appropriées ni préparation (touristes...);
- **Isolés vulnérables** (isolated and vulnerable) : physiquement ou socialement isolés (p. ex. personnes âgées ou handicapées) ce qui limite leur capacité à répondre adéquatement.

Chaque archétype a donc un comportement différent, avec ses propres caractéristiques : motivation sous-jacente, niveau de conscience des risques, connaissances, stratégie de réponse, et donc ses propres besoins d’informations associés. Les auteurs de l’étude notent aussi que ces archétypes ne sont pas forcément liés à des caractéristiques socio-démographiques, car les individus peuvent changer de comportement en fonction des circonstances. Nous discutons maintenant le type d’architecture agent requise pour modéliser ces comportements.

3 Architecture agent

Différentes architectures peuvent être utilisées dans les simulations à base d’agents, offrant divers niveaux de complexité et niveaux d’abstraction, et différents types de raisonnement. Le choix de l’architecture doit donc être adapté à l’application visée (Balke & Gilbert, 2014). Notre application à la gestion de crise demande un modèle réaliste du comportement réel de la population afin que les résultats des simulations soient valides et fiables pour les aider les décisions des gestionnaires du risque. Nous justifions ici l’intérêt d’une architecture BDI pour modéliser fidèlement les comportements de la population identifiés ci-dessus.

3.1 Pré-requis pour l’architecture d’agent cognitif

Les sept archétypes de comportement ci-dessus diffèrent sur un certain nombre de caractéristiques qui influencent les décisions et actions des individus pendant un incendie et doivent donc être modélisées. Chaque archétype a :

- une **intention** principale (fuir ou défendre) qu’il est (plus ou moins) engagé à accomplir
- des **désirs** potentiellement conflucuels (combattre le feu mais protéger sa vie, fuir mais seulement si nécessaire, prendre le chemin le plus court mais éviter les embouteillages, partir tôt mais finir des choses avant, regarder l’incendie mais éviter les dangers...)
- des **croyances** sur soi-même (confiance en soi), sur le feu (conscience des risques, expérience), sur l’environnement (état des routes, position des abris), et sur les autres (position des amis et famille, comportement attendu des pompiers ou des voisins). Ces croyances peuvent être incomplètes et éventuellement inexactes (fausse croyance de sécurité, position inconnue ou erronée des abris).

En conséquence chaque archétype nécessite certains types d'informations qui sont pertinents par rapport à leur intention (progression de l'incendie, position des abris, trafic sur les routes...).

Les individus peuvent **réagir** aux stimuli de leur environnement (alerte incendie, ordre d'évacuation, proximité immédiate du feu...) ou **agir proactivement** en utilisant des **plans** pour accomplir leur intention. Leur comportement peut aussi être influencé par leurs **émotions** (stress paralysant, anxiété de perdre des proches ou ses moyens de subsistance, espoir irrationnel que le feu disparaisse ou évite la région, peur du feu ou peur pour ses proches poussant à fuir...). Par conséquent l'architecture d'agent choisie doit être une architecture de type cognitive, suffisamment complexe (comportement réactif et proactif basé sur les buts et plans), et fournir des concepts de haut niveau tels que les croyances, désirs, intentions et émotions.

3.2 Architecture BDI (Belief-Desire-Intention)

Le modèle BDI (Belief, Desire, Intention) est précisément basé sur la psychologie populaire et propose de modéliser les agents en termes de leurs croyances, désirs et intentions (Rao & Georgeff, 1995). Ce modèle offre donc le bon niveau d'abstraction pour facilement intégrer des données comportementales issues d'études de terrain ou d'experts du domaine, mais aussi pour expliquer les résultats des simulations aux gestionnaires de crise et au public. En effet le comportement des agents est décrit en termes de plans, pertinents dans différents contextes, et appliqués pour atteindre différents buts. Le modèle BDI standard peut aussi être étendu pour tenir compte des émotions (Adam *et al.*, 2009). Les architectures cognitives comme SOAR (Rosenbloom *et al.*, 1993) ou ACT-R (Anderson, 1983) offrent aussi le bon niveau d'abstraction, mais les concepts sous-jacents plus complexes rendent plus difficile l'explication du comportement généré, ce qui est gênant pour notre application à l'aide à la décision.

L'architecture BDI a été largement utilisée en génie logiciel orienté agent (Wooldridge, 2008; Padgham & M. Winikoff, 2004) mais plus rarement en modélisation et simulation orientées agent. Des outils existent pourtant (par exemple CoJACK (Evertsz *et al.*, 2008)) mais ils sont conçus par et pour des informaticiens et finalement peu utilisés par des modélisateurs issus d'autres disciplines. Par ailleurs ceux-ci préfèrent en général des modèles microscopiques plus simples permettant l'émergence de comportements macroscopiques complexes. Ces réticences sont notamment dues à la peur de la difficulté du passage à l'échelle des modèles plus complexes, ou à l'absence d'outils simples dédiés aux non-informaticiens pour leur faciliter la création de modèles complexes, deux problèmes pourtant contournables (Adam *et al.*, 2011). Ici nous nous intéressons en particulier au deuxième problème.

4 Méthodologie et outils

Dans cette section nous répondons donc à ce manque d'outils en proposant d'utiliser la méthodologie TDF (Tactics Development Framework) et l'outil associé pour faciliter la conception (y compris par des non-spécialistes) de modèles BDI de la population en situation de crise. Nous montrons comment cette méthodologie peut être adaptée de son domaine d'application initial (la capture de connaissances des experts militaires) vers notre propre domaine d'application (la modélisation du comportement de la population civile pendant une crise).

4.1 Tactics Development Framework (TDF)

La méthodologie TDF (Evertsz *et al.*, 2015) est une extension de la célèbre méthodologie Prometheus pour le développement logiciel orienté agent (Padgham & M. Winikoff, 2004), en y ajoutant la notion de tactiques : des moyens spécifiques d’accomplir une mission, incluant à la fois des comportements réactifs et délibératifs. Elle a été créée pour des applications militaires, et a fait l’objet d’une démonstration pour la modélisation de tactiques de guerre sous-marine à partir d’interviews d’experts militaires (Evertsz *et al.*, 2014). La méthodologie TDF et l’outil associé sont très utiles pour aider les non-informaticiens à créer leurs propres modèles, ou au moins à participer à leur élaboration et à faire des retours à une phase précoce du processus de modélisation. Les modèles sont élaborés en plusieurs itérations de trois phases successives, avec validation progressive entre chaque phase et chaque itération :

1. **Spécification du Système** : identification des artefacts au niveau du système, c’est-à-dire les missions, buts, scénarios, percepts, actions, données, acteurs et rôles ;
2. **Conception Architecturale** : spécification interne du système, regroupement des agents en rôles, et spécification de leurs interactions (protocoles et messages) ;
3. **Conception Détaillée** : définition interne des agents (diagrammes de plans, événements internes, messages envoyés et reçus, et données utilisées).

4.2 Adaptation de la méthodologie du domaine militaire au domaine civil

TDF a été créée pour modéliser des tactiques militaires. Ici nous voulons l’appliquer pour modéliser les tactiques de la population civile en situation de crise. Nous expliquons ci-dessous les différences entre la modélisation de tactiques militaires et civiles, et comment adapter la méthodologie en conséquence.

Notre but n’est pas d’obtenir un modèle *normatif* des meilleures tactiques afin de programmer une entité autonome (par exemple un drone), mais plutôt de fournir un modèle *descriptif* des tactiques réelles de la population, afin de créer une population simulée crédible permettant la validation des politiques de gestion de crise. Les tactiques modélisées ne sont donc pas les meilleures tactiques, mais les tactiques effectivement employées par les civils, qui pourraient être non-optimales voire dangereuses. En effet le comportement des civils n’est pas imposé par des procédures, protocoles ou tactiques définis dans un manuel, mais basé sur leurs choix individuels, et donc plus aléatoire, potentiellement inattendu, incohérent ou irrationnel.

En conséquence il est aussi plus difficile d’obtenir des informations sur leurs tactiques, car les civils peuvent n’avoir aucune idée de ce qu’ils feraient en situation de crise. C’est pourquoi au lieu de conduire des interviews (comme cela a été le cas avec les experts militaires), nous avons extrait notre connaissance des tactiques de la population civile à partir des enquêtes conduites dans le Victoria après les incendies de février 2009. Nous choisissons ici de nous appuyer sur les résultats de ces enquêtes conduites par des gestionnaires de crise bien informés sur le contexte local spécifique, et donc plus à même d’obtenir des informations de la part de la population.

5 Conception de notre modèle

La méthodologie TDF se base sur des scénarios décrivant différentes exécutions possibles, afin de susciter la **spécification interne du système**. Nous décrivons ci-dessous un tel scénario pour un incendie de forêt.

- Avertissement de risque d'incendie
- Certains habitants quittent la zone
- Un incendie démarre, alerte diffusée aux habitants de la zone
- Davantage d'habitants quittent la zone pour éviter le feu
- D'autres habitants se préparent à combattre le feu, vérifient leur équipement et rassemblent de l'aide
- Les touristes et certaines personnes restent sur place et continuent leurs activités normales ou regardent l'incendie
- L'incendie progresse vers la ville
- Les autorités envoient régulièrement des informations mises à jour sur la progression de l'incendie, la météo, la position des abris et les meilleures routes d'évacuation
- Les alertes et l'imitation des voisins qui évacuent poussent plus d'habitants à évacuer eux aussi vers les abris
- A cause des nombreuses évacuations en même temps, les routes sont encombrées
- Le feu se rapproche de la ville, les autorités envoient un ordre d'évacuation aux habitants (SMS, appel téléphonique, ou porte à porte si possible)
- Certains résidents attendaient cet ordre personnel et décident maintenant d'évacuer
- D'autres résidents qui ont décidé de défendre leur propriété refusent d'évacuer
- Les pompiers et les résidents qui sont restés combattent l'incendie
- Après un certain temps, l'incendie est contrôlé
- Les habitants sont avertis quand il est possible de rentrer chez eux

Conception architecturale : ce scénario implique différents rôles des civils, correspondant aux 7 archétypes de comportement identifiés ci-dessus, et interagissant entre eux et avec les gestionnaires de crise (pompiers, etc).

Conception détaillée : chacun de ces rôles a sa propre mission (défendre sa vie et/ou sa propriété), et donc ses propres perceptions pertinentes et actions disponibles (cf figure 2), et sa propre hiérarchie de buts (cf figure 1)

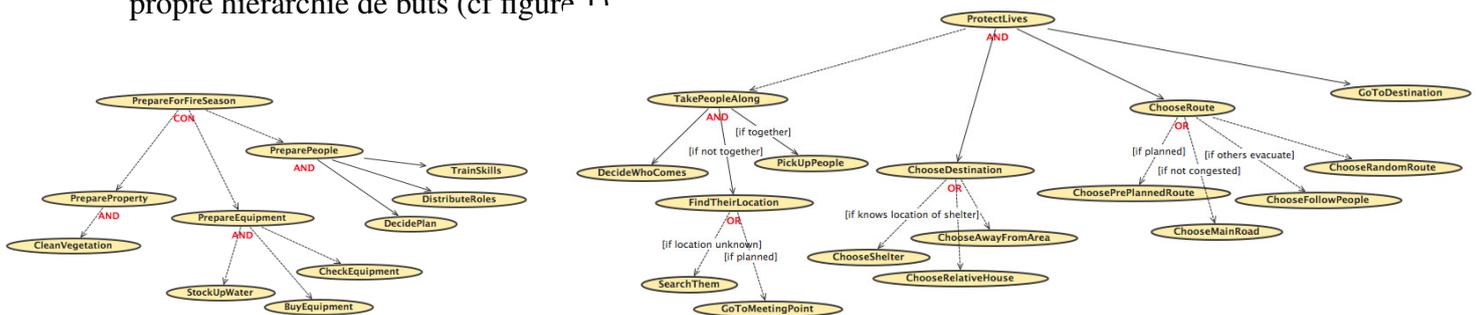


FIGURE 1 – Arborescence des buts “se préparer” (rôle défenseur) et “protéger des vies” (rôle d’éviteur).

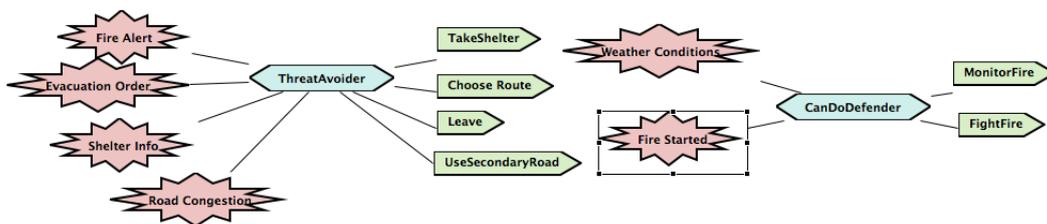


FIGURE 2 – Aperçu des perceptions (étoiles roses) et actions (flèches vertes) des rôles “éviteur de menace” et “défenseur compétent”.

Nous avons donc conçu un modèle préliminaire de comportement tactique de la population face aux incendies, en faisant un premier passage superficiel sur les 3 phases de TDF, non détaillées plus faute de place. Après validation, nous raffinerons ce modèle de manière itérative.

6 Conclusion

Cet article fournit quelques contributions et premiers pas vers une future application socialement bénéfique qui nécessite cependant encore du travail.

Modèle de la population.

Nous avons fourni un modèle BDI précis du comportement de la population du Victoria dans un incendie, en conceptualisant et désambiguïsant des profils de comportement identifiés par des enquêtes conduites par diverses agences de gestion de crise. La prochaine étape consiste à évaluer la fidélité de ce modèle à la réalité, ce qui est facilité par sa représentation graphique avec l'outil TDF, permettant d'impliquer les gestionnaires de crise et d'obtenir leurs réactions dès cette phase précoce. Nous implémenterons ensuite notre modèle dans GAMA, une plateforme de simulation libre focalisée sur la simplicité d'écriture des modèles (par des non informaticiens) et le réalisme des données géographiques sous-jacentes (Grignard. *et al.*, 2013). GAMA permet aussi de développer des simulations à large échelle. Le modèle implémenté pourra alors être évalué plus profondément en comparant ses prédictions avec les données et statistiques de comportement rassemblées par les agences après chaque incendie. Après validation, cette simulation sera exécutée sur différents scénarios afin d'identifier les meilleures stratégies pour les gestionnaires de crises, en leur offrant l'avantage de se baser sur des comportements réalistes (et validés) de la population plutôt que sur des comportements attendus.

Méthodologie.

Nous avons adapté une méthodologie conçue pour la modélisation des tactiques des experts militaires, et l'avons utilisée pour la modélisation des tactiques des civils en situation de crise. Cela prouve la plus large applicabilité de cette méthodologie, et répond au manque d'outils pour l'intégration d'agents BDI dans les simulations. Une telle méthodologie (et l'outil associé) devrait permettre aux gestionnaires de crises et autres acteurs de concevoir eux-mêmes de tels modèles, et de modifier et maintenir facilement des modèles existants. La prochaine étape consistera à évaluer cette méthodologie via une étude auprès d'utilisateurs, après avoir identifié les critères d'évaluation pertinents. Les modèles générés devront aussi être évalués en termes de compréhensibilité (est-ce que les gestionnaires peuvent répondre à des questions sur leur contenu ?) et de facilité de maintenance (les gestionnaires peuvent-ils facilement modifier le modèle, leurs changements sont-ils propagés correctement par l'outil ?).

Boîte à outils intégrée

A plus long terme, nous voulons développer une boîte à outils complète pour concevoir des modèles et implémenter des simulations sociales avec des comportements et prises de décisions complexes de la part des agents, ce qui est essentiel pour la validité de telles simulations. Nos travaux futurs vers cet objectif incluent l'extension de la capacité de génération automatique de code de TDF vers d'autres langages BDI (seul Jack est supporté pour l'instant) et l'extension d'un framework développé au RMIT pour l'intégration d'agents BDI dans des simulations (Chen *et al.*, 2014; Padgham *et al.*, 2014) afin de gérer aussi la plateforme GAMA. Ce framework permet en outre de concevoir des simulations modulaires, avec possibilité d'y brancher divers composants (MATSIM pour la gestion du trafic, Phoenix pour la simulation du feu...).

Remerciements

Le premier auteur remercie le LIG pour le financement d'une mission à Melbourne début 2015, et le RMIT pour son accueil pendant cette période. Elle remercie en particulier John Thangarajah et Rick Evertsz pour leur aide sur la méthodologie et l'outil TDF qu'ils ont développés.

Références

- ADAM C., GAUDOU B., HICKMOTT S. & SCERRI D. (2011). Agents BDI et simulations sociales. *Revue d'Intelligence Artificielle (RIA) - numéro spécial simulation multi-agents*, **25**(1).
- ADAM C., HERZIG A. & LONGIN D. (2009). A logical formalization of the occ theory of emotions. *Synthese*, **168**(2), 201–248.
- ANDERSON J. (1983). *The architecture of cognition*. Cambridge, MA : Harvard University Press.
- BALKE T. & GILBERT N. (2014). How do agents make decisions ? a survey. *Journal of Artificial Societies and Social Simulation (JASSS)*, **17**(4), 13.
- BECK E., DUGDALE J., TRUONG H. V., ADAM C. & COLBEAU-JUSTIN L. (2014). Crisis mobility of pedestrians : from survey to modelling, lessons from lebanon and argentina. In *ISCRAM-Med*, volume 196 of *LNBIP*.
- CHEN Q., WILSHER A., SINGH D. & PADGHAM L. (2014). Adding bdi agents to matsim traffic simulator (demonstration). In *AAMAS*, p. 1637–1638.
- EVERTSZ R., RITTER F. E., BUSETTA P., PEDROTTI M. & BITTNER J. L. (2008). CoJACK – Achieving Principled Behaviour Variation in a Moderated Cognitive Architecture. In *BRIMS*.
- EVERTSZ R., THANGARAJAH J., YADAV N. & LI T. (2014). Tactics development framework (demonstration). In *AAMAS*, p. 1639–1640.
- EVERTSZ R., THANGARAJAH J., YADAV N. & LI T. (2015). Agent oriented modelling of tactical decision making. In BORDINI, ELKIND, WEISS & YOLUM, Eds., *AAMAS*, Istanbul, Turkey : IFAAMAS.
- FIRE SERVICES COMMISSIONER (2013). *Review of the community response in recent bushfires*. Rapport interne, NOUS group. <http://goo.gl/wJcGn3>.
- GRIGNARD. A., TAILLANDIER P., GAUDOU B., HUYNH N., VO D.-A. & DROGOUL A. (2013). Gama v. 1.6 : Advancing the art of complex agent-based modeling and simulation. In *PRIMA*.
- MASA GROUP (2015). SYNERGY tool. http://www.masagroup.net/products/masa_synergy/. Accessed June 2015.
- MCNEILL I., DUNLOP P., SKINNER T. & MORRISON D. (2014). *Information Processing Under Stress : Community Reactions*. Rapport interne, Bushfire CRC, Australia. <http://goo.gl/TkcC4d>.
- MINH L. V., ADAM C., GAUDOU B., VINH H. T. & TAILLANDIER P. (2011). Simulation of emotion dynamics in a group of agents in an evacuation situation. In *Postproc. of PRACSYS*, volume 7057 of *LNAI*. Springer.
- PADGHAM L. & M. WINIKOFF M. (2004). *Developing intelligent agent systems : a practical guide*. Wiley.
- PADGHAM L., NAGEL K., SINGH D. & CHEN Q. (2014). Integrating BDI Agents into a MATSim Simulation. In *ECAI*, p. 681–686.
- RAO A. & GEORGEFF M. (1995). Bdi agents : From theory to practice. In *First ICMAS*, p. 312–319.
- RHODES A. (2014). *Why don't they do what we think they should ? Understanding People's Response to Natural Hazards*. Rapport interne, Emergency Management Victoria - AFAC.
- ROSENBLUM P. S., LAIRD J. E. & NEWELL A. (1993). *The SOAR papers : research on integrated intelligence*. Cambridge, MA : MIT Press.
- SCERRI D., HICKMOTT S., BOSOMWORTH K. & PADGHAM L. (2012). Using modular simulation and agent based modelling to explore emergency management scenarios. *The Australian Journal of Emergency Management*, **27**(3), 44–48.
- TEAGUE B., MCLEOD R. & PASCOE S. (2009). *Final Report*. Rapport interne, 2009 Victorian Bushfires Royal Commission. <http://goo.gl/dtXnHE>.
- TRUONG H. V., BECK E., DUGDALE J. & ADAM C. (2013). Developing a model of evacuation after an earthquake in Lebanon. In *ISCRAM Vietnam*.
- VAN RUIJVEN T. (2011). Serious games as experiments for emergency management research : A review. In *8th International ISCRAM Conference*, Lisbon, Portugal.
- WOOLDRIDGE M. (2008). *An introduction to multiagent systems*. Wiley.

Strategic Decision Making In Yacht Match Racing: Stochastic Game Approach

Lamia Belouaer¹, Matthieu Boussard², Patrick Bot¹

¹ IRENAV, Institut de Recherche de l'École navale, Brest, France
lamia.belouaer@ecole-navale.fr
patrick.bot@ecole-navale.fr

² MASA GROUP, Paris, France
matthieu.boussard@masagroup.net

Résumé :

A sailing yacht match race is a race between two yachts with rather similar performances competing against each other with a given set of racing rules, where the tactical decisions from the skippers are decisive and complex as many parameters come into play such as the wind fluctuations and the opponent's actions, both items being partly expected and partly stochastic. Strategy plays a major role and sailors continuously have to make their decisions according to the wind variations and the opponent's position and actions. This paper shows how the strategic decision-making in sailing yacht match racing corresponds to the settings of stochastic games, considering the wind fluctuations and the opponent's actions. Even if the model is simplified with strong assumptions with respect to the real game of match racing, the new proposed formalism with a two-player stochastic game approach permits some insights into strategic decision making and will be helpful for sailors training after some future developments to get closer to the real race.

Mots-clés : Stochastic Game, Zero-sum Game, Sailing Match Race

1 Introduction

To win the race, sailors must not only be skilled to sail the yacht fast in the prevailing sailing conditions, but also be able to make efficient strategic decisions. Spatial inhomogeneities of the wind may be present on the race course, and local and temporary wind fluctuations usually occur during a race which sailors have to use to optimize their route to win the race. The strategy is an important part of the race and is often predominant (Philpott *et al.*, 2004) particularly when the yachts have similar performances and sailors are good enough to sail their yacht at the optimal performance. Thus, to win a race, the naive strategy is to look for the route minimizing the time to complete the course, which is the matter of weather routing. However, when racing against an opponent in an uncertain wind, the opponent's behavior should be considered and a better strategy is to try to maximize the probability to reach the finish line before the opponent (Tagliaferri *et al.*, 2014).

This paper presents theoretically, a stochastic game model of strategic decision-making in a match racing upwind leg based on the dynamic and uncertain wind and the opponent actions. The model proposed in the present work considers two non-cooperative players. Each player corresponds to a yacht and wants to win. A new state results from the current state, the selected yachts' actions and the dynamic and uncertain wind. To solve this game is finding the winning strategy. To our knowledge there is no model for match racing based on game theory taking into account the opponent's behavior and the wind fluctuations at the same time.

2 Match Racing and Simplified Problem Addressed

A sailing yacht match race is a duel between two similar yachts under a given set of rules, mostly on an upwind-downwind course, where the aim is not necessarily to complete the course in the shortest possible time, but to reach the finish line before the opponent. The winner is the one who reaches the upwind mark first.

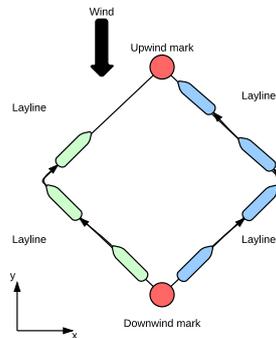


FIGURE 1 – Schematic representation of the course. The y axis is oriented in the direction of the upwind mark (initial wind direction).

For brevity's sake, only an upwind leg is considered and the race course may be represented as shown on Figure 1. We focus on the strategic decision-making and do not address the yacht performance nor the crew ability to trim and drive the yacht. Hence, we will consider identical yachts always sailing upwind at their best VMG (Velocity Made Good) which is assumed to be known. Therefore, each yacht has two possible actions : to tack or not to tack. Other strong assumptions made are that we have not considered yet the racing rules nor the effect on a yacht's performance of the proximity of the opponent. Thus, only situations where both yachts are not too close to each other are realistically modelled.

2.1 Weather routing

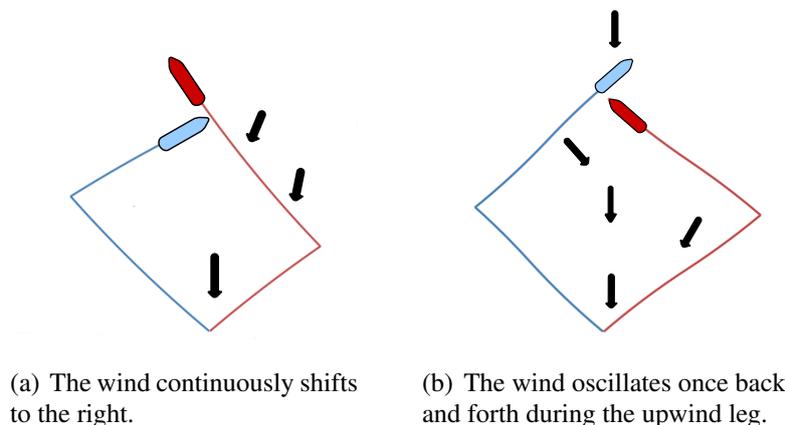


FIGURE 2 – Wind fluctuations in two basic situations.

Weather routing is finding the fastest path by considering yacht performance and the wind and current variations. In an upwind leg, a yacht cannot sail straight into the wind and has to zig-zag towards the upwind mark (Figure 1).

In real situations, the wind direction and speed vary in space and time, and the optimal route includes several tacks to take advantage of the wind shifts, provided the gain exceeds the loss due to tacking. Two typical simple situations are described to illustrate some of the basics in weather routing (Figure 2) :

1. The wind is continuously shifting to one side -say right- during the upwind leg ; the optimal route includes one tack and starts going right, as the opposite strategy -going left- leads to a longer path (Figure 2(a)).
2. The wind oscillates with firstly a shift to the right, then shifting back to its initial direction, then shifting to the left and finally shifting back to its initial direction, during the upwind leg ; the optimal route includes one tack and starts going left while the wind is shifted to the right, then tacking when the wind is shifted to the left, to always maximize the projected velocity towards the upwind mark (Figure 2(b)).

2.2 Adversarial situations

When racing against an opponent in an uncertain wind, the opponent’s behavior should be considered and the strategy should include some risk management. Yacht’s relative locations and behaviors are key elements during the race.

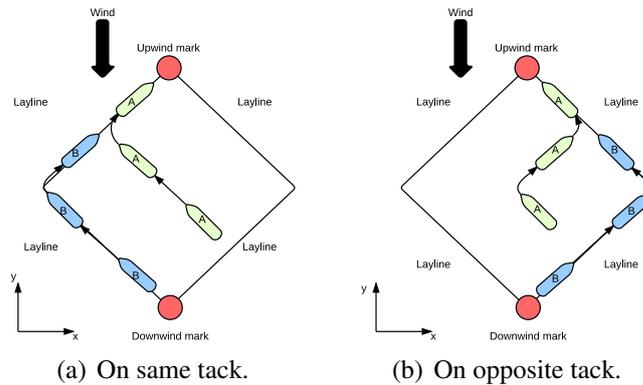


FIGURE 3 – Adversarial Situations. Yacht A behavior.

Let us consider for example yachts *A* and *B* where *A* is leading and *B* is trailing (Figure 2.2). We consider yacht *A* behavior. To achieve its goal and considering an upwind leg, yacht *A* reacts to yacht *B* behavior.

1. *B* on the same tack ; *A* continues until the layline and tacks once to limit the tacking cost (Figure 3(a)).
2. *B* on the opposite tack ; *A* tacks to minimize risk. In this case, *A* achieves one more tack but minimises the risk by a loose covering (Figure 3(b)).

Therefore, the strategy is to choose a route which maximizes the probability to reach the upwind mark before the opponent, as shown in (Tagliaferri *et al.*, 2014).

3 Match Racing as a Stochastic Game

We tackle our decision making problem using a stochastic game model. Formally, a stochastic Game (Shapley, 1953) is a tuple $\langle Ag, \{A_i : 1, \dots, n\}, \{R_i : 1, \dots, n\}, S, T \rangle$:

1. Ag is a finite set of players indexed $1, \dots, n$;
2. A_i is a finite set of actions available to player i and $\langle a_1, \dots, a_n \rangle \in \times_i A_i$ denotes a vector of the joint action of the players ;
3. $R_i : S \times \vec{A} \rightarrow \{-1, 0, 1\}$ is a reward function for the i th player ;
4. S is a finite set of states ;
5. $T : S \times \vec{A} \times S \rightarrow [0, 1]$ is a transition function. It indicates the probability of moving from a state $s \in S$ to a state $s' \in S$ by executing the joint action \vec{a} .

3.1 Players, actions and states

Actors in a game are the players whose intents are to either maximize gains or minimize losses. To model the match racing as a game, we assume that a player represents the skipper and his yacht. Thus, sailing match racing is a game with two players ($n = 2$). b_i denotes a player where $i = \{1, 2\}$. During the race, at a given speed, each player b_i has only two possible actions : **to tack** or **not to tack**. Executing a tack action forces the player to turn through the wind and slow down. We assume a known speed loss for tacking. A_i denotes the set of actions for each player b_i , $(A_i)_{i \in \{1,2\}} = \{\text{to tack, not to tack}\}$. $\vec{a} = (a_1, a_2)$ denotes the joint action of players b_1 and b_2 ($\vec{a} \in \{\vec{a}_{(t,t)}=(\text{tack, tack}), \vec{a}_{(t,n)}=(\text{tack, not to tack}), \vec{a}_{(n,t)}=(\text{not to tack, tack}), \vec{a}_{(n,n)}=(\text{not to tack, not to tack})\}$).

Our game consists in a finite set of states $S = \{s^0, s^1, \dots, s^h\}$. Each state corresponds to the yachts situations and the wind direction and speed. Formally, a state s is a tuple $s = \langle Y_1, Y_2, W \rangle$:

1. Yacht situation (Y_i) corresponds to the yacht (b_i) internal state. Yacht situation describes location, direction and speed of a given yacht. Formally, $Y_{b_i} = \langle pos, \vec{v} \rangle$, where ; (1) $pos = \langle x, y \rangle$ denotes the current location in the race and (2) $\vec{v} = \langle \theta, v \rangle$ denotes the current yacht direction and speed.
2. Wind direction and speed, $W \in \{-45^\circ, -40^\circ, \dots, 0^\circ, \dots, 45^\circ\} \times \mathbb{R}^+$, where 0° represents the direction to the upwind mark and the other states represent shifts of $\pm 5^\circ$ from that direction (Tagliaferri *et al.*, 2014).

3.2 Transition

The transition function captures the stochastic nature of the yacht's motion. Our game transits from one state to another according to a probability distribution. The state transition probability is a function of both the players' actions, and the current state. Formally, the transition function $T(s^k, \vec{a}, s^{k+1}) = Pr(s^{k+1} | s^k, \vec{a})$ is the probability that the system transits to state s^{k+1} when the joint action \vec{a} is taken at state s^k . The transition function is based on the predicted yacht positions. The transition function $T(s^k, \vec{a}, s^{k+1}) = Pr(shift)$ corresponds to the probability that the wind shifts. Therefore, we focus on changes in wind direction that affect the race. We adopted the wind transition defined in (Tagliaferri *et al.*, 2014).

3.3 Reward

During the race the challenge for yacht b_1 is to stay ahead the opponent b_2 . We define this advantage as the estimated time to reach the upwind mark from the current location. During the race, when a yacht b_1 makes a decision to tack or not to tack, it evaluates the remaining time to reach the upwind mark. Formally, we define, in the same way for the two yachts, a function denoted $\Gamma_1(s)$ estimating the time to reach the upwind mark (UM). This function depends on the current yacht location relative to the upwind mark ($dist(pos^{UM}, Y_i.pos)$), where ; (1) Y_i denotes the current state (s) of yacht b_i and (2) $pos^{UM} = (x^{UM}, y^{UM})$ denotes the location of the upwind mark. At each state s , according to $\Gamma_i(s)$, there are three cases : (1) yacht b_1 takes less time than the opponent b_2 to reach the upwind mark. Yacht b_1 is the leader, (2) yacht b_1 takes more time than the opponent b_2 to reach the upwind mark. Yacht b_1 is the follower and (3) both players are to a tie. We define, in the same way for both yachts, the pay-off function $p_i(s)$ for each state :

$$p_1(s) = \begin{cases} 1 \Leftrightarrow \Gamma_1(s) < \Gamma_2(s) & \text{if } b_1 \text{ is the leader} \\ -1 \Leftrightarrow \Gamma_1(s) > \Gamma_2(s) & \text{if } b_1 \text{ is the follower} \\ 0 \Leftrightarrow \Gamma_1(s) = \Gamma_2(s) & \text{otherwise.} \end{cases}$$

To win the race, a yacht b_i must reach the upwind mark first. We set the reward for the goal state to a positive or negative value. The reward function R_i is defined in the same way for both yachts :

$$R_i = \begin{cases} 1 & \text{if } b_i \text{ wins} \\ -1 & \text{if } b_i \text{ loses} \\ 0 & \text{otherwise.} \end{cases}$$

3.4 Decision model

At each step k , we formulate the state s^k as a two-dimensional matrix game G^k (at each state the player is faced to a game). A matrix game G^k indicates pay-offs for b_1 and b_2 . We allow one player b_1 to choose the rows and the other b_2 the columns. The entries in the matrix are the pay-offs to the row player. An important property for this game is $p_1(s) = -p_2(s)$ which means that a pay-off for one player is a cost for the other. At each state s^k , each player b_i has a finite set of actions at his disposal to choose from $(A_i)_{i \in \{1,2\}} = \{\text{to tack, not to tack}\}$. The resulting bi-matrix therefore contains the matrix game for both players.

$$G^k = \begin{array}{cc} & \begin{array}{cc} \text{to tack} & \text{not to tack} \end{array} \\ \begin{array}{c} \text{to tack} \\ \text{not to tack} \end{array} & \left(\begin{array}{cc} p_1(s^k) & , & p_1(s^k) \\ p_1(s^k) & , & p_1(s^k) \end{array} \right) \end{array}$$

Different states can lead to the same game G . Let this set be S_G . When acting, two players transit from one state s^k to another state s^{k+1} leading to a transition from one game to another. Transitions from game G^k to another game G^{k+1} depend on the outcome of G^k and the joint action given by the matrix game G^k . The probability of transiting from game G^k when acting with a joint action \vec{a} is the probability to transit from one state of S_{G^k} to a state of $S_{G^{k+1}}$. Players simultaneously choose a row and a column of the matrix game causing player b_1 to win the pay-off $p_1(s^k)$ from player b_2 who loses the same amount. Then, the game moves to another state,

with a probability that depends on the selected joint action and the current state. Formally, for each state s^k ; $Pr(G^k, \vec{a}, G^{k+1}) = \sum_{s^{k+1} \in S_{G^{k+1}}} T(s^k, \vec{a}, s^{k+1})$.

For each matrix game, the game value is the unique solution of G^k with game values given as $V(k)$. However, for a long-term race, the strategy should maximize the gain expectation. To this end, we propose a value function for each game G^k at each state s^k , as follows ;

$$V(k) = \max_{\vec{a}} \begin{cases} \sum_{s^k, G^{k+1}} Pr(G^k, \vec{a}, G^{k+1}) V(k+1) & k < h \\ R_i & k = h. \end{cases}$$

$Pr(G^k, \vec{a}, G^{k+1})$ denotes the probability of transiting from game G^k when acting with a joint action \vec{a} . Solving this problem we can use a value iteration algorithm (Littman, 1994). There is a mixed strategy for player b_1 such that b_1 's average gain is at least V no matter what b_2 does and there is a mixed strategy for player b_2 such that b_2 's average loss is at most V no matter what b_1 does. Also, if $V = 0$, the game is neutral. If $V > 0$ the game is said to be favorable to player b_1 , otherwise if $V < 0$ the game favors player b_2 . The row player is trying to match the column player and the column player is trying to guess the opposite of the row player. The value of the game may be calculated as either the minimum of what the row player can achieve knowing the strategy of the column player or the maximum of what the column player can hold the row player to, knowing the strategy of the row player.

3.5 Sailing yacht match racing example

We consider two yachts b_1 and b_2 . Both yachts have exactly the same distance to reach the upwind mark. b_1 is sailing on starboard tack and is located to the left of b_2 sailing on port tack. The yachts speed is $v = 13kt$ and the speed loss when tacking is $3kt$. The wind speed is constant. We focus on a finite horizon h ($h = 10$, $\delta t = 30s$), where the wind shifts to the left ($shift = -5^\circ$) twice, at times $k = 4$ and $k = 8$. We focus on b_1 behavior. We assume the following initial state :

$$s^0 = \begin{cases} Y_{b_1} = \langle 15, 130, -45^\circ, 13 \rangle \\ Y_{b_2} = \langle 25, 130, +45^\circ, 13 \rangle \\ W = \langle 0^\circ, windspeed \rangle. \end{cases}$$

At step $k = 1$, the wind direction remains the same ($W^1 = 0^\circ$). There are 4 possible situations and 1 possible matrix game (G_1^1) according to joint actions $\vec{a}_{(t,t)}$, $\vec{a}_{(t,n)}$, $\vec{a}_{(n,t)}$ and $\vec{a}_{(n,n)}$:

$$G_1^1 = \begin{array}{cc} & \begin{array}{cc} \text{to tack} & \text{not to tack} \end{array} \\ \begin{array}{c} \text{to tack} \\ \text{not to tack} \end{array} & \left(\begin{array}{cc} 0 & -1 \\ 1 & 0 \end{array} \right) \end{array}$$

1. if both yachts b_1 and b_2 tack ($\vec{a}_{(t,t)}$), there is no leader at step $k = 1$;
2. if yacht b_1 tacks and yacht b_2 doesn't tack ($\vec{a}_{(t,n)}$), b_2 is the leader at step $k = 1$;
3. if yacht b_1 doesn't tack and yacht b_2 tacks ($\vec{a}_{(n,t)}$), b_1 is the leader at step $k = 1$;
4. if yachts b_1 and b_2 do not tack ($\vec{a}_{(n,n)}$), there is no leader at step $k = 1$;

For each time step k , we repeat the same operations in order to compute states in the next step $k + 1$ according to joint actions and current state, considering possible wind shifts. This allows the matrices games to be determined. This process provides a decision tree. A winning strategy for yacht b_1 consists in finding a path in the global decision tree from s^0 to s^h . Formally, computing a winning strategy consists in computing recursively $V(k)$ up to the final state. The computed value assigns to each state the maximum expected value expressing the probability to win and considers for this state the action maximizing this probability.

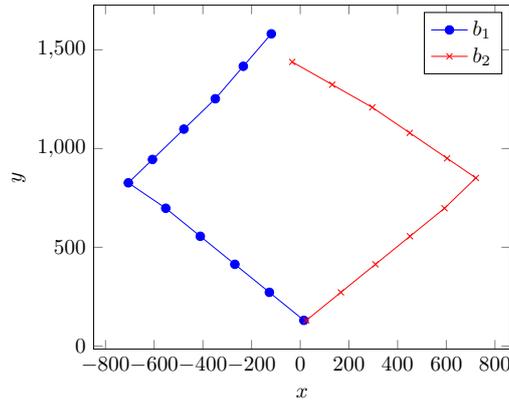


FIGURE 4 – One winning strategy for yacht b_1 .

During the first part of the race, at steps from $k = 0$ up to $k = 3$ with the initial wind ($W = 0^\circ$), both yachts b_1 and b_2 do not tack. Thus, there is no leader in time step $k = 4$. In step $k = 4$ the wind shifts to the left ($shift = -5^\circ$). Again, both yachts b_1 and b_2 do not tack. Yacht b_2 becomes the leader as his course is closer to the direction of the upwind mark. At time step $k = 5$, the wind is to the left ($W = -5^\circ$), both yachts b_1 and b_2 tack. Yacht b_2 remains the leader at step $k = 6$ but the gain towards the mark is higher for b_1 . From this step, yacht b_1 takes the lead. Indeed, the wind direction is more favorable to b_1 than to b_2 . This is reflected into the following matrices games. During the first part of the race, at steps from $k = 0$ up to $k = 4$, it is the same matrix game, such as :

$$G_1^1 = G_1^2 = G_1^3 = G_1^4 = \begin{matrix} & \text{to tack} & \text{not to tack} \\ \text{to tack} & \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \\ \text{not to tack} & \end{matrix}$$

According to these matrices games and Figure 4, both yachts b_1 and b_2 have decided not tack. At time step $k = 5$ and time step $k = 6$, the wind is to the left ($W = -5^\circ$), For any considered joint action, yacht b_2 is the leader. This is reflected into the following matrices games, such as :

$$G_1^5 = G_1^6 = \begin{matrix} & \text{to tack} & \text{not to tack} \\ \text{to tack} & \begin{pmatrix} -1 & -1 \\ -1 & -1 \end{pmatrix} \\ \text{not to tack} & \end{matrix}$$

At step $k = 7$, as long as yacht b_1 does not tack, he is the leader. This is reflected into the following matrices games, such as :

$$G_1^7 = \begin{array}{cc} & \begin{array}{cc} \text{to tack} & \text{not to tack} \end{array} \\ \begin{array}{c} \text{to tack} \\ \text{not to tack} \end{array} & \left(\begin{array}{cc} 0 & -1 \\ 1 & 1 \end{array} \right) \end{array}$$

At steps from $k = 8$ up to $k = 10$, for any considered joint action, yacht b_1 remains the leader. This is reflected into the following matrices games, such as :

$$G_1^8 = G_1^9 = G_1^{10} = \begin{array}{cc} & \begin{array}{cc} \text{to tack} & \text{not to tack} \end{array} \\ \begin{array}{c} \text{to tack} \\ \text{not to tack} \end{array} & \left(\begin{array}{cc} 1 & 1 \\ 1 & 1 \end{array} \right) \end{array}$$

The outcomes from the model on this simple scenario are consistent with the well-known strategy basics that if the wind shifts to one side during an upwind leg, the yacht positioned to that side takes an advantage, as seen in Section 2.

4 Conclusions and Future Work

This work introduces a theoretical model of stochastic game between two non-cooperative players is proposed and coupled to the stochastic wind MDP model developed in (Tagliaferri *et al.*, 2014), to account for both wind fluctuations and yachts' reactions to their opponent position and actions. The method is applied to a simple race scenario and shows results in agreement with the basic sailing strategy knowledge.

Strong assumptions to simplify the problem have been made, and further developments are needed to extend the practical applicability to the real match racing game. The main considered improvements are to enhance the pay-off and transition functions according to : (1) Modification of each yacht performance due to the vicinity of the other yacht altering the wind, (2) Modeling the racing rules and yachts' reactions to obey the rules or penalize them otherwise, ...

Nevertheless, in this highly complex game, this work proposes a new formalism to model a stochastic game to investigate the decision making with respect to the opponent's position and actions which should help integrate this issue in more complete models of match racing strategy. Moreover, it is believed that the simplified model proposed in this work can already give some insight into match racing strategy when yachts are far enough from each other in order that racing rules and wind shadow do not come into play.

Références

- LITTMAN M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the eleventh international conference on machine learning*, volume 157, p. 157–163.
- PHILPOTT A. B., HENDERSON S. & TEIRNEY D. (2004). A simulation model for predicting yacht match race outcomes. *Operations Research*, **52**(1), 1–16.
- RICHARDS P., AUBIN N. & LE PELLEY D. (2014). Wind tunnel investigation of the interaction between two sailing yachts. *23rd International HISWA Symposium*.
- RICHARDS P., LE PELLEY D., JOWETT D., LITTLE J. & DETLEFSEN O. (2012). Wind tunnel investigation of the interaction between two sailing yachts.
- SHAPLEY L. S. (1953). Stochastic games. *Proceedings of the National Academy of Sciences of the United States of America*, **39**(10), 1095.
- TAGLIAFERRI F., PHILPOTT A., VIOLA I. & FLAY R. (2014). On risk attitude and optimal yacht racing tactics. *Ocean Engineering*, **90**, 149–154.

Évolution dans l'industrie du métier d'ingénieur cognitif ou d'ingénieur de la connaissance entre 1985 et 2015

Alain BERGER¹

ARDANS SAS
6 rue Jean Pierre Timbaud, "Le Campus" Bâtiment B1,
78180 Montigny-le-Bretonneux, France
aberger@ardans.fr
www.ardans.fr

Résumé : Les premiers ingénieurs de la connaissance ou ingénieurs cognitifs sont apparus dans l'industrie avec l'arrivée des systèmes experts dans le milieu des années 1980. Les mutations des technologies, les évolutions du rôle de l'informatique dans les organisations, la confusion issue de la vulgarisation du "knowledge management" font que ce métier d'ingénieur de la connaissance reste méconnu et ce malgré toute sa valeur ajoutée. L'article constitue un regard sur l'évolution de ce métier empreint d'humilité et d'humanité dans la mutation de l'environnement industriel de ces trente dernières années en France.

Mots-clés : Ingénieur Cognitif, Ingénieur de la Connaissance, Système à base de connaissance, *Knowledge Management*, Pérennisation d'expertise, Évolution du métier de l'ingénieur dans l'industrie.

1 Il était une fois...

Il était une fois une discipline enseignée à l'université de Paris qui s'appelait l'INTELLIGENCE ARTIFICIELLE (ou IA) et que j'ai eu le bonheur de découvrir dès l'initiation en premier cycle avec le Professeur Jean-Louis Laurière. À la fin de l'année de DEA¹, j'ai opté pour la voie de l'industrie, quand d'autres restaient dans des laboratoires de l'université ou entraient dans des services IA de grandes entreprises privées ou publiques.

À cette période, l'informatique était une réelle épopée. Il y a ceux qui rejoignaient cette filière pour l'aspect lucratif ou simplement alimentaire, et d'autres pour les horizons nouveaux et lointains plus merveilleux les uns que les autres. Quand le monde de la gestion ressemblait à un terminal à caractères de type VT100 comprenant 40 lignes sur 80 colonnes, je choisis le monde des machines LISP (Dandelion de Xerox, Symbolics ou TI Explorer) dotées de souris à plusieurs boutons et d'écrans avec des résolutions dignes du XXI^{ème} siècle !

Les acteurs industriels étaient nombreux et actifs (Aérospatiale, Air France, Banque de France, BSN, Bull, CEA, CNES, CNET, Compagnie Générale d'Electricité, Avions Marcel Dassault, DGA, Electricité de France, Elf, Framatome, Gaz de France, Philips, PSA Peugeot Citroën, Renault, SNCF, Sollac, Thomson, etc.) et les premières sociétés de services en ingénierie spécialistes IA éclosaient (Cognitech, Framentec, ITMI, CRIL, Ingenia, Tecsi...) ainsi que des premiers éditeurs (Cediag, Ilog, Cosytec, Euro Decision, O2 Technology, Prologia...) voire constructeur (AMAIA).

1. Le DEA pour Diplôme d'Études Approfondies était la cinquième année après le baccalauréat qui suivait la maîtrise et celle qui initiait à la recherche pour aller jusqu'à la thèse.

Les premières méthodes jaillissaient de la recherche comme de l'industrie : KOD (Vogel, 1988), SAGACE (J.M, 1990), KADS (Schreiber *et al.*, 1993), REX (Malvache, 1993), CORPUS (Grundstein, 1996), MKSM (Ermine *et al.*, 1996), MEREX (Corbel, 1997), GAMES (Grundstein, 2000), ...

Quel foisonnement de talents, avec des parcours de formation très variables : des mathématiques, à l'algorithmique, de l'architecture à la médecine, dans des écoles d'ingénieurs, des laboratoires d'université en France comme aux Etats Unis... un réel brassage d'humains curieux à l'idée de valoriser les savoirs et savoir-faire détenus par les « experts ». Pour autant, après l'enthousiasme des années 1980-1995, le lancement d'opération de systèmes expert dans l'industrie chutait drastiquement. D'autres technologies dite « Collecticiel » puis basées sur internet allaient prendre le relais.

1.1 Cognitif vs Ingénieur de la connaissance

Les mots « Cognitif » et « Ingénieur de la Connaissance » apparaissent respectivement avec la société Cognitech en 1983 et avec la société Framentec en 1984. L'histoire retiendra que « Cognitif » est même déposé à l'INPI par Jean Michel Truong (Fondateur de Cognitech) quand la traduction littérale de « *Knowledge Engineer* » est retenue par la seconde.

1.1.1 Cognitif

Pour être très précis, Jean Michel Truong² propose les définitions suivantes :

- **Cognitif** : discipline scientifique et pratique technique, branche de l'informatique, qui a pour objet l'acquisition et la représentation formelle des connaissances et des modes de raisonnement, en vue de leur simulation à l'aide d'ordinateurs.
- **Cognitif** : spécialiste de la cognitive.

La conséquence d'une telle définition est que certains observateurs ont considéré que le « Cognitif » était une sorte de guru particulièrement habile et talentueux dans la mise en forme des règles dans le langage des moteurs d'inférence des systèmes experts.

1.1.2 L'ingénierie de la connaissance : l'outil de la mémoire collective

1.1.2.1 La genèse de cette définition

J'ai encore ce souvenir d'échanges avec Michel Grundstein (Grundstein *et al.*, 1988) dans les années 88-90, où il pointait une différence entre le cognitif qui déploie une approche influencée par l'« outil » quand l'ingénieur de la connaissance s'intéressait d'abord à « l'expert en situation de travail ».

Marqué par cette vision, il m'a paru évident qu'Ardans pratique ce métier de l'ingénierie de la connaissance où *in fine* la technique n'est là que pour révéler l'humain et l'aider à accoucher, à formaliser ce qu'il porte implicitement ou inconsciemment dans sa pratique.

Ardans pratique l'ingénierie de la connaissance principalement dans les classes de situations organisationnelles suivantes : une expertise individuelle, une expertise d'un service ou des ser-

2. www.jean-michel-truong.com/systemes_experts/page/introuvable.html

vices d'un département d'une entreprise, une expertise distribuée dans un réseau d'entités qui appartiennent à différentes organisations.

1.1.2.2 Ingénierie de la connaissance

On définit (Berger & Cotton, 2011) par **ingénierie de la connaissance** (IC), ce métier de l'Intelligence Artificielle qui est né avec la technologie des systèmes experts dans les années 1970, lors de la rédaction des règles et des premières bases de connaissances "déclaratives". Cette discipline aujourd'hui couvre tout un cycle depuis l'émergence d'un élément de connaissance, sa détection, sa structuration, son mûrissement en termes de contenu, son expression (via une définition claire, non ambiguë, la rédaction étant appuyée par des illustrations si nécessaire), son applicabilité (en termes de domaine d'usage, de droit à en connaître en termes de publication ou de d'habilitation, de durée de vie ou de péremption), et bien sûr de validation (appréciation d'expert, justification, degré de preuve). Toutes les étapes de cette maturation sont liés à une interaction, une discussion à la fois libre et orientée entre l'« expert » et l'« ingénieur de la connaissance ». Par le fruit de l'écoute, de la reformulation, la parole de l'expert se déclinera à des fins d'illustration, de justification, de relation, en formule, en texte, en schéma, en renvoi vers une note, un document, un résultat...

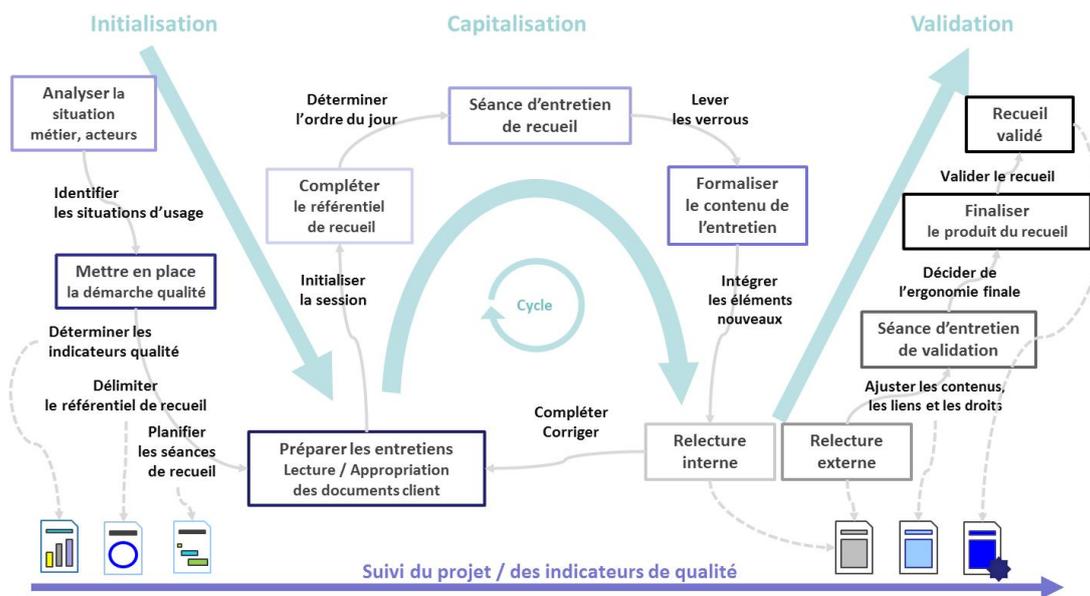


FIGURE 1 – Le cycle en Ω de capitalisation des connaissances.

Le recueil d'expertise correspond à tout un processus, tel que décrit par le cycle en Ω (cf. figure 1 (Coppens *et al.*, 2006; Besson & Berger, 2015)), qui est maîtrisé par l'ingénieur de la connaissance.

Cette transformation de l'oral vers l'écrit se traduit en numérique bien évidemment.

Lorsque l'on travaille sur la **mémoire collective** d'un domaine métier, il convient aussi d'orchestrer les différents modèles qui porteront les éléments de connaissances, de poser les liens

entre les éléments en relation, et de classer ces éléments par rapport à des concepts ou des ontologies intelligibles, distinctes, complémentaires et non contestables.

1.2 Les caractéristiques de l'Ingénieur de la connaissance

A quoi ressemble aujourd'hui ce type de profil que l'on nomme « ingénieur de la connaissance » ? Comme la question est fréquente et comme il n'existe toujours pas de cursus académique indiscutable, il est intéressant d'apprécier cette proposition en sept facettes :

1. **Habilitation** : cette caractéristique qui paraît subjective est aussi une clé de l'industrie. L'ingénieur de la connaissance va avoir à disposition des informations sensibles, singulières, qui sont un élément de performance de l'organisation. L'ingénieur de la connaissance doit être un individu avec des dimensions « éthique », de probité, de rigueur, de respect, des règles implicites et explicites de l'organisation. L'habilitation est une dimension extrêmement prégnante dans la période que nous vivons.
2. **Scientifique** : les ingénieurs de la connaissance sont principalement des scientifiques. Il est bien clair que le métier s'applique d'abord pour les métiers de l'industrie, de la science, ou de la finance. Il est certain que l'ingénieur de la connaissance doit disposer de talents épistolaires et avoir l'aptitude à rédiger de manière claire, non ambiguë et synthétique.
3. **Techniques d'entretien** : la discussion, le dialogue ont pour but, par les reformulations successives, d'aider l'expert à expliciter ses savoirs, ses expériences, ses convictions... L'art qui consiste à obtenir de l'expert à trois reprises, et sans le froisser, le « je ne sais pas » qui assure à l'ingénieur de la connaissance d'avoir atteint une limite à son instruction, est une discipline qui nécessite de la formation et de la pratique (Berger, 2015; Delmas, 2015). Cette technique est entraînée par un moteur qui se nomme « curiosité³ ». Comment expliciter et révéler de la connaissance sans grande curiosité, sans faim de connaître ?
4. **Techniques de rédaction** : la communication avec l'expert doit se traduire fidèlement dans la restitution pour un futur consommateur de cette connaissance formalisée. L'écriture doit donc être fidèle, précise et exhaustive pour être validée par l'expert, et intelligible, non ambiguë et argumentée pour être appréciée par son destinataire. Si l'« écrivain public » rédige pour le compte de celui qui ne sait pas écrire, l'ingénieur de la connaissance a le devoir de comprendre les fondements de la position de l'expert afin de pouvoir restituer l'essence de son savoir le plus fidèlement possible. Il ne s'agit pas forcément d'une simple rédaction, mais plutôt d'une reformulation dont la valeur ajoutée est liée aux trois caractéristiques majeures que sont, la fidélité à la vision de l'expert, la consistance par rapport au patrimoine de connaissance nourri par les nouveaux éléments de connaissance, la clarté dans la restitution du message cognitif délivrée au futur lecteur.
5. **Techniques de modélisation** : Comment remettre en ordre toute une avalanche de phrases afin de proposer des restitutions qui permettent à un candide de trouver la bonne réponse

3. Merci à Jean-François Gelin qui, dans notre échange, argumentait son propos par cette citation (Gelb, 2004) de Léonard de Vinci (1452-1519) « toutes nos connaissances ont pour origine notre perception ».

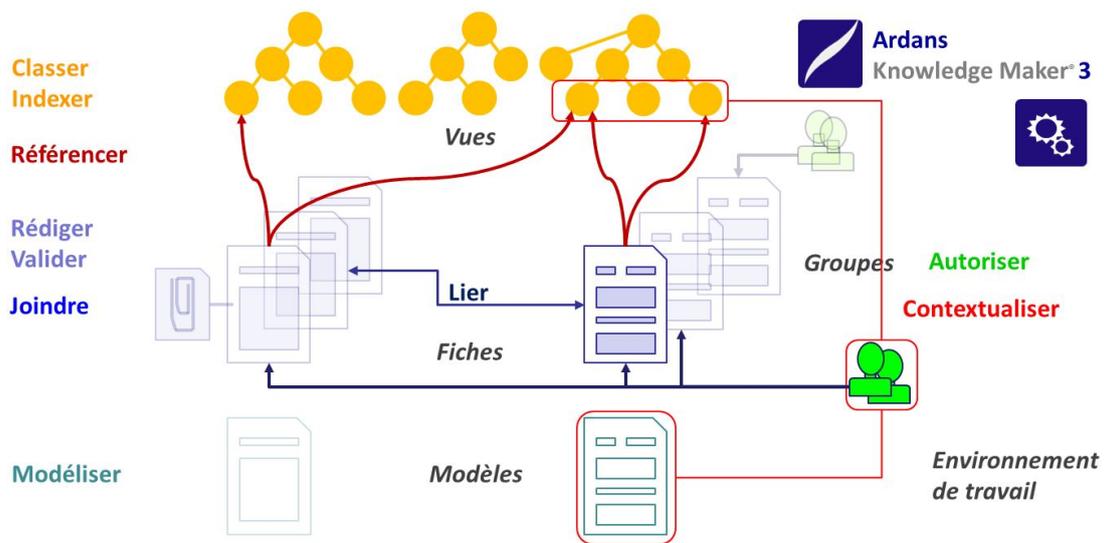


FIGURE 2 – Éléments de modélisation pour un IC dans l'industrie avec Ardans®.

à son interrogation... en trois clics ? Là est le défi. Entre les systèmes experts des années 1975-1995 et les plates-formes d'ingénierie de la connaissance d'aujourd'hui (cf. figure 2), les outils de modélisations et de structuration ont évolué (Mariot *et al.*, 2007; Besson & Berger, 2015) et les résultats de la formalisation ne sont pas les mêmes.

6. **Informatique** : Ces techniques de modélisation vont jusqu'à avoir une appétence dans certains outils informatiques pour explorer voire introspecter les bases de connaissances (Vexler *et al.*, 2013a) afin d'en apprécier leur consistance et leur qualité. Il y a aussi la maîtrise des outils de valorisation de la connaissance recueillie (animations, schémas, photographie, vidéos...) et l'interopérabilité entre la base de connaissance et d'autres applications de l'environnement de travail de l'utilisateur ciblé pour consommer l'expertise colligée (outils de CAO, de calcul, de gestion documentaire ou métier...). Les travaux du W3C sur le web sémantique sont aussi des pistes avec de nouveaux outils potentiels à explorer⁴.
7. **Humain** : Si l'ingénieur de la connaissance travaille sur l'expérience et le savoir technique de l'expert, c'est sa compétence humaine qui est particulièrement impressionnante. Sans connaissance sur le domaine *a priori*, l'ingénieur de la connaissance apprend à partir des éléments mis à disposition par l'expert. Il s'imprègne des documents principaux fournis par l'expert puis va préparer des entretiens où il confronte avec l'expert ses perceptions. C'est en établissant un dialogue avec l'expert basé sur des reformulations que le binôme accouche de l'expertise. Ensemble, ils vont structurer, formaliser, qualifier et valider les contenus qui seront consignés dans les éléments de connaissance qui constitueront la fourniture... Les relations dans ce binôme évoluent au fur et à mesure que le

4. Concernant les spécifications du W3C dans le cadre du Web sémantique, on note l'émergence d'interfaces linguistiques (langages contrôlés comme CNL(Kuhn, 2014), ACE (Fuchs & Kaljurand, 2008), ...) destinés à la reformulation des concepts et règles logiques (avec *modus ponens*, modalités, quantification et multitude d'opérateurs) en référence à des modules ontologiques.

travail avance. La relation humble et respectueuse qui prévaut initialement évolue vers une complicité quand le travail en commun révèle et rassemble des éléments précieux de l'expérience de l'expert. Cette dernière aptitude qui relève du savoir-être, porte tant le besoin de curiosité, que l'humilité de celui qui va questionner.

1.3 Et l'expert dans le dispositif ?

L'ingénieur de la connaissance pratique son art conjointement avec l'expert qui au départ de l'opération est naturellement dubitatif : « pensez-vous vraiment récupérer en quelques heures une vie de travail ? » A vrai dire, ce n'est pas une vie de travail qui est le but de ces prestations. Ce sont plutôt les bonnes réponses aux questions pointues d'aujourd'hui, les retours d'expérience de projets majeurs et aussi la vision de demain qui sont attendues. L'expert sera surpris de constater qu'*in fine* il en saura plus sur ce qu'il sait déjà. Il sera ravi de découvrir le résultat de la reformulation qu'il aura validé et qui lui aurait pris dix fois plus de temps s'il l'avait fait seul. Sans compter le plaisir d'être à l'origine d'un contenu bien rédigé, joliment illustré, aisément manipulable et très pratique pour retrouver des bonnes réponses ! L'expert doit être rassuré, il ne sera pas dépossédé de son expertise : elle lui sera reconnue !

2 Et maintenant...

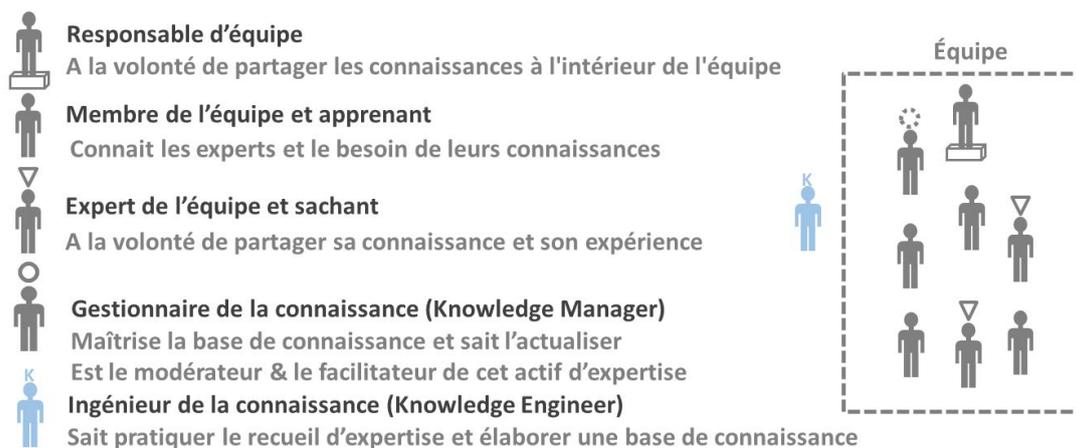


FIGURE 3 – Les parties prenantes d'un système de pérennisation d'expertise dans l'industrie.

2.1 Les schémas actuels d'intervention

Si les technologies informatiques ont particulièrement évolué dans le dernier quart de siècle⁵, force est de constater que les organisations ont été impactées dans cette période par le transfert

5. J'invite le lecteur à découvrir la vision de l'évolution de l'ingénierie de la connaissance sur cette période avec le regard avisé et académique de Nathalie Aussenac (Aussenac-Gilles, 2014) et fr.slideshare.net/aussenac/aussenac-confinviteic2014-histoire-ic25ans.

de pouvoir marqué par une toute puissance de la direction financière au détriment de la direction technique. La conséquence de ce constat est que le temps s'égrénant, les « anciens » quittent l'entreprise et que leurs savoirs et savoir-faire n'ont pas été transmis, où s'ils l'ont été, rien ne garantit que leurs successeurs n'aient eu le temps d'en maîtriser toutes les dimensions. Heureusement, il existe des organisations où l'horizon ne se limite pas au *quarter* et où la pérennisation de la connaissance est appréciée comme un actif moins onéreux à faire vivre qu'à avoir à recréer. Il est implicite que le mot « connaissance » employé ici, correspond à une expertise qui n'existe pas, de près ou de loin, dans l'univers accessible par le commun des mortels disposant d'une liaison internet. Dans le monde de l'industrie voire de la défense, le droit à en connaître est une notion qui précise bien le caractère précieux du savoir ; on est alors assez loin des concepts de démarches de production participative présentées comme vertueuses de « *crowd sourcing* » ou de « *crowdcrafting* ».

Pour revenir au rôle de l'ingénieur de la connaissance dans l'industrie, on s'appuiera sur les deux figures 3 et 4 qui présentent les acteurs associés à une opération classique de mise en place d'un système de pérennisation d'expertise dans l'industrie et son déroulement.

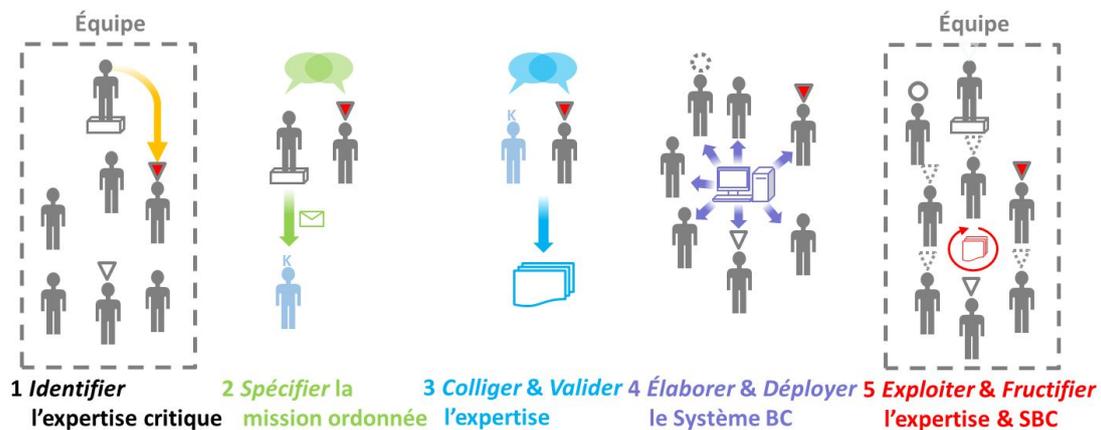


FIGURE 4 – Synopsis d'implantation industrielle d'un système de pérennisation d'expertise.

L'ingénieur de la connaissance est un acteur externe à l'équipe où travaille l'expert. Il intervient à partir de l'ordre de mission jusqu'à la livraison du dispositif à cette équipe. On observe que l'ingénieur de la connaissance s'immerge dans le « métier » au départ, puis apprécie l'équipe et ses membres qui auront à s'approprier le contenu puis à le faire fructifier dans le temps. L'analyse de la situation d'usage futur de cette consommation de connaissance est aussi un point que l'ingénieur de la connaissance doit anticiper et intégrer dans sa mission.

2.2 Les cartographies

Avant de conclure sur ce regard lié au métier de l'ingénierie de la connaissance, nous observons l'intérêt de nos clients à mesurer la qualité des bases d'expertise et à mieux appréhender les facettes qui composent l'expertise (pour des besoins de qualification de compétences requises pour intégrer une équipe par exemple) voire les trous de connaissance (ou « a-connaissance »

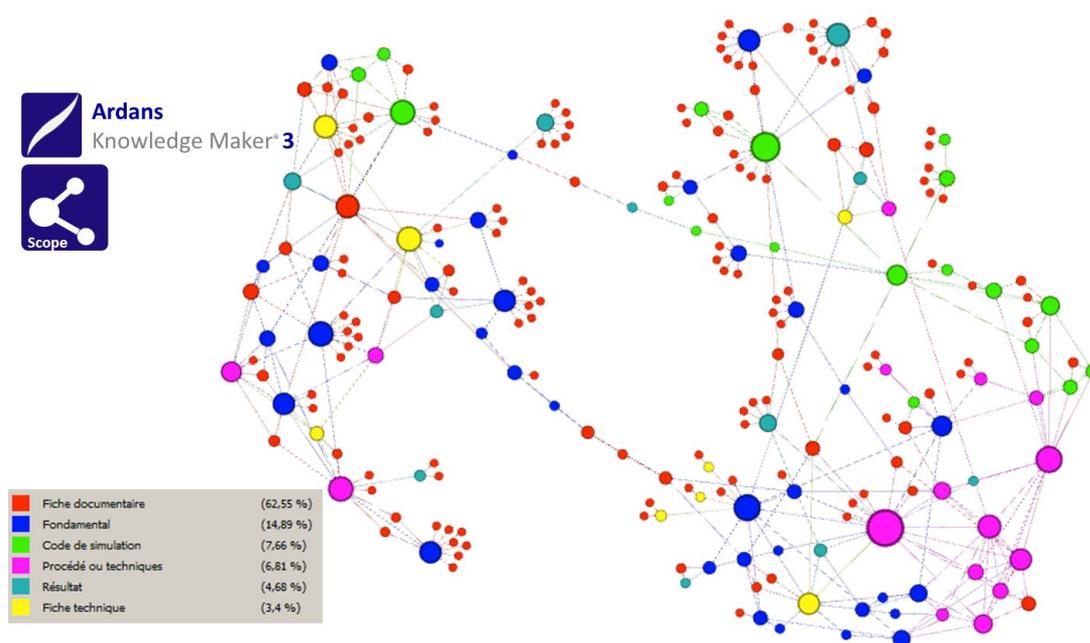


FIGURE 5 – Distribution des éléments de connaissances, leurs liens et leur nature de modèle.

(Vexler *et al.*, 2013b)). Ces cartographies sont une restitution riche et aussi un outil précieux pour l'IC.

Voici un exemple basé sur un projet réel d'opération de pérennisation d'expertise dans le prolongement des travaux présentés par François Vexler (Vexler *et al.*, 2013a,b). A des fins didactiques les trois cartes présentées sont superposables et portent des filtres complémentaires sur ce même corpus d'expertise pérennisé. Dans la cartographie présentée dans la figure 5, chaque cercle représente un élément de connaissance capitalisé. La couleur est liée à la nature du modèle qui a été utilisé pour recueillir l'expertise. Comme pour MATEO (Mathieu *et al.*, 2014), dans cette capitalisation, 6 modèles ont été élaborés (Fiche documentaire, Fondamental, Code de simulation, Procédé ou technique, Résultat, Fiche technique). La taille du cercle est liée au nombre de connexion qui arrivent à l'élément de connaissance considéré. L'absence d'élément isolé et la nature de la distribution permettent d'apprécier la qualité de la base.

Ici les deux parties gauche et droite, ainsi que les deux ponts constituent une signature remarquable de cette base.

En se penchant sur la figure 6, on valorise tout l'intérêt d'une démarche d'ingénierie de la connaissance rigoureuse sur la qualité de la validation réalisée par l'expert.

Pour un élément de connaissance, chaque contenu est validé individuellement. Mais les liens d'accrochages aux arborescences de classification (ontologies, points de vue...) qui structurent le langage du métier et les liens de voisinages aux autres éléments de connaissance, sont aussi des relations qualifiées et validées par l'expert.

Lorsque ce réseau constitué est présenté à des algorithmes d'agrégation, on voit apparaître des différentes facettes constitutive de l'expertise formalisée.

Ce point devient un élément riche pour les caractéristiques « compétences » à maîtriser et

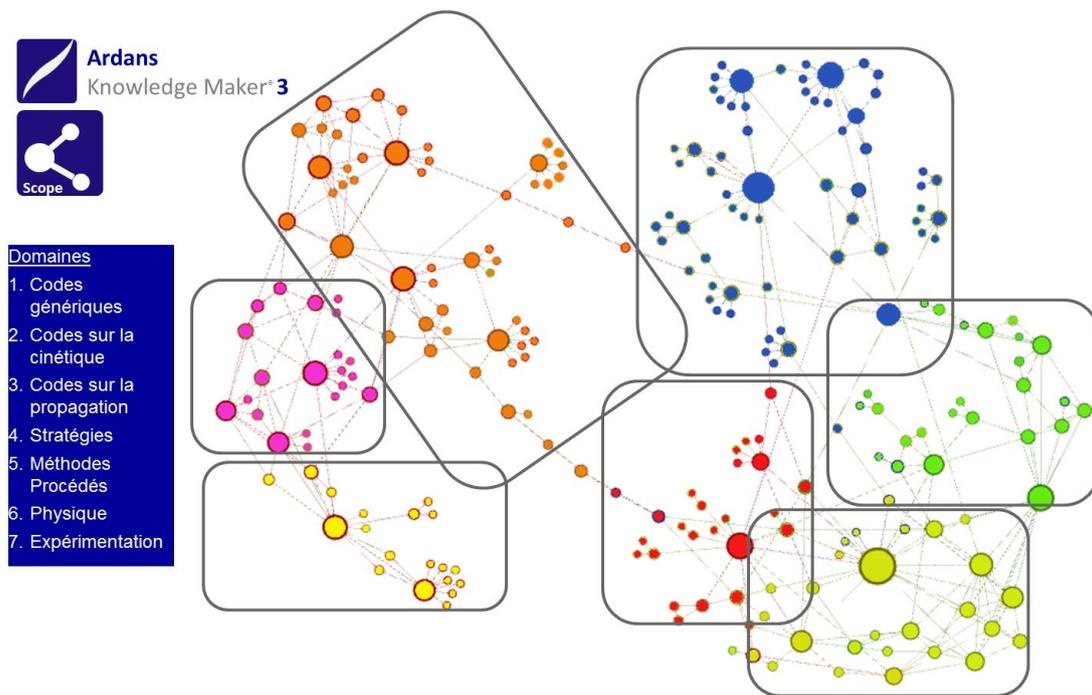


FIGURE 6 – Restitution d'un algorithme type Atlas pour faire émerger des agrégats d'EC.

donc à transférer vers des nouveaux acteurs sur le métier de l'expert.

Cette carte de la figure 7 correspond à un histogramme de la constitution de la base.

Elle reflète tout l'apport de la démarche d'ingénierie de la connaissance dans un projet de pérennisation d'expertise.

Le consultant et l'expert ont travaillé ensemble sur une partie déjà bien formalisée dans un premier temps : côté droit.

Puis, avec une technique de symétrie - l'image des ailes du papillon est très évocatrice -, l'IC et l'expert se sont penchés symétriquement sur la partie, côté gauche, où l'expert n'avait pas eu le temps de formaliser et de rédiger autant de notes techniques sur ce sujet extrêmement pointu.

Au cours de ce deuxième temps, l'expert s'est remémoré des éléments pour enrichir la partie droite. Alors qu'au milieu de la prestation l'expert était satisfait du travail sur cette partie droite, il s'est avéré qu'elle fut enrichie jusqu'à la fin de la réalisation de la prestation.

Cette représentation corrobore ce que constatait déjà Michel Grundstein 'Travailler sur la connaissance génère de la connaissance !' (sic).

3 Conclusion

Le métier d'ingénieur de la connaissance est donc lié à une discipline qui a évolué ces trente dernières années. La forte raréfaction des opérations de réalisation de système expert d'une part, et les arrivées d'outils de collaboration, puis de plates-formes dédiées à la pérennisation des expertises n'y sont pas étrangères. Avec ces nouveaux moyens, le métier a atteint un degré de qualité dans la mesure des contenus recueillis qui offre des analyses objectives nouvelles.

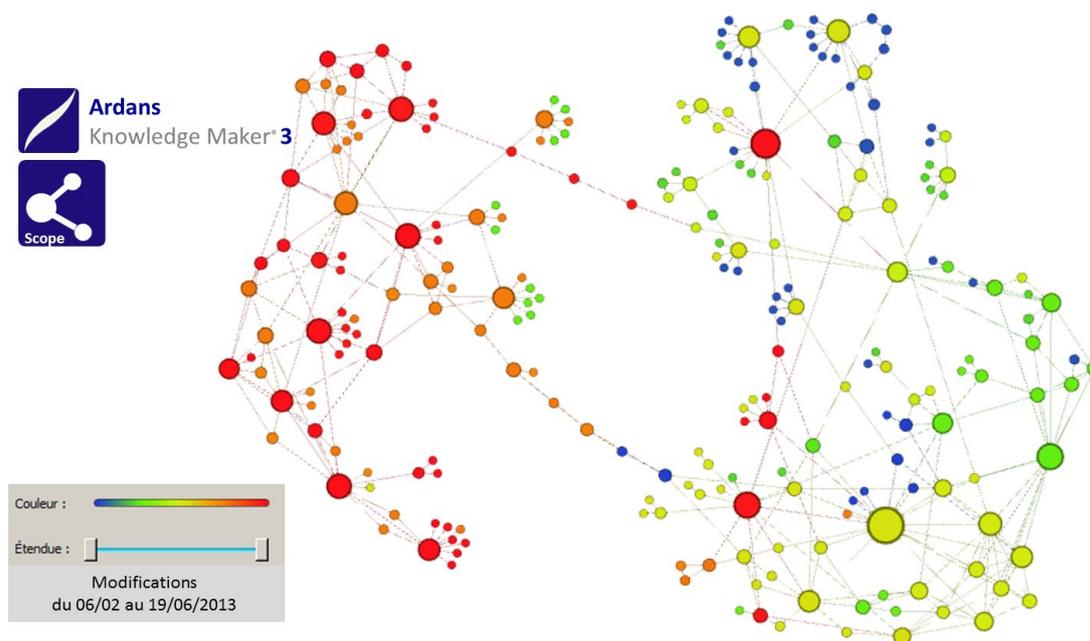


FIGURE 7 – Chronographie de recueil d'expertise : la connaissance génère de la connaissance.

Les résultats des actions de pérennisation de connaissance sont d'une part un recueil validé de l'expertise, mais aussi une abstraction des domaines sous-jacents qui la compose. L'ingénieur de la connaissance dispose aujourd'hui de moyens cartographiques pour aider les experts à mieux restituer leurs savoirs et pour accompagner les managers à apprécier la complexité des disciplines qui concourent à cette expertise.

Ce métier qui est au confluent des sciences humaines (psychologie, communication, management), de l'informatique, de la technique et de la langue est particulièrement exigeant pour celle ou celui qui le pratique. Il mérite d'être mieux reconnu dans les organisations et d'être enseigné avec soin dans les établissements académiques.

C'est certainement l'un des métiers où l'on réalise parmi les plus belles rencontres d'êtres passionnés : ces humains qui sont reconnus comme des « Experts ».

Références

- AUSSENAC-GILLES N. (2014). De la pénurie à la surabondance de connaissances : évolution de l'ingénierie des connaissances en France depuis 25 ans (www.irit.fr/CMS-DRUPAL7/IC2014/sites/irit.fr.CMS-DRUPAL7/IC2014/files/ConfInvitée1_Aussenac_histoireIC25ans.pdf). Conférencier invité.
- BERGER A. (2015). Expertise's capture in knowledge engineering : elements of return of experience of ardans. *EIRMA : SIG KM*.
- BERGER A. & COTTON J. (2011). Construire une mémoire collective de l'entreprise : la gestion des connaissances (www.ardans.fr/doc/AFIA_Communication.pdf). *AFIA n° 72*, p. 70–73.

- BESSON V. & BERGER A. (2015). To initiate a corporate memory with a knowledge compendium : ten years of learning from experience with the ardans method. *Revue des Nouvelles Technologies de l'Information*, Extraction et Gestion des Connaissances, RNTI-E-28, 401–412.
- COPPENS C., LAROQUE-MALBERT J., BERGER A., MARIOT P. & COTTON J. (2006). La capitalisation des connaissances dans l'industrie : implanter un référentiel métier et le déployer : l'exemple d'Icare (www.irit.fr/SDC2006/cdrom/contributions/Alainberger_SDC2006.pdf). *Actes de SdC 2006*.
- CORBEL J. (1997). *Méthodologie et retour d'expérience : démarche MEREX de Renault*. Paris : Hermès.
- DELMAS D. (2015). Acquérir des connaissances expertes par des techniques d'interview adaptées. *Université de Genève OVSM*.
- ERMINE J., CHAILLOT, M. AND BIGEON P., CHARENTON B. & MALAVIELLE D. (1996). MKSM a method For knowledge management. In S. ED., Ed., *International Symposium on the Management of Industrial and Corporate Knowledge (ISMICK'96)*, p. pp.288–302, Rotterdam , Neederlands.
- FUCHS N. & KALJURAND, K. AND KUHN T. (2008). Attempto controlled english for knowledge representation. volume Number 5224 of *Notes in Computer Science*, p. 104–124. Springer.
- GELB M. (2004). *How to think like Leonardo da Vinci*. London : Thorson.
- GRUNDSTEIN M. (1996). "CORPUS," An Approach to Capitalizing Company Knowledge. In A. PROCEEDINGS., Ed., *The Fourth International Workshop on Artificial Intelligence in Economics and Management*, Tel-Aviv, Israel.
- GRUNDSTEIN M. (2000). From capitalizing on company knowledge to knowledge management. *Knowledge Management, Classic and Contemporary Works by Daryl Morey, Mark Maybury, Bhavani Thuraisingham*, **chapter 12**, 261–287.
- GRUNDSTEIN M., DE BONNIERES P. & PARA S. (1988). *Les Systèmes à Base de Connaissances, Systèmes Experts pour l'Entreprise*. Paris La Défense : AFNOR Gestion.
- J.M P. (1990). SAGACE : une représentation des connaissances pour la supervision de procédés, Systèmes Experts de deuxième génération. In *EC2 (Ed.)*, p. paper 8, Avignon, France.
- KUHN T. (2014). A survey and classification of controlled natural languages. *Computational Linguistics*, **40**(1), 121–170.
- MALVACHE, P. AND PRIEUR P. (1993). Mastering Corporate Experience with the REX Method, Management of Industrial and Corporate Memory. In *International Symposium on the Management of Industrial and Corporate Knowledge (ISMICK'93)*, p. pp.33–41, Compiègne, France.
- MARIOT P., GOLBREICH C., , COTTON J. & BERGER A. (2007). Méthode, Modèle et Outil Ardans de capitalisation des connaissances (www.med.univ-rennes1.fr/lim/doc_166.pdf). *RNTI EGC 2007*, RNTI(E-12), 187–206.
- MATHIEU D., VEXLER F., BERGER A., HAIRAUT L. & SIMONETTI P. (2014). Retour d'expérience d'une opération de capitalisation de connaissance sur le dispositif d'analyse moléculaire MATEO. (www.irit.fr/CMS-DRUPAL7/IC2014/sites/irit.fr.CMS-DRUPAL7.IC2014/files/Berger_IC2014.pdf). *Session spéciale "Projet et Industrie"*.
- SCHREIBER G. , WIELINGA B. & J B. (1993). *KADS : A Principled Approach to Knowledge-based System Development*. San Diego : Academic Press.
- VEXLER F., BERGER A., COTTON J. & BELLONI A. (2013a). Eléments d'appréciation et d'analyse d'une base de connaissance : l'expérience industrielle d'Ardans. (http://eric.univ-lyon2.fr/aide/actesAIDE_EGC2013ENLIGNE.pdf). *Actes Atelier AIDE @ EGC'2013, 13^{ème} Conférence Francophone sur l'Extraction et la Gestion des Connaissances*, Actes-Aide(EGC2013), 59–72.
- VEXLER F., BERGER A., COTTON J. & BELLONI A. (2013b). Représentation et visualisation de connaissances expertes : une préoccupation industrielle. *Actes Journées EGC- AFIHM*, .
- VOGEL C. (1988). *Génie cognitif*. Paris : Masson.

Dix ans : L'âge de la maturité pour la plate-forme WebLab

Stephan Brunessaux¹, Patrick Giroux¹

¹ Airbus Defence and Space, Val de Reuil
 {stephan.brunessaux,patrick.giroux}@airbus.com

Résumé : Depuis maintenant dix ans, Airbus Defence and Space développe et met en œuvre une plate-forme d'intégration baptisée WebLab. Cette plate-forme, mise à disposition en open source via le consortium OW2¹, est utilisée dans de nombreux projets pour réaliser des applications qui exploitent des technologies de fouille de documents multimédia et de traitement automatique des langues. Elle permet de composer des solutions répondant à différents besoins métier en assemblant des briques logicielles d'origines diverses et variées (composants du commerce, logiciels open-sources, modules développés par des laboratoires de recherche, etc.). Longtemps utilisée comme plate-forme de référence pour le développement de prototypes ou de démonstrateurs dans le cadre de projets de R&D, WebLab est devenue une infrastructure robuste et efficace sur laquelle reposent des applications industrielles et des solutions opérationnelles considérées aujourd'hui comme des offres majeures d'Airbus Defence and Space.

Mots-clés : Plate-forme d'Intégration, Fouille de données, Web Intelligence, Web Sémantique, Extraction et gestion des connaissances, Traitement Automatique du Langage, Traitement de la parole.

1 Qu'est-ce que WebLab ?

WebLab² est le nom de la plate-forme d'intégration de composants de fouille de documents que développe Airbus Defence and Space depuis maintenant dix ans (Giroux, 2008).

L'objectif de cette plateforme est de fournir un ensemble de moyens et d'outils logiciels destinés à la construction de système d'informations et d'applications dans le domaine du renseignement d'origine source ouverte (OSINT). Elle a été conçue pour répondre à différents besoins métiers (renseignement militaire, intelligence économique, veille stratégique ou scientifique, gestion d'archive ou de « bigdata », etc.) et elle est plus particulièrement dédiée au traitement de contenus multimédia et d'informations non-structurées (texte, image, audio et vidéo). Les services de WebLab permettent de convertir ces données non-structurées issues des sources ouvertes en information structurée accessible et compréhensible par des applications informatiques.

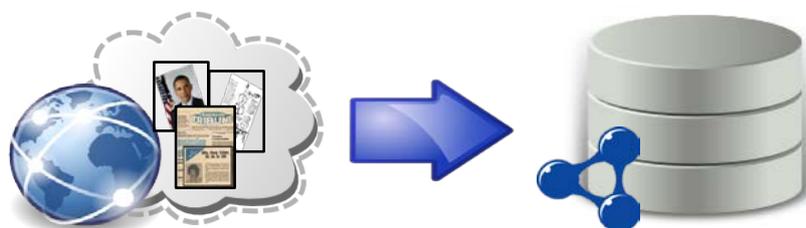


Figure 1: De l'information non structurée à la connaissance

¹ forge.ow2.org/projects/weblab/

² www.weblab-project.org

Face à l'enjeu important que représente le traitement d'informations non structurées, il est donc essentiel de pouvoir mettre en place des solutions faisant appel aux technologies les plus efficaces à chaque étape des processus. Il est également fondamental d'avoir la capacité de faire évoluer ces solutions au fur et à mesure des avancées technologiques et scientifiques car, dans ce domaine, le nombre d'outils, leurs performances et leurs niveaux de maturité évoluent à un rythme très soutenu.

Les travaux relatifs au développement de la plate-forme WebLab furent initialisés en 2005 et se poursuivent aujourd'hui encore pour soutenir bon nombre de projets de recherche dans lesquels Airbus Defence and Space est impliqué. Grâce à WebLab, Airbus Defence and Space peut se positionner dans ces projets comme l'architecte/intégrateur des prototypes de démonstration. Son rôle est ainsi de fédérer des travaux de R&D menés par différents partenaires et d'unifier les résultats obtenus pour construire des applications cohérentes et innovantes.

Par ailleurs, lors des dernières années, l'architecture technique et certains composants de la plate-forme ont atteint un niveau de maturité industrielle qui les rend parfaitement éligibles pour la réalisation de projets à caractère opérationnel. Ceci vient récompenser le travail et les investissements consentis par Airbus Defence and Space pour se doter de cet outil qui s'avère être un atout majeur pour la société.

2 Une plate-forme d'intégration

La plate-forme WebLab se compose d'un socle technique et propose également un catalogue de composants logiciels qui offrent des fonctionnalités de collecte de données sur des sources ouvertes (Internet, TV ou radio, presse écrite, etc.) ou dans des entrepôts privés, de traitement automatique des contenus (extraction d'informations, analyse sémantique, classification, transcription speech-to-text, traduction, segmentation, reconnaissance d'écriture, détection de motifs, etc.), de capitalisation (stockage, indexation, enregistrement dans des bases de connaissance, etc.) et d'exploitation de l'information (recherche avancée, visualisation, synthèse graphique, aide à la décision, etc.).

Dans ces domaines, bon nombre d'outils sont aujourd'hui disponibles qui répondent, de façon plus ou moins satisfaisante, à un sous-ensemble du besoin inhérent à l'exploitation des informations en sources ouvertes. Les outils spécialisés proposent des fonctionnalités performantes mais souvent très ciblées (robot de collecte, analyseur morphosyntaxique, traducteur automatique, moteur de recherche, etc.). Quelques outils ou suites plus « généralistes » couvrent un spectre plus large mais présentent des limitations en termes de performance. Par ailleurs, ils sont souvent construits sur des architectures « propriétaires » ce qui les rend difficilement extensibles et ne permet pas de les coupler avec d'autres outils.

Dans un souci d'efficacité et de rationalisation des coûts de développement, il semble assez naturel et pragmatique de sélectionner les meilleurs outils et de les assembler pour répondre à l'ensemble des besoins identifiés. Cette approche consiste donc à évaluer chaque élément de solution par rapport aux services qu'il peut potentiellement offrir, à retenir le meilleur élément dans chaque domaine fonctionnel (l'outil offrant le meilleur niveau de service) et à le considérer comme l'un des composants d'une solution globale couvrant, si possible, l'intégralité du besoin. Dans cette logique, il est fort probable que certains outils seront sous-utilisés et il n'est pas certain que la totalité du besoin puisse être couverte. La complémentarité est donc essentielle (Brunessaux, 2011).

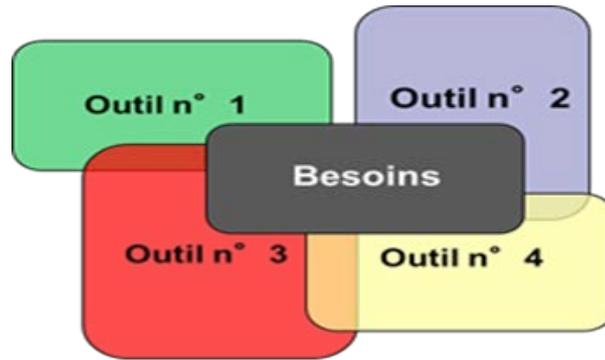


Figure 2: Couverture du besoin par un ensemble d'outils

Mais pour que la solution résultante de cet assemblage puisse satisfaire le besoin, encore faut-il que ces composants puissent cohabiter dans le même environnement et aussi qu'ils puissent interopérer. Dans cette optique, le besoin d'interopérabilité constitue une réelle difficulté car les composants sélectionnés ont, bien sûr, été développés séparément en utilisant des technologies différentes. Cette agrégation d'outils techniquement disparates nécessite la mise en place d'une structure d'accueil commune qui fournit des mécanismes d'interaction et d'échange entre les composants. Ce socle technique doit permettre:

- de disposer d'une infrastructure informatique sur laquelle pourront être déployés les différents outils à intégrer en tant que composants d'une solution globale,
- de s'appuyer sur un intergiciel ouvert et extensible pour assurer la médiation entre les composants intégrés quelles que soient leurs origines et les technologies qu'ils utilisent,
- de s'appuyer sur un modèle de connecteur permettant de faire communiquer les composants entre eux et d'assurer leur interopérabilité technique,
- de s'appuyer sur un modèle de référence pour normaliser les échanges entre composants et assurer leur interopérabilité sémantique,
- de construire des chaînes de traitements mettant en jeu un ensemble d'outils interopérables.

Les outils intégrés offrent donc un ou plusieurs services et l'invocation de chaque service correspond à une étape dans le processus automatique mis en œuvre. A chacune des étapes du processus ainsi défini, les documents sont annotés par les services mis en œuvre et leurs descriptions s'enrichissent progressivement au fur et à mesure de la progression dans la chaîne. Chaque service ajoute donc de nouveaux éléments de connaissance qu'il attache au document traité (Caron, 2014).

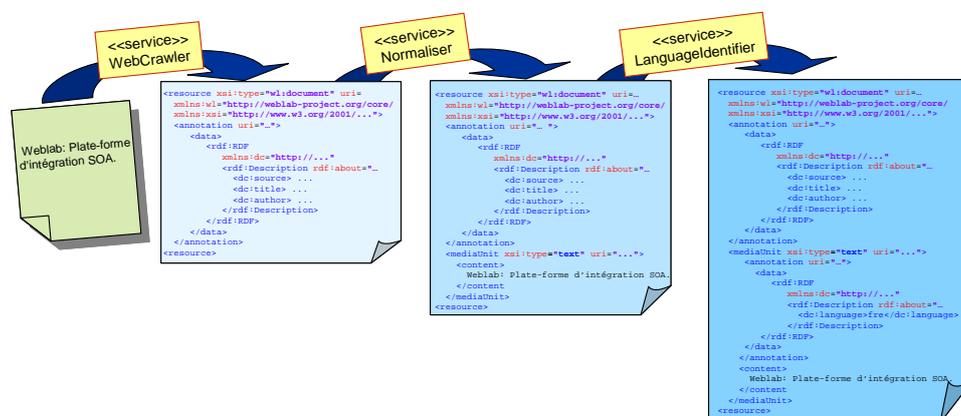


Figure 3 Exemple de chaîne de traitement WebLab

3 Une architecture modulaire

La modularité est l'une des qualités exigées pour l'architecture de la plate-forme car elle doit permettre:

- l'intégration de briques fonctionnelles d'origines diverses,
- l'assemblage de ces briques pour constituer un système cohérent permettant de satisfaire le besoin opérationnel,
- l'évaluation comparative de différentes briques offrant des fonctionnalités similaires dans un même cadre d'utilisation et dans un même environnement technique de mise en œuvre,
- le remplacement d'une brique par une autre brique jugée plus performante à un instant donné de la vie du système.

La plate-forme WebLab est structurée selon 3 niveaux :

- le WebLab-Core qui se présente comme un socle technique «open source» fournissant une infrastructure technique de communications entre les services, des mécanismes d'orchestration de ces services, un modèle d'échange d'informations entre les services ainsi que des mécanismes de composition d'IHM basés sur une technologie de type portail;
- les services WebLab qui désignent les modules fonctionnels réalisant un traitement spécifique sur des documents ou sur des données «métier». Les services WebLab peuvent être fournis par des composants du commerce, «open source» ou de laboratoire. Plus de 80 services WebLab ont déjà été réalisés à ce jour par «encapsulation» de composants. Airbus Defence and Space propose un catalogue de services développés « en interne » pour répondre à des besoins particuliers. Dans les projets collaboratifs de R&D, chaque partenaire délivre un ou plusieurs services Weblab qui concrétise ses travaux et sa contribution au projet (VITALAS³, SAIMSI⁴, VIRTUOSO⁵, TWIRL⁶, AXES⁷, etc.). (Fluhr, 2013)
- enfin, les applications WebLab, sont dédiées à un besoin «métier» particulier et sont construites par composition de services WebLab. Elles sont développées en exploitant les mécanismes du WebLab Core et en combinant différents services WebLab afin de traiter une problématique donnée. Weblab permet de répondre à des besoins fonctionnels dans des domaines « métier » divers et variés (Giroux, 2012).

³ <http://vitalas.ercim.eu/>

⁴ http://www.geolsemantics.com/projet_SAIMSI.php

⁵ <https://www.eurosint.eu/virtuoso-project>

⁶ <https://itea3.org/project/twirl.html>

⁷ <http://forge.ow2.org/projects/openaxes/>

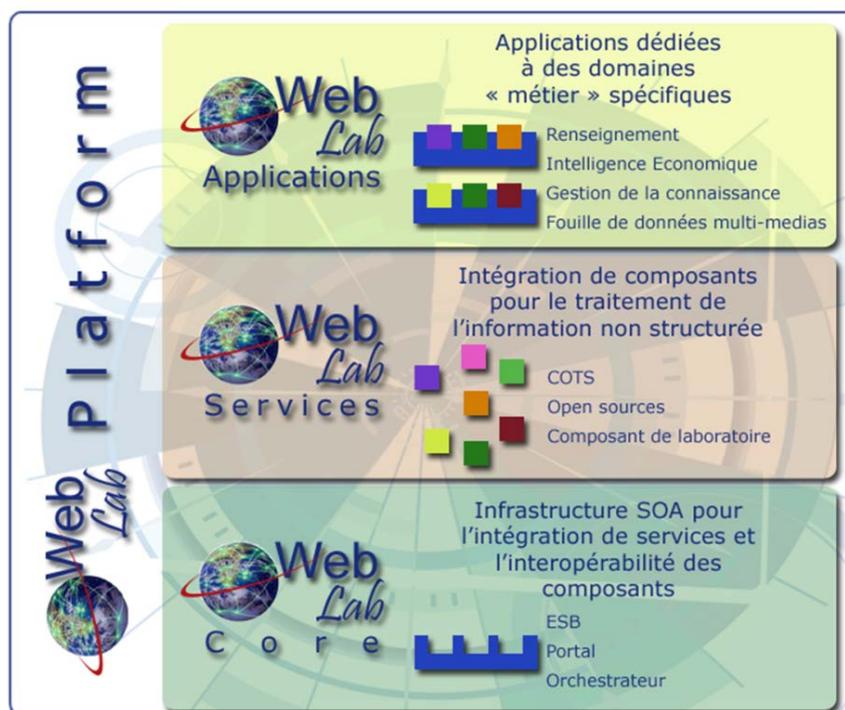


Figure 4: Les trois niveaux de la plate-forme WebLab

Le Weblab Core est la pierre angulaire de la plate-forme. Pour permettre les échanges de messages entre services, il fournit un modèle de référence, basé sur les standards du Web Sémantique (XML, RDF, RDFS, OWL, SPARQL, ...), permettant à des composants logiciels hétérogènes d'échanger efficacement les informations et les connaissances issues des traitements qu'ils effectuent sur les contenus multimédias (Caron, 2012). Il propose les mécanismes nécessaires pour normaliser les interactions entre les composants et simplifier la construction de chaînes de traitement dans lesquelles ils sont mis en œuvre pour contribuer à un objectif global.

Le modèle d'échange fait l'objet d'une documentation technique détaillée. Il se traduit par un schéma de données permettant de caractériser explicitement :

- un document multimédia par une référence à son contenu binaire dans son format natif,
- un contenu normalisé dans un format pivot compréhensible par tous les services. Ce contenu peut être inclus dans les données échangées ou accessible dans un référentiel partagé par les services,
- un ensemble de métadonnées qui fournissent des descriptions techniques et/ou sémantiques du contenu du document et qui prennent la forme d'annotations généralement générées comme résultats de traitement.

La couche WebLab Services propose un ensemble de « briques » réutilisables et composables pour construire rapidement des applications adaptées à un besoin particulier. Ces briques prennent la forme de services qui couvrent un large spectre de fonctionnalités et de composants d'IHM qui permettent de piloter les services et d'en exploiter les résultats.

Les services peuvent être sélectionnés puis assemblés et orchestrés selon un processus métier dont la description est réalisée grâce à des outils d'édition graphique.

L'IHM est constituée par assemblage de composants qui s'exécutent au sein d'un portail Web personnalisable par l'utilisateur final. Pour des besoins nouveaux ou spécifiques, les services et composants d'IHM peuvent être créés de toutes pièces ou développés en intégrant des composants du commerce ou open source.

4 L'offre WebLab

En 2015, l'un des enjeux majeurs du projet WebLab réside dans la nécessité de proposer à la communauté un environnement de travail encore plus efficace et plus simple d'utilisation. L'objectif est de donner à chaque utilisateur de la plate-forme une bonne compréhension des principes sur lesquels elle repose, de faciliter la compréhension des techniques mises en œuvre et d'offrir des outils capables de masquer la complexité technologique inhérente à l'intégration de composants hétérogènes.

Un autre enjeu important concerne les performances et, en particulier, l'aptitude des applications WebLab à traiter des données volumineuses et hétérogènes pour les exploiter « en boucle courte » c'est-à-dire dans un délai contraint par la mise à disposition massive et continue de nouvelles informations. On peut noter que l'on retrouve ici le paradigme des 3V (Volume, Variété, Vitesse) souvent utilisé pour caractériser le phénomène Big Data dont tout le monde parle aujourd'hui (St. Brunessaux, 2014).

Enfin, des solutions « clef en main » doivent pouvoir être proposées pour satisfaire des besoins métiers spécifiques. Des services dédiés et des chaînes de traitement adaptées doivent être implantés sur le socle WebLab existant. Une réflexion doit être menée pour faciliter le développement d'extensions ou de « plug-in » qui entrent dans la composition d'applications pouvant parfois sortir du cœur de métier Airbus Defence & Space. Trois domaines semblent aujourd'hui être particulièrement concernés par les capacités du WebLab : le Renseignement d'Origine Humaine (HUMINT) ou Source Ouverte (OSINT), la veille en Intelligence Économique et la gestion d'archives audio-visuelles.

Aujourd'hui, l'offre WebLab de Airbus Defence and Space se décline en plusieurs volets :

- Le cœur du système (en bleu) est disponible en « open-source » par l'intermédiaire du consortium OW2 dont Airbus Defence and Space est membre,
- Un ensemble de services et une application de démonstration sont également disponibles en open source sur le site du projet WebLab d'OW2
- Différentes applications « verticales » sont proposées au catalogue Airbus Defence and Space parmi lesquelles
 - WebLab for Competitive Intelligence
 - WebLab Media-Mining
 - WebLab for OSINT and HUMINT (ACTINT® MediaMining)



Figure 5: L'offre WebLab

5 Enjeux et perspectives

Même si la plate-forme WebLab a fait ses preuves dans de nombreux projets, les équipes d'Airbus Defence and Space continuent de la faire évoluer dans le cadre d'une roadmap ambitieuse qui prévoit de nombreuses améliorations dans les mois à venir.

La version 2.0 de la plate-forme WebLab a été mise à disposition en open source sur le site OW2 en Mars 2014. La technologie BPEL initialement choisie pour définir les chaînes de traitement a été délaissée au profit d'une approche basée sur les EIP (Enterprise Integration Pattern). La solution exploite désormais les capacités d'un « Middleware » orienté message basé sur le standard JMS (Java Message Server). Cette solution technique donne la possibilité de déposer les données collectées dans une file d'attente (Queue) afin de permettre à la chaîne de traitement de les absorber au fil de l'eau et de les traiter en asynchrone en fonction de la disponibilité des ressources.

Le bus de services existant qui était basé sur la technologie JBI (Java Business Integration) a été remplacé par une solution plus adaptée basée sur le logiciel Libre Apache-Camel. Ce composant a été étendu par l'intermédiaire d'un connecteur permettant d'assurer une compatibilité avec le modèle d'échange WebLab. Cette architecture remaniée est aujourd'hui utilisée pour le développement d'une nouvelle génération d'applications WebLab dans les différents domaines cibles. Elle offre une bonne « scalabilité » et permet, dans la grande majorité des cas, de répondre aux besoins. Mais, les équipes Airbus DS s'intéressent aussi aux technologies « Big Data » pour anticiper les besoins des nouvelles applications qui sont amenées à traiter des volumes sans cesse croissants de données. Certains services de l'architecture actuelle ont déjà fait l'objet de travaux liés à l'introduction de Hadoop.

Le Data Analytics, l'analyse des réseaux sociaux, les modèles prédictifs, l'exploitation visuelle des données, etc. sont autant de sujets qui constituent des opportunités de valorisation et d'enrichissement des capacités de WebLab.

Les travaux de recherche se poursuivent également pour aller plus loin au niveau technologique et scientifique. Le département accueille depuis plusieurs années des doctorants en contrat CIFRE pour conduire des travaux de recherche qui concourent à enrichir fonctionnellement ou technologiquement la plate-forme WebLab (Grilhères, 2009), (Saval, 2011), (Dupont, 2011), (Doucy, 2011), (Mombrun, 2012), (Bondu, 2013), (Serrano, 2014), (Noel, 2014).

Références

- SY. BRUNESSAUX, S. CANTARELL et all. (2011). HERISSON: a WebLab-based platform for assessment and experimentation of information processing technologies. ICSSEA 2011, Paris, France.
- ST. BRUNESSAUX (2014). WebLab in the Land of Big Data - OW2'con, Issy-Les-Moulineaux, France
- E. BONDU (2013). Contribution à la veille stratégique: Système d'aide à la capitalisation et l'exploitation de connaissances expertes Thèse de Doctorat. LITIS, Rouen, France.
- C. CARON, B. AMANN et all. (2014). WePIGE: The WebLab Provenance Information Generator and Explorer, EDBT/ICTD, Athens, Greece.
- C. CARON, J. GUILLAUMONT et all. (2012). WebLab : une plateforme open-source et collaborative dédiée à la capitalisation de connaissances. Démonstration à la 12ème Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances (EGC 2012), Bordeaux, France.
- J. DOUCY (2011). Méthodologie pour l'orchestration sémantique de services, application au traitement de documents multimédia. Thèse de Doctorat. LITIS, Rouen, France.
- G. DUPONT (2011). Apprentissage Implicite pour la recherche d'information. Thèse de Doctorat. LITIS, Rouen, France.
- C. FLUHR et all. (2013). SAIMSI Suivi Adaptatif Interlingue et MultiSources des Informations, WISG 2013, Troyes, France,

- P. GIROUX, ST. BRUNESSAUX et all. (2008). WebLab: An integration infrastructure to ease the development of multimedia processing applications. ICSSEA 2008, Paris, France.
- P. GIROUX, O. CORBY et all. (2012). Earth Modeling using Web Services - Chapitre de l'ouvrage: Shared Earth Modeling Knowledge Driven Solutions for Building and Managing Subsurface 3D Geological Models par M.Perrin & JF.Rainaud).
- B. GRILHERES (2009). Apprentissage pour l'adaptation d'une plate-forme d'intelligence informationnelle. Thèse de Doctorat. LITIS, Rouen, France
- Y. MOMBRUN (2012). Évaluation de l'information disponible sur Internet : Application au renseignement d'origine sources ouvertes. Thèse de Doctorat. LITIS, Rouen, France.
- R. NOEL (2014). Contribution à la veille stratégique : DOWSER, un système de découverte de sources Web d'intérêt opérationnel. Thèse de Doctorat. LITIS, Rouen, France.
- A. SAVAL (2011). Modèle Temporel, Spatial et Sémantique pour la Découverte de Relations entre événements. Thèse de Doctorat. GREYC, Caen, France.
- L. SERRANO (2014). Vers une capitalisation des connaissances orientée utilisateur - Extraction et structuration automatiques de l'information issue de sources ouvertes. Thèse de Doctorat. GREYC, Caen, France.

Conception de « Scènes de Vie » pour le Peuplement d'Environnements Virtuels

Caroline Chopinaud et Etienne de Sevin

MASA Group

75002 Paris

<http://www.masagroup.net/>

{caroline.chopinaud, etienne.de-sevin}@masagroup.net

Résumé : La conception de personnages non-joueurs capables d'exhiber des comportements crédibles, dans l'objectif de doter les environnements virtuels d'une véritable « ambiance de vie » dans laquelle l'utilisateur peut aisément s'immerger, est un défi majeur pour les jeux actuels. La qualité de l'expérience utilisateur réside en partie dans la cohérence des « scènes de vie » qui sont jouées collectivement dans l'environnement et des interactions qui sont proposées au travers de ces scènes. Pour simplifier le travail de conception de telles scènes complexes, nous développons un outil dédié aux game designers leur permettant de créer et exécuter des scènes de vie. Cet outil se base sur le concept de « smart zone » pour définir les caractéristiques et gérer de façon automatisée le déroulement des scènes. L'objectif est de fournir une solution complète de création de scènes et de gestion des comportements des PNJ permettant aux designers de garder le contrôle et la compréhension des comportements réalisés.

Mots-clés : agents, systèmes adaptatifs, comportements collectifs, environnement virtuel, jeux

1 Introduction

Les applications et jeux immersifs proposent aux utilisateurs d'évoluer dans un environnement virtuel avec lequel ils peuvent interagir. L'immersion ressentie par le joueur passe par la présence de personnages ou groupes de personnages non-joueurs (PNJ) ayant des comportements crédibles vis-à-vis de la situation proposée (Johansson, 2013). Les PNJ se doivent de réagir de façon cohérente aux actions du joueur sous peine de nuire à l'expérience proposée. Des jeux tels que *Assasins'Creed*, *WatchDogs* (Ubisoft) ou *Skyrim* (Bethesda) tirent parti de la présence forte de PNJ qualifiés de « personnages ambiants » ou de « second plan » pour orienter le scénario du jeu et proposer aux joueurs un environnement riche et peuplé. Néanmoins, les comportements proposés restent assez répétitifs et limités dans leur adaptation et leur réaction par rapport aux joueurs, les rendant peu crédibles sur la durée. Cela s'explique en partie par le fait que les studios de jeu ont peu de moyens à investir dans la mise en œuvre de mécanismes et d'architectures d'intelligence artificielle qui leur permettraient d'améliorer considérablement les comportements des PNJ. Le processus de mise en place d'une telle ambiance est en effet long et complexe : d'une part le designer est responsable de spécifier les comportements des personnages et d'autre part un développeur se charge de l'implémentation. Ainsi, de nombreux échanges et itérations sont généralement nécessaires pour obtenir des comportements et des scènes d'ambiance en accord avec les attentes du designer (Becroft et al., 2011).

Dans ce contexte, nos travaux visent à apporter une solution permettant la mise en place d'ambiance dans les environnements virtuels par la définition de *scènes de vie*. Une scène de vie est représentée par un ensemble de personnages non-joueurs qui exécutent des comportements individuels dans l'objectif de produire un comportement collectif pour réaliser une ambiance donnée. L'exécution de ces scènes crée une impression de vie dans

l'environnement interactive pertinente en tenant compte des actions du joueur et du contexte courant du jeu. Elles ont pour vocation première de peupler les environnements de jeux (vidéo ou sérieux), simulations (formation, entraînement) ou mondes virtuels (communication) pour fournir au joueur une expérience interactive pertinente. Le concept de scène de vie se rapproche de celui de *pattern of life* (Folsom-Kovarik et al., 2013) utilisé dans les simulations d'entraînement militaire, ou des comportements ambiants (Cutumisu et al., 2006).

Notre solution permet de structurer la conception d'ambiance en deux niveaux : la définition des comportements individuels des PNJ d'une part, et la définition du comportement des scènes d'autre part. Pour ce faire, nous proposons un outil de conception de scènes intégré dans Unity3D.

Il existe peu d'outils à destination des designers pour le peuplement d'environnements virtuels à l'aide de PNJ ambiants. L'outil REACT intégré à Unity3D utilise un modèle de Behavior Tree pour simplifier la conception des comportements mais il ne permet pas de s'abstraire d'une étape d'implémentation. L'outil BehaviorShop (Heckel, Youngblood, & Hale, 2009) résout une grande partie des difficultés de conception grâce à l'utilisation d'une architecture de subsumption assez intuitive mais qui à ce jour ne permet d'accéder qu'à un ensemble restreint de comportements possibles. La simulation de foule est également une solution utilisée pour peupler rapidement des environnements (Yersin et al., 2009). Dans l'industrie plusieurs sociétés (e.g. Golaem, Autodesk, Havok) proposent des solutions de conception de comportements individuels ou de foule à grande échelle. Dans tous les cas ces solutions ne permettent pas de répondre à la problématique de conception d'ambiance scénarisées riches et interactives.

SceneMaker (Gebhard et al., 2003) propose un langage de script simple, à l'instar de Improv (Perlin & Goldberg, 1996) et de SCREAM (Prendinger & Ishizuka, 2002), pour permettre à des experts non-programmeurs de créer des contenus riches correspondant que l'on peut rapprocher des scènes de vie. Les scènes peuvent être écrites grâce à ce langage qui se veut proche du langage naturel, pour ensuite être traduites en différents plans de scènes interconnectés. Les enchaînements de scènes sont définis à l'aide de machine à états finis hiérarchiques, ce qui a l'avantage d'être assez intuitif et puissant en termes d'expressivité des transitions. Néanmoins, le concepteur se doit d'avoir une certaine expertise en programmation pour construire ces automates et en particulier en ce qui concerne la spécification des transitions.

Dans cet article, nous présentons les concepts et caractéristiques sur lesquels utilisés par notre outil de conception de scènes de vie qui vise à être accessible directement par les (game)-designers. Cet outil se fonde principalement sur le concept de *smart zone* pour la définition et la gestion du déroulement des scènes.

2 Les Smart Zones pour la conception des Scènes de Vie

2.1 Concept de Smart Zones

Une *smart zone* est une représentation concrète d'une scène de vie localisée dans l'environnement virtuel, réactive à la présence des joueurs et exécutée de façon indépendante pour gérer la durée de vie des scènes. Le concept de smart zone est inspiré des Smart Objects (Kallmann & Thalmann, 1999), souvent utilisés dans les jeux vidéo pour gérer les interactions agents-objets (comme dans The Sims, EA, 1999). Sur ce principe, une smart zone gère les comportements collectifs ainsi que l'assignation de comportements individuels aux PNJ présents dans la zone. Une smart zone contient donc la description de toutes les caractéristiques d'une scène de vie et la façon dont les PNJ impliqués vont exécuter leurs comportements de façon synchronisée. Quand un PNJ entre dans une smart zone, il a accès aux caractéristiques de la scène et peut exécuter automatiquement les comportements attendus.

Des approches récentes utilisent des concepts similaires comme les « Smart Events » (Stocker et al., 2010) (Shoulson & Badler, 2011) qui s'intéresse à la gestion de PNJ d'ambiance « extras » orientés événements pour des problématiques de narration, ou les « Smart Areas » (Černý et al., 2014) pour la gestion des comportements d'un PNJ en fonction de sa localisation et les « SmartBuildings » (Blank et al., 2011) pour gérer des foules liée à des bâtiment et des objets dans les simulations d'entraînement. Ces solutions, méthodes de conception et parfois outils sont des solutions qui chacune dans leur domaine permettent de gérer une partie ou une forme d'ambiance dans les environnements. Mais elles ne fournissent pas d'outil de conception d'ambiance sous la forme de groupe de PNJ exhibant des comportements collectifs scénarisés et contrôlables par le game designer.

L'intérêt des smart zones réside dans le fait d'avoir un contrôle décentralisé des comportements des PNJ. En effet, il n'est pas nécessaire de contrôler la situation courante du jeu dans son ensemble mais uniquement de se focaliser sur chaque smart zones instanciées dans l'environnement. Il est aussi alors possible de construire des bibliothèques de scènes que l'on peut instancier et paramétrer dans de nouveaux environnements. En intégrant ce concept de smart zone nous visons à proposer une solution qui unifie les différents concepts d'événements, de zones et d'objets pour fournir un outil complet de conception d'ambiance.

2.2 Conception des Scènes de Vie

Une scène de vie est localisée dans l'environnement, dans le temps et s'exécute en fonction d'événements incluant la présence du joueur. Chaque PNJ impliqué dans une scène remplit un rôle. Chaque rôle est défini par un ensemble de comportements qui seront exécutés pour les PNJ pour réaliser la scène attendue. Les comportements sont séquencés et organisés par type de rôle pour créer le comportement collectif attendu dans la scène ou une partie de la scène. Ainsi, créer une scène de vie dans un environnement virtuel revient à définir la smart zone correspondante et de fournir les informations suivantes :

- Les *rôles* qui seront assignés aux PNJ,
- Les *comportements* associés aux rôles,
- Les *séquences* de comportements définies pour l'orchestration de la scène,
- Les *déclencheurs* pour l'activation de la scène,
- La *localisation* de la scène de vie dans l'environnement (position, taille et forme).

Enfin, les smart zones sont instanciées et positionnées dans l'environnement. Ce dernier est considéré comme la zone monde et les autres zones instanciées dans l'environnement vont être placées par superposition. La gestion de potentielle superposition se fait par la définition d'une priorité entre zones qu'il convient également de définir.

2.3 Gestion du déroulement des Scènes de Vie

Dans cette section, nous allons présenter l'architecture des smart zones (Fig. 1) qui permet de gérer l'instanciation, le déclenchement, l'exécution et l'arrêt d'une scène ainsi que l'assignation des rôles et des comportements aux PNJ entrant dans une zone.

2.3.1 Gestion des Déclencheurs

Un déclencheur de scène correspond à un ensemble de règles d'activation qui spécifient les critères de démarrage d'une scène. Une règle d'activation se fonde sur la détection d'événements tels que un instant donné, la présence d'un certain PNJ ou du joueur à un endroit spécifique, une interaction.... Ce premier module de l'architecture s'assure ainsi que le contexte courant permet à la scène de s'exécuter. Le module prend en entrée tous les déclencheurs définis pour la scène et vérifie si toutes les conditions sont remplies.

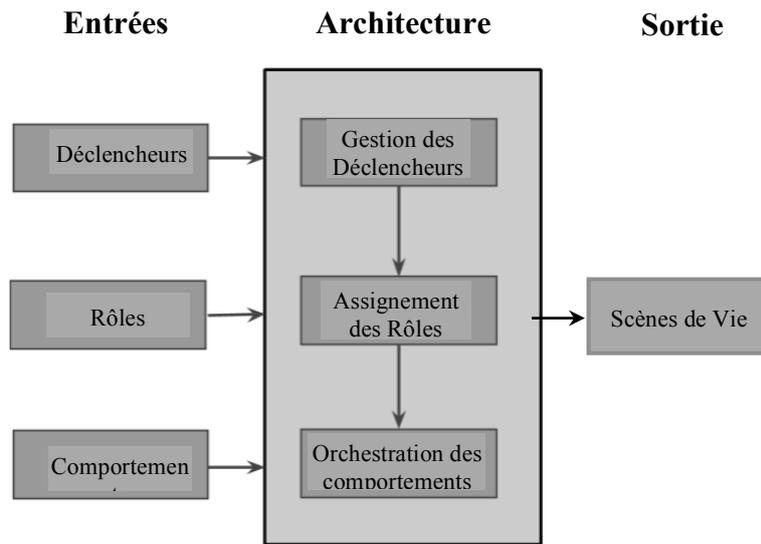


FIGURE 1 – Architecture d'une smart zones

2.3.2 Assignment des Rôles

Ce module est responsable de l'assignement des rôles pour chaque PNJ présents dans la zone. Un PNJ ne peut endosser qu'un seul rôle à la fois. Afin de gérer automatiquement l'affectation des rôles, nous les avons classés en trois catégories : les rôles principaux, secondaires et extras. Les rôles principaux sont indispensables pour l'exécution de la scène. Ils sont utilisés pour diriger et synchroniser le comportement de la scène. Les rôles secondaires, optionnels, interagissent avec les rôles principaux et entre eux pour créer le comportement collectif de la scène. Enfin, les rôles extras servent essentiellement à ajouter du contenu mais ne sont pas pris en compte dans l'orchestration du comportement collectif de la scène. Si tous les rôles principaux sont assignés, la scène peut être exécutée. A chaque fois qu'un PNJ entre dans une smart zone, le module détermine en fonction des rôles encore disponibles quel rôle doit lui être assigné. Si plus aucun rôle n'est disponible, le PNJ ne rejoint pas la scène et garde le rôle qu'il avait dans la scène en cours moins prioritaire (par défaut la zone monde).

2.3.3 Orchestration des Comportements

Ce module est responsable de l'assignement automatique des comportements aux PNJ, à partir de l'orchestration des comportements par rôle dans un séquenceur (Fig. 2). En fonction de son rôle un PNJ exécute la séquence de comportements associée. Le séquenceur orchestre dynamiquement les rôles des PNJ dans le temps et les synchronise dans l'objectif d'exécuter des comportements collectifs cohérents.

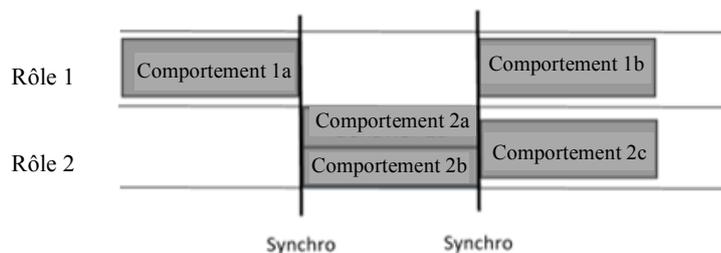


FIGURE 2 – Séquenceur pour l'orchestration des comportements

3 Implémentation de l'outil dans Unity3D

Un premier prototype de l'outil utilisant le concept de smart zone a été développé dans le logiciel 3D de création de jeux Unity3D, dont l'utilisation est largement répandue dans l'industrie du jeu vidéo. Le moteur d'exécution des smart zones fondé sur l'architecture présentée dans la section précédente a été quant-à-lui développé dans le logiciel MASA Life à l'aide de composants fournis par ce logiciel (C++) et de modèle de décisions utilisant des Behavior Trees. Ainsi, l'outil propose l'interface nécessaire à la définition des caractéristiques des scènes de vie qui sont regroupées dans les smart zones correspondantes. Ces zones sont placées dans l'environnement via l'interface de Unity3D (cercles sur Fig. 3). L'outil propose également une interface pour le séquenceur (Fig. 4) permettant de placer rapidement en fonction des rôles les comportements à exécuter pour la scène.

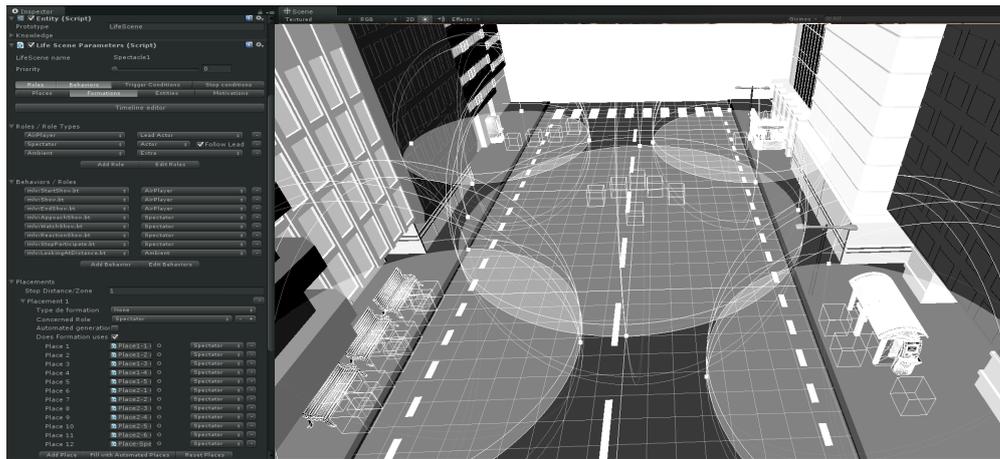


FIGURE 3 – Instanciation des scènes dans Unity3D – Exemple de rue commerçante

Pour démontrer l'utilisabilité de notre outil et l'apport du concept de smart zone pour la gestion d'ambiance via des scènes de vie, nous avons développé un environnement virtuel intégrant un scénario de rue commerçante. Dans un premier temps, les comportements individuels exécutés dans les scènes ont été créés grâce au logiciel MASA LIFE qui permet la modélisation et l'exécution de comportements à l'aide du mécanisme des Behavior Trees. Dans un second temps, les scènes ont été créés grâce à l'outil dans Unity3D.



FIGURE 4 – Le séquenceur dans Unity3D – Exemple d'orchestration des comportements

4 Conclusion

Les scènes de vie permettent de réaliser des ambiances de vie dans des environnements virtuels avec des personnages autonomes exhibant des comportements crédibles, moins répétitifs et interactifs. Nous proposons un outil destiné aux game designers utilisant le concept de smart zone pour gérer automatiquement les comportements des PNJ dans les scènes. Cet outil permet de déplacer la complexité de conception d'ambiance de vie dans

chaque zone et vise à améliorer la réutilisabilité des scènes. Le prototype implémenté dans Unity3D sera sujet à évaluation auprès de game designers dans le cadre de la conception d'un jeu vidéo et par des étudiants en game design. Une expérimentation avec deux groupes d'étudiants (avec et sans l'outil) sera menée pour déterminer l'apport de l'outil dans le processus de conception du jeu. Enfin, certaines limitations doivent être encore levées comme la prise en compte du joueur et l'impact de ses actions sur les scènes ainsi qu'une réelle prise en compte des smart zones dans la décision et la réalisation des buts individuels des PNJ, leur permettant alors de raisonner sur les scènes et décider de leur participation éventuelle.

5 Remerciements

Ces travaux sont réalisés dans le cadre du projet OCTAVIA (PIA-FSN-2012) en partenariat avec BSE Group, Black Sheep Studio et le CNAM. Le projet est financé par la Direction Générale des Entreprises (DGE) et labélisé par le pôle CAP DIGITAL

Références

- BECROFT, D., BASSETT, J., MEJÍA, A., RICH, C., & SIDNER, C. (2011). AIPaint: A Sketch-Based Behavior Tree Authoring Tool. Dans *Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)* (pp. 02-07).
- BLANK, B., PASTERNAK, G., CRANE, A., & BROADBENT, A. (2011). SmartObjects and Agents for Autonomous High Population Simulated Environments. Dans *Interservice/Industry Training, Simulation, and Education Conference (IITSEC)* (pp. 1-10).
- ČERNÝ, M., PLCH, T., MARKO, M., ONDRACEK, P., & BROM, C. (2014). Smart Areas: A Modular Approach to Simulation of Daily Life in an Open World Video Game. Dans *6th International Conference on Agents and Artificial Intelligence (ICAART 2014)* (pp. 703-708).
- CUTUMISU, M., SZAFRON, D., SCHAEFFER, J., MCNAUGHTON, M., ROY, T., ONUCZKO, C., & CARONARO, M. (2006). Generating ambient behaviors in computer role-playing games. *Intelligent Systems*, 21(5), 19-27.
- FOLSOM-KOVARIK, J., SCHATZ, S., & JONES, R. (2013). Scalable Models for Patterns of Life. Dans *Innovative Applications of Artificial Intelligence Conference* (pp. 1582 - 1584).
- GEBHARD, P., KIPP, M., KLESEN, M. & RIST, T. (2003). Authoring for Adaptive, Interactive Performances. Dans *Proc. of Autonomous Agent and Multi-Agent Systems*, Melbourne, Australia.
- HECKEL, F., YOUNGBLOOD, G., & HALE, D. (2009). BehaviorShop: An Intuitive Interface for Interactive Character Design. Dans *Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*.
- MAGNUS, J. (2013). Do Non Player Characters dream of electric sheep? : A thesis about Players, NPCs, Immersion and Believability. *Stockholm University*, Phd Thesis.
- KALLMANN, M., & THALMANN, D. (1999). Modeling objects for interaction tasks. Dans *Computer Animation and Simulation* (pp. 73-86).
- PERLIN, K. & GOLDBERG, A. (1996). Improv: A System for Scripting Interactive Actors in Virtual Worlds. *Computer Graphics*, 29(3)
- PRENDINGER, H., ISHIZUKA, M (2002). SCREAM: Scripting emotion-based agent minds. Dans *proc. of Autonomous Agent and Multi-Agent Systems*, ACM Press, (pp 350-351).
- SHOULSON, A., & BADLER, N. I. (2011). Event-Centric Control for Background Agents. Dans *International Conference on Interactive Digital Storytelling, ICIDS* (pp. 193-198).
- STOCKER, C., SUN, L., HUANG, P., QIN, W., ALLBECK, J. M., & BADLER, N. I. (2010). Smart events and primed agents. Dans *Intelligent Virtual Agents* (pp. 15-27).
- YERSIN, B., MAÏM, J., PETTRÉ, J., & THALMANN, D. (2009). Crowd patches: populating large-scale virtual environments for real-time applications. Dans *Interactive 3D graphics and games* (pp. 207-214).

Deux applications des logiques non monotones : Décision à bord d'un sous marin, Représentation et découverte pour la cellule cancéreuse

Andrei Doncescu 1, Pierre Siegel 2

¹ Université Paul Sabatier – Inserm – CNRS, CRCT UMR 1037, Toulouse, France
andrei.doncescu@inserm.fr

² Aix Marseille Université - CNRS, LIF UMR 7279, 13288, Marseille, France.
pierre.siegel@univ-amu.fr

Résumé : Dans la vie réelle et pour nombre d'applications on est emmené à travailler avec des connaissances incomplètes, incertaines, révisables ou partiellement fausses. La logique classique ne permet pas de prendre en compte ces informations. Mais il faut compléter la connaissance pour prendre des décisions. L'Intelligence Artificielle a commencé à essayer de résoudre ces problèmes il y a une trentaine d'années, en particulier en étudiant des logiques non-monotones et des formalismes pour la révision. Nous présentons deux applications dans ce cadre. La première, étudiée avec la DCNS Toulon, porte sur la modélisation du comportement d'un opérateur à bord d'un sous marin en temps de guerre. La deuxième, étudiée avec l'ICR Toulouse, est l'étude des voies de signalisation dans la cellule cancéreuse. Ces applications sont liées car les techniques théoriques pour les résoudre sont semblables. D'autre part dans les deux cas la complexité algorithmique dans le pire des cas est trop importante. Pour un passage à l'échelle, on réduit cette complexité dans la pratique.

Mots-clés : Logique formelle et outils pour l'IA, Langages de programmation pour l'IA, Passage à l'échelle, Aide à la décision, Représentation des connaissances, Logique des défauts, Voies de Signalisation.

1. Introduction

Nous présentons ici deux applications de l'IA, ainsi que des techniques théoriques et pratiques pour les traiter. Ces deux applications demandent de raisonner (et aussi de décider) avec des informations incomplètes, révisables, incertaines ou même fausses. Il faut aussi gérer la complexité algorithmique pour passer à l'échelle dans la pratique.

La première application a été étudiée avec la DCNS Toulon dans le cadre de la bourse CIFRE d'Isabelle Toulgoat (Toulgoat 2011). Elle étudie la modélisation du comportement et de la décision d'un officier de quart à bord d'un sous marin (SM) en temps de guerre. Un SM est très silencieux et ne peut utiliser que des sonars passifs. Les informations connues de l'environnement viennent donc principalement de ce qui est entendu. Les distances, cap, vitesse... des navires qui l'entourent peuvent être très imprécises.

D'un autre côté l'officier peut être amené à prendre des décisions avec des règles contradictoires. Un exemple élémentaire est la situation où un SM suit un but dans son baffle. Le baffle est un angle derrière le but, où le SM n'est pas détectable à cause du bruit de l'hélice ; on a donc intérêt à y rester. D'un autre côté si le SM n'est pas nucléaire, il faut qu'il remonte régulièrement à l'air libre pour recharger ses batteries. Ici les deux règles sont contradictoires et il faut décider rapidement quelle manœuvre effectuer. Pour être simulée, cette décision peut aussi prendre en compte le caractère de l'officier : prudent, moins prudent ou même kamikaze. Ce travail a été implémenté en Prolog, interfacé avec le simulateur ATANOR de la DCNS et appliqué à un scénario impliquant deux SM ennemis.

La deuxième application, étudiée avec l'Institut Claudius Regaud de Toulouse est la modélisation des voies de signalisation dans la cellule cancéreuse. En simplifiant, une cellule est une machine à produire des protéines/gènes. Elle doit survivre et se dupliquer un nombre raisonnable de fois. Le cancer arrive quand la cellule est altérée, mais ne meurt pas et se duplique trop.

Pour un informaticien une cellule peut être représentée par quelque chose qui ressemble à un graphe orienté. Ce graphe décrit les interactions entre les gènes : activations, inhibitions, liaisons... Ici le premier problème qui se pose, formellement proche de la prise de décision dans un SM, est qu'une telle représentation peut être incohérente si on regarde l'ensemble du graphe : pour certaines configurations, un gène peut être à la fois inhibé et activé. Le deuxième problème, plus difficile, est que les interactions connues viennent d'expériences, longues et coûteuses faites par les biologistes. De fait on ne connaît qu'une petite partie des interactions et il faut essayer de compléter le graphe. En logique on parle d'abduction. Un des buts est d'essayer de savoir si une molécule (un médicament) a une chance de réparer ou de tuer la cellule cancéreuse. L'intérêt d'une telle approche in-silico est de cibler des expériences intéressantes à faire. Bien entendu, il ne s'agira que d'hypothèses et ensuite les biologistes reprennent la main en expérimentant.

Pour les deux applications, se pose donc la question de la prise en compte d'informations incomplètes ou incertaines. Cette question a commencé à être étudiée théoriquement en Intelligence Artificielle vers 1980. Une partie des études a donné lieu à la formulation de logiques non classiques, en particulier les logiques non monotones. Une logique classique est *monotone* si tout ce qui est déductible d'une information sera toujours déductible en ajoutant de l'information supplémentaire. Une logique *non monotone* perd cette propriété. Ceci permet en particulier de relaxer les règles et de trouver plusieurs solutions contradictoires : des *extensions*. On utilise ici la logique des défauts. D'un autre côté le *raisonnement abductif* va permettre de compléter l'information.

Ensuite il faut implémenter les algorithmes qui mettent en œuvre ces formalismes logiques. Dans le cas général la complexité algorithmique est beaucoup trop importante, ce qui limite à des très petits exemples. Pour pouvoir passer à l'échelle, nous avons réduit la puissance d'expression du langage en utilisant principalement des clauses de Horn. Nous avons aussi donné des algorithmes de calcul d'extension efficaces et défini un langage de représentation qui semble être une clef pour contrôler l'explosion combinatoire.

2 Modélisation de la décision d'un opérateur à bord d'un sous marin

Le but d'une simulation de combat naval est d'évaluer les performances opérationnelles de navires militaires ou de forces navales. Le groupe DCNS possède un atelier de simulation, ATANOR, qui permet de modéliser des navires et des équipements du système de combat, et de dérouler un scénario en temps simulé en faisant interagir les différents modèles.

L'objectif du travail a été de développer un système permettant de modéliser le comportement d'un opérateur dans les simulations de performances et de l'intégrer à ATANOR. Ce système devait répondre à plusieurs exigences :

- Permettre de modéliser naturellement les lois de comportement de l'opérateur.
- Permettre de raisonner avec des informations incomplètes et révisables.
- Proposer plusieurs possibilités d'action pour une même situation. Il faut aussi choisir entre ces propositions, en sélectionnant une ou plusieurs actions.
- Permettre l'ajout de nouvelles règles de comportement, sans avoir à modifier la représentation des connaissances et donc sans remettre en question les règles précédemment établies (à la différence des réseaux de Pétri, dans lesquels les modifications sont compliquées).
- Pouvoir raisonner avec des règles assez générales, sans avoir à compiler de manière très précise toutes les informations.
- Et bien entendu avoir de très bons temps de calcul, car il est nécessaire d'effectuer de nombreuses simulations avec ATANOR.

2.1 Un scénario de détection et de pistage d'un sous-marin

On modélise la décision d'un officier de quart dans un sous-marin en fonction des événements perçus sur la situation tactique. Pour modéliser dans le cadre des simulations de performance, des sous-marinières ont été interrogés et ont décrit des scénarios qui ont permis de définir des règles. Pour un scénario faisant intervenir deux SM, voici quelques exemples simplifiés de règles :

- (1) Tant que le SM n'a rien détecté, il poursuit une trajectoire aléatoire de recherche dans sa zone de patrouille : le SM avance par tronçons rectilignes successifs et effectue des abattées d'écoute (des changements de cap) à des intervalles de temps irréguliers. Cette manœuvre lui permet de vérifier qu'il n'est pas pisté. Cette règle est appliquée tant que le SM n'a pas de nouvelles informations.
- (2) Si le SM détecte un autre sous-marin, le but, il enclenche les actions suivantes : manœuvres d'anticollision, manœuvres de mesure de distance, ralliement du poste de pistage, pistage en restant dans le baffle du but.
- (3) Si le SM se fait détecter en effectuant la mesure de distance, le ralliement du poste de pistage ou le pistage, il dérobe : il place le but dans un gisement de 150° en augmentant sa vitesse, pour faire en sorte que son adversaire perde le contact.
- (4) Si le SM est à propulsion non nucléaire et que 4 heures se sont écoulées depuis la dernière recharge de batterie, alors il se met au schnorchel : pour cela, il remonte à l'immersion périscopique et si il ne détecte rien, il sort son tube à air et recharge pendant 20 minutes, avec une vitesse de 10 noeuds maximum.
- (5) Il est impossible de poursuivre une trajectoire aléatoire et de remonter.

Ces règles peuvent entrer en compétition, à un même moment, le sous-marin peut avoir besoin d'effectuer plusieurs actions contradictoires. Par exemple il peut avoir besoin de remonter au schnorchel, poursuivre l'action de pistage et d'éviter un rocher.

2.2 Modélisation logique des règles de comportement

En logique classique, les règles (1), (4) et (5) simplifiées peuvent se représenter par :

$$(1') \forall t \forall sm (TrajAleatoire(sm, t) \wedge \neg Detecte(sm, t)) \rightarrow TrajAleatoire(sm, t + 1)$$

$$(4') \forall t \forall sm (\neg Nucleaire(sm) \wedge Recharge(sm, < t - 4h)) \rightarrow Schnorchel(sm, t + 1)$$

$$(5') \forall t \forall sm \neg (TrajAlea(sm, t) \wedge Schnorchel(sm, t))$$

La place manque pour donner les règles (2) et (3). Ici t représente le temps, sm un sous marin quelconque, et \neg est la négation. Si on est à un instant t , où le SM doit poursuivre la trajectoire aléatoire et doit aussi remonter, on a une inconsistance logique. Il est impossible d'effectuer les deux actions en même temps. C'est le problème de la logique classique où les formules/règles sont considérées comme certaines et non révisables. Il faut alors relaxer ces règles. Intuitivement, on a envie de remplacer, par exemple, la règle (1) par :

*Tant que SMI n'a rien détecté, et qu'il est **possible** de poursuivre une trajectoire aléatoire, alors il poursuit la trajectoire.*

Il faut alors formaliser le **possible**. Pour ceci on utilise, la logique des défauts.

2.3 Logiques non monotones et Logique des défauts

Les logiques classiques, comme la logique du premier ordre ou les logiques modales sont monotones : si $E \vdash F$ alors $E \cup E' \vdash F$. En d'autres termes ce qui est inféré n'est pas révisable avec l'ajout de nouvelles informations. Mais dans la pratique, il est courant d'invalider des conclusions précédemment établies. Une logique non monotone le permet.

2.3.1 Logiques des défauts

La logique non monotone la plus fréquemment utilisée est la *logique des défauts* (Reiter 80). Cette logique peut être vue comme une généralisation de la négation par échec en Prolog. C'est aussi une généralisation du formalisme ASP apparu plus tard.

Une théorie des défauts est définie par un couple $\Delta = (D, W)$. L'ensemble W est un ensemble de faits/informations toujours vrais, représentés par des formules de logique du premier ordre. L'ensemble D est un ensemble de *défauts*, qui sont des règles d'inférence spécifiques exprimant l'incertitude. Un défaut est une expression de la forme :

$$d = A(X) : B(X) \vdash C(X)$$

Ici $A(X)$, $B(X)$ et $C(X)$ sont des formules et X est l'ensemble des variables libres de d . On appelle $A(X)$ le *prérequis*, $B(X)$ la *justification* et $C(X)$ le *conséquent*. Intuitivement, le défaut d signifie : si $A(X)$ est vrai, et il s'est *possible/consistant* que $B(X)$ soit vrai, alors $C(X)$ est vrai. Si $B(X) = C(X)$, le défaut est *normal*. Un défaut normal va représenter une formule du type « Généralement, si $A(X)$ est vrai alors $B(X)$ est vrai ». Un défaut est semi-normal si $B(X)$ infère $C(X)$. Ici seuls des défauts normaux ou semi-normaux sont utilisés.

2.3.2 Extensions

Le but est de trouver les *extensions* d'une théorie par défauts $\Delta = \{W, D\}$. Une extension E est un ensemble cohérent de formules obtenu en ajoutant, sous conditions, à W un ensemble maximal de conséquents de D . Les extensions vont, par exemple, représenter les manœuvres possibles à effectuer à l'instant $t+1$. La place pour définir formellement les extensions manque. Mais, pratiquement, pour le cas simple, où tous les défauts sont normaux, on part de W et on choisit un défaut $d = A : B \vdash B$ non encore utilisé. On vérifie que $\neg B$ n'est pas inféré par W . Si c'est le cas on ajoute B à W et on réitère l'opération jusqu'à ce que tous les défauts aient été utilisés ou rejetés.

Par exemple, on part de la configuration de la section 2.2 : SM n'est pas nucléaire, est en phase de recherche aléatoire qu'il est sensé poursuivre et doit remonter au Schnorchel. On aura une théorie $\Delta = \{W, D\}$ avec W qui donne l'état courant au temps t et $D = \{d1, d2\}$

$$W = \{ \neg Nucleaire(SM), \neg Detecte(SM, t), Schnorchel(SM, t), TrajAlea(SM, t), \\ \forall t \forall sm \neg (TrajAlea(sm, t) \wedge Schnorchel(sm, t)) \}$$

$$d1 = ((TrajAlea(SM, t)) \wedge \neg Detecte(SM, t)) : TrajAlea \vdash TrajAlea(SM, t + 1)$$

$$d2 = \neg Nucleaire(SM) : Schnorchel(SM, t + 1) \vdash Schnorchel(SM, t + 1)$$

On remarque que W contient la règle (5) qui dit que *TrajAlea* et *Schnorchel* sont mutuellement exclusifs. A cause de cette règle, il est alors impossible d'utiliser en même temps $d1$ et $d2$ pour construire une extension. Si $d1$ est utilisé, *TrajAlea*($SM, t + 1$) est ajouté à l'extension et l'exclusion mutuelle interdit d'utiliser $d2$ pour ajouter *Schnorchel*. De même si $d2$ est utilisé on n'utilise pas $d1$. Nous obtenons deux extensions (solutions) contradictoires l'une contient *TrajAlea*($SM, t + 1$) et l'autre *Schnorchel*($SM, t + 1$).

Il reste à choisir l'extension qui va être retenue pour la consigne de déplacement du SM . Pour l'exemple réduit le choix est facile, le SM remonte au Schnorchel. Dans l'application réelle il y a d'autres règles qui disent, que si on détecte un autre bateau il ne faut pas se faire détecter, et que la remontée est très bruyante. Le choix est alors moins évident. Les choix ne sont pas donnés par le système logique pur et la décision se fait en pondérant les extensions. Pour l'application, des fonctions de choix ont été définies en associant à chaque défaut un poids qui traduit son importance. Par exemple une manœuvre

d'anticollision sera fortement pondérée. Sans entrer dans les détails, à partir de ces poids on pondère les extensions. Et on choisit la meilleure qui donne la consigne à appliquer.

2.3.3 Interfaçage avec la plateforme ATANOR et résultats

Une interface a été réalisée entre notre programme et l'atelier de simulation ATANOR. ATANOR envoie au programme des informations qui sont :

- des valeurs certaines (cap, vitesse, immersion, position...) sur notre sous-marin SM1 auquel on applique les règles de comportements.
- des valeurs approximatives sur le SM2 but : détection, position...

Ensuite les extensions sont calculées, l'une d'entre elles est choisie, on renvoie les consignes de cap, vitesse et immersion à ATANOR. Puis on recommence.

La figure 1 donne un exemple d'exécution. La ligne brisée donne la trajectoire de SM1, décidée par le programme. La ligne droite est la trajectoire réelle de SM2 (non connue de SM1) déterminée par ATANOR. Le nuage de points est la trajectoire estimée de SM2 par SM1. SM1 transite dans la zone et ce transit est modélisé avec des tronçons, dont le cap est sélectionné de façon à garder un cap moyen. Dans ce scénario, SM1 exécute dans un premier temps les abattées d'écoutes tant qu'il n'a rien détecté. Lorsqu'il détecte le but, il exécute l'enchaînement des actions suivant : anticollision, mesure de distance, ralliement du poste de pistage, pistage. La trajectoire estimée, est au début éloignée de la trajectoire réelle du but. La manœuvre de mesure de distance permet d'obtenir une meilleure estimation de cette trajectoire, qui se confond à la fin avec la trajectoire réelle.

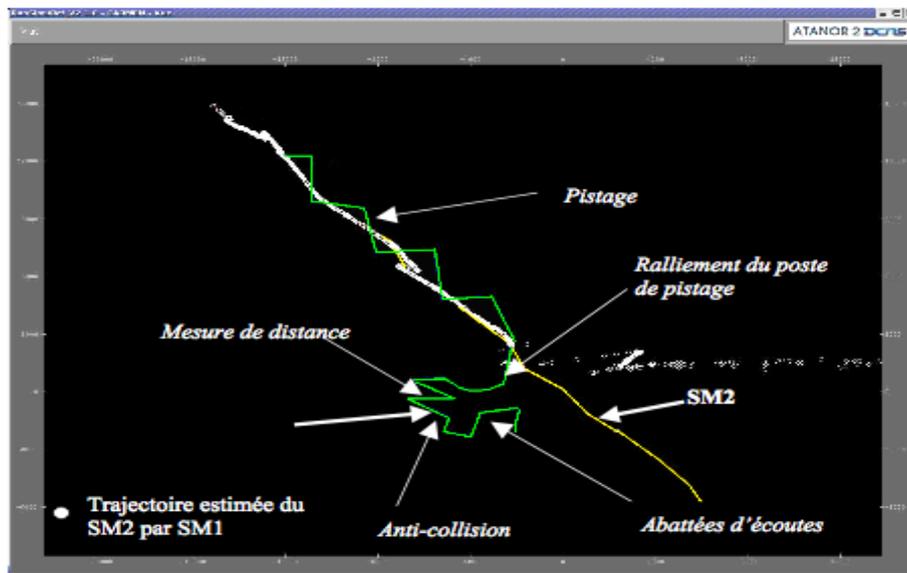


Figure 1.

L'exécution représentait environ trois heures de navigation et demandait moins d'une seconde de calcul avec un algorithme, peu optimisé, écrit en Prolog. Ce temps de calcul représentait environ le quart du temps demandé pour les seuls calculs d'ATANOR. Les simulations ont été validées par des sous-marinières, qui ont retrouvé leurs décisions.

3. Représentation des voies de signalisations de la cellule cancéreuse

Pour cette application le but est de modéliser des interactions dans la cellule cancéreuse. Ce travail est basé sur un exemple réel : les données bibliographiques de la réponse à DSB représentées par une carte d'interaction de Pommier et al. (Pommier 2005, 2006) Fig. 2.

Un des problèmes de cette représentation qu'il est très difficile d'ajouter une nouvelle interaction ou protéine sans une réévaluation complète.

3.1 Un petit exemple

La figure 3 donne un exemple d'école simplifié. Elle montre des interactions dans une cellule entre le cancer, les ultraviolets et des gènes.

- La flèche \rightarrow représente une activation. Par exemple $UV \rightarrow cancer$ signifie que les ultraviolets provoquent le cancer. Et $\rightarrow p53$ dit que p53 est activé.
- L'inhibition est représentée par le symbole \neg . Par exemple $A \neg Cancer$.
- Deux gènes peuvent se lier pour créer un composé. Ici P53 et Mdm2 se lient pour créer un composé qui active B.

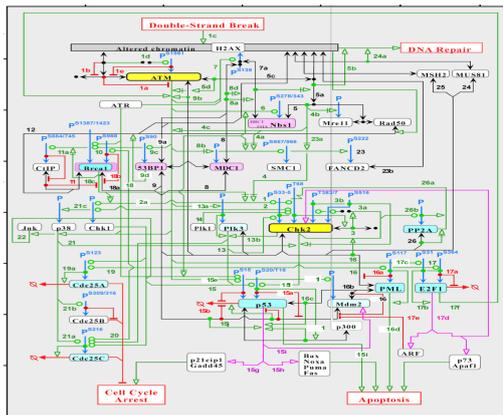


FIGURE 2 - Carte de Pommier.

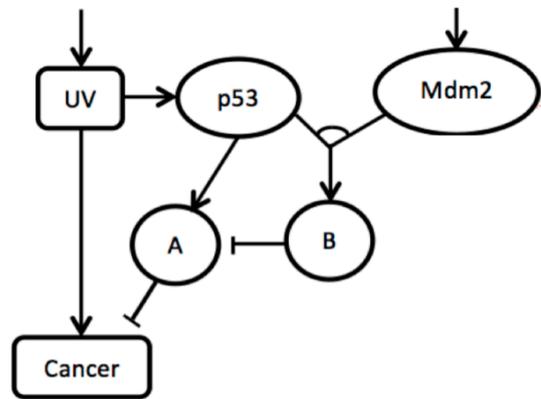


FIGURE 3 – Un exemple d'école.

Le graphe causal Fig. 3, montre par le biais de différents mécanismes, non représentés ici, que les ultraviolets donnent le cancer ($UV \rightarrow cancer$). D'autre part UV active la protéine P53 ($UV \rightarrow P53$) et P53 active la protéine A ($P53 \rightarrow A$). On a aussi que B inhibe la protéine A ($B \neg A$) et que A bloque le cancer ($A \neg cancer$). En outre P53 et Mdm2 se lient et le complexe obtenu active B, et B inhibe A.

Cette explication est incohérente. En effet si il y a des UV, le cancer est déclenché. Mais les UV activent aussi P53 qui active A et A inhibe le cancer. Le cancer est alors, à la fois, déclenché et inhibé, ce qui est incompatible. Le problème se complique si Mdm2 est présent car P53 et Mdm2 vont se lier et le complexe obtenu active B. Et B bloque A.

Pour lever les incohérences, on utilise la logique des défauts et on obtient des extensions. Chaque extension est une solution d'activations/inhibitions cohérente.

Evidemment l'exemple est trop simple, il sert juste à montrer les problèmes de base et des solutions. Pour des systèmes biologiques réels, contenant de nombreuses voies avec des milliers de gènes, il est clair qu'il faut gérer la complexité algorithmique.

3.2. Formalisme et représentation

La description formelle de l'aspect dynamique des interactions des protéines est basée sur le langage de la logique du premier ordre. Par exemple $stimulation(UV,ADN)$ dit que l'ADN est soumis à des ultraviolets. Et $verre \rightarrow \neg stimulation(UV)$ dit que le verre protège des ultraviolets. On peut aussi pondérer les règles, introduire le temps, donner des valeurs de concentration des protéines ou des seuils. Avec la logique des défauts, il est aussi possible de dire quels sont les faits certains, représentés par des formules de logique du premier ordre et quelles sont les règles révisables, représentées par des défauts.

La logique est un langage universel et l'avantage d'un cadre logique est la capacité de tout représenter de manière naturelle. Mais pour passer à l'échelle et contrôler l'explosion combinatoire, il faut limiter la puissance d'expression : pour les règles certaines on va se restreindre aux *clauses de Horn*

3.2 Clauses de Horn et algorithmes efficaces

Une *clause de Horn* est une disjonction de littéraux contenant au plus un littéral positif. C'est aussi une formule du type $P_1 \wedge \dots \wedge P_n \rightarrow Q$ ou les P_i sont des littéraux positifs. Le premier avantage des clauses de Horn est d'être une manière naturelle et lisible de donner des règles. L'autre avantage, fondamental ici, est que pour le problème SAT les algorithmes sont exponentiels dans le pire des cas. Mais, avec des clauses de Horn on a des algorithmes linéaires pour la taille de la donnée. Dans notre application les règles certaines sont donc des clauses de Horn. Pour les défauts ce sont des clauses quelconques.

Le choix des clauses de Horn pour les règles dures peut paraître brutal car elles ne représentent pas les informations disjonctives positives telles que $P \vee Q$. Mais ces disjonctions sont rares dans la pratique. Pour l'application nous n'en avons pas eu besoin. Pour les SM la seule utilisation est de choisir une valeur (un cap, une vitesse...) parmi un ensemble de valeurs. Mais dans ce cas on peut utiliser un traitement spécifique. Si on doit vraiment quitter le cadre des clauses des Horn, il est aussi possible d'utiliser des techniques très efficaces qui viennent de la programmation par contraintes et de la résolution pratique de problèmes NP-complets : par exemple le renommage, les cardinalités, les symétries (Benhamou et Siegel, 2010) ou *les strong backdoors* (Ostrowski et al.2010)...

3.3 Comment compléter les voies de signalisation avec l'abduction

On part de la figure 3 et on ajoute une protéine C qui, quand elle est activée inhibe B. On a aussi X dont le rôle est inconnu. On pose la question : *Est ce que X peut servir à bloquer le cancer en activant B ?* Le but est de savoir si un futur médicament pourrait éventuellement agir efficacement. Les flèches en pointillées de la figure 4 donnent une solution analytique : Mdm2 active X et X se lie avec Mdm2 pour donner un composé qui bloque C. Ce raisonnement simple, reflète une approche biologique de découverte d'hypothèses (Inoue, Doncescu et Nabeshima, 2013). Pour des cas réels pouvant contenir plusieurs milliers de gènes ce processus de découverte doit être automatisé.

La découverte d'interactions manquantes est un problème *d'abduction*. La logique utilise la déduction $F \vdash R$ qui dit que R est inféré par F. L'abduction généralise la déduction. L'information F est incomplète et l'abduction revient à ajouter à F un ensemble cohérent H d'hypothèses qui permet d'inférer R : $F \wedge H \vdash R$ et $F \wedge H$ est consistant

L'abduction est fondamentale en IA. La difficulté réside dans la mise en œuvre des algorithmes. Même limité au calcul propositionnel, la complexité théorique est en Σ_2^P ce qui serait inacceptable pour les problèmes réels. Mais pour des applications réelles, l'expérience montre que l'on peut obtenir des algorithmes très efficaces.

L'abduction va servir à rechercher des interactions manquantes. Pour ceci on utilise la logique des défauts. Si on cherche à savoir ce qui peut activer X, on ajoute un défaut sans prérequis $d = : active(\alpha, X) \vdash active(\alpha, X)$. Ici α est une variable qui peut être instanciée par n'importe quelle protéine. Ceci peut être généralisé à tous les prédicats.

Avec cette technique en calculant les extensions qui contiennent *bloque(cancer)* on obtient 8 extensions. Deux seulement Fig. 4 et Fig. 5 sont intéressantes. Les flèches en pointillé donnent les liens trouvés automatiquement par abduction.

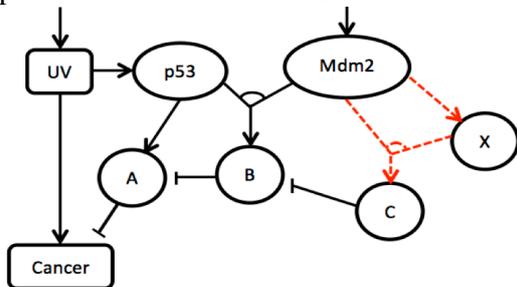


Figure 4

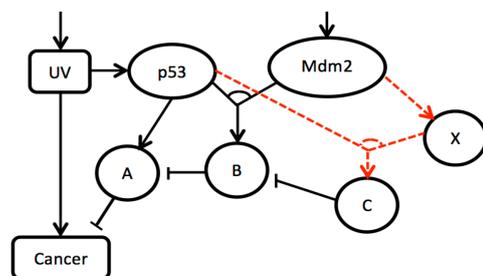


Figure 5

3.4 Résultats

Tout ceci a été appliqué à la carte de Pommier Fig. 1. Cette carte a été traduite en 206 règles dans notre formalisme. Les règles peuvent être dures ou par défaut et il est facile de changer leur nature pour tester différentes configurations. Des premiers résultats de découverte ont été obtenus. Par exemple (Pommier, 2006) simplifie la carte initiale à la main. Nous avons pu simplifier cette carte initiale automatiquement. Les deux cartes simplifiées se ressemblent beaucoup. Les différences portent sur quelques gènes, mais elles suggèrent qu'une interaction entre deux gènes jouent un rôle critique dans la tumorigenèse.

L'algorithme de calcul d'extensions, est écrit en SWI Prolog. Comme les clauses sont de Horn et que les défauts sont normaux, l'arbre de recherche est optimisé. Il est facile de calculer les extensions sans doublet dans un temps très court. Par exemple, pour la carte complète de Pommier, avec la plupart des règles par défaut, il y a deux extensions. Le calcul se fait en 500 000 LIPS et 0,4 secondes de temps CPU sur MacBook (Doncescu, Le et Siegel, 2012a,2012b,2013,2014a,2014b). Ceci renforce notre opinion qu'il est possible d'utiliser une logique non monotone et de faire de l'abduction sur des applications réelles de taille importante.

D'autres formalismes ont été testés, en particulier des techniques basées sur la logique modale, sur SAT, sur SOLAR (Doncescu et al. 2007a, 2007b) et sur les « équations différentielles ». Les résultats sont mitigés, ici aussi la place manque pour être plus précis.

References

- Benhamou, B., Nabhani, T., Siegel, P. (2010) *Reasoning by symmetry in non-monotonic logics*. Proc. 13th international workshop on Non-Monotonic Reasoning, NMR 2010, Toronto, Canada, May 2010.
- Doncescu, A., Yamamoto, Y., Inoue, K. (2007a) *Biological systems analysis using Inductive Logic Programming*. Proc. of the 21st International Conference on Advanced Information Networking and Applications (AINA 2007), pages 690-695, IEEE Computer Society, 2007.
- Doncescu, A., Inoue, K., Yamamoto, Y. (2007b) *Knowledge-based discovery in systems biology using CF-induction*. *New Trends in Applied Artificial Intelligence*. Proc. 20th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA / AIE 2007), Lecture Notes in Artificial Intelligence, volume 4570, pages 395-404, Springer, 2007.
- Doncescu, A., Le, T., Siegel, P. (2012a) Default Logic for Diagnostic of Discrete Time Systems. Proc. 8th BWCCAb013 - 8th pages 488-493, Compiègne, France, Oct 2012.
- Doncescu, A., Siegel, P. (2012-b) The Logic of Hypothesis Generation in Kinetic Modeling of System Biology, Proc.23rd ICTAI, IEEE, pages 927-929, Boca Raton, Florida, USA, Nov. 2012
- Doncescu, A., Le, T., Siegel, P. (2013) Utilization of Default Logic for Analyzing a Metabolic System in Discrete Time. Proc.13th ICCSA 2013, p. 130-136, Ho Chi Min, Vietnam, June 2013.
- Doncescu, A., Siegel, P., Le T. (2014a) Relevance of Information in Cell Signaling Pathways Using Default Logic. Proc. BIOCAMP'14, , pages 16-b2, ISBN : 1-60132-265-8.
- Doncescu, A., Siegel, P., Le T. (2014b) : Representation and Efficient Algorithms for the Study of Cell Signaling Pathways Proc ICAI'14, vol 1, pages 504-510, 21-24 July 2014, ISBN : 1-60132-274-7.
- Ostrowski, R., Paris, L., Sais, L., Siegel, P. (2010) *Computing Horn Strong Backdoor Sets Thanks to Local Search*. Proc Proc. ICTAI'06, p. 139-143, IEEE Computer Society, Washington D.C., US, nov. 2006
- Pommier, Y. et al. (2005) *Targeting Chk 2Kinase : Molecular Interaction Map and Therapeutic Rationale*. *Current pharmacy design*, 11(22) : 2855- 72, 2005.
- Pommier, T. et al. (2006) *Chk2 Molecular Interaction Map and Rationale for Chk2 Inhibitors* Clin Cancer Res. 2006 May1; 12(9) : 2657-61.
- Reiter, R. (1980) *A Logic for Default Reasoning*. Art.Int.13(1-2) : 81-132 ,1980.
- Toulgoat I. (2011) *Modélisation de combat humain dans les simulations de combat naval*. Thèse, Toulon, Janvier 2011.
- Toulgoat I, Siegel, Lacroix Y, Botto J. (2010a) *Operator Decision in naval Action's Simulation*", 13th international workshop on Non-Monotonic Reasoning, NMR'10, Sydney, May 2010.
- Toulgoat I, Siegel P., Lacroix Y., Botto J. (2010b) *Operator decision modeling : application to a scenario involving two submarines*, COMPIT10 Gubbio, Italy, 12-14 April 2010.

Une approche Bayésienne pour l'apprentissage de règles séquentielles de classification

Elias Egho¹, Dominique Gay¹, Marc Boullé¹, Nicolas Voisine¹, Fabrice Clérot¹

ORANGE LABS, Lannion, France
 prenom.nom@orange.com

Résumé : Nous proposons un nouveau cadre pour extraire des règles séquentielles de classification. Après avoir formalisé l'espace de règles à explorer, nous suggérons un nouveau critère Bayésien pour évaluer l'intérêt des règles séquentielles et un algorithme d'extraction efficace et libre de tout paramètre utilisateur. Les expériences montrent que (i) le nouveau critère identifie les règles intéressantes et robustes ; (ii) l'utilisation des règles extraites comme nouveaux descripteurs de base de données permet d'atteindre de meilleures performances prédictives que les méthodes concurrentes basées sur les règles séquentielles de classification.

Mots-clés : Fouille de données séquentielles, Règles de Classification

1 Introduction

La classification supervisée de données séquentielles (Xing *et al.*, 2010) est pertinente pour un large spectre de domaines d'applications, pour la biologie (Deshpande & Karypis, 2002), l'analyse de textes (Sebastiani, 2002) ou encore l'analyse de données web (Tan & Kumar, 2002). La classification supervisée basée sur la fouille de règles séquentielles est très populaire en raison de leur interprétabilité. Une règle séquentielle est une expression de la forme $\pi : s \rightarrow c_i$ où s est une séquence, prémisse de la règle, et c_i une valeur d'attribut classe. π peut s'interpréter comme : “*lorsqu'un objet respecte la séquence d'évènements s alors c'est souvent un objet de classe c_i* ”. Ainsi, un nouvel objet entrant qui supporte π se verra affecté la classe c_i .

La plupart des Classifieurs Basés sur les Règles Séquentielles (CBRS) adoptent une stratégie similaire : d'abord, un ensemble de règles est extrait en fonction d'une mesure d'intérêt, puis, soit un classifieur dédié, e.g., une liste de décisions, est construit à partir d'un sous-ensemble de règles (Tseng & Lee, 2005; Zhou *et al.*, 2013), soit les règles sont directement utilisées comme nouveaux descripteurs dans un processus de classification supervisée (Lesh *et al.*, 1999; Deng & Zaïane, 2010; Lam *et al.*, 2012). Si les approches existantes ont généralement de bonnes performances prédictives, on peut toutefois mettre en évidence deux de leurs faiblesses :

L'instabilité des mesures d'intérêt. Considérons deux mesures communément utilisées pour l'évaluation de règles de classification, la confiance (Lesh *et al.*, 1999; Tseng & Lee, 2005) et le taux d'accroissement (Growth Rate : GR) (Ji *et al.*, 2005; Deng & Zaïane, 2010). Sur la figure 1, nous montrons que les règles extraites en fonction de ces mesures ne sont pas individuellement

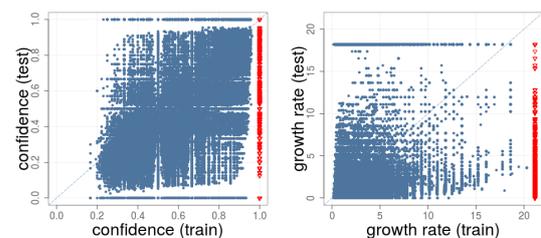


FIGURE 1 – Comparaison des valeurs de confiance et GR des règles séquentielles de classification dans une expérience 50%-apprentissage/ 50%-test pour les données skater. ∇ pour le top-1000 des règles en fonction de leur valeur de confiance, resp. GR, sur les données d'apprentissage.

robustes : ici, pour la base de données skater (Mörchen & Ultsch, 2007), chaque point représente une règle, son abscisse est sa valeur pour la mesure sur la base de données d'apprentissage et son ordonnée sa valeur en test. Nous observons un nuage de points très diffus dans le repère apprentissage/test, ce qui indique que les valeurs de confiance et de GR sont très instables des données d'apprentissage aux données test.

La malédiction du paramétrage. la plupart des approches existantes nécessitent l'ajustement de paramètres : un seuil de fréquence minimum et/ou un seuil pour la mesure d'intérêt lors de la phase d'extraction ; puis le nombre de règles à utiliser dans l'ensemble final utilisé. Malheureusement, le paramétrage est crucial puisque les performances des classifieurs en est fortement dépendant (Coenen & Leng, 2007) mais il n'est pas tâche facile – en particulier chaque domaine d'application peut nécessiter son propre paramétrage.

Dans cet article, nous proposons une solution à ces deux problèmes en suggérant (i) un nouveau critère robuste pour l'évaluation des règles, (ii) un algorithme d'extraction de règles efficace et libre de tout paramètre utilisateur. L'évaluation empirique en section 4 démontre la pertinence de notre approche : le critère permet d'identifier les règles intéressantes et robustes, et l'utilisation des règles extraites comme nouveaux descripteurs nous permet d'obtenir de meilleures performances prédictives que l'état de l'art des CBRS.

2 Contexte & Définitions

Soit $\mathcal{I} = \{e_1, e_2, \dots, e_m\}$ un ensemble fini de m items distincts. Une **séquence** s sur \mathcal{I} est une liste ordonnée $s = \langle s_1, \dots, s_{\ell_s} \rangle$ où $s_i \in \mathcal{I}$; ($1 \leq i \leq \ell_s, \ell_s \in \mathbb{N}$). Une séquence $s' = \langle s'_1 \dots s'_{\ell_{s'}} \rangle$ est une **sous-séquence** de $s = \langle s_1 \dots s_{\ell_s} \rangle$, qu'on note $s' \preceq s$, si il existe des indices $1 \leq i_1 < i_2 < \dots < i_{\ell_{s'}} \leq \ell_s$ tels que $s'_z = s_{i_z}$ pour tout $z = 1 \dots \ell_{s'}$ et $\ell_{s'} \leq \ell_s$. On dit que s est une **super-séquence** de s' . $\mathbb{T}(\mathcal{I})$ représente l'ensemble de toutes les séquences possibles sur \mathcal{I} . Soit $\mathcal{C} = \{c_1, \dots, c_j\}$ un ensemble fini fini de j classes distinctes. Une **base de données séquentielles labellisées** \mathcal{D} est un ensemble fini de n triplets (sid, s, c) , où sid est un identifiant de séquence, s une séquence ($s \in \mathbb{T}(\mathcal{I})$) et c une valeur de classe ($c \in \mathcal{C}$). L'ensemble $\mathcal{D}_{c_i} \subseteq \mathcal{D}$ contient les n_{c_i} séquences de même classe c_i . Le **support** ou la **fréquence** d'une séquence s dans \mathcal{D} , noté $f(s)$, est défini par $f(s) = |\{(sid', s', c') \in \mathcal{D} | s \preceq s'\}|$. Nous dénotons la valeur $n - f(s)$ par $\bar{f}(s)$; et le support de s dans \mathcal{D}_c est noté $f_c(s)$ alors que $\bar{f}_c(s)$ vaut $n_c - f_c(s)$.

Definition 1 (Modèle de règle séquentielle de classification (MRSC))

Soit \mathcal{D} une base de données séquentielles labellisées à j classes. Une règle séquentielle de classification π est une expression de la forme : $\pi : s \rightarrow f_{c_1}(s), f_{c_2}(s), \dots, f_{c_j}(s)$ où s est une séquence, appelée prémisses de la règle et $f_{c_i}(s)$ est le support de s dans chaque \mathcal{D}_{c_i} , $i = 1 \dots j$. L'espace des modèles à explorer est l'ensemble de tous les MRSCs ainsi définies. Chacune d'entre elles est caractérisée par la hiérarchie de paramètres suivante : les items qui constituent la prémisses de la règle, l'ordre d'occurrence des items dans la prémisses et la distribution des classes dans la prémisses (dans le corps) et hors de la prémisses (hors du corps).

Un cadre Bayésien pour l'évaluation de motifs séquentiels. Considérant le cadre Bayésien introduit par Boullé (2006), le problème de fouille de règles peut être reformulé comme un problème de sélection de modèles. Afin de choisir le "meilleur" modèle de règle dans l'espace

de modèle, nous utilisons une approche Bayésienne, dite Maximum A Posteriori (MAP) : nous cherchons à maximiser $p(\pi|\mathcal{D})$, la probabilité a posteriori d'une règle π connaissant les données \mathcal{D} . Considérant la règle Bayes, $p(\pi|\mathcal{D})$ prend la forme suivante :

$$p(\pi|\mathcal{D}) = \frac{p(\pi, \mathcal{D})}{p(\mathcal{D})} = \frac{p(\pi) \times p(\mathcal{D}|\pi)}{p(\mathcal{D})}$$

Puisque $p(\mathcal{D})$ est constant dans le problème d'optimisation courant, cela revient à maximiser l'expression $p(\pi) \times p(\mathcal{D}|\pi)$. Le critère d'évaluation, appelé *cost*, est défini par le logarithme négatif de $p(\pi|\mathcal{D})$ et s'exprime comme suit :

$$cost(\pi) = -\log(\underbrace{p(\pi)}_{\text{prior}} \times \underbrace{p(\mathcal{D}|\pi)}_{\text{vraisemblance}}) \propto -\log(\underbrace{p(\pi|\mathcal{D})}_{\text{posterior}})$$

En considérant (i) un a priori hiérarchique sur les paramètres d'une règle (comme indiqué dans sa définition) pour $p(\pi)$ et (ii) une distribution multinomiale pour la probabilité des données connaissant le modèle de règle ($p(\mathcal{D}|\pi)$), il est possible d'obtenir une expression exacte de la fonction *cost*. Le manque de place ne nous permet pas de détailler chaque terme de la fonction *cost*, toutefois, notons que ce critère s'obtient de manière similaire pour les règles de classification dans les données transactionnelles (Gay & Boullé, 2012). L'expression complète et exacte de la fonction de *cost* pour une règle est alors :

$$\begin{aligned} cost(\pi) = & \log(m+1) + \log(\ell_{max}+1) + \log\left(\frac{m^{k_s}}{k_s!}\right) + \log(k_s^{\ell_s}) \\ & + \log\left(\binom{f(s)+j-1}{j-1}\right) + \log\left(\binom{\overline{f(s)}+j-1}{j-1}\right) \\ & + \log(f(s)!) - \sum_{i=1}^j \log(f_{c_i}(s)!) + \log(\overline{f(s)}!) - \sum_{i=1}^j \log(\overline{f_{c_i}(s)}!) \end{aligned}$$

où ℓ_s est le nombre d'items dans la séquence s , k_s le nombre d'items distincts de la séquence s , ($k_s \leq \ell_s$) et ℓ_{max} le nombre items de la plus longue séquence de \mathcal{D} .

Comme l'amplitude des valeurs de *cost* dépend du nombre n de séquences et du nombre m d'items dans les données, nous suggérons un critère normalisé, appelé *level*, qui sera notre mesure d'intérêt pour évaluer et comparer des règles.

Definition 2 (Level)

Le *level* d'une règle π est défini par : $level(\pi) = 1 - \frac{cost(\pi)}{cost(\pi_\emptyset)}$ où π_\emptyset est le modèle de règle nul (i.e., la règle par défaut avec une prémisse vide).

Le *level* permet d'identifier la frontière dans l'espace de règles entre les règles pertinentes et celles qui ne le sont pas. En effet, les règles π telles $level(\pi) \leq 0$ sont moins probables que la règle nulle π_\emptyset . Leur utilisation pour caractériser les classes est plus coûteux, en terme de *cost*, que π_\emptyset : de telles règles sont considérées comme non-pertinentes. Au contraire, les règles telles que $0 < level(\pi) \leq 1$ constituent les motifs intéressants. En fait, les règles à fort *level* (ou faible *cost*) sont les plus probables connaissant les données et indiquent qu'il y a une corrélation entre la prémisse de la règle et l'attribut classe.

3 Stratégie et algorithme d'extraction de règles séquentielles de classification

L'extraction de motifs séquentiels (Agrawal & Srikant, 1995) est un problème NP-difficile. Bien que les propriétés du seuil minimum de fréquence et des représentations condensées (Manila & Toivonen, 1996) permettent de réduire le temps de calcul, le problème reste difficile pour les grandes bases de données. Comme le *level* ne possède pas de propriétés (comme celles de la fréquence) pour faciliter l'extraction de motifs, une exploration exhaustive pour obtenir les règles à *level* positif est inenvisageable. En effet, la taille de l'espace de recherche est exponentiel en m , le nombre d'items : $\sum_{i=1}^{\ell_{max}} m^i \equiv O(m^{\ell_{max}})$. Pour cette première approche, nous préférons donc une formulation plus simple et réaliste du problème, à savoir "Extraire avec diversité un sous-ensemble de règles à *level* positifs".

Notre algorithme d'extraction de règles, appelé **MiSeRe** (pour *Mining Sequential Classification Rules*), est libre de tout paramètre utilisateur, il emploie une stratégie aléatoire de recherche basée sur les objets séquences qui promeut la diversité et enfin il est dit "any-time" – plus on lui accorde de temps de calcul, plus il apprend. La principale difficulté est de calculer efficacement les distributions de classes $f_{c_1}(s), \dots, f_{c_j}(s)$. Pour ce faire, nous utilisons la représentation en BitSet des séquences et des opérations Booléennes présentées par Ayres *et al.* (2002). Ainsi, extraire une règle est effectué en $O(\ell_s \times n \times \log(n))$.

Algorithme 1 : MiSeRe

```

input :  $\mathcal{D}$ , une base de données séquentielles labellisées
output :  $\mathcal{R}$ , un ensemble de règles à level positif
1  $\mathcal{S} = \{s = \langle s_1 \rangle; s_1 \in \mathcal{I}\}$ ;
2  $\mathcal{R} = \{\pi : s \rightarrow f_{c_1}(s), \dots, f_{c_j}(s); s \in \mathcal{S} \wedge level(\pi) > 0\}$ ;
3 while  $\neg TempsLimite$  do
4    $s = ChoisirSéquenceAléatoire(\mathcal{D})$ ;
5    $ads = CalculerNombreSousSéquences(s)$ ;
6   for  $i = 1$  to  $\log(ads)$  do
7      $s' = GénérerSousSéquencesAléatoires(s)$ ;
8      $\pi : s' \rightarrow f_{c_1}(s'), \dots, f_{c_j}(s')$ ;
9     if  $level(\pi) > 0 \wedge \pi \notin \mathcal{R}$  then
10       $\mathcal{R} = \mathcal{R} \cup \{\pi\}$ 
11 return  $\mathcal{R}$ ;
```

Procédure de classification. Nous suggérons d'utiliser les règles extraites par MiSeRe comme nouveaux descripteurs pour recoder la base de données séquentielles initiale \mathcal{D} en une base de données transactionnelles binaires. Un nouvel attribut binaire est créé pour chaque règle π et prend la valeur 1 pour un objet (sid, s, c) si s est une super-séquence de la prémisse de π et 0 sinon. Cette procédure présente deux avantages : (i), il existe un grand nombre de classifieurs disponibles applicables à ces données recodées ; (ii), dans certains cas d'étude, les séquences ne sont qu'une partie de la description des objets à classer en plus d'attributs numériques ou catégoriels classiques : ainsi, remplacer la partie séquentielle des données par des attributs binaires pertinents enrichit les données avant l'utilisation d'un classifieur.

4 Validation expérimentale

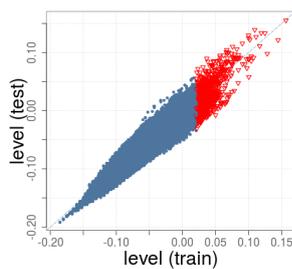
Nous présentons un résumé des résultats de l'évaluation empirique de notre approche. Une visualisation détaillée et interactive des résultats ainsi que le code JAVA de MiSeRe sont disponibles sur le site qui accompagne ce travail (Egho *et al.*, 2015) : <http://misere.co.nf>. Les expériences sont destinées à nourrir la discussion autour des points suivants : la stabilité du *level*, les performances prédictives de notre procédure de classification, la comparaison des performances prédictives de MiSeRe avec l'état de l'art des CBRS.

Pour l'évaluation empirique, nous utilisons 11 bases de données réelles communément utilisées par la littérature en classification de séquences. Ces données présentent une large variété en termes de nombre de séquences, d'items, de longueur de séquences et de classes et sont issues de divers domaines d'applications (voir Table 1).

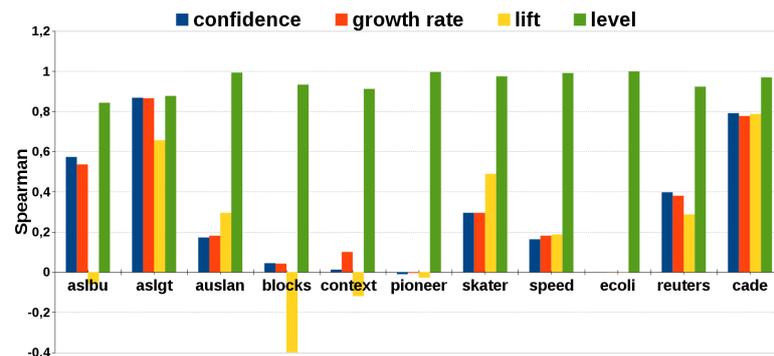
Stabilité du critère *level*. Pour évaluer la stabilité d'une mesure d'intérêt, nous menons des expériences apprentissage / test : chaque base est divisé en deux parties égales, 50% pour l'apprentissage en fonction de la mesure d'intérêt et 50% pour le test pour évaluer la mesure. Pour chaque règle extraite les valeurs de mesure en apprentissage et en test sont comparées. Nous extrayons les motifs séquentiels fréquents de la base d'apprentissage avec cSPADE (Zaki, 2000) (seuil de fréquence minimum de 2% et gap maximum entre items de 2). Des règles sont construites à partir de ces motifs et nous calculons les valeurs de *level*, confiance, GR et lift de ces règles en apprentissage et en test. Remarquons qu'en figure 1, relatif à l'exemple d'introduction, les valeurs de *level* des règles extraites sont sensiblement plus stables que la confiance et GR. Les mêmes observations tiennent pour les autres bases de données. Pour une vue globale de la stabilité des mesures étudiées, nous étudions la concordance des rangs des valeurs des mesures dans une expérience apprentissage/test. Pour chaque base, nous ordonnons les règles en fonction de leur valeur de mesure en apprentissage puis en test. Ensuite, la concordance des rangs d'apprentissage et de test est analysée au moyen du coefficient de corrélation de Spearman (Myers & Well, 2003). La figure 2b montre que les rangs de valeurs *level* en apprentissage et en test sont très corrélés (valeur de coefficient proche de 1) ; le *level* est donc stable alors que les autres mesures sont très instables (faible corrélation des rangs en apprentissage/test).

Données	# Séquences	# Items	Longueur Max	# Classes
<i>aslbu</i>	441	140	58	7
<i>aslgt</i>	3493	47	197	40
<i>auslan</i>	200	12	24	10
<i>blocks</i>	210	8	24	8
<i>context</i>	240	54	272	5
<i>pioneer</i>	160	92	127	3
<i>skater</i>	530	41	260	6
<i>speed</i>	530	41	260	7
<i>ecoli</i>	106	16	28	2
<i>reuters</i>	5459	14529	533	8
<i>cade</i>	15000	111766	19763	12

TABLE 1 – Description des données expérimentales.



(a) Résultat de l'expérience apprentissage/test pour le *level* sur les données de la figure 1 en introduction.



(b) Analyse de la concordance des rangs par le coefficient de Spearman dans une expérience apprentissage/test.

FIGURE 2 – Etude comparative de la stabilité des critères *level*, confiance, GR et lift.

Pour confirmer la robustesse du *level*, nous menons l'expérience suivante : pour chaque base de données, nous assignons une classe $c \in \mathcal{C}$ de manière aléatoire à chaque objet, puis nous

laissons tourner MiSeRe pendant 30 minutes : aucune règle n'est extraite puisque qu'elles ont toutes un *level* négatif. Au contraire, pour la moitié des bases disponibles, certaines règles à fort GR, confiance ou lift sont extraites alors qu'il n'y a rien à découvrir. Ainsi, le *level* est bien une mesure robuste, évite les motifs non-pertinents et le sur-apprentissage.

Performance prédictive de notre approche. Pour évaluer les performances prédictives de notre approche, nous utilisons plusieurs classifieurs standards sur les données recodées avec les règles obtenues avec MiSeRe : le Bayésien naïf (NB), la forêt aléatoire (RF), l'arbre de décision (C4.5 alias J48), une machine à vecteurs supports (SVM), deux approches de k plus proches voisins k Star et IBk accessibles avec WEKA (Hall *et al.*, 2009) – que nous utilisons avec les paramètres par défaut. Nous utilisons aussi le Bayésien naïf sélectif¹ (SNB) de Boullé (2007). Les résultats de précision sont obtenus par validation croisée stratifiée en 10 parties : bien sûr MiSeRe n'exploite que la partie apprentissage. Bien que MiSeRe soit de type “anytime”, pour ces expériences, nous faisons le choix (que nous motivons par la suite) d'extraire 2^{10} , i.e., 1024 règles. Les résultats détaillés de précision moyenne sont disponibles dans (Egho *et al.*, 2015). Pour comparer les performances de plusieurs classifieurs sur plusieurs bases de données nous utilisons le test de Friedman couplé au test post-hoc de Nemenyi (au seuil de significativité $\alpha = 0.05$), suggérés par Demšar (2006). L'hypothèse nulle est rejetée, indiquant que les classifieurs comparés en sont pas équivalents en termes de précision. Le résultat de cette expérience est résumé par le diagramme de différence critique de performance en figure 3, où l'on peut observer le rang moyen de chaque classifieur sur les 11 bases de données. Bien qu'aucun classifieur ne soit meilleur que tous les autres, nous pouvons identifier deux groupes : {SVM, J48, k Star, IBk , NB} au sein duquel il n'y a pas de différence significative de performance, et {SNB, RF}, performances sont meilleures que les autres bien qu'on ne puisse pas dire qu'ils sont significativement meilleurs que SVM. Notre recommandation est donc de combiner MiSeRe avec soit le SNB ou la RF. Puisque le SNB est libre de tout paramètre utilisateur, c'est celui que nous utiliserons par la suite.

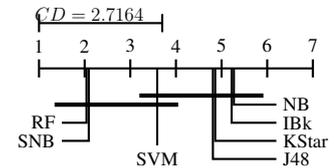


FIGURE 3 – Diagramme de différence critique de performance entre divers classifieurs sur des données recodées par les SCRMes extraites par MiSeRe.

Efficacité de MiSeRe. Notre méthode d'extraction, *MiSeRe*, est contrôlée par une contrainte de temps de calcul pendant lequel un certain nombre de règles est extrait. Nous étudions l'évolution de la performance prédictive de *SNB-MiSeRe* en terme de précision en fonction du nombre de règles extraites (pour un nombre de règles $\rho = 2^\alpha$; $\alpha \in [0; 14]$, voir Figure 4). Nous observons que la performance augmente avec le nombre de règles, puis se stabilise après quelques dizaines de règles pour les petites bases et quelques centaines (voire 1024) pour les plus grandes bases. De manière générale, après quelques centaines de règles, le système *SNB-MiSeRe* atteint ces performances maximales. Il en est de même pour les autres classifieurs couplés à MiSeRe (Egho *et al.*, 2015).

Nous étudions aussi l'efficacité de notre algorithme *MiSeRe*. La figure 4b présente le temps de calcul en fonction du nombre de règles extraites. Pour la plupart des bases de données, l'extraction d'un millier de règles est effectuée en moins de 80 secondes. Pour la base *ecoli*,

1. <http://www.khiops.com>

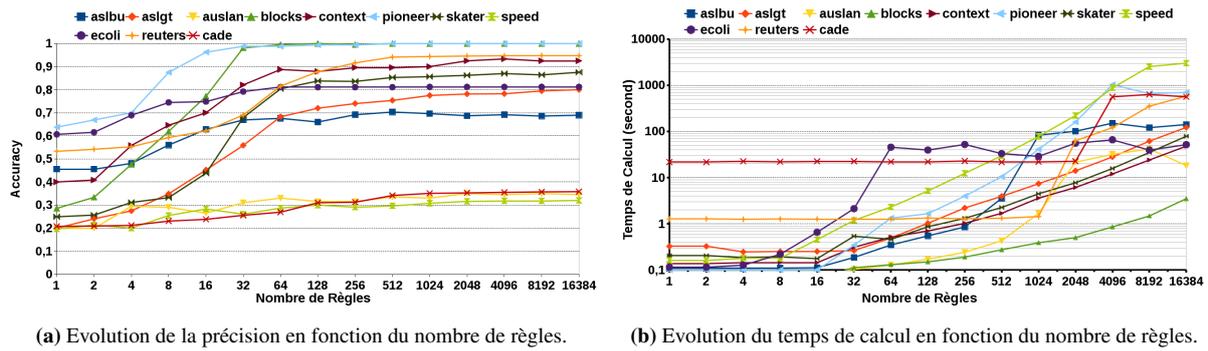
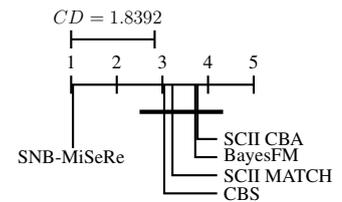


FIGURE 4 – Evolution des performances de MiSeRe.

seulement 53 règles à *level* positif sont extraites, c'est pourquoi *MiSeRe* a un temps d'exécution fixe ($\simeq 60s$ après 53 règles). Pour les données *cade*, de type texte, le temps d'exécution est d'environ 20s jusqu'à 2048 règles. En effet, la base contient 111776 mots distincts, donc les 2048 règles extraites ont une prémisse singleton. Ce temps de calcul correspond à la première étape (Lignes 1-2) de l'algorithme 1.

MiSeRe et l'état de l'art. Nous comparons les performances prédictives de *SNB-MiSeRe* avec quatre classifieurs à base de règles séquentielles de l'état de l'art : SCII Match, SCII CBA (Zhou *et al.*, 2013), BayesFM (Lesh *et al.*, 1999) et CBS (Tseng & Lee, 2005). Puisque les concurrents ne fournissent que des résultats de précision, ce sera notre critère de comparaison de performance. Les expériences sont menées avec les paramètres utilisés dans les articles originaux et les résultats de précision moyenne sont obtenus par validation croisée stratifiée en 10 parties. Les résultats détaillés sont accessibles dans (Egho *et al.*, 2015). A nouveau, nous appliquons le test de Friedman couplé au test post-hoc de Nemenyi. Le résultat de ces tests est présenté en figure 5. Le diagramme de différence critique de performance est sans appel : il y a bien une différence significative de performance entre *SNB-MiSeRe* qui est classé premier et ses concurrents. On observe les mêmes résultats pour *MiSeRe* couplé avec les autres classifieurs standards utilisés précédemment (Egho *et al.*, 2015).


 FIGURE 5 – Diagramme de différence critique de performance entre *SNB-MiSeRe* et ses concurrents.

5 Conclusion

Nous proposons une solution complète pour la classification supervisée à base de règles. Dans un cadre Bayésien, nous définissons un nouveau critère d'évaluation pour les règles séquentielles ; nous présentons aussi *MiSeRe*, un algorithme efficace pour extraire des règles pertinentes et robustes. L'usage de ces règles combiné à des classifieurs standards démontre de meilleures performances prédictives que les méthodes de l'état de l'art basées sur les règles. Notons que le critère suggéré est indépendant de l'algorithme *MiSeRe* et peut être donc appliqué aux règles séquentielles extraites par un autre algorithme. De plus, la généralité de la méthode employée nous permet d'envisager des extensions du nouveau critère pour des données séquentielles plus complexes, comme des séquences multi-dimensionnelles.

Références

- AGRAWAL R. & SRIKANT R. (1995). Mining sequential patterns. In *ICDE'95*, p. 3–14.
- AYRES J., FLANNICK J., GEHRKE J. & YIU T. (2002). Sequential pattern mining using a bitmap representation. In *KDD'02*, p. 429–435 : ACM.
- BOULLÉ M. (2006). MODL : A bayes optimal discretization method for continuous attributes. *Machine Learning*, **65**(1), 131–165.
- BOULLÉ M. (2007). Compression-based averaging of selective naive bayes classifiers. *Journal of Machine Learning Research*, **8**, 1659–1685.
- COENEN F. & LENG P. H. (2007). The effect of threshold values on association rule based classification accuracy. *Data & Knowledge Engineering*, **60**(2), 345–360.
- DEMŠAR J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, **7**, 1–30.
- DENG K. & ZAIANE O. R. (2010). An occurrence based approach to mine emerging sequences. In *DaWaK'10*, p. 275–284.
- DESHPANDE M. & KARYPIS G. (2002). Evaluation of techniques for classifying biological sequences. In *PAKDD'02*, p. 417–431.
- EGHO E., GAY D., BOULLÉ M., VOISINE N. & CLÉROT F. (2015). MiSeRe : Mining Sequential Classification Rules. <http://misere.co.nf>.
- GAY D. & BOULLÉ M. (2012). A bayesian approach for classification rule mining in quantitative databases. In *ECML/PKDD'12*, p. 243–259.
- HALL M. A., FRANK E., HOLMES G., PFAHRINGER B., REUTEMANN P. & WITTEN I. H. (2009). The WEKA data mining software : an update. *SIGKDD Explorations*, **11**(1), 10–18.
- JI X., BAILEY J. & DONG G. (2005). Mining minimal distinguishing subsequence patterns with gap constraints. In *IEEE ICDM'05*, p. 194–201.
- LAM H. T., MOERCHEN F., FRADKIN D. & CALDERS T. (2012). Mining compressing sequential patterns. In *SDM'12*, p. 319–330.
- LESH N., ZAKI M. J. & OGIHARA M. (1999). Mining features for sequence classification. In *ACM SIGKDD'99*, p. 342–346.
- MANNILA H. & TOIVONEN H. (1996). Multiple uses of frequent sets and condensed representations (extended abstract). In *KDD'96*, p. 189–194.
- MÖRCHEN F. & ULTSCH A. (2007). Efficient mining of understandable patterns from multivariate interval time series. *Data Mining and Knowledge Discovery*, **15**(2), 181–215.
- MYERS J. L. & WELL A. D. (2003). *Research Design and Statistical Analysis*. New Jersey : Lawrence Erlbaum Associates.
- SEBASTIANI F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, **34**(1), 1–47.
- TAN P. & KUMAR V. (2002). Discovery of web robot sessions based on their navigational patterns. *Data Mining and Knowledge Discovery*, **6**(1), 9–35.
- TSENG V. S. & LEE C. (2005). CBS : A new classification method by using sequential patterns. In *SDM'05*, p. 596–600.
- XING Z., PEI J. & KEOGH E. J. (2010). A brief survey on sequence classification. *SIGKDD Explorations*, **12**(1), 40–48.
- ZAKI M. J. (2000). Sequence mining in categorical domains : Incorporating constraints. In *CIKM'00*, p. 422–429 : ACM.
- ZHOU C., CULE B. & GOETHALS B. (2013). Itemset based sequence classification. In *ECML/PKDD'13*, p. 353–368.

TAAABLE : un système de raisonnement à partir de cas qui adapte des recettes de cuisine

Emmanuelle Gaillard, Jean Lieber et Emmanuel Nauer

UNIVERSITÉ DE LORRAINE, LORIA — 54506 Vandœuvre-lès-Nancy, France
CNRS — 54506 Vandœuvre-lès-Nancy, France
INRIA — 54602 VILLERS-LÈS-NANCY, FRANCE prénom.nom@loria.fr

Résumé : TAAABLE est un système de raisonnement à partir de cas (RÀPC) qui adapte des recettes de cuisine pour répondre à des requêtes culinaires. TAAABLE combine différentes approches issues des systèmes à base de connaissances telles que le RÀPC, la représentation des connaissances, l'extraction de connaissances à partir de données, et le traitement automatique des langues. Dans cet article, nous décrivons brièvement les principaux modules du système : le système de RÀPC, les différents processus d'adaptation, les conteneurs de connaissances exploités et la façon de les acquérir.

Mots-clés : raisonnement à partir de cas, ontologies, annotation sémantique, wiki sémantique, RDF(S), extraction de connaissances, traitement automatique des langues.

1 Introduction

TAAABLE est un système de raisonnement à partir de cas (RÀPC) qui retourne des recettes de cuisine en réponse à des requêtes posées par des utilisateurs, à partir d'un ensemble limité de recettes. Si aucune recette ne satisfait exactement la requête, TAAABLE recherche des recettes *similaires* et les adapte pour satisfaire la requête. TAAABLE a été initialement développé pour participer au *Computer Cooking Contest* (<http://computercookingcontest.net/>), une compétition annuelle organisée dans le cadre des conférences ICCBR (*International Conference on Case-Based Reasoning*). TAAABLE participe à cette compétition depuis sa création en 2008 et a déjà remporté plusieurs prix. Le fonctionnement du système TAAABLE est décrit en section 2 ; les principes de mise en œuvre sont décrits en section 3. Pour de plus amples détails, nous renvoyons à (Cordier *et al.*, 2014). Nous concluons par une discussion sur les liens entre TAAABLE et le domaine de l'intelligence artificielle (IA) et notamment comment les travaux menés dans cette application de cuisine ont donné naissance à des outils génériques utilisables dans d'autres domaines.

2 Description du système TAAABLE

TAAABLE est un système de RÀPC qui retourne des recettes de cuisine qui satisfont un ensemble de contraintes données par un utilisateur, telle que la présence ou l'absence d'ingrédients, le type et l'origine du plat, la compatibilité avec certains régimes (végétarien, sans alcool, etc.). La FIGURE 1 présente l'interface de TAAABLE (accessible à <http://tuuurbine.loria.fr/taaable/>) sur un exemple d'interrogation demandant un cocktail contenant de la bière, du jus d'ananas, de la grenadine mais sans liqueur. La recette « Cocktail mexicanos » contenant du sirop de fraise, de la bière, du jus d'ananas et pas de liqueur est retournée avec l'adaptation qui consiste à remplacer le sirop de fraise par du sirop de grenadine.

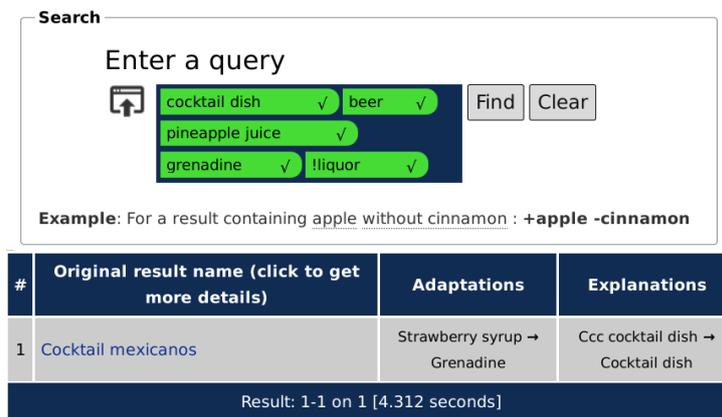


FIGURE 1 – Exemple d’adaptation dans l’interface du système TAAABLE.

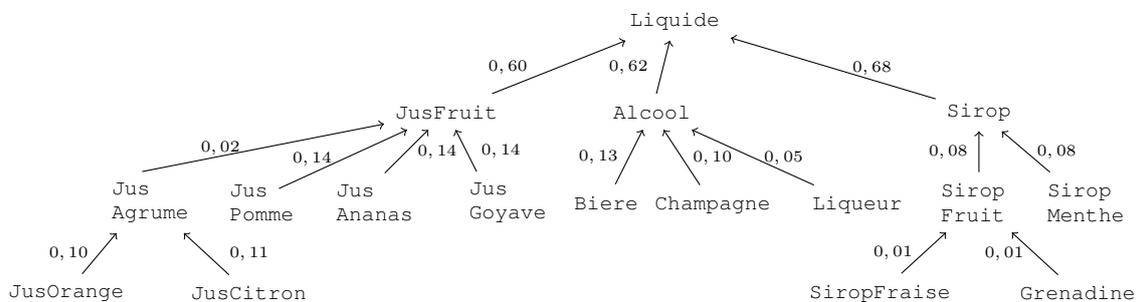


FIGURE 2 – Un extrait de la hiérarchie des aliments avec les coûts de généralisation utilisés comme connaissance de remémoration.

3 Mise en œuvre du système TAAABLE

3.1 Processus de remémoration

Comme la plupart des systèmes de RÀPC (Riesbeck & Schank, 1989), TAAABLE utilise une ontologie pour retourner les cas sources qui sont les plus similaires au problème cible (i.e. la requête). À partir de la requête qui est composée d’une conjonction de contraintes, le système recherche dans la base de recettes disponibles (i.e. la base de cas) si des recettes satisfont les contraintes. Si tel est le cas, elles sont retournées à l’utilisateur, sinon le système recherche des recettes similaires (i.e. des recettes qui satisfont partiellement la requête cible) et va adapter ces recettes pour en créer de nouvelles. La recherche de recettes similaires est guidée par une ontologie composée de plusieurs hiérarchies de spécialisation, sur les aliments, les types de plats, les origines des plats, etc. La recherche consiste à relaxer les contraintes en généralisant la requête de l’utilisateur. Dans les hiérarchies, chaque lien qui relie un concept à ses parents est associé à un coût qui traduit l’effort d’adaptation (cf. FIGURE 2). Le coût de généralisation d’une requête est calculé à partir de ces coûts de généralisation. L’objectif est de trouver la généralisation la plus spécifique (i.e. dont le coût est minimal) pour laquelle il existe des recettes dans la base de cas. Dans l’exemple de la FIGURE 1, la généralisation de Grenadine en SiropFruit permet de retrouver la recette “Cocktail mexicanos”.

3.2 Adaptation des ingrédients

TAAABLE exploite actuellement trois types d’approches pour adapter une recette aux ingrédients souhaités ou non par l’utilisateur.

Le premier type d’adaptation consiste, après généralisation de la requête dans la phase de remémoration, à utiliser une phase de spécialisation pour rechercher dans les cas sources, quels ingrédients jouent le rôle d’ingrédients qui viennent d’être généralisés. Dans l’exemple de la recette « Cocktail mexicanos », l’ingrédient qui joue le rôle de `SiropFruit` est `SiropFraise`. Le système dispose donc d’un chemin `SiropFraise` \sqsubseteq `SiropFruit` \sqsupseteq `Grenadine` qui lui permet de proposer de remplacer le sirop de fraise par le sirop de grenadine.

Le deuxième type d’adaptation exploite des règles de substitutions d’ingrédients par d’autres ingrédients dans un contexte donné (Gaillard *et al.*, 2011). Par exemple, remplacer `JusAnanas` et `SiropFraise` par `JusOrange` et `Grenadine` est un exemple de règle d’adaptation dans le contexte des `Cocktail`. L’adaptation à partir de règles est en réalité directement intégrée au processus de remémoration en recherchant des cas indexés par les ingrédients substituants pour une requête contenant les ingrédients substitués, par exemple, en cherchant des recettes contenant `JusAnanas` et `SiropFraise` pour une requête sur `JusOrange` et `Grenadine`.

Le troisième type d’adaptation utilise un opérateur de révision des croyances (Cojan & Lieber, 2014). Le cas source, la requête et les connaissances du domaine sont représentés en logique propositionnelle. Le principe de l’opérateur de révision est de modifier *a minima* le cas source afin qu’il satisfasse la requête.

3.3 Adaptation des quantités

La substitution d’ingrédients par d’autres ingrédients nécessite de gérer l’adaptation des quantités, pour conserver un certain équilibre dans la recette adaptée. Ainsi, lors d’une substitution de la bière par du champagne (liquide plus alcoolisé et plus sucré que la bière) dans un cocktail, il est intéressant de conserver le taux d’alcool, le taux de sucre, ainsi que le volume initial du cocktail. TAAABLE utilise pour cela un autre moteur de révision dans un formalisme de conjonctions de contraintes linéaires. Les contraintes qui sont impliquées dans la construction du résultat sont le cas source (e.g. la quantité initiale de chaque ingrédient de la recette à adapter), le résultat de l’adaptation (e.g. la quantité de bière est 0), et les connaissances du domaine (e.g. la quantité d’alcool par volume de champagne est 12 g, la quantité de sucre dans le cocktail est égal à la quantité de sucre apportée par chaque ingrédient, etc.). Étant donné cet ensemble de contraintes, le moteur de révision calcule les nouvelles valeurs de quantités en minimisant les changements des valeurs du cas source. Par exemple, 12 cl de bière seront remplacés par 5, 5 cl de champagne, la quantité du sirop de fraise passera de 3 cl à 1, 9 cl (en raison d’un apport plus important de sucre par le champagne, par rapport à la bière) et la quantité de jus de fruit passera de 12 cl à 20 cl pour conserver la quantité de liquide dans le cocktail.

3.4 Adaptation du texte de préparation

Lorsqu’on remplace un ingrédient par un autre, le texte contenant les instructions de préparation doit également être adapté. Différents niveaux d’adaptation sont possibles, pouvant aller de la simple substitution de chaînes de caractères (e.g. remplacer « *sirop de fraise* » par « *sirop de grenadine* » dans une recette de cocktail) à des substitutions plus complexes. Par exemple,

(Dufour-Lussier *et al.*, 2010) détaille comment adapter des parties complètes de textes. Cette adaptation utilise une représentation formelle du texte de la préparation sous forme d’arbres obtenue à partir de méthodes de traitement automatique des langues et quelques heuristiques particulières (Dufour-Lussier *et al.*, 2014).

3.5 La base de connaissances

Les connaissances exploitées par TAAABLE sont stockées, en accès libre, dans WIKITAAABLE (<http://wikitaaable.loria.fr/>), un wiki sémantique qui favorise l’édition collaborative des connaissances. On peut distinguer quatre types de connaissances :

1. une ontologie du domaine de la cuisine, composée d’environ 3200 concepts, organisés en plusieurs hiérarchies avec spécialisation mutiple : sur les aliments (cf. FIGURE 2), les types de plats, les origines des plats, etc., et d’une base terminologique qui associe à chaque concept une forme textuelle préférée et des variantes terminologiques ;
2. la base de cas qui contient actuellement 1666 recettes décrites chacune par leur titre, le type de plat produit, les ingrédients requis, les étapes de préparation, etc. ;
3. les connaissances d’adaptation qui sont des règles de la forme “*dans un contexte C, remplacer X_i par Y_i* ” où C est une conjonction de concepts de l’ontologie ou une recette particulière, X_i sont les ingrédients substitués et Y_i les ingrédients substituants ;
4. les connaissances de remémoration qui sont stockées sous la forme d’un coût associé aux relations de généralisation/spécialisation et aux règles d’adaptation (cf. FIGURE 2).

L’acquisition des connaissances a été réalisée manuellement, automatiquement et semi-automatiquement selon les types de connaissances.

L’ontologie du domaine et la base terminologique associée résulte principalement d’une expertise manuelle à partir de ressources disponibles sur le web tels que le *Cook’s thesaurus* (<http://www.foodsubs.com/>), l’*USDA Nutrient database* (<http://www.nal.usda.gov/>), Wikipedia et différents sites de recettes de cuisine. Une approche exploitant l’analyse formelle de concepts, une méthode de classification qui permet d’organiser des objets par rapport à leurs propriétés (Ganter & Wille, 1999) a également été utilisée pour structurer plus finement la hiérarchie des aliments en introduisant des concepts intermédiaires en fonction de comment les aliments sont utilisés dans les recettes (Dufour-Lussier *et al.*, 2011). Par exemple, le concept `FromageFrais` peut être subdivisé en plusieurs sous-concepts, par exemple, en `FromageFraisQuiPeutEtreTranché` (e.g. *Mozzarella*), `FromageFraisQuiPeutEtreBattuEtMêlé` (e.g. *Mascarpone*), etc.

L’alimentation de la base de cas fait appel à un processus automatique d’annotation sémantique pour passer des recettes sous forme textuelle à une représentation formelle. L’annotation des ingrédients consiste à associer un quadruplet de la forme (`<quantité>`, `<unité>`, `<aliment>`, `<modifieur>`) à chaque ligne d’ingrédient. Par exemple, “*1kg de tomates, coupées en tranches*” se verra associé à (1, kg, Tomate, en tranches). Ce processus d’annotation est intégré au wiki avec des procédures de complétion et de validation du résultat, afin d’ajouter les aliments absents de l’ontologie, de nouvelles formes terminologiques, ou encore corriger une recette contenant des erreurs de typologie. L’annotation du texte de la préparation est un processus automatique, comme mentionné en section 3.4.

L'acquisition de règles d'adaptation peut se faire par l'édition manuelle d'une nouvelle page de règle d'adaptation dans le wiki. De plus, lorsque TAAABLE propose une adaptation, l'utilisateur peut aussi évaluer l'adaptation proposée. S'il la juge correcte, l'adaptation est ajoutée au wiki. S'il la juge incorrecte, un processus d'acquisition de connaissances d'adaptation guidé par l'échec est déclenché (Badra *et al.*, 2009). Cette approche d'extraction de règles d'adaptation, qui utilise la recherche de motifs fréquents, est également reprise plus globalement pour extraire des règles d'adaptation génériques (Gaillard *et al.*, 2012), i.e. applicable dans un contexte assez large (e.g. les cocktails) et des règles d'adaptations spécifiques pour adapter une recette particulière à des contraintes additionnelles (Gaillard *et al.*, 2011), e.g. adapter la recette "Gâteau au chocolat" avec la contrainte "sans œuf".

Pour plus de détails sur la construction de la base de connaissances, voir (Cordier *et al.*, 2014).

4 Conclusion et discussion

TAAABLE est un système de RÀPC dont la construction a eu recours à un nombre important de principes, méthodes et techniques du domaine de l'IA tel que la représentation de connaissance, la construction d'ontologie, l'extraction de connaissance à partir de données, l'annotation sémantique, le raisonnement. Une ontologie pour représenter la connaissance du domaine de la cuisine, ainsi qu'une base terminologique établissant les liens entre textes et concepts de l'ontologie ont été construites. Ces ressources, stockées dans un wiki sémantique, sont utilisées par un processus d'annotation qui permet de construire, à partir de recettes textuelles, une représentation formelle également stockée dans le wiki. L'acquisition de connaissances a également fait appel à des processus d'extraction de connaissances à partir de données, pour enrichir l'ontologie en y ajoutant des concepts afin d'obtenir une structuration plus riche, ou encore pour acquérir des règles d'adaptation. Un processus de RÀPC ainsi que différentes approches pour l'adaptation des ingrédients, des quantités, du texte, qui exploitent ces connaissances, ont été mis en œuvre.

Le travail réalisé depuis 2008 autour de TAAABLE a conduit à la création de deux outils génériques qui sont distribués sous licence libre GPL :

- TUUURBINE (<http://tuuurbine.loria.fr/>), un système de RÀPC générique pour le web sémantique (Gaillard *et al.*, 2014) qui raisonne sur des connaissances stockées en RDF. TUUURBINE utilise des technologies du web sémantique telles que RDF(S), des RDF store, SPARQL et implémente un mécanisme d'inférence générique qui consiste à remémorer des cas similaires et à remplacer des caractéristiques de ces cas pour répondre à une requête donnée. La recherche de cas similaires utilise le principe de généralisation/spécialisation à partir de coûts de généralisation et de règles d'adaptation tel que ceux décrits en 3. La version actuelle de TAAABLE est en fait une instantiation de TUUURBINE qui utilise les connaissances stockées dans le RDF store associé à WIKITAAABLE (WIKITAAABLE utilisant *Semantic Media Wiki* qui stocke ses données sémantiques dans un RDF store).
- REVISOR (<http://revisor.loria.fr/>), une bibliothèque d'opérateurs de révision et d'adaptation par révision (Cojan *et al.*, 2013). La version courante de TAAABLE utilise des éléments de REVISOR pour l'adaptation des ingrédients et pour l'adaptation des quantités.

Actuellement, les ressources (connaissances et logiciels) disponibles à travers le système TAAABLE servent de support à de nouveaux travaux de recherche et une thèse est en cours sur le raisonnement exploitant des connaissances provenant d'une e-communauté, en gérant la fiabilité des connaissances par des interactions sociales. Des évaluations mesurant la satisfaction de l'utilisateur ont été réalisées (Gaillard *et al.*, 2015).

Références

- BADRA F., CORDIER A. & LIEBER J. (2009). Opportunistic Adaptation Knowledge Discovery. In L. MCGINTY & D. C. WILSON, Eds., *8th International Conference on Case-Based Reasoning - ICCBR 2009*, volume 5650, p. 60–74, Seattle, United States : Springer. The original publication is available at www.springerlink.com.
- COJAN J., DUFOUR-LUSSIER V., HERMANN A., LE BER F., LIEBER J., NAUER E. & PERSONENI G. (2013). Révisor : un ensemble de moteurs d'adaptation de cas par révision des croyances. In *JIAF - Septièmes Journées de l'Intelligence Artificielle Fondamentale - 2013*, Aix-en-Provence, France.
- COJAN J. & LIEBER J. (2014). Applying Belief Revision to Case-Based Reasoning. In *Computational Approaches to Analogical Reasoning : Current Trends*, volume 548 of *Studies in Computational Intelligence*, p. 133 – 161. Springer.
- CORDIER A., DUFOUR-LUSSIER V., LIEBER J., NAUER E., BADRA F., COJAN J., GAILLARD E., INFANTE-BLANCO L., MOLLI P., NAPOLI A. & SKAF-MOLLI H. (2014). Taaable : a Case-Based System for personalized Cooking. In S. MONTANI & L. C. JAIN, Eds., *Successful Case-based Reasoning Applications-2*, volume 494 of *Studies in Computational Intelligence*, p. 121–162. Springer.
- DUFOUR-LUSSIER V., LE BER F., LIEBER J. & NAUER E. (2014). Automatic case acquisition from texts for process-oriented case-based reasoning. *Information Systems*.
- DUFOUR-LUSSIER V., LIEBER J., NAUER E. & TOUSSAINT Y. (2010). Text adaptation using formal concept analysis. In I. BICHINDARITZ & S. MONTANI, Eds., *18th International Conference on Case-Based Reasoning - ICCBR 2010*, volume 6176 of *Lecture Notes in Artificial Intelligence*, p. 96–110, Alessandria, Italy : Springer-Verlag.
- DUFOUR-LUSSIER V., LIEBER J., NAUER E. & TOUSSAINT Y. (2011). Improving case retrieval by enrichment of the domain ontology. In *19th International Conference on Case Based Reasoning - ICCBR'2011*, London, United Kingdom.
- GAILLARD E., INFANTE-BLANCO L., LIEBER J. & NAUER E. (2014). Tuurbine : A Generic CBR Engine over RDFS. In *Case-Based Reasoning Research and Development*, volume 8765, p. 140 – 154, Cork, Ireland.
- GAILLARD E., LIEBER J. & NAUER E. (2011). Adaptation knowledge discovery for cooking using closed itemset extraction. In *The Eighth International Conference on Concept Lattices and their Applications - CLA 2011*, p. 87–99.
- GAILLARD E., LIEBER J., NAUER E. & CORDIER A. (2015). How Managing the Knowledge Reliability Improves the Results of a Reasoning Process. In *European Conference on Knowledge Management*, Udine, Italy.
- GAILLARD E., NAUER E., LEFEVRE M. & CORDIER A. (2012). Extracting Generic Cooking Adaptation Knowledge for the TAAABLE Case-Based Reasoning System. In *Cooking with Computers workshop @ ECAI 2012*, Montpellier, France.
- GANTER B. & WILLE R. (1999). *Formal Concept Analysis*. Heidelberg : Springer.
- RIESBECK C. & SCHANK R. (1989). *Inside Case-Based Reasoning*. Hillsdale, New Jersey : Lawrence Erlbaum Associates, Inc.

Modélisation, Résolution et Expérimentation de méthodes de Planification et d’Ordonnancement Collaboratif pour le Commandement de Brigade

Christophe Guettier¹, Willy Lamal², Israel Mayk³ and Jacques Yelloz¹

¹ Sagem Defense & Security, {christophe.guettier, jacques.yelloz}@sagem.com

² Direction Générale de l’Armement, willy.lamal@dga.defense.gouv.fr

³ US Army RDECOM, israel.mayk.civ@mail.mil

Adapté et traduit de "Design and Experiment of a Collaborative Planning Service for NetCentric International Brigade Command", IAAI 2015

Abstract

Les environnements opérationnels complexes nécessitent d’importantes capacités de commandement et un niveau élevé d’interopérabilité entre les systèmes d’information de la coalition. Ce papier synthétise les résultats constatés lors d’une série d’expérimentations utilisant un service d’information collaboratif intégrant différentes méthodes de préparation de mission automatisée. L’intérêt opérationnel réside d’une part dans l’amélioration de la compréhension mutuelle en interarmes, interarmées ou interalliés et d’autre part dans les capacités d’assistance entre unités (appui logistique, évacuation sanitaire entre autres). Les premières expérimentations sont nationales tandis que les dernières sont réalisées en coopération avec les USA et l’Allemagne. Le service d’information utilise un outil de planification et d’ordonnancement basé sur la programmation par contrainte : ORTAC (pour Optimisation des Ressources Tactiques et de l’Action au Contact), développé par la Direction Générale de l’Armement (DGA) et SAGEM. Le service est également intégré dans le démonstrateur MCG (Mission Command Gateway).

Introduction

En opération, la flexibilité et la sécurité du plan, l’organisation des soutiens et la compréhension mutuelle sont des problèmes critiques. Les systèmes de commandement et de contrôle (C2) doivent échanger de manière intelligente le long de la chaîne de commandement. Grâce à l’introduction des réseaux sans fil IP, il est maintenant possible de partager de l’information cohérente entre les unités, et ce, à différents échelons. Cependant, lorsque les modes d’actions sont planifiés de manière collaborative, il faut tenir compte des différents critères d’optimisation (durée de la mission, sécurité du dispositif, adéquation des ressources) et éviter des conflits entre actions qui pourraient apparaître entre unités amies. C’est le cas en particulier des actions de soutien et d’appui mutuels entre unités amies.

Les premières expérimentations (Phoenix 2007 et 2008) portent sur l’outil de planification et d’ordonnancement ORTAC, permettant de sélectionner automatiquement les modes d’actions et les itinéraires de progression et les transmettre à l’équipe de commandement. L’outil permet également d’ordonner les actions dans le temps, en coordonnant les différentes unités. L’outil de planification et d’ordonnancement ORTAC combine les techniques de

résolution de contraintes avec des stratégies de résolution élaborées pour résoudre, optimiser des plans ou en limiter les conflits automatiquement.

Les dernières expérimentations (C4ISR On the Move 2010 et 2012) évaluent la dimension collaborative du problème en intégrant l’outil dans un web service, lui-même intégré dans l’architecture du C2 du CERDEC (US Army Communications-Electronics Research Development and Engineering Center), nommé Mission Command Gateway (MCG). MCG permet de synthétiser la situation opérationnelle jusqu’au niveau tactique (comprenant les véhicules, les drones l’infanterie et la cavalerie). Il permet également au commandement de préparer des parties de plan de mission de manière collaborative, de les soumettre et de les valider lorsque l’ensemble est cohérent. Dans cette version, ORTAC peut être activé automatiquement ou à la demande durant le processus collaboratif de planification opérationnelle.

La dernière expérimentation représente la conclusion de cinq années de travail. Les résultats démontrent la faisabilité de prises de décisions accélérées et plus sûres. Ils démontrent également la faisabilité d’une approche par résolution de contraintes pour résoudre les plans d’engagement, dans un environnement système distribué.

Contexte Opérationnel

Dans les armées modernes, deux pratiques de commandement mettent en oeuvre les processus de planification. La première approche suit un processus descendant, suivant la hiérarchie c’est le principe "Battle Command". La hiérarchie détermine alors les ressources et les éléments de coordination pour toutes les actions jusqu’au plus bas niveau. La seconde approche suit une approche Mission Command, favorisant la prise de décision par le subsidiaire. Ceci signifie que le commandement donne les objectifs au niveau inférieur sans prendre en compte les actions de plus bas niveau. Dans les deux cas, les plans ne doivent pas présenter d’actions contradictoires mais elles doivent être coordonnées. Les plans doivent être cohérents des objectifs de mission, des positions et de la situation opérationnelle globale. Enfin, les actions doivent être ordonnées dans le temps de manière cohérente. Le processus de planification est initié à partir d’un ordre préparatoire dit "Warning Order" (WARNO). Les informations sur le

Unité	Durée habituelle	Personnel
Brigade	1 - 6 heures	300
Régiment/Bataillon	1 heure	70
Compagnie/Escadron	5 - 15 mins	16
Section/Peloton	second - 5 mins	4

Figure 1: Durées usuelles pour la planification et la replanification

plan et l'ordonnancement des actions sont représentées dans un Ordre Opérationnel ("Operational Order" ou OPORD). Lorsque la mission évolue en raison de contingences ou d'imprévus, l'OPORD est redéfini par un ordre partiel ou Fragmentary Order (FRAGO), permettant la mise à jour de la situation et la replanification des ordres périmés dans la chaîne de commandement.

En coalition, les plans initiaux et leurs mises à jour doivent être partagés, mutuellement compris et validés, sans ambiguïtés ou contradictions. En raison de la difficulté à construire des plans rapidement, les opérations sont planifiées au niveau de la division. Au plan international, les différents plans ne doivent pas entrer en conflit et certaines actions doivent être définies en commun. Nous montrons dans ce papier que certains conflits peuvent être résolus au niveau temporel (e.g., timings et/ou ordonnancement) ou spatial (e.g., zones d'opérations et/ou lignes de coordination spécifiés dans les ordres OPORD ou FRAGO). Ces conflits résultent généralement d'interprétations ou compréhensions différentes d'une situation tactique. Enfin, les échanges d'information (situation, plans, ordres, compte-rendus) doivent être cohérents. En particulier, dans plusieurs cas, la tenue de situation est mise à jour durant le processus de planification.

La figure 1 donne un ordre de grandeur en nombre d'unités et de temps pour élaborer un OPORD ou un FRAGO.¹

Le but des expérimentations est d'évaluer les délais et la qualité des décisions, la réduction des risques et l'amélioration des soutiens et appuis mutuels en coalition. Enfin, la compréhension mutuelle peut être améliorée en détectant et indiquant automatiquement les actions en conflit aux utilisateurs.

Exemple de situation et de plan en coalition

Dans le scénario (Fig. 2), la 4ème Brigade expérimentale internationale², notée 4IBCT (International Battle Combat Team), intègre des bataillons français et Allemands ainsi qu'une logistique et des appuis US. L'OPORD est (fortement) résumé dans cette section.

¹Les niveaux brigades combine différentes armes (artillerie, cavalerie, infanterie, logistique...), tandis que le division inclut aussi les forces aériennes et navales. Les niveaux Régiment et en dessous sont dédiés au domaine tactique

²dans ce scénario, l'opération en coalition est réalisée au niveau brigade à titre expérimental.

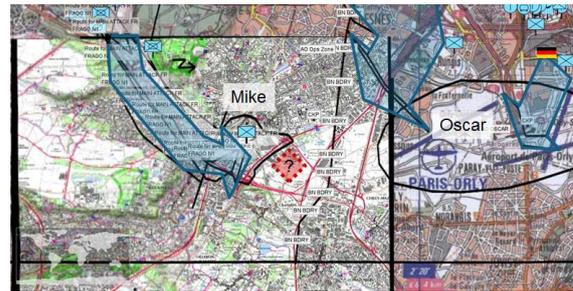


Figure 2: Vue graphique de l'ordre opérationnel, positions ennemies et amies, axes de progression amis pour les 3 bataillons. Les conflits entre plans peuvent apparaître aux limites des zones d'action.

Bien que pour des raisons pratiques, l'action se déroule dans le sud de Paris, le scénario est purement fictif et représentatif d'une mission de maintien de la paix.

Forces ennemies: ce sont des unités de milice "Lexoviennes", localisés entre ORY and MSY. Modes d'actions: embuscades, pièges et attaques de bâtiments administratifs, à l'aide de mortiers, snipers et roquettes, à partir de zones fortement compartimentées.

Forces amies: définies en termes de position, objectifs (OBJ) et de domaine de responsabilités ("Area of Responsibility" - AoR). Le domaine d'opération ("Area of Operation" - AO) inclut OSCAR, CHARLIE et VICTOR, et les objectifs (OBJ) OSCAR, et MIKE.

- Infanterie mécanisée (MECH) Française: 3/44 MECH BN(FR) sur AoR VICTOR
- Infanterie (INF) Allemande: 2/44 INF BN(GE) située à 5kms ouest de MSY, sur AoR OSCAR.
- Infanterie US : 1/44 INF BN(US) située à 10kms au nord d'ORY, sur AoR CHARLIE.

Mission

Dans le but de conduire l'évacuation de civils de OBJ MIKE vers OSCAR, la 4IBCT doit

1. Sécuriser les AoR VICTOR, CHARLIE, OSCAR puis OBJ MIKE.
2. Organiser deux points de rassemblement pour les réfugiés dans OBJ MIKE.
3. Reconnaître et sécuriser l'axe de OBJ MIKE vers OSCAR.
4. Assurer l'évacuation du convoi routier de OBJ MIKE vers OSCAR

Liason entre OBJ OSCAR, AoR VICTOR, OBJ MIKE

Prêt à:

- Prévenir toute action des Lexoviens dans l'AO et sur les axes routiers
- Evacuer les réfugiés civils vers l'aéroport.

chaque bataillon sous forme de tactes assignées pour les unités manoeuvrantes:

3/44 MECH BN(FR)

1. Contrôler OBJ MIKE et ses accès. Reconnaître et Sécuriser les axes d’accès.
2. Organiser deux points de regroupement de réfugiés sur deux sites distincts. Reconnaître et sécuriser deux axes entre OBJ MIKE et OBJ OSCAR. Liaison avec 3 MECH BN(FR).
3. Evacuer les réfugiés vers OBJ OSCAR sur deux axes par route.

2/44 INF BN(GE)

1. Contrôler OBJ OSCAR et ses accès. Reconnaître et Sécuriser les axes d’accès.
2. Coordonner avec la 2INF BN(GE) l’évacuation des convois dans OBJ MIKE. Faire des points de liaison sur les axes d’évacuation avec 2INF BN(GE) vers OBJ OSCAR.
3. Protéger le débarquement des réfugiés sur OSCAR à l’arrivée.
4. Protéger les convois depuis OBJ MIKE vers OSCAR et organiser l’embarquement en avion avec les autorités locales.

1/44 INF BN(US)

1. Etablir la route principale de ravitaillement depuis AoR VICTOR vers OBJ MIKE.
2. Etre prêt à conduire un déminage vers OBJ MIKE, conduire une reconnaissance vers OBJ OSCAR et Liaison avec 3MECH BN(FR).
3. Etre prêt à appuyer 2INF BN(GE) et 3MECH BN(FR) en déminage.
4. Dans le but de protéger les forces durant l’évacuation, sur ordre, neutraliser les obstacles sur les ponts et routes nationales.

Modélisation et résolution du problème de planification et d’ordonnancement

Une formulation simplifiée et informelle de la spécification d’entrée de la mission peut être donnée pour la structure du terrain, les conditions initiales, les objectifs de mission, les capacités et les contraintes de coordination. Les éléments suivants sont connus hors ligne et caractérisent la spécification d’entrée:

- La structure du terrain est définie comme un ensemble de positions liées par des axes de progression. Chaque position est géoréférencée et deux positions distantes sont liées par une distance.
- Les conditions initiales correspondent aux positions ressources disponibles.
- Les actions sont déterminées selon une position du terrain et l’unité qui l’atteint. Elles peuvent être sujettes à des préconditions.
- Tout objectif (primaire ou secondaire) correspond à une position avec éventuellement une action nécessaire à exécuter.
- Les contraintes de coordination résultent des règles d’engagement. Elles imposent des synchronisations entre les unités ainsi que l’exécution d’actions en parallèle.

représentation.

Modèle de planification de mission

Le modèle de planification et d’ordonnancement a été largement exposé en (Guettier 2007), (Lucas et al. 2010) et (Lucas and Guettier 2012). Il peut représenter les déplacements, observations, gestion des appuis et logistique de l’infanterie et de la cavalerie. Pour être exécutée, toute action doit vérifier un ensemble de préconditions (en termes de ressources, timing ou déplacement). Une coordination logique permet la synchronisation des différentes unités dans l’espace et le temps. Dans (Guettier 2007), trois types de logique de coordination ont pu être proposés:

- Soutien et appui: une unité doit exécuter une action depuis une position, lorsqu’une autre peut l’appuyer.
- Composition d’actions: deux unités exécutent deux actions respectives dans la même fenêtre de temps.
- Exclusions d’actions: deux unités ne peuvent exécuter deux actions respectives simultanément (en utilisant une fenêtre de temps) sur deux positions prédéfinies.

Dans le cadre de la planification collaborative, des extensions de l’approche sont présentées pour résoudre les conflits entre unités.

Les positions du terrain et les routes disponibles sont représentées en utilisant un graphe orienté $G(X, U)$ où l’ensemble U des arêtes représente les chemins possibles et l’ensemble X de sommets représente les points de navigation (Fig. 3). Un sommet est noté x , une arête peut être notée u ou (x, x') . L’itinéraire d’un plan de mission est défini par un ensemble de variables définies dans $\{0, 1\}$ sur les arêtes: l’ensemble de variables $\varphi_u \in \{0, 1\}$ modélise un chemin de $start \in X$ vers $end \in X$, où u appartient au plan ssi la variable de décision $\varphi_u = 1$. Le plan résultant peut être représenté $\Phi = \{u | u \in U, \varphi_u = 1\}$. D’une position initiale à la finale, la cohérence du plan est imposée par les équations de conservation de flots, soit $\omega^+(x) \subset U$ et $\omega^-(x) \subset U$ sont les arêtes entrantes et sortantes du sommet x , respectivement.

$$\sum_{u \in \omega^+(start)} \varphi_u = 1, \quad \sum_{u \in \omega^-(end)} \varphi_u = 1, \quad (1)$$

$$\sum_{u \in \omega^+(x)} \varphi_u = \sum_{u \in \omega^-(x)} \varphi_u \leq 1 \quad (2)$$

Comme les variables de flots sont dans $\{0, 1\}$, l’équation (2) assure la connectivité et l’unicité du chemin, tandis que l’équation (1) impose les conditions limite sur le début et la fin. Ce modèle détermine une chaîne linéaire alternant points de navigation et axes de progression le long des arêtes du graphe. La formulation de la longueur du chemin (3) où la variable D_x exprime la date d’arrivée à la position x (voir Fig. 3). La variable $d_{(x', x)}$ représente la durée pour effectuer la manoeuvre depuis la position x' vers x . La durée D_x cumule la durée de l’action et de la navigation entre les deux points de navigation. On a:

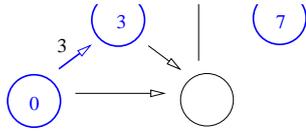


Figure 3: Sur le graphe, les arêtes bleues sont des flots positifs indiquant la route planifiée, et propagent une métrique opérationnelle (temps, ressources)

$$D_x = \sum_{(x',x) \in \omega^-(x)} \varphi_{(x',x)}(d_{(x',x)} + D_{x'}) \quad (3)$$

$$\forall (x, x') \in U, d_{(x,x')} \in \mathbb{N}, l_{(x,x')} \leq d_{(x,x')} \leq u_{(x,x')} \quad (4)$$

Les bornes $u_{(x,x')}$ et $l_{(x,x')}$ sont des constantes du problème, et utilisées en (4), et peuvent différer entre deux arêtes. Si une unité ne passe pas par le sommet x , alors $D_x = 0$. Comme présenté dans (Guettier 2007), différentes contraintes et fonctions de coût peuvent être formulées de cette manière.

Algorithme de résolution

Toute la formulation du problème et le développement des stratégies de résolution ont été implantée en Programmation Logique avec Contraintes en utilisant l’algèbre sur les domaines finis CLP(FD), dans l’environnement SICStus Prolog. La stratégies de résolution a été présentée en (Lucas and Guettier 2012)(Guettier 2007) et utilise une hybridation de différentes méthodes. Elle garantit à l’utilisateur la complétude de la recherche, l’optimalité de la solution et la preuve d’optimalité. La stratégie combine un algorithme de propagation de contraintes (implanté par le SICS), une méthode d’évaluation - séparation et une sonde (probe) définie statiquement résultant d’un méta-heuristique. La sonde permet de limiter les retours-arrières et de focaliser rapidement sur les solutions prometteuses. Celle-ci est construite à l’aide d’un algorithme de colonies de fourmis (Ant Colony Algorithm - ACO) résolvant une relaxation du problème initial (Lucas et al. 2010). ACO développe alors une population de fourmis à travers l’espace d’états et améliore itérativement la recherche, basée sur une heuristique et un modèle de mémoire collective *phéromone*. En pratique, ACO produit un chemin pour chaque entité, en relaxant les contraintes d’ordonnancement et les conflits. La méthode exploratoire est alors structurée à l’aide de métriques établies entre cette première solution relaxée, et une solution vérifiant l’ensemble des contraintes. Cette phase permet d’ordonner l’exploration des variables de décision et des valeurs associées. Les variables sont triées selon les valeurs des métriques résultant de la sonde. De fait la solution relaxée n’est pas exploitée par la stratégie globale, mais ses propriétés le sont. Ces calculs sont de complexité polynomiale et l’approche garantit les propriétés de complétude et optimalité.

scénarios de planification centralisée. Ces premières expérimentations optimisent la durée d’exécution de l’ensemble de la mission. Ces scénarios sont représentatifs de missions de maintien de la paix dans un contexte d’opérations réseau-centré et de niveau bataillon, compagnie ou escadron. Les instances comprennent entre 4 et 8 unités et utilisent de l’infanterie comme de la cavalerie (ce qui a une incidence sur les temps de déplacement). (Fig. 10). Chaque action constituant le scénario comprend des préconditions sur les positions des amis ainsi que sur une ressource capacitaire (temps d’exposition pour chaque unité).

1. Reconnaissance aux abords: déploiement d’un dispositif de reconnaissance aux abords d’une large zone urbaine. Le problème réside dans le choix des positions clés et la protection mutuelle des positions.
2. Renforcement d’un dispositif d’aide ONU: déploiement d’un dispositif de protection et soutien sur une intervention humanitaire de l’ONU. Le problème consiste à rejoindre, de manière coordonnée, les zones d’implantation de l’ONU en zones urbaines et en sécurité.
3. Inspections de sites sensibles en zones urbaines: différents sites doivent être inspectés en vue d’un désarmement. Le problème consiste à ordonner les actions d’inspection en ayant verouillé les accès.
4. Sécuriser une zone d’extraction de civils: déployer un dispositif et manoeuvrer pour localiser des réfugiés et sécuriser une zone de rassemblement de civils.

Deux scénarios supplémentaires sont évalués dans le cadre d’un véritable déploiement terrain, lors des expérimentations Phoenix 2007 et 2008, correspondant à un cas d’école en matière de reconnaissance. Les problèmes sont résolus sur des processeurs standards de 1Ghz sous Windows. La preuve d’optimalité est obtenue en général (sur 80% des instances) dans un temps raisonnable inférieur à 5 secondes après avoir obtenu l’optimal. Les temps de résolution sont acceptables pour l’utilisateur (fig. 4). Ceux-ci dépendent surtout de la taille du graphe et de la complexité des contraintes de coordination entre unités.

Scenario	Nombre d’ Unités	Temps de résolution pour l’optimalité
Simulation		
Reconnaissance	8	< 0.5s
Renforcer l’ONU	8	< 0.25s
Inspection de sites	8	< 1s
Sécuriser une zone	8	< 1s
Expérimentation terrain		
Reconnaissance (2007)	5	< 0, 1s
Reconnaissance (2008)	5	< 0, 5s

Figure 4: Temps de résolution pour les différents scénarios expérimentaux

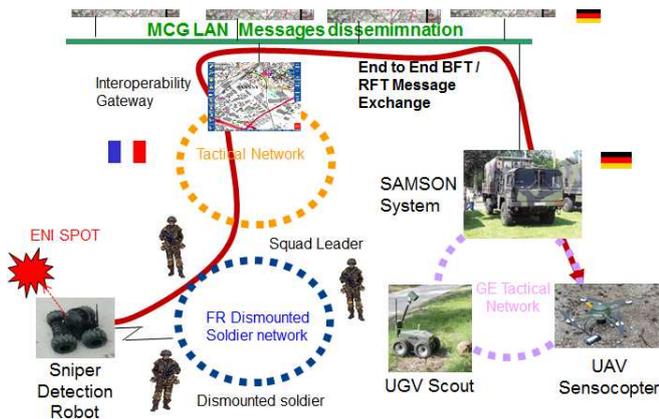


Figure 5: Architecture expérimentale en réseau, incluant les niveaux Brigade / Bataillon, les réseaux tactiques, drones, robots, et systèmes soldats connectés

Aide à la Planification Collaborative et Distribuée

Cette partie étend le principe de planification au travail collaboratif en coalition. Pour cela, l'outil de planification et d'ordonnancement ORTAC est intégré dans l'architecture de services de l'US Army, MCG. Cet outil permet un travail collaboratif distribué et asynchrone, les acteurs pouvant contribuer à l'élaboration d'un plan, d'un compte-rendu ou d'un ordre à distance. Différentes fonctions de tenue de situation opérationnelle sont intégrées dans MCG. L'ensemble est évalué dans une expérimentation en Octobre 2012 comprenant des bataillons fictifs US, Français et Allemands de niveau Brigade (cf. l'exemple introductif).

Cadre expérimental

L'architecture opérationnelle déployée pour cette expérimentation émule une chaîne de commandement depuis la Brigade jusqu'au niveau tactique. Elle permet l'échange vertical d'ordres et de rapports. Les commandements des Bataillons (Français, US et Allemand) et de la Brigade internationale utilisent l'environnement MCG, tandis que les niveaux tactiques (compagnie, escadron), utilisent les systèmes de commandement nationaux. De plus, pour représenter une activité sur le terrain, plusieurs systèmes réels sont présents et communiquent avec le commandement. Le reste des forces fictives est simulé. En dépit des simplifications, le système résultant est représentatif des principales activités d'une brigade (fig. 5). Un système de communications "IP de bout en bout" permet la communication de type diffusion ou bien "point à point" entre tout les acteurs.

Système d'information et de commandement

MCG est basé sur une architecture orientée service. Un service est un groupe de fonctions défini par ses interfaces ex-

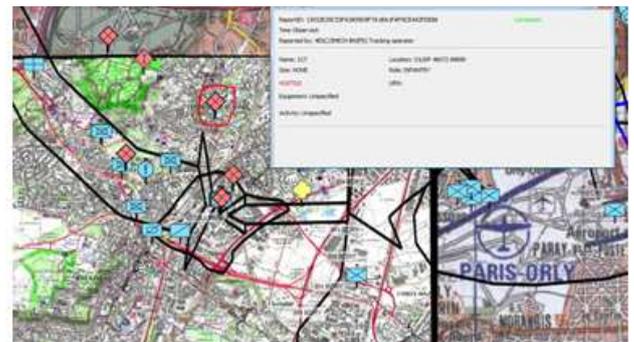


Figure 6: Situation opérationnelle avec positions amies et contacts ennemis.

ternes et dont la qualité est régie par un contrat couvrant les données d'entrées ou sorties, le contrôle et les propriétés du service. MCG dispose de fonctions d'orchestration (Workflow Orchestration Service - WOS) permettant les communications et synchronisation internes et externes des services (Fig. 7). MCG permet aussi d'assurer une continuité entre différents modèles de traitements, d'échanges de données ou d'activation (publication/souscription, flot de données, transactions, ...).

Service de tenue de situation opérationnelle Durant une mission les informations tactiques sont remontées, puis fusionnées et consolidées par des fonctions de tenue de situation opérationnelle. Celle-ci évolue pendant la mission et doit être partagée en temps utilisateur. La plateforme MCG traduit les messages entre les différentes représentations utilisées. La plupart des messages sont des ordres, des compte-rendus, des reports de positions amis, neutre, non identifiés ou ennemis. Quand la situation change, en général en raison d'un retard, d'un danger ou d'une menace, la chaîne de commandement doit replanifier et transmettre un FRAGO, tout en évitant d'engendrer de nouveaux conflits entre unités amies (Fig. 6).

Services de planification collaborative Le processus de planification collaboratif et distribué de MCG permet à tout commandant de préparer un plan et de le soumettre à sa hiérarchie, en consultant la situation opérationnelle sur une cartographie numérique. Celle-ci peut valider la proposition ou renvoyer un retour avec des suggestions de modification. Lorsque tous les plans des subsidiaires sont validés, la hiérarchie peut transmettre les ordres comme les OPORD (en début de mission) ou FRAGO (en cours de mission). Le processus se propage alors le long de la chaîne de commandement. Ce processus pose deux problèmes, d'une part les plans doivent être construits de manière concurrente pour être efficaces, et des accès simultanés à des variables partagées sont nécessaires à implanter. Pour cela,

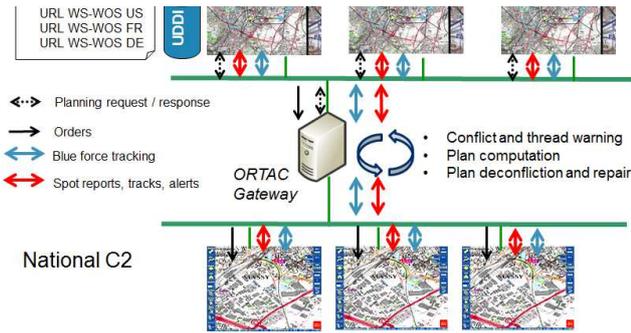


Figure 7: Architecture informatique des différents postes de commandement et serveur ORTAC

l’ingénierie doit garantir l’absence d’interblocage. D’autre part la situation opérationnelle évolue pendant le processus de planification. De plus, les utilisateurs ne doivent pas être surchargés et converger vers une solution partagée. Par conséquent, au delà de la cohérence des données, le paradigme collaboratif nécessite une aide automatique comme ORTAC pour renforcer la coordination des plans et supprimer les conflits rapidement.

ORTAC peut être activé à la demande ou automatiquement, lorsque des conflits ou dangers sont détectés à partir des modules de gestion de situation opérationnelle (et mise à jour grâce aux reports de positions amies ou ennemies). Les données manipulées en entrée sont les positions courantes pour les forces amies, les objectifs, les menaces comme les conflits potentiels entre unités. ORTAC optimise le calcul des plans en minimisant les conflits ou la durée des missions. Le service ORTAC peut détecter automatiquement un conflit ou une menace, résoudre et proposer à l’utilisateur une alternative. Ceci permet par exemple de prévenir les accidents ou tirs fratricides.

Détection et résolution des conflits

Cette section propose l’extension du modèle et des algorithmes d’ORTAC afin de résoudre automatiquement les différents conflits lors du processus de planification. Soit i et j deux unités différentes, elles sont considérées en conflit lorsque leur date de passage sur un même point de navigation sont concomitantes dans un intervalle temporel $[-W_x, W_x]$. Une variable C_x dans $\{0, 1\}$ traduit un sommet X subissant un tel conflit (Fig. 8). On considère la fonction de coût finale $\Omega(X)$, correspondant au nombre total de conflits détectés (6) pour tout les noeuds X .

$$C_x \Leftrightarrow \exists(i, j)$$

$$[D_x^i - W_x, D_x^i + W_x] \cap [D_x^j - W_x, D_x^j + W_x] \neq \emptyset \quad (5)$$

$$\Omega(X) = \sum_{x \in X} C_x \quad (6)$$

Au niveau de la stratégie de résolution, la partie séparation

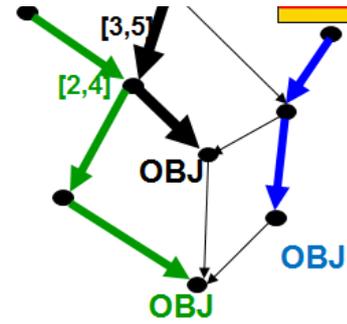


Figure 8: Si la route bleue est en conflit avec la noire, les dates de passage doivent appartenir à des intervalles "surs" (resp. [2,4] and [3,5], pour $W_x = 1$) ou bien les itinéraires doivent être redéfinis

- évaluation minimise le nombre de conflits entre plans, ce qui correspond à la fonction $\Omega(V)$. La stratégie favorise la déconflition temporelle et effectue les retours arrières sur les variables temporelles en premier. Quand les alternatives temporelles ont été épuisées, la stratégie réévalue les itinéraires avec la sonde.

Résultats

Les expérimentations se sont déroulées sur trois jours complets comprenant les activités de planification et d’ordonnancement depuis un OPORD initial suivi de FRAGO. L’expérimentation impliquait des données produites par les systèmes de terrain. Les charges réseaux n’ont pas eu d’impact sur les processus de décision. La figure(9) représente l’exécution de l’expérience pour le deuxième jour (en 5 heures d’expérimentation environ), ce qui est largement inférieur au temps habituel de décision.

Le service ORTAC a été utilisé pour définir les axes de progression associés à un ordre. Les points de passages obligés étaient utilisés de manière occasionnelle, notamment pour suivre la géométrie des axes. La figure (10) donne le nombre de décisions par rapport au nombre de requêtes effectives. Cela montre que plusieurs tentatives sont nécessaires avant de proposer et valider un plan correct. La détection et résolution positive de conflits a été observé à deux reprises durant l’expérimentation entre 1-44IN BN(US) et 3MECH BN (FR) le deuxième jour et entre US 1-44IN BN et 2IN BN (GE), le troisième jour. Pour utiliser pleinement la déconflition, le concept de validation de plan est important pour déclencher la détection et la résolution. Les temps de résolution sont compatibles avec les attentes des utilisateurs, bien que la déconflition demande plus de puissance CPU.

Travaux Connexes

Plusieurs laboratoires de recherche se sont intéressés à la planification automatique de mission, s’appuyant essen-

+1h05	FR MECH BN FR MECH BN	Reconnaissance vers OBJ OSCAR Contact ennemi
+1h10	4 IBCT 4 IBCT	Envoi FRAGO1 2 INF BN Evacuation Ambassade
+2h10	All BN GE IN BN	Envoi FRAGO Mouvement au point de liaison MIKE
+3h42	GE IN BN	Contact ennemi
+4h00	4 IBCT 4 IBCT	Envoi FRAGO2 2 INF BN Evacuation Ambassade
+4h30	US 1-44IN BN US/FR FR MECH BN US 1-44IN BN	Envoi FRAGO Conflit résolu! Envoi FRAGO Mouvement vers OBJ MIKE

Figure 9: Synthèse du journal d’expérimentation du jour 2; le processus de planification enchaîne l’OPORD de la Brigade. Les FRAGO sont transmis à chaque re planification de mission. Un conflit a été détecté et résolu entre bataillons Français et US.

Day	1	2	3	Temps de résolution
Planification (OPORD)	3	3	3	< 0.5s
Replanification (FRAGO)	4	4	2	< 0.75s
Deconfliction (FRAGO)	0	1	1	< 30s
Nombre Total de requêtes	18	22	25	

Figure 10: Les trois premières lignes donnent le nombre d’activation du service ORTAC et fournit une décision (OPORD or FRAGO). La dernière ligne donne le nombre total d’activations.

tiellement sur un cadre spécialisant les Hierarchical Task Networks (HTN). Par exemple, MACBeth (Goldman et al. 2002), combine la résolution de contraintes et les HTN pour spécifier les invariants temporels et la consommation de ressources. Il utilise aussi un algorithme permettant de couper l’espace de recherche. La coordination de tâches ou plans opérationnels a aussi été le sujet de différents travaux, par exemple (Myers 2002) et (Myers 2006), utilisant des métriques exploitées pour le contrôle d’exécution. Seulement les travaux tels que (Allsopp et al. 2003), impliquant des expérimentations, sont vraiment comparables.

Conclusion

Ce travail est le fruit d’un accord unique entre les USA, l’Allemagne et la France puisque c’est la première tentative dans l’histoire d’une intégration au niveau Brigade de ces trois forces. L’expérimentation finale confirme l’intérêt des travaux préliminaires dans le domaine: la Brigade Internationale pouvait améliorer les délais de décision et meilleures que les pratiques existantes. D’un point de vue technique, ces expérimentations démontrent la faisabilité de l’approche. Par rapports aux premières expérimentations internationales (Guettier et al. 2011), de nouvelles techniques de résolution et déconfliction ont été utilisées.

portant entre les équipes US, françaises et Allemandes. Les auteurs remercient vivement tout les participants, développeurs, PME et en particulier le BWB Allemand, Diehl BGT Defense, l’US Army CERDEC, Drexel University, EOIR Technologies, Inc, Cougar Software Inc., l’Ambassade des Etats Unis à Paris, la DGA Française et Sagem. Merci également à Mark Boddy (Adventium Lab) et Eric Jacopin (Centre de Recherche des Ecoles de Saint Cyr Coetquidan) pour leurs conseils.

References

- Allsopp, D.; Beautement, P.; Kirton, M.; Tate, A.; Bradshaw, J.; Suri, N.; and Burstein, M. 2003. The coalition agents experiment: Network-enabled coalition operations. *Special Issue on Network-enabled Capabilities, Journal of Defence Science* 8(3):130–141.
- Goldman, R.; Haigh, K.; Musliner, D.; and Pelican, M. 2002. Macbeth: A multi-agent constraint-based planner. In *Proceedings of the 21st Digital Avionics Systems Conference*, volume 2, 7E3:1–8.
- Guettier, C.; Yelloz, J.; Mayk, I.; and Lamal, W. 2011. Interoperable joint planning and execution web service with titan. In *Proceedings of IEEE MILCOM Military Communications Conference*.
- Guettier, C. 2007. Solving planning and scheduling problems in network based operations. In *Proceedings of Constraint Programming (CP)*.
- Lucas, F., and Guettier, C. 2012. Hybrid solving technique for vehicle planning. In *Proceedings of Military Communication Conference (MILCOM)*.
- Lucas, F.; Guettier, C.; Siarry, P.; de La Fortelle, A.; and Milcent, A.-M. 2010. Constrained navigation with mandatory waypoints in uncertain environment. *International Journal of Information Sciences and Computer Engineering (IJISCE)* 1:75–85.
- Myers, K. 2002. Active coordination of distributed human planners. In *Proceedings of the 6th Int. Conference on Artificial Intelligence Planning and Scheduling*.
- Myers, K. 2006. Metatheoretic plan summarization and comparison. In *Proceedings of the 16th Int. Conference on Automated Planning and Scheduling*.

Simulations multi-agents de l'activité humaine : application dans le contexte énergétique résidentiel français

Thomas Huraux^{1,2}, Nicolas Sabouret³, Yvon Haradji¹, François Sempé

¹ EDF R&D, Département ICAME, France
yvon.haradji@edf.fr

² LIP6, CNRS UMR 7606 – Sorbonne Universités, UPMC Univ Paris 06, France
thomas.huraux@lip6.fr

³ LIMSI-CNRS, UPR 3251 – Université Paris-Sud, France
nicolas.sabouret@limsi.fr

Résumé : Dans cet article, nous présentons la plateforme SMACH qui permet aux experts d'EDF de représenter et étudier l'activité des habitants dans leurs foyers pour la mettre en relation avec la consommation électrique. En se basant sur la modélisation des habitants, de leurs tâches et des appareils présents dans le logement, la plateforme utilise le paradigme des systèmes multi-agents pour permettre à la fois des simulations automatiques et des simulations participatives où des clients EDF jouent leurs propres rôles dans le foyer.

Mots-clés : systèmes multi-agents, comportements humains, simulation participative, consommation électrique

1 Introduction

Une difficulté majeure pour les producteurs d'électricité comme EDF, à mesure que la consommation énergétique augmente, est de pouvoir s'adapter en temps réel à la demande des clients pour maintenir une fréquence stable. En cas de déséquilibre (une demande trop forte par rapport à la production ou le contraire), le système peut se retrouver en *black out* : risques de délestage fréquence-métrique, emballements des machines de production synchrones, *etc.* En France, la production électrique repose en grande partie sur le nucléaire (plus de 76% de la production). Cette production est stable puisque peu modulable au niveau des centrales et le démarrage d'une tranche (c.-à-d. un réacteur et son système de production associé) est lent. Pour faire face aux pics de demande, les producteurs utilisent donc l'hydraulique de retenue (barrages) et les centrales thermiques à énergie carbonées. Ces centrales à énergies carbonées ont un démarrage rapide et une puissance facilement ajustable mais elles produisent beaucoup de CO_2 et contribuent au réchauffement climatique (André & Royer (1999)). C'est pourquoi EDF s'intéresse de plus en plus à comprendre et réduire les pics de consommation au niveau des consommateurs, pour limiter le recours à ce type de centrales afin de pouvoir mobiliser des ressources moins réactives mais surtout moins polluantes.

Dans ce contexte, deux observations peuvent être faites : le bâtiment représente 44%¹ de la consommation en énergie finale (Fig. 1) et la consommation de ce secteur est caractérisée par une grande variabilité dûe aux comportements humains qui sont, par nature, un phénomène complexe. En effet, la dynamique du comportement humain dépend des caractéristiques de

1. chiffre du Ministère de l'écologie

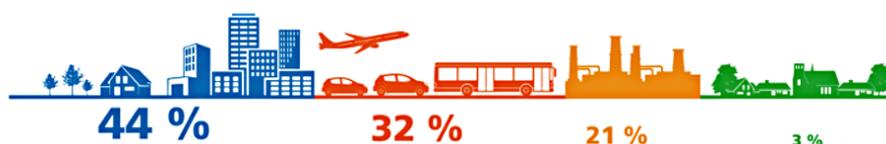


FIGURE 1 – Consommation finale par secteur (source EDF)

l'habitant, de ses perceptions et interactions avec l'environnement et les autres habitants, ce qui rend sa prédiction par le calcul difficile (Batty & Torrens (2001)). C'est pourquoi comprendre et analyser les comportements des usagers dans les foyers est une problématique majeure pour la réduction de la consommation énergétique.

Deux solutions sont envisageables pour étudier et anticiper les comportements. D'une part l'expérience en situation réelle permet aux ergonomes d'étudier l'activité humaine. D'autre part la simulation peut permettre, en fonction des simplifications informatiques, de reproduire artificiellement des phénomènes de consommation liés aux comportements humains (Kashif *et al.* (2012); Amouroux *et al.* (2013)). Cette approche à base de simulation fournit aux experts de nouveaux outils pour mettre en relation comportements et consommation électrique. La difficulté est alors de pouvoir considérer le système dans son ensemble, de pouvoir reproduire les phénomènes émergents qui intéressent les experts, tout en le simplifiant suffisamment pour être compréhensible et manipulable par ces experts. Pour répondre à cette difficulté, nous proposons la plateforme de modélisation et de simulation SMACH.

2 De l'ergonomie à la simulation multi-agent

Ces dernières années, nous avons développé au sein d'EDF la plate-forme SMACH (Haradji *et al.* (2012); Amouroux *et al.* (2013); Huraux *et al.* (2014)). L'idée de cet outil est de permettre aux experts du domaine de simuler l'activité quotidienne des foyers. En particulier, nous essayons d'anticiper comment l'organisation des activités s'adapte face à de nouvelles contraintes (prix de l'électricité, nouveaux systèmes techniques, événements inattendus, *etc.*).

L'approche que nous présentons dans cet article repose sur deux propriétés fondamentales : (1) modéliser simplement le comportement humain dans le contexte de la consommation électrique des foyers et (2) proposer des mécanismes réalistes d'auto-adaptation (lors des simulations) de l'activité du foyer. Afin de satisfaire ces deux propriétés tout en maintenant une accessibilité aux experts du domaine, nous avons choisi une description de l'activité humaine basée autour des principes de l'action/cognition située (Relieu *et al.* (2004); Salembier & Pavard (2004); Salembier & Zouinar (2004); Haué (2004)).

Nous considérons que reproduire un comportement humain en simulation, c'est reproduire pour une étude donnée une suite d'échanges entre l'individu simulé et son environnement. La réussite de cet objectif repose sur la validation des comportements produits par la simulation. En ce sens, la validation représente un enjeu important. Le niveau de validité des comportements est l'un des choix de modélisation à réaliser en fonction des besoins du simulateur. Dans le cadre de nos travaux, cette étape a nécessité plusieurs ajustements. Au départ basée sur une description très précise, nous avons choisi aujourd'hui de reproduire l'activité sous la forme d'un enchaînement de tâches par l'habitant. Le modèle est centré sur certaines caractéristiques

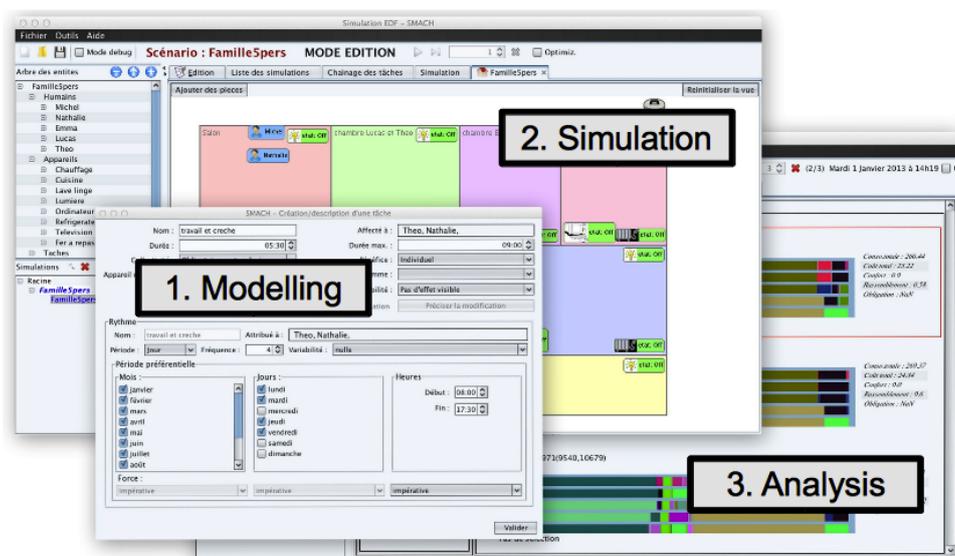


FIGURE 2 – Interface de la plateforme SMACH

de l'activité humaine (priorité de réalisation, pré-condition entre les tâches, période préférentielle dans la journée,...) et sur des éléments de coordination plutôt que sur des mécanismes fins de décision (e.g. un raisonnement sur les états internes des habitants comme la fatigue et la faim, leurs intentions, ...). Nous défendons l'idée que le réalisme peut ne pas être visé lorsqu'il n'est pas nécessaire à l'objectif d'aide à la décision (Haradji *et al.* (2012)). Dans la section suivante, nous présentons le fonctionnement de la plateforme SMACH.

3 La plateforme SMACH

Comme illustré sur la figure 3, l'utilisation de la plateforme peut se résumer en trois grandes étapes : (1) Modélisation d'un foyer par la description de ses habitants, tâches et appareils sous la forme d'un scénario (2) Réalisation de simulations où chaque habitant est représenté par un agent (3) Analyse des résultats sous forme de traces d'activité et de courbes de consommation.

3.1 Création d'un scénario

Nous appelons scénario la description d'un foyer dans la plateforme SMACH. Un scénario est défini comme un ensemble d'habitants, de tâches et d'appareils électriques.

L'élément central est la notion de tâche. Ces tâches peuvent nécessiter des appareils ménagers, qui sont définis dans un environnement. Nous parlerons d'*actions* pour la réalisation des tâches par les habitants. Une tâche peut avoir un rythme correspondant à une certaine régularité dans sa réalisation. Un rythme est caractérisé par une fonction périodique qui associe chaque moment de la simulation à une probabilité de réalisation.

Chaque membre du foyer est représenté par un agent habitant. Les principales propriétés d'un habitant sont un ensemble d'actions qu'il peut effectuer, un âge, un sexe et un niveau de

frilosité.

Un appareil électrique est caractérisé par une consommation électrique. Certains appareils disposent de programmes de consommation. Ces programmes sont définis comme des courbes de consommation provenant de la base de données REMODECE² caractérisant la consommation d'appareils en situation réelle. Les appareils peuvent avoir une puissance en veille (comme la télévision, l'ordinateur, ...) et un thermostat (chauffage).

En se basant sur les données d'un foyer réel ou sur les caractéristiques d'une typologie de foyer, l'utilisateur de la plateforme renseigne les éléments du scénario. Cette étape se fait à l'aide d'une interface de modélisation développée avec les ergonomes partenaires et pensée spécifiquement pour les experts du domaine de l'énergie.

3.2 Simuler la vie quotidienne des foyers

Dans SMACH, chaque membre du foyer est représenté par un agent. Ces agents sont synchrones et sélectionnent, à chaque pas de temps, l'action à exécuter en se basant sur leur priorité. De plus, un niveau de confort thermique est calculé pour chaque agent en utilisant l'équation de Fanger (van Hoof (2008)).

Les habitants d'un foyer peuvent communiquer pour échanger des informations ou demander de l'aide. Pour ce faire, nous utilisons un protocole de communication multi-agent inspiré de FIPA-ACL (FIPA consortium (2003)). Les performatifs principaux sont : *es-tu prêt à ... ?*, *que fais-tu ?*, *as-tu fait ... ?*, *peux-tu faire ... ?* et leur réponse *je fais ...*, *non*, *oui*, *plus tard*.

Un agent habitant sélectionne l'action à effectuer en fonction de ses priorités qui dépendent notamment de ses communications (*i.e.* incitation des autres agents à faire quelque chose). La priorité d'une action est réévaluée à chaque pas de temps et l'agent exécute l'action ayant la priorité maximale parmi les actions réalisables. Une action est réalisable si toutes ses préconditions sont satisfaites (*e.g.* l'action *repasser* nécessite la réalisation préalable de l'action *laver le linge*) et si l'appareil électrique éventuellement nécessaire est disponible (*e.g.* l'action *repasser* nécessite un fer à repasser). Les agents peuvent également réaliser des actions pour tenter d'améliorer leur confort thermique (changer les températures de consigne, ouvrir une fenêtre, ...) (Plessis *et al.* (2014)).

Pendant la simulation, l'utilisateur peut suivre l'évolution de la consommation instantanée, les températures dans le logement, le confort des individus et leurs diagrammes d'activité. Cette représentation graphique de l'activité (Fig. 3.3.2), où chaque couleur correspond à une tâche, permet de visualiser le comportement d'un habitant durant la journée. Ainsi, ce diagramme permet de retrouver l'activité précise d'un habitant à chaque instant de la simulation.

3.3 Mettre en relation activité humaine et consommation électrique

La plateforme a été conçue avec les ergonomes d'EDF de manière à permettre aux experts de simuler l'activité d'un foyer. Une fois les caractéristiques des occupants et de l'habitat renseignées, il est possible de simuler l'organisation du foyer (par exemple l'organisation d'une famille le matin avant de partir au travail) et de lancer de multiples simulations produisant ainsi une diversité de comportements.

2. REMODECE : European database on residential consumption - <http://remodece.isr.uc.pt>

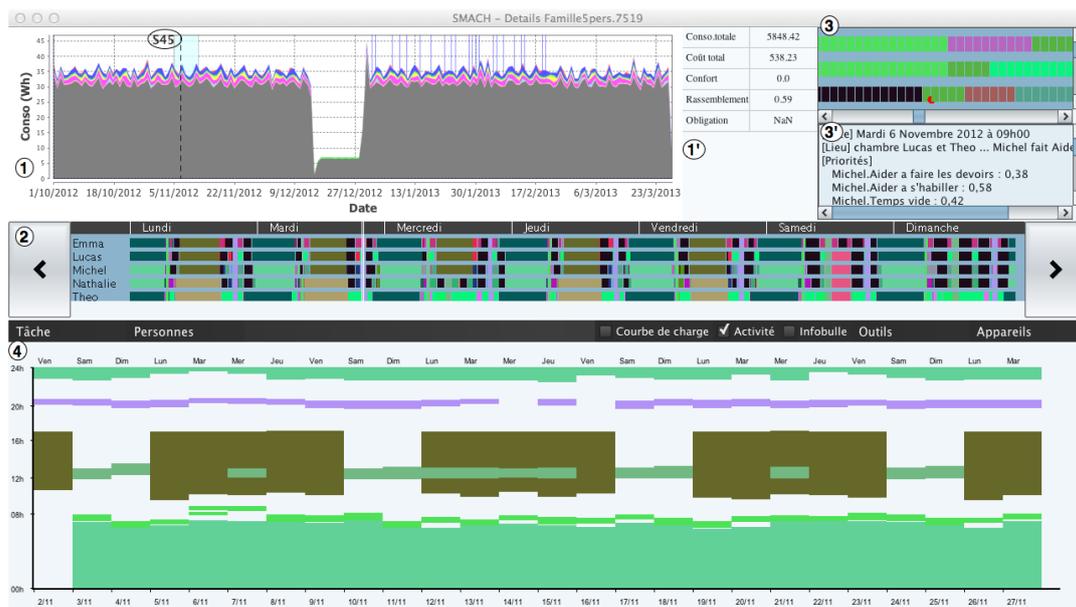


FIGURE 3 – IHM en mode analyse

Chaque simulation est liée à une courbe de consommation électrique permettant aux experts de comparer et d'évaluer les comportements en relation avec leurs consommations. La plateforme permet ensuite de revenir sur la modélisation du foyer pour tester de nouvelles caractéristiques. Cet aller-retour au sein de la plateforme entre modélisation et simulation permet aux experts métiers d'anticiper une consommation d'énergie résultant de comportements humains.

L'IHM d'analyse de simulation de SMACH (Fig. 3.3) permet de voir simultanément les éléments suivants : (1) la consommation électrique globale avec une couleur par appareil ; (2) l'activité des habitants au cours d'une semaine (S25 sélectionnée en (1)) ; (3 et 3') le détail de l'activité par minute et (4) les rythmes effectifs des actions (l'axe horizontal représente l'écoulement des jours alors que l'axe vertical celui des heures de la journée).

4 Simulations participatives

La plateforme SMACH permet également la réalisation de simulations participatives dont l'utilisation a été montrée pertinente dans le cadre des SMA appliqués aux sciences humaines et sociales (Guyot & Drogoul (2004)). Leur intérêt est double : d'une part valider les modèles utilisés et, d'autre part, fournir un outil semi-automatique pour l'étude des comportements dans l'habitat.

4.1 Entretiens préalables avec des clients EDF

La simulation participative commence par la réalisation d'entretiens préalables dans des foyers. Ces entretiens, conduits par des ergonomes, permettent de dresser un portrait de l'activité dans le foyer afin de la transcrire sous forme de scénario dans SMACH. Un ou plusieurs



FIGURE 4 – La simulation participative dans SMACH

membres du foyer dressent un modèle de leur vie quotidienne : l'organisation du foyer, leurs habitudes, les repas, *etc.*

4.2 L'humain face à son avatar

Lors de la simulation participative, une personne réelle est placée face à une simulation de son quotidien où les autres membres de son foyer sont incarnés par des agents. En plus de donner ses retours et impressions sur la simulation, le sujet peut prendre le contrôle de son avatar en cours de simulation afin de modifier ses activités ou encore certaines actions liées au confort (éclairage, chauffage, vêture, ...). Son implication dans la simulation a pour but d'affiner les comportements produits automatiquement. Cela permet notamment d'éliciter le modèle d'activité pouvant comporter des oublis ou erreurs introduites lors des entretiens préalables ou pendant la réalisation du scénario par le modélisateur.

La figure 4 présente l'interface de simulation participative de la plateforme SMACH. Comme pour les simulations automatiques, nous recueillons l'ensemble des traces d'activités des avatars et des agents. Ces données nous permettent là encore d'étudier les comportements autour de l'énergie et de lier activité humaine et consommation électrique.

4.3 Mise en œuvre

Nous avons réalisé en fin d'année 2014 une expérimentation de simulation participative sur une dizaine de foyers bretons. Nous nous appuyons sur le projet OPCO mené par EDF R&D³

3. dans le cadre du dispositif "Une Bretagne d'avance"

pour des études sociologiques et statistiques afin d'évaluer l'impact d'un pilotage de la charge par des actions sur le chauffage en fonction de différentes variables d'influence (température extérieure, type de jour de la semaine, ...). Cette expérimentation nous a permis de récupérer des données quantitatives sur la consommation électrique de plus de 300 clients résidentiels. Nous avons sélectionné un échantillon de ces clients pour la réalisation de simulations participatives.

Dans le cadre de nos travaux, les résultats attendus sont d'une part la validation des comportements simulés et, d'autre part, l'évaluation des consommations par mesures de distance aux profils de consommation mesurés dans OPCO. La validation des comportements est basée sur l'observation par les ergonomes de la prise en main des personnes lors de la simulation participative, de leurs remarques et ressentis.

Suite à cette expérimentation, notre objectif est de fournir aux experts EDF un outil fonctionnel permettant de mieux comprendre les liens entre pratiques et consommation dans une logique d'efficacité énergétique. Concrètement, les modèles de consommation seront calibrés et la plateforme sera adaptée (IHM, type de résultats retournés, *etc*) à une ou plusieurs applications métiers comme la création de nouvelles offres tarifaires, la génération d'activités humaines pour les études sur la thermique des bâtiments, jeux sérieux, ...

5 Conclusion

Dans cet article, nous avons présenté la plateforme SMACH dont l'objectif est de combiner les techniques de simulation multi-agent avec des outils permettant aux modélisateurs, experts de la gestion de l'énergie et du comportement humain, d'observer, de comprendre, d'analyser et de modifier les comportements des habitants simulés.

Les simulations participatives permettent d'une part de valider le modèle de comportement des agents en confrontant les clients EDF aux situations simulées. Et d'autre part de comparer les consommations produites aux données réelles mesurées sur le terrain pour valider les modèles de consommation dans SMACH. Nos premiers résultats de comparaisons entre les courbes de consommation issues des simulations participatives et les courbes réelles sont encourageants. Les données réelles de consommation étant antérieures aux données simulées, nous avons notamment pu identifier certains écarts importants de consommation qui se sont avérés correspondre à des bouleversements récents dans l'organisation du foyer (changement d'emploi, nouveaux membres dans la famille, ...).

Nous travaillons actuellement sur la question de la simulation mettant en jeu plusieurs disciplines et expertises à l'aide de SIMLAB, un modèle multi-agent multi-niveau (Huraux *et al.* (2014)).

Références

- AMOUROUX E., HURAUX T., SEMPÉ F., SABOURET N. & HARADJI Y. (2013). Smach : Simuler l'activité humaine pour limiter les pics de consommation électrique. *Journées Francophones sur les Systèmes Multi-Agents (JFSMA)*, p. 51–60.
- ANDRÉ J.-C. & ROYER J.-F. (1999). Les fluctuations à court terme du climat et l'interprétation des observations récentes en termes d'effet de serre.
- BATTY M. & TORRENS P. M. (2001). Modeling complexity : the limits to prediction. *Cybergeo : European Journal of Geography*.

- FIPA CONSORTIUM (2003). *FIPA Communicative Act Library Specification and FIPA ACL Message Structure Specification*. Rapport interne, Foundation for intelligent physical agents.
- GUYOT P. & DROGOUL A. (2004). Designing multi-agent based participatory simulations. In *Proceedings of the 5th Workshop on Agent Based Simulations*.
- HARADJI Y., POIZAT G. & SEMPÉ F. (2012). L'activité humaine et la conception d'une plate-forme de simulation sociale. *Congrès de la SELF (Société d'Ergonomie de Langue française)*, p.41.
- HAUÉ J.-B. (2004). Intégrer les aspects situés de l'activité dans une ingénierie cognitive centrée sur la situation d'utilisation. *Activités*, **1**(2), 170.
- HURAUX T., SABOURET N. & HARADJI Y. (2014). A Multi-Level Model for Multi-Agent Based Simulation. In *Proc. of the 6th International Conference on Agents and Artificial Intelligence (ICAART)*, Angers, France.
- KASHIF A., PLOIX S., DUGDALE J. & LE X. H. B. (2012). Simulating the dynamics of occupant behaviour for power management in residential buildings. *Energy and Buildings (online pre-print)*.
- PLESSIS G., AMOUROUX É. & HARADJI Y. (2014). Coupling occupant behaviour with a building energy model - a fmi application. *Proceedings of the 10th International Modelica Conference*.
- RELIEU M., SALEMBIER P. & THEUREAU J. (2004). Introduction au numéro spécial activité et action/cognition située. *Activités*, **1**(2), 3–10.
- SALEMBIER P. & PAVARD B. (2004). Analyse et modélisation des activités coopératives situées. évolutions d'un questionnement et apports à la conception. *Activités*, **1**(1), 87.
- SALEMBIER P. & ZOUINAR M. (2004). Intelligibilité mutuelle et contexte partagé. *Inspirations conceptuelles et réductions technologiques*. *Activités*, **1**(2), 64–85.
- VAN HOOFF J. (2008). Forty years of Fanger's model of thermal comfort : comfort for all ? *Indoor Air*, **18**(3), 182–201.

Un système de RÀPC pour prévoir l'allocation de parcelles agricoles

Florence Le Ber¹, Laura Martin², Alain Mille³, Marc Benoît²

¹ ICUBE, Université de Strasbourg, ENGEES, CNRS, France
florence.leber@engees.unistra.fr

² SAD ASTER, UR 055, INRA Mirecourt, France
Marc.Benoit@mirecourt.inra.fr

³ LIRIS, Université Lyon1, UMR CNRS 5205, Lyon, France
alain.mille@univ-lyon1.fr

Résumé : Cet article présente un prototype de raisonnement à partir de cas développé dans le cadre d'une application en agronomie, pour prévoir l'implantation dans les exploitations agricoles d'une nouvelle culture énergétique, le miscanthus. Peu d'expériences étaient disponibles pour construire ce modèle, ce qui nous a conduits à intégrer des variantes dans le module de raisonnement, permettant ainsi des usages différents. Le modèle et ses usages sont discutés. **Mots-clés** : Raisonnement à partir de cas, Agronomie, Explication.

1 Introduction

Parmi les modèles à base de connaissances, le raisonnement à partir de cas (RÀPC), introduit par Riesbeck & Schank (1989), permet d'appréhender des domaines où les connaissances générales sont mal formalisées et où l'expertise repose sur l'expérience. Classiquement, le RÀPC est vu comme un mode de résolution de problèmes s'appuyant sur la réutilisation de solutions de problèmes déjà résolus. Un *cas* est la représentation d'un épisode de résolution de problème : il encode un problème et une solution de ce problème ainsi que des informations sur les liens entre ce problème et cette solution. L'ensemble des cas d'un système de RÀPC est regroupé dans une *base de cas* et un cas de cette base est appelé *cas source* (dont la partie problème est appelée *problème source*). Reasonner à partir de cas c'est résoudre un problème, appelé *problème cible*, en s'appuyant sur la base de cas et sur des connaissances du domaine d'application. Ce raisonnement s'effectue en deux étapes principales, la *remémoration* et l'*adaptation*. La remémoration consiste à sélectionner un cas source jugé « similaire » au problème cible. L'adaptation utilise le cas source remémoré pour résoudre le problème cible, souvent en modifiant la solution associée au cas source et en s'appuyant sur la différence entre le problème source et le problème cible. Cependant, dans certains domaines, le modèle générique de cas, problème-solution, exigeant une forme fixe aussi bien pour la partie problème que la partie solution, n'est pas directement opérationnel. Il faut alors introduire une étape d'élaboration du problème cible, comme présenté par Fuchs *et al.* (2006), permettant de reformuler les choses afin de pouvoir mobiliser les connaissances effectivement disponibles pour l'adaptation d'une solution existante.

Par ailleurs, dans de nombreuses applications du RÀPC, la définition et la modélisation des cas et des connaissances nécessaires aux étapes de remémoration et d'adaptation requièrent un long travail d'analyse commun entre experts du domaine et informaticiens (voir par exemple l'expérience relatée par Le Ber & Brassac (2008)). La notion de cas ne s'applique alors pas de la même façon, mais garde sa pertinence si on considère que l'on cherche à expliquer une situation

à partir de l'explication trouvée dans une situation similaire : il s'agit d'une approche formalisée par Schank *et al.* (1994) permettant de formaliser des domaines en cours d'exploration comme ici ou relevant de l'argumentation comme en jurisprudence par exemple.

Nous présentons ici une démarche et un outil de RÀPC développé dans le domaine agronomique, dans une situation où l'expérience même est rare, puisqu'il s'agit de l'introduction d'une nouvelle culture, le miscanthus, dans des exploitations agricoles. Le modèle du RÀPC s'applique donc à une expérience en construction. Les questions à résoudre sont : quel est le processus de décision qui conduit un agriculteur à planter du miscanthus dans une parcelle ? ce processus est-il généralisable ou adaptable pour prévoir ce que fera un autre agriculteur sur une autre parcelle ? Un prototype, dénommé SAMM, a été réalisé et expérimenté à partir d'enquêtes menées en Bourgogne, dans une zone où l'exploitation du miscanthus s'est développée il y a peu (Martin, 2014). L'article est organisé comme suit : après un état de l'art, nous exposons la problématique agronomique ; nous détaillons ensuite le modèle SAMM, puis en discutons avant de conclure.

2 Travaux connexes

Notre travail s'inscrit dans les travaux liant intelligence artificielle et agronomie. Plus précisément, il s'inspire de Osty *et al.* (2008) qui ont discuté de l'intérêt du RÀPC comme outil de modélisation pour l'agronomie des territoires (Benoît *et al.*, 2012) en s'appuyant sur une expérience de développement d'un système pour l'analyse et la comparaison d'enquêtes en exploitations. Plus largement, il existe de nombreux systèmes liant RÀPC et sciences environnementales, la plupart adoptant des approches très numériques, proches de méthodes d'apprentissage. On citera en particulier Du *et al.* (2012), qui intègrent des relations spatiales entre voisins pour mesurer la similarité entre parcelles et prédire leur utilisation (bâti, forêt ou culture). Des travaux plus anciens avaient déjà utilisé le RÀPC pour l'analyse de données géographiques, par exemple pour la classification des sols (Holt & Benwell, 1996).

Dans tous ces travaux fondés sur des approches numériques, il est peu fait mention des connaissances mises en jeu par les acteurs, comme nous avons entrepris de le faire ici. Le travail décrit par Osty *et al.* (2008) se fondait sur des connaissances d'acteurs préformalisées par des chercheurs. Girard *et al.* (2001) ont pris en compte des connaissances d'acteurs dans une approche de modélisation similaires au RÀPC, mais sans aller au développement d'un système. De fait, les systèmes incorporant explicitement des expériences des acteurs, comme celui décrit par Bosch *et al.* (1997) sur les *rangeland* neo-zélandais, sont rares.

Le travail présenté ici s'inscrit en effet dans une situation d'utilisation de l'expérience, de type *Explanation Based Reasoning* (Schank *et al.*, 1994). L'approche d'un raisonnement analogique fondé sur les explications est bien documenté dans la littérature avec plusieurs synthèses toutes produites dans les années 2004-2005. Roth-Berghofer (2004) reprend les fondements de la notion d'explication aussi bien au niveau philosophique que dans le cadre d'un système expert. Il distingue les explications cherchant à répondre aux questions : qu'est ce concept (qui intervient dans la décision), quelle est la justification d'un fait, comment une décision s'est-elle mise en place, dans quel but telle décision a-t-elle été produite, quel est le contexte cognitif de telle décision. Les auteurs énumèrent quelques bonnes propriétés de *bonnes* décisions : fidélité, suffisance, facilité d'élaboration, efficacité. Dans leur synthèse, Sørmo *et al.* (2005) s'intéressent spécifiquement à la capacité d'un système expert à fournir les explications permettant à

un utilisateur de s'approprier le raisonnement et ses conclusions. Le RÀPC se prête particulièrement bien à cet exercice en gardant explicitement les explications spécifiques à chaque cas. Un cas, exprimé dans le registre de l'utilisateur, avec des explications faisant appel à des concepts partagés, permet de plus facilement réutiliser un cas pour en résoudre un autre tout en fournissant les clés de compréhension du domaine. L'ouvrage préfacé par Leake & Mcsherry (2005), reprend ces différentes façons de considérer, représenter et raisonner sur les explications.

3 Problématique

Le *Miscanthus x giganteus* – dit miscanthus – est une plante herbacée originaire d'Asie, cultivée en France depuis 2006 (Schnitzler, 2011). Cette plante est pérenne, elle est cultivée pendant 15 à 20 ans sur une même parcelle, et peut produire une biomasse importante tout au long de ce cycle. Elle est cultivée pour des usages exclusivement non alimentaires, comme écomatériau, combustible pour produire de la chaleur et pour le paillage.

L'extension de l'implantation du miscanthus en France est importante et peut constituer une tendance lourde au vu de l'intérêt récent des dirigeants politiques. Cet intérêt s'explique en partie par son potentiel pressenti en termes de bilan énergétique et de bilan environnemental : le miscanthus nécessite peu d'apports, fertilisants ou produits de traitement, et peu d'interventions de l'agriculteur. En revanche le miscanthus a des exigences en matière de substrat, il se développe préférentiellement dans les sols profonds et humides et peut se trouver en concurrence avec d'autres cultures, comme le maïs. Ainsi, modéliser la localisation du miscanthus dans les territoires permet d'identifier et d'anticiper les risques d'une concurrence d'usage des sols et plus globalement de la réorganisation des territoires, en cas de forte demande énergétique.

Les connaissances sur les processus d'allocation du miscanthus sont encore peu nombreuses. Les travaux existants portent principalement sur les facteurs d'adoption du miscanthus dans les exploitations agricoles, en particulier sur les facteurs économiques. Plus récemment d'autres facteurs ont été étudiés, comme la taille de l'exploitation agricole, le capital humain, les contraintes techniques ou en ressources, l'accès à l'information et la confiance (Bocquého, 2012). Or modéliser la localisation du miscanthus requiert de prendre en compte les choix des agriculteurs et de coupler des variables biophysiques et humaines. Il est nécessaire pour cela de recueillir et d'analyser un grand nombre de données et d'informations plus ou moins complexes.

À cette fin, des enquêtes ont été réalisées en 2011 puis renouvelées l'année suivante auprès de treize exploitations agricoles. Pour établir les règles d'allocation du miscanthus, il a été demandé aux enquêtés de décrire leur exploitation îlot par îlot, en explicitant leurs décisions d'allocation de culture. Nous disposons ainsi de la description fine des îlots et des objectifs de l'enquêté. Cette description permet de collecter des valeurs relatives et subjectives nécessaires à prendre en compte pour allouer le miscanthus dans le parcellaire des agriculteurs (Martin *et al.*, 2012). Nous recherchons aussi les règles d'allocation spatiales du miscanthus choisies par les agriculteurs dans la situation réelle et dans deux mises en situation imaginées : 1) les îlots réellement implantés en miscanthus n'existent pas – il est alors demandé à l'enquêté où le miscanthus aurait été implanté ; 2) l'enquêté souhaiterait implanter d'autres îlots en miscanthus – il lui est alors demandé dans quels îlots supplémentaires et sous quelles conditions.

4 Le modèle SAMM

Le système construit doit permettre de prévoir si un agriculteur va décider ou non d'implanter du miscanthus dans une de ses parcelles. Pour cela on dispose d'éléments de connaissances extraits des enquêtes, qui se présentent de la façon suivante : un ensemble de parcelles dans lesquelles a été implanté ou non du miscanthus, chaque parcelle est décrite par un ensemble descripteurs et assortie d'une ou plusieurs règles exprimant le choix de l'agriculteur. Par exemple (1) dans l'expression suivante « *Une parcelle pas drainée : très mauvaise, enfin humide, humide, très humide (...) j'ai tout le temps vu en jachère (...) ce n'est pas le même prix : je dis "hop, je fais le miscanthus dedans"* », on voit que la décision de l'agriculteur se fonde sur le caractère humide de la parcelle et sur son occupation précédente par une jachère peu rémunératrice.

Ces éléments ont permis de développer le prototype SAMM, dont la base de connaissances, le module de raisonnement, et enfin l'implantation informatique sont présentés ci-dessous.

4.1 Base de connaissances

4.1.1 Base de cas

Dans notre application, un cas est défini comme une expérience particulière d'insertion territoriale du miscanthus décidée par un agriculteur. Le couple problème-solution est une parcelle agricole et son potentiel d'implantation du miscanthus. Il est représenté par un vecteur d'attributs-valeurs, composé de deux parties (cf. figure 1) :

1. la partie problème décrit la parcelle par 32 variables influençant le potentiel d'implantation du miscanthus : ces variables relèvent de six catégories (géométrique, d'accessibilité, etc.) et sont décrites par 159 valeurs – ce jeu des descripteurs est appelé DP ;
2. la partie solution d'un cas décrit le potentiel d'implantation du miscanthus par une seule variable prenant trois valeurs : non implantable en miscanthus (valeur 0), implanté en miscanthus (valeur 1) et implantable avec conditions (valeur 2).

La base de cas du prototype SAMM comprend 82 parcelles pour lesquelles le potentiel d'insertion territoriale du miscanthus a été explicité par les agriculteurs lors des enquêtes en exploitations agricoles ; il s'agit donc de parcelles pour lesquelles la solution est connue. L'organisation de cette base de cas est plate, mais, pour faciliter la remémoration des cas sources, des index sont utilisés. Ce second jeu de descripteurs est appelé DI. Il correspond aux éléments qui ont effectivement pesé dans la décision de l'agriculteur, l'hydromorphie de la parcelle et son occupation précédente dans l'exemple (1) cité ci-dessus.

4.1.2 Connaissances de similarité et d'adaptation

Les connaissances de similarité ont été élaborées à partir des résultats des enquêtes sur les décisions des agriculteurs. Différents niveaux et types de similarité sont considérés :

- un niveau numérique global : le nombre de descripteurs partagés entre les cas source et cible, noté n ; on peut considérer l'ensemble DP ou l'ensemble DI ;
- un niveau sémantique local : les valeurs des descripteurs communs sont comparées deux à deux ; si elles sont égales, la distance locale est nulle ; si elles sont différentes, la distance

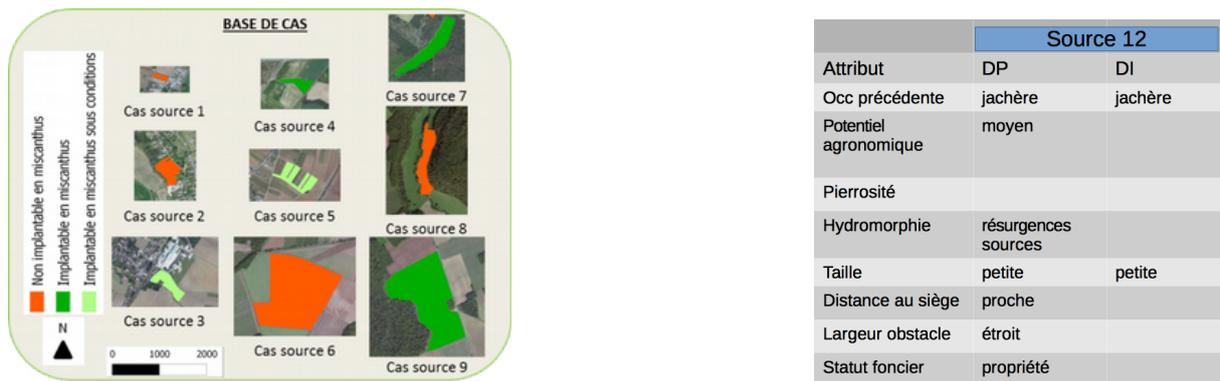


FIGURE 1 – Quelques cas sources (parcelle, potentiel d'implantation) et détail des attributs d'un cas ; le potentiel d'implantation est représenté ici par une couleur (0/rouge, 1/vert, 2/vert clair)

varie entre 5 et 20, selon qu'elles influencent de la même façon ou non le potentiel d'implantation ; par exemple la valeur d'hydromorphie "terre humide" est similaire à la valeur "résurgences sources", mais distante de la valeur "terres séchantes" ;

- un niveau sémantique global : la moyenne des mesures locales à savoir, distance globale (source, cible) = $\sum_{i=1}^n \text{distance locale (source}_i, \text{cible}_i) / n$ où source_i (resp. cible_i) est un descripteur du problème source (resp. cible).

Les connaissances d'adaptation reposent sur les règles de décision des agriculteurs. Ces règles sont de deux types : les règles de décision génériques et non situées dans l'espace (dites aspatiales) et les règles de décision spatialisées (liées à une parcelle). Nous disposons de 96 règles de la forme $\langle \text{prémisses (descripteurs), conclusion (potentiel d'insertion)} \rangle$.

4.2 Module de raisonnement

4.2.1 Remémorer des cas sources

La remémoration consiste à identifier un ou plusieurs cas sources de la base de cas susceptibles d'aider à la résolution du problème cible. Elle se déroule en trois temps : l'appariement du problème cible aux problèmes sources, l'évaluation de la similarité entre les problèmes et la sélection finale des cas sources. Dans le prototype SAMM, l'appariement du problème cible aux cas sources consiste en un appariement de vecteurs. L'évaluation de la similarité entre chaque problème source et le problème cible s'appuie sur les mesures définies ci-dessus. La sélection est réalisée selon différentes combinaisons de ces mesures (cf. figure 2).

Quatre algorithmes de remémoration ont été définis. Ils se caractérisent par les descripteurs utilisés pour la comparaison des cas et la procédure de remémoration choisie :

- il y a deux ensembles de descripteurs permettant de comparer les cas : le premier, DP, est composé de l'ensemble des descripteurs du problème source ; le second jeu, DI, est composé des descripteurs qui indexent le cas source ;
- il y a deux procédures : (i) la première filtre d'abord les cas sources minimisant la distance globale puis les cas sources maximisant le nombre de descripteurs comparés et (ii) la seconde, à l'inverse, filtre dans un premier temps les cas sources maximisant le nombre de

descripteurs comparés puis les cas sources minimisant la distance globale.

```

Comparaison cas source 10499801 avec cas cible 10385231
distance pb entre odsP_jachere et odsP_jachere : 0
distance pb entre hydr_sechant et hydr_partiellement hydromorphe : 20
distance pb entre forme_pointue et forme_pointue et convexe : 10
distance pb entre foncS_0 et foncS_1 : 20
distanceIndex entre odsP_jachere et odsP_jachere : 0
distanceIndex entre forme_pointue et forme_pointue et convexe : 10
nombre d'éléments pb comparés : 4
distance globale pb : 12.5
nombre d'éléments index comparés : 2
distance globale index : 5.0
    
```

FIGURE 2 – Exemple de résultats obtenus après comparaison d'un cas source à un cas cible

4.2.2 Adapter la solution des cas sources

Dans les systèmes de RÀPC existants, l'adaptation s'opère quand au moins un descripteur du problème source est dissemblable du problème cible. Les connaissances utilisées s'appuient sur les relations de dépendance, dont l'acquisition peut être manuelle ou automatique. L'adaptation est ensuite réalisée par substitution de la solution source ou par transformation de la solution.

Dans le prototype SAMM, l'adaptation se fait par substitution en utilisant des règles décrites ci-dessus. Ces règles sont applicables dès lors que leurs prémisses sont similaires aux descripteurs du problème cible, i.e. avec une distance locale inférieure à 5. La substitution se fait après une étape de sélection des différentes solutions proposées par les règles pouvant être appliquées à un même problème cible. Cette sélection repose sur un algorithme d'adaptation. En pratique, trois algorithmes d'adaptation ont été implantés dans le système. L'algorithme pessimiste (ADAPT0) sélectionne en priorité les règles dont la conclusion vaut 0 ; l'algorithme optimiste (ADAPT12) sélectionne les règles dont la conclusion vaut 1 ou 2 ; l'algorithme prudentiel (ADAPT3) sélectionne les règles applicables au cas cible ayant les plus grandes prémisses. Le choix de l'algorithme est laissé à l'utilisateur selon ses objectifs de prédiction.

4.3 Implantation et tests

Le prototype SAMM a été implanté dans l'environnement de développement Eclipse¹. Dans l'état actuel, l'utilisateur peut charger différentes connaissances et bases de cas, permettant notamment d'appliquer le modèle sur d'autres territoires, sous réserve de certaines précautions d'usages. L'utilisateur peut aussi facilement tester différents algorithmes et paramètres de remémoration et d'adaptation grâce à un volet de configuration.

Des tests ont ensuite été conduits pour comprendre le fonctionnement du système. Ils ont été élaborés puis mis en œuvre à partir de différents paramètres d'expérimentation parmi lesquels :

- le choix de l'algorithme de remémoration, décliné en quatre modalités correspondant aux combinaisons des mesures et procédures définies ci-dessus ;
- le choix des règles à utiliser, avec deux modalités : les règles de l'agriculteur du cas source remémoré et les règles des agriculteurs de tous les cas sources de la base ;

1. www.eclipse.org

- le choix de l'algorithme d'adaptation, décliné en trois modalités, correspondant aux trois algorithmes présentés ci-dessus.

Les tests ont été menés en divisant la base de cas en deux sous-ensembles, le premier constituant les cas sources à remémorer, le second les problèmes cibles à résoudre. Les résultats obtenus sur ces problèmes cibles sont ensuite comparés à leur solution originelle (le potentiel d'implantation de la parcelle). Après analyse il apparaît que certaines procédures (ou combinaisons de paramètres) sont plus efficaces que d'autres, au sens où elles permettent de mieux retrouver les solutions originelles, sur ce jeu de données (Martin, 2014).

5 Discussion et conclusion

À ce stade et d'un point de vue thématique, pour les agronomes, le développement de SAMM a permis deux avancées majeures : (i) formaliser les règles de décision des agriculteurs à partir des résultats d'enquêtes complexes ; (ii) construire, en vue de prospective, des gammes de scénarios d'allocation du miscanthus. Toutefois, une difficulté demeure : les cas sont décrits par un petit nombre d'attributs, ce qui rend les appariements fragiles (les cas cible et source ne partagent le plus souvent que 2 ou 3 descripteurs). Ceci renforce la nécessité de bien choisir les règles qui permettent l'adaptation ou tout au moins de les choisir en connaissance de cause. Par exemple pour la prospective, on peut envisager un point de vue bas (les exploitants n'implantent du miscanthus que dans certaines parcelles, les plus adaptées) – on utilise alors l'algorithme pessimiste ADAPTO – ou haut (les exploitants implantent dès qu'ils le peuvent) – on utilise l'algorithme optimiste ADAPT12.

De fait, on se trouve ici dans une situation d'utilisation de l'expérience qui relève d'un raisonnement à partir d'explications. En effet, s'il semble possible d'élucider quelques explications générales sur les décisions (les règles énoncées par les agriculteurs pour expliquer leur décision), les cas eux-mêmes sont difficiles à adapter et les agriculteurs ne se sont pas prêtés au jeu d'utiliser un cas « similaire » pour dire ce qui est différent entre son cas et un autre cas pouvant expliquer pourquoi la décision est différente ou la même. La complexité du processus de raisonnement et l'importance de donner la main au décideur plaident dans le sens d'explorer ces différences avec les décideurs. Ce qui ressort vraiment clairement du travail réalisé, c'est que la description du cas et de ses explications est une excellente façon de rendre concrètes les connaissances issues des enquêtes. L'utilisation du raisonnement est intéressante spécifiquement pour les interactions qu'il permet, plutôt que pour sa capacité de prédiction. Le prototype est utile pour construire de nouvelles boucles d'enquête et de réalisation de prototypes qui seront cette fois vraiment orientés sur la prédiction.

La notion d'explication semble donc la plus intéressante à considérer dans l'exploration de connaissances décisionnelles et en particulier, dans ce contexte évolutif de l'introduction d'une nouvelle culture dans les territoires d'exploitations agricoles.

Remerciements

Cet article s'appuie sur le travail de thèse de Laura Martin. Elle a bénéficié de l'accueil et du temps de nombreux agriculteurs, ainsi que de M. Béjot (Bourgogne Pellets), M. Garnaud (SCA Déshydratation de la Haute-Seine) et Mmes Pierson et Petit (Chambre d'Agriculture de

Bourgogne). Xavier Dolques (ICube) a implanté le prototype SAMM.

Références

- BENOÎT M., RIZZO D., MARRACCINI E., MOONEN A.-C., GALLI M., LARDON S., RAPEY H., THENAIL C. & BONARI E. (2012). Landscape agronomy : a new field for addressing agricultural landscape dynamics. *Landscape Ecology*, **27**(10).
- BOCQUÉHO G. (2012). Risque, temps et adoption des cultures pérennes énergétiques : exemple du cas français. Thèse de AgroParisTech.
- BOSCH O. J. H., GIBSON R. S., KELLNER K. & ALLEN W. J. (1997). Using case-based reasoning methodology to maximise the use of knowledge to solve specific rangeland problems. *Journal of Arid Environments*, **35**, 549–557.
- DU Y., LIANG F. & SUN Y. (2012). Integrating spatial relations into case-based reasoning to solve geographic problems. *Knowledge-Based Systems*, **33**, 111–123.
- FUCHS B., LIEBER J., MILLE A. & NAPOLI A. (2006). Une première formalisation de la phase d'élaboration du raisonnement à partir de cas. In *14ème séminaire sur le raisonnement à partir de cas*.
- GIRARD N., BELLON S., HUBERT B., LARDON S., MOULIN C.-H. & OSTY P.-L. (2001). Categorising combination of farmers' land use practices : an approach based on examples of sheep farms in the south of France. *Agronomie*, **21**, 435–459.
- HOLT A. & BENWELL G. (1996). Case-Based Reasoning and Spatial Analysis. *Journal of the Urban and Regional Information Systems Association*, **8**, 27–36.
- LE BER F. & BRASSAC C. (2008). Étude longitudinale d'une procédure de modélisation de connaissances en matière de gestion du territoire agricole. *Revue d'Anthropologie des Connaissances*, **2**(2), 151–168.
- LEAKE D. & MCSHERRY D. (2005). Introduction to the Special Issue on Explanation in Case-Based Reasoning. *Artificial Intelligence Review*, **24**(2), 103–108.
- MARTIN L. (2014). Modéliser l'insertion territoriale du *Miscanthus x giganteus* à partir des décisions des agriculteurs : une approche exploitant le modèle du raisonnement à partir de cas. Thèse de l'Université de Lorraine.
- MARTIN L., WOHLFAHRT J., LE BER F. & BENOÎT M. (2012). L'insertion territoriale des cultures biomasses pérennes. Etude de cas sur le miscanthus en Côte d'Or (Bourgogne, France). *L'Espace Géographique*, (2), 138–153.
- OSTY P.-L., LE BER F. & LIEBER J. (2008). Raisonnement à partir de cas et agronomie des territoires – constructions croisées. *Revue d'Anthropologie des Connaissances*, **2**(2), 169–193.
- RIESBECK C. K. & SCHANK R. C. (1989). *Inside Case-Based Reasoning*. Hillsdale, New Jersey : Lawrence Erlbaum Associates, Inc.
- ROTH-BERGHOFER T. (2004). Explanations and case-based reasoning : Foundational issues. In *Advances in Case-Based Reasoning*, LNCS 3155, p. 389–403. Springer.
- R. C. SCHANK, A. KASS & C. K. RIESBECK, Eds. (1994). *Inside Case-Based Explanation*. Hillsdale, New Jersey : LEA.
- SCHNITZLER A. (2011). Miscanthus : L'homme cultive-t-il un nouvel envahisseur ? Rapport, Université Paul Verlaine de Metz, Laboratoire des Interactions Ecotoxicologie, Biodiversité, Ecosystèmes.
- SØRMO F., CASSENS J. & AAMODT A. (2005). Explanation in Case-Based Reasoning – Perspectives and Goals. *Artificial Intelligence Review*, **24**(2), 109–143.

Densité de trafic émergente pour des véhicules intelligents communicants guidés par heuristique

Philippe Morignot¹, Oyunchimeg Shagdar^{1,2}, Fawzi Nashashibi¹

¹ INRIA, équipe-projet RITS, Domaine de Voluceau, B.P. 105, 78150 Le Chesnay, France
 prenom.nom@inria.fr

² VEDECOM, 77, rue des Chantiers, 78000 Versailles, France

Résumé : Dans cet article, nous étudions la densité et le comportement émergent du trafic, composé de plusieurs centaines de véhicules intelligents, en fonction de la communication véhicule-à-véhicule (V2V) (pour qu'un véhicule perçoive le trafic) et d'heuristiques de planification de chemin dynamique (pour qu'un véhicule atteigne sa destination) en environnement urbain. Les modes de communication idéale / réaliste / aucune sont croisés avec les heuristiques boussole / vers-plus-peuplé / vers-moins-peuplé pour mesurer la vitesse moyenne de trajet de chaque véhicule, modélisé par un automate à états finis. Le modèle de communication V2V, basé sur des modèles de propagation de signal et sur MAC (*medium access control*), est présenté. Nos expériences, des simulations comprenant jusqu'à 400 véhicules en environnement urbain réaliste, montrent que la communication et les heuristiques conduisent à une meilleur vitesse moyenne globale qu'une absence de communication ; et que, à chaque fois qu'il y a un chemin secondaire, fuir le trafic conduit à des performances globales meilleures que suivre le trafic.

Mots-clés : Véhicules intelligents, Systèmes Multi-Agents, Communication.

1 Introduction

La conduite copérative automatisée fait l'objet de nombreux travaux de recherche dans des domaines scientifiques comme la robotique, l'intelligence artificielle, les systèmes multi agents et la communication. Le but est d'obtenir une conduite collective à la fois sûre et intelligente. Si des études montrent le potentiel des communications V2V (véhicule-à-véhicule) pour une conduite automatisée plus sûre et plus souple (Sakaguchi *et al.* (1999), Bento *et al.* (2012), Xu *et al.* (2002)), il n'est pas encore clair de savoir si les technologies de standardisation peuvent intégrer les strictes demandes de la conduite automatisée. De plus, si la décision dans le contrôle de véhicules individuels est basée sur les communications V2V, la performance de ces communications devrait beaucoup influencer la "qualité" du comportement collectif. Plus encore, puisque les décisions liées à la planification de chemin influent sur la densité de trafic, elles influent donc sur la performance des communications sans fil. A notre connaissance, l'interdépendance entre communication et comportement de véhicules automatisés est encore peu explorée.

Xu *et al.* (2002) ont étudié comment le taux de pénétration de la communication améliore l'effort de freinage de véhicules individuels. Sakaguchi *et al.* (1999) ont défini un protocole de communications entre et à l'intérieur de pelotons de véhicules pour le contrôle sécuritaire des pelotons. Mehani & de La Fortelle (2007) définissent des échanges de messages de haut niveau entre véhicules et superviseur (l'infrastructure), qui sont requis pour la planification de trajectoire sur des routes qui se croisent.

Populaire *et al.* (1993) et Briot & Demazeau (2001) ont étudié la communication sous le paradigme des systèmes multi agents, mais définissent des protocoles de communication pour le langage naturel en soi (*les actes de langage*) et ignorent l'émission / réception physique de messages.

Dans cet article, nous combinons différents modes de communication V2V avec différentes heuristiques de planification de chemin dynamique, sur une population de plusieurs centaines

de véhicules intelligents, pour observer une éventuelle convergence (émergence ou phénomène global non prévu par chaque agent individuel) vers un trafic fluide. Pour étudier l'influence de la communication sur la qualité du comportement collectif, des modèles de communication V2V, particulièrement un modèle de propagation signal et un modèle MAC (Medium Access Control), sont utilisés.

La section suivante présente notre modèle et la section 3 présente nos résultats expérimentaux ; la dernière section résume notre contribution.

2 Modèle

2.1 Environnement

Nous considérons un environnement urbain avec des rues orientées exclusivement nord-sud et est-ouest (type quartier de Manhattan). Les rues perpendiculaires se croisent aux intersections ; les rues parallèles sont séparées par la même distance. Pour ne pas limiter le mouvement des véhicules, nous supposons que les rues bouclent aux bordures de l'environnement (environnement circulaire), i.e., l'extrémité la plus à droite (resp. en haut) d'une rue est-ouest (resp. nord-sud) rejoint son extrémité gauche (resp. en bas). Une rue est composée de deux voies, droite et gauche ; les véhicules circulent sur la voie de droite (conduite nord-américaine). Les feux de croisement sont modélisés aux 4 voies entrantes de chaque intersection, 2 par 2, tous avec la même phase (pas d'onde verte).

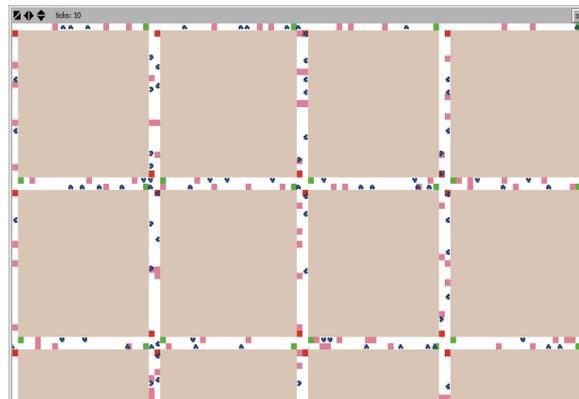


FIGURE 1 – Topologie de l'environnement : les véhicules intelligents (icônes bleues) circulent sur la voie de droite des rues (pixels blancs) jusqu'à une destination (pixels roses), chaque croisement possède un feu de signalisation par voie entrante (pixels verts / rouges).

2.2 Véhicules intelligents

2.2.1 Automate à états finis

Le comportement de chaque véhicule intelligent est modélisé par un automate à états finis (une boucle sur 5 états séquentiels) :

1. Aller tout droit sur sa voie, et accélérer (augmenter la vitesse de 1 pixel/tic par tic) jusqu'à la vitesse limite (identique pour tous les véhicules), ou freiner à cause d'autres véhicules (évitement d'obstacles) ou de feux rouges (respect du code de la route) ;
2. Si un véhicule intelligent atteint un croisement, communication possible avec les autres véhicules dans les 4 directions (N / S / E / O) pour déterminer les véhicules perçus ;

3. Choisir une direction de sortie parmi les 4 possibles (N / S / E / O) ;
4. Suivre un plan de 1- à 4-étapes (voir 2.2.2) pour aller à cette sortie, en évitant une collision avec d'autres véhicules à ce croisement ;
5. Aller à l'état 1, jusqu'à ce que le temps limite (fin de simulation) soit écoulé.

Un véhicule intelligent reste sur une voie (état 1) sans dépasser, i.e., nous supposons qu'il y a une ligne continue au milieu de chaque rue.

Quand un véhicule intelligent a atteint sa destination (état 1), une nouvelle destination aléatoire lui est assignée (tout point sur une voie qui n'est pas à un croisement), pour que les voitures roulent constamment.

2.2.2 Croisement

Les véhicules intelligents suivent un plan de 1- à 4- étapes pour traverser un croisement et atteindre la voie de sortie qu'ils ont choisi par heuristique (état 4 de l'automate précédent). Les rues sont composées de deux voies et les véhicules circulent sur la voie de droite d'une rue, ce qui conduit à 4 pixels par croisement, notés de 1 à 4 (voir Fig. 2). Si un véhicule, $V1$, doit faire un virage à droite, il va tout droit jusqu'au pixel 1, puis fait un virage à droite, puis va tout droit (plan en 1 étape) ; Si $V1$ doit aller tout droit au croisement, il va tout droit jusqu'au pixel 1, puis jusqu'au pixel 2, puis va tout droit (plan en 2 étapes) ; Si $V1$ doit faire un virage à gauche, il va tout droit jusqu'au pixel 1, puis jusqu'au pixel 2, puis fait un virage à gauche jusqu'au pixel 3, puis va tout droit (plan en 3 étapes) ; Si $V1$ doit faire demi-tour, il va tout droit jusqu'au pixel 1, puis jusqu'au pixel 2, puis fait un virage à gauche au pixel 3, puis fait un second virage à gauche au pixel 4, puis va tout droit (plan en 4 étapes).

Chaque changement de direction, pour un virage à droite ou à gauche, est réalisé en additionnant / retranchant 90 degrés à la direction courante du véhicule (voir e.g. Au & Stone (2010) pour des modèles de trajectoire plus précis).

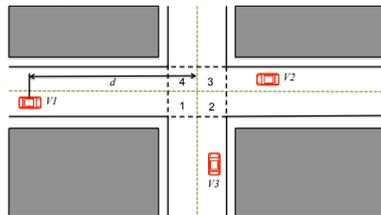


FIGURE 2 – 4 pixels à un croisement, notés de 1 à 4, pour un plan jusqu'à 4 étapes pour la traversée d'un croisement. Les 3 autres cas (N / S / O) peuvent être obtenus par rotation.

Les mêmes règles qu'en 2.2.1 sur l'accélération / freinage sont valables pour la traversée d'un croisement : ceci est suffisant pour assurer un évitement d'obstacle, et donc la sécurité pendant cette phase. Par exemple, nous avons observé que des véhicules ayant choisi de faire un virage à gauche à un croisement, attendent sur leur pixel 2, dans le plan en 3 étapes précédent, que d'autres véhicules passent sur la voie opposée, avant d'aller tout droit à leur pixel 3 lorsque plus aucun véhicule ne passe, et ensuite aller tout droit vers leur voie de sortie.

2.2.3 Heuristiques

A un croisement (état 4 de l'automate en 2.2.1), chaque véhicule intelligent peut toujours choisir entre 4 directions : à droite, tout droit, à gauche et derrière (le véhicule fait demi-tour). Juste avant d'entrer dans un croisement, chaque véhicule intelligent utilise une heuristique,

identique pour tous les véhicules, pour choisir une des 4 sorties du croisement menant à sa prochaine voie.

Boussole. Le véhicule intelligent est supposé équipé d'un capteur GPS, qui permet de déterminer une ligne droite (vol d'oiseau) entre sa position courante et sa destination. Cette heuristique choisit l'angle le plus petit en valeur absolue entre la direction d'une voie de sortie N / S / E / O et cette droite idéale, implémentant ainsi le chemin le plus court. Aucune communication V2V n'est nécessaire.

Vers le moins peuplé ("Anti Fourmi"). Comme avant, les 4 angles en valeur absolue formés par les 4 directions de sortie avec le droite idéale sont calculés. Si l'angle le plus petit correspond au chemin le plus court (voir heuristique précédente), le deuxième plus petit angle pourrait aussi conduire le véhicule à sa destination, e.g., aller tout droit et prendre un virage à droite est presque équivalent à faire un virage à droite puis faire un virage à gauche — le chemin est seulement un peu plus long. Pour choisir entre ces 2 angles, le véhicule effectue une communication V2V pour percevoir le nombre d'autres voitures dans chacune de ces deux directions. Si un des 2 angles possède un trafic strictement inférieure à l'autre, cet angle-là est choisi (ce peut donc être le deuxième) : cette heuristique cherche ainsi à éviter le trafic. Sinon (trafic identique pour les 2 angles), le premier meilleur angle est choisi (heuristique Boussole).

Vers le plus peuplé ("Fourmi"). Cette heuristique, proche dans l'esprit des algorithmes d'optimisation par colonie de fourmis Dorigo & Stützle (2004), est identique à la précédente sauf sur le choix entre les deux meilleurs angles : si un des 2 meilleurs angles possède un trafic strictement inférieur à l'autre, le *deuxième* meilleur angle est choisi. Cette heuristique, l'opposée de la précédente (un test est inversé), choisit l'angle le plus peuplé entre ces deux meilleurs angles.

Cependant, lorsque les deux dernières heuristiques choisissent le deuxième meilleur angle plusieurs fois de suite, cela peut conduire les véhicules à dévier "trop" de leur chemin vers leur destination et de la rater. Aussi, lorsqu'un véhicule est "assez proche" (e.g., un paté de maison) de sa destination, il bascule de l'heuristique "Fourmi" ou "Anti Fourmi" à l'heuristique "Boussole". Cette zone d'attraction de la destination force l'heuristique "Boussole" vers la fin du chemin d'un véhicule.

2.3 Communication V2V

2.3.1 Propagation du signal en environnements urbains

En raison de sa simplicité, le modèle two-ray ground (TRG) est couramment utilisé pour l'environnement LOS (Line of Sight). Nous étudions l'applicabilité de TRG à un scénario d'intersection de route en comparant les modèles aux résultats du modèle de lancer de rayon pour le système IEEE 802.11p pour la bande de fréquence 10 MHz et le puissance de transmission 20 dBm Bib (2010). TRG calcule la qualité du signal en utilisant un modèle d'espace libre pour les distances proches et un modèle à 2 rayons pour les distances lointaines :

$$L_{trg}[dB] = \begin{cases} L_{fs} = 20 \log_{10}(4\pi \frac{d}{\lambda}), & \text{si } d \leq d_c \\ L_{tr} = 20 \log_{10}(\frac{d^2}{h_t h_r}), & \text{sinon} \end{cases} \quad (1)$$

Ici, d est la distance entre l'émetteur et le récepteur, λ est la longueur d'onde, et h_t and h_r sont les hauteurs d'antenne du transmetteur et du récepteur, respectivement. d_c est l'étendue de la zone de Fresnel $d_c = 4\pi h_t h_r / \lambda$.

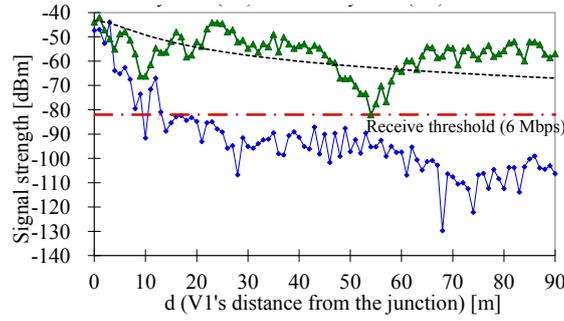


FIGURE 3 – Les puissances reçues à V2 et V3 de V1 sont calculées en utilisant les modèles ray-tracing et TRG.

Sur Fig. 3, l'estimation de lancer de rayon est faite pour un scénario de croisement de route en supposant que les bâtiments hauts sont à chaque coin (voir Fig. 2). Les puissances de réception aux noeuds V2 et V3, qui sont de 15 m et 30 m depuis l'intersection respectivement, sont calculées pour des signaux émis par V1. Comme montré sur la figure, V1 et V2 sont sur la même route, qui coupe celle de V3. Malgré le fait que 2 rayons seulement soient pris en compte, TRG peut estimer grossièrement la puissance du signal qui est sur la même route que le transmetteur (V2). La caractéristique de la puissance de réception à V3, d'un autre côté, baisse rapidement sous le seuil de réception de 6 Mbps de taux de codage (qui est le réglage typique pour des applications de sécurité) dès que la distance de V1 depuis la jonction est supérieure à 10 mètres, ce qui est dû à l'évidence à un blocage de signal. Nos résultats indiquent que la puissance du signal du récepteur peut être estimée par le modèle TRG pour des véhicules qui sont sur la même route que le transmetteur, et qu'il est difficile d'attendre des communications entre véhicules sur des routes qui se croisent, à moins que la distance à la jonction soit très petite.

2.3.2 Modèle MAC

L'ETSI a défini les trames CAM (*cooperative awareness message*) pour l'échange d'information périodique entre véhicules et infrastructure, ce qui permet aux usagers de la route de connaître l'existence et l'état d'un autre usager Bib (2009). Nous proposons que les trames CAM servent effectivement à des applications de planification de chemin. Cette sous section présente un modèle MAC simple pour caractériser la performance de transmission CAM entre véhicules. Comme nous considérons un seul type de trafic d'information et que les CAMs sont des paquets d'émission, il suffit d'exprimer le comportement du protocole par une chaîne de Markov à une dimension. Dans ce cas, la probabilité de réception réussie d'une trame CAM donnée à un récepteur donné est calculée comme

$$P_s = (1 - \tau)^{N-1}. \quad (2)$$

Où N est le nombre d'usagers et τ est la probabilité d'accès au canal, qui peut être formulé comme :

$$\tau = q \times \left[1 + \frac{W - 1}{2(1 - p_k)} \right]^{-1}. \quad (3)$$

Ici, W est la taille minimale de la fenêtre de contention, p_b est la probabilité du canal d'être occupé, et q est la probabilité d'un paquet en attente au noeud. Les détails du modèle Markov sont présentés dans Shagdar *et al.* (2013).

3 Expérimentations

Des expériences avec le modèle de la section 2 ont été réalisées en utilisant l'environnement logiciel NetLogo Wilensky (1999), un outil graphique de simulation multi agents — des expériences réelles avec des centaines de véhicules intelligents automatisés semblent impossibles de nos jours. La figure 1 illustre le scénario urbain lorsque la distance entre deux rues parallèles est supposée être de 200 mètres.

Nous avons mené des investigations sur les performances des comportements "Fourmi" et "Anti Fourmi", lorsque les véhicules 1) ne communiquent pas, 2) ont une information complète sur tous les autres véhicules (communication idéale, i.e., aucune erreur de communication), et 3) communiquent suivant la technologie IEEE 802.11p (communication réaliste). La communication "réaliste" est implémentée en suivant les modèles PHY (propagation de signal) et MAC introduits dans la section précédente. En appliquant le modèle TRG, nous obtenons une portée de transmission de 500 mètres et une portée de perception de 800 mètres pour les véhicules sur la même rue. Sans perdre beaucoup en généralité, nous supposons que les véhicules à un croisement sont dans la portée de transmission les uns des autres. Dans nos expériences, les véhicules individuels se comportent selon un automate à états finis (voir Section 2) et envoient des messages CAM (200 octets) toutes les 100 ms. La probabilité de succès pour un message CAM d'un véhicule donné est calculée pour les autres véhicules dans la portée de transmission selon le modèle MAC (Eqs. (3) – (2)).

Chaque point des figures 4 et 5 représente une simulation jusqu'au tic d'horloge 10 000.

3.1 Influence de la communication

La figure 4 (a) compare l'écart-type du nombre de véhicules sur des rues : dans la communication idéale, la stratégie "Fourmi" crée une distribution de véhicules extrêmement biaisée (e.g., la plupart des véhicules sont dans la même rue), alors que la stratégie "Anti Fourmi" montre un usage de la route plus équilibré. C'est prévisible parce que dans la stratégie "Fourmi", les véhicules sont attirés par les autres véhicules, et créent ainsi des routes encombrées. Au contraire, dans la stratégie "Anti Fourmi", les véhicules essaient de prendre des rues désertes, créant ainsi une situation où il n'y a ni route trop encombrée ni route déserte.

Pour la communication "réaliste", si l'écart-type est encore une fois plus grand pour "Fourmi" que pour "Anti Fourmi", la différence est petite, ce qui montre que les deux stratégies échouent à produire le comportement collectif espéré. Le comportement collectif peut être très différent du comportement "attendu" s'il y a des erreurs de communication.

La figure 4 (b) compare la probabilité de succès d'une communication réaliste pour les cas "Fourmi" et "Anti Fourmi" : la probabilité de succès se dégrade avec l'augmentation du nombre de véhicules (augmenter "N" dans Eq. (2) conduit à une dégradation de la probabilité de succès). Plus généralement, on s'attend à ce que, puisque le comportement "Fourmi" crée des routes encombrées, l'erreur de communication est plus importante dans le cas "Fourmi" que dans le cas "Anti Fourmi". Cependant, si la probabilité de succès est meilleure pour "Anti Fourmi", la différence est très petite, ce qui est expliqué par la figure. 4 (a). A cause des erreurs de communication, les stratégies "Fourmi" et "Anti Fourmi" ne produisent pas de comportements prévisibles, i.e., celui de la communication "idéale". La distribution des véhicules n'est pas trop biaisée pour le cas "Fourmi", donc "N" dans Eq. (2) n'est pas très différent pour les cas "Fourmi" et "Anti Fourmi".

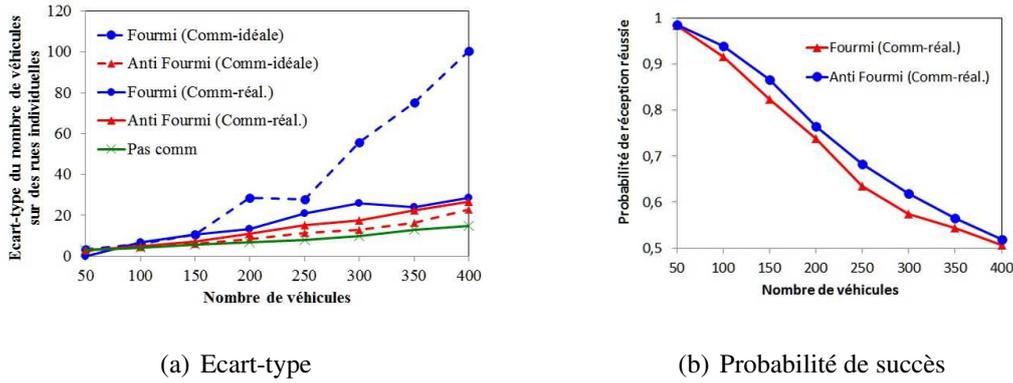


FIGURE 4 – (a) Comparaison de l'écart-type du nombre de véhicules sur les rues. (b) Probabilité de succès des trames CAM.

3.2 Comportements émergents

Pour chaque simulation et à chaque tic d'horloge, la moyenne de la vitesse instantanée vers la destination (mesurée en pixel / tic) de chaque véhicule est calculée. Cette vitesse, mise à jour à chaque fois qu'un véhicule atteint sa destination, est la distance manhattan (incluant l'environnement circulaire / bouclé) entre la destination précédente (ou le point de départ, au tic = 0) et la destination courante, divisé par le nombre de tics écoulés pour qu'un véhicule fasse le trajet d'une destination à l'autre.

Pour chaque simulation (chaque point de Fig. 5 (a) et (b)), une stabilisation de la vitesse-vers-destination vers une certaine valeur est observée, après une montée abrupte au début de la courbe, durant 2 à 3000 tics. Pour éviter cette augmentation et ne comparer que des valeurs stabilisées, nous traçons (i) la moyenne des valeurs sur les 6000 derniers tics parmi 10 000 (voir Fig. 5 (a)), ce qui laisse 4000 tics pour passer cette montée, et (ii) la dernière valeur (voir Fig. 5 (b)).

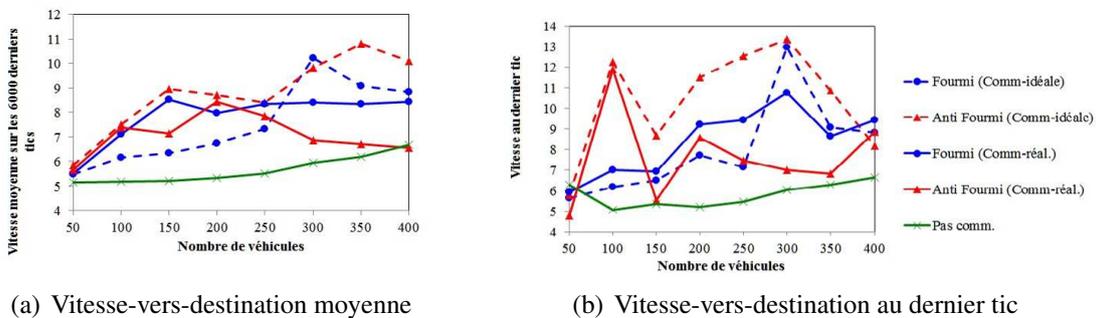


FIGURE 5 – (a) Moyenne sur les 6000 derniers tics d'horloge de la vitesse-vers-destination instantanée moyenne de chaque véhicule, en fonction du nombre de véhicules. (b) Dernière valeur (au tic = 10000) des mêmes mesures que dans (a).

Les figures 5 (a) et (b) suggèrent que : (1) L'influence de la communication est importante sur les performances du comportement émergent, i.e., "pas de communication" fournit la pire vitesse-vers-destination, des véhicules suivant aveuglément leur chemin vers leur destination sans communiquer est pire que toute autre heuristique de planification de chemin avec communication. (2) Lorsque la communication est idéale, le comportement "Anti Fourmi" est nettement meilleur que "Fourmi", i.e., lorsqu'il y a le choix, s'échapper du trafic conduit à de meilleures vitesses-vers-destination que suivre le trafic. (3) Lorsque la communication est réaliste, la différence entre les comportements "Anti Fourmi" et "Fourmi" est plus petite en comparaison de la communication idéale, mais l'heuristique "Fourmi" se comporte légèrement mieux

pour un grand nombre de voitures.

4 Conclusion

Dans cet article, nous avons étudié les performances du trafic émergent de plusieurs centaines de véhicules uniformes, en fonction des modes de communication V2V et d'heuristiques de planification dynamique de chemin en environnement urbain. Une implémentation du modèle suggère que la communication et les heuristiques conduisent à une meilleure vitesse moyenne émergente (i.e., un trafic global avec de meilleures performances) qu'une absence de communication (i.e., des véhicules ne faisant que suivre aveuglément leur boussole jusqu'à destination) ; et que fuir le trafic conduit à des performances globales meilleures que suivre le trafic.

Nos travaux futurs comprennent la différenciation des véhicules (rôle / groupe / agent) et une topologie différente pour l'environnement (e.g., rues courbes).

Références

- (2009). ETSI TR 102 638 ; Intelligent Transport Systems (ITS) ; Vehicular communications ; Basic set of Applications. Draft V1.0.4.
- (2010). IEEE Standard for Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirement, Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 802.11p-2010.
- AU T.-C. & STONE P. (2010). Motion planning algorithms for autonomous intersection management. In *Bridging the Gap Between Task and Motion Planning*.
- BENTO L. C., PARAFITA R. & NUNES U. (2012). Intelligent traffic management at intersections supported by v2v and v2i communications. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, p. 1495–1502 : IEEE.
- J.-P. BRIOT & Y. DEMAZEAU, Eds. (2001). *Principes et Architecture des Systèmes Multi-Agents*. Traité IC2, Hermès.
- DORIGO M. & STÜTZLE T. (2004). *Ant Colony Optimization*. MIT Press.
- MEHANI O. & DE LA FORTELLE A. (2007). Trajectory planning in a crossroads for a fleet of driverless vehicles. In *Computer Aided Systems Theory—EUROCAST 2007*, p. 1159–1166. Springer.
- POPULAIRE P., DEMAZEAU Y., BOISSIER O. & SICHMAN J. (1993). Description et implémentation de protocoles de communication en univers multi-agents. *Ieres Journées Francophones sur l'IAD et les SMA, AFCET & AFIA, Toulouse, IRIT*.
- SAKAGUCHI T., UNO A. & TSUGAWA S. (1999). Inter-vehicle communications for merging control. In *Vehicle Electronics Conference, 1999.(IVEC'99) Proceedings of the IEEE International*, p. 365–370 : IEEE.
- SHAGDAR O., DANIEL A. & PRIMAK S. (2013). Beacon delivery over practical v2x channels. In *ITS Telecommunications (ITST), 2013 13th International Conference on*, p. 68–73 : IEEE.
- WILENSKY U. (1999). NetLogo, available online at <http://ccl.northwestern.edu/netlogo>. *Center for Connected Learning and Computer-Based Modeling. Northwestern University, Evanston*.
- XU Q., HEDRICK K., SENGUPTA R. & VANDERWERF J. (2002). Effects of vehicle-vehicle/roadside-vehicle communication on adaptive cruise controlled highway systems. In *Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th*, volume 2, p. 1249–1253 : IEEE.

Problèmes d’optimisation dans les jeux avec GHOST

Florian Richoux

LINA UMR 6241, Université de Nantes, France
florian.richoux@univ-nantes.fr

Résumé : Cet article présente GHOST, un solveur d’optimisation combinatoire qu’un développeur de jeux de stratégie en temps réel (RTS) peut utiliser comme une boîte noire pour résoudre tout problème modélisé comme un problème de satisfaction/optimisation de contraintes. Nous montrons une manière de modéliser trois différents problèmes de RTS dans ce formalisme, chacun de ces problèmes appartenant à un niveau d’abstraction spécifique, en utilisant le jeu RTS StarCraft comme environnement de test. Sur chacun de ces trois problèmes, GHOST retourne des solutions de très bonne qualité en l’espace de quelques dizaines de millisecondes.
Mots-clés : Game AI, stratégie, temps réel, contrainte, solveur, optimisation, combinatoire.

1 Introduction

On peut voir les jeux comme une simplification du monde : le terrain est plus petit, les règles plus simples et moins nombreuses, rendant ainsi les possibilités plus limitées. Cependant, les jeux sont assez riches pour proposer des environnements dynamiques complexes restant difficile pour un ordinateur à appréhender, faire de la prédiction, avoir une compréhension globale de la situation courante et ainsi de prendre une décision. Ceci est en particulier vrai lorsque l’information est incomplète, forçant l’ordinateur à inférer l’état global du jeu à partir de fragments d’informations. C’est le cas avec les jeux de stratégie en temps réel (RTS), où le brouillard de guerre tel que l’a théorisé Clausewitz masque les coups des adversaires. Les RTS constituent ainsi un terrain favorable à la conception de techniques d’intelligence artificielle qui pourraient par la suite s’appliquer avec succès à d’autres applications que les jeux.

De nombreuses techniques d’IA sont utilisées dans les RTS. Sans en faire une liste exhaustive, nous pouvons citer l’apprentissage par renforcement, le *case-based reasoning*, la programmation bayésienne, les *potential fields*, etc. Nous recommandons au lecteur les articles d’enquêtes de Ontañón *et al.* (2013) et Robertson & Watson (2014) pour avoir un balayage complet des techniques d’IA modernes mises en œuvre dans les RTS.

Cependant, il y a peu de travaux en IA pour RTS utilisant la programmation par contraintes à travers la modélisation de problèmes de satisfaction/optimisation de contraintes (CSP/COP). Entre autres, nous retrouvons des algorithmes de *branch and bound* pour l’optimisation de *build order* (Churchill & Buro, 2011), c’est-à-dire la planification de séquences d’actions afin d’atteindre un but fixé le plus rapidement possible. Nous trouvons également des modèles de génération procédurale de cartes esthétiques résolus par un algorithme évolutionniste (Lara-Cabrera *et al.*, 2014). Les CSP/COP ont l’avantage d’offrir un cadre pratique et homogène capable de modéliser un grand nombre de problèmes combinatoires et d’optimisation, et proposent un ensemble varié d’algorithmes pour leur résolution.

Ce formalisme est largement utilisé en IA pour résoudre par exemple des problèmes de recherche de chemin, d’ordonnancement et de logistique. Les algorithmes de résolution de CSP/COP sont habituellement conçus pour être généraux, capable de traiter de n’importe quel problème modélisé dans ce formalisme. Outre sa généralité, il est également intuitif de modéliser un problème avec un CSP/COP. Tout ceci amène des conditions idéales pour la conception

2 Architecture de GHOST

GHOST est une bibliothèque écrite en C++¹, sous licence GNU GPL v3, conçue pour traiter de n’importe quel problème de RTS du moment qu’il soit modélisé par un CSP ou un COP. Il s’agit d’une génération et d’une extension du solveur ad-hoc développé pour optimiser la construction de murs défensifs dans le célèbre RTS StarCraft (Richoux *et al.*, 2014).

Avant de rentrer plus dans les détails de GHOST, nous rappelons très brièvement ce que sont formellement un CSP et un COP. Un CSP est un triplet (V, D, C) où V est un ensemble de variables, D l’ensemble des domaines de chaque variable (c’est-à-dire l’ensemble de valeurs que peut prendre chaque variable), et C l’ensemble des contraintes sur les variables de V. Le but est de trouver une valeur dans D pour chaque variable dans V de manière à ce que chaque contrainte de C soit satisfaite. Un COP est la même chose, avec en plus une fonction objectif à maximiser ou minimiser.

GHOST vise deux types d’utilisateurs :

- L’utilisateur de base, qui souhaite simplement utiliser GHOST pour résoudre un problème pré-encodé et disponible dans la bibliothèque. Cet utilisateur a seulement besoin d’instancier ses variables, leur domaine, ses contraintes et éventuellement sa fonction objectif pour décrire l’instance de son problème. Suite à quoi il n’a plus qu’à appeler la fonction `solve` pour lancer la recherche d’une solution. Tout ceci ne requiert que cinq lignes de C++.
- L’utilisateur avancé, qui souhaite résoudre un problème qui n’est pas proposé par la bibliothèque. GHOST a été conçu afin de faciliter l’implémentation de nouveaux problèmes sans devoir modifier une seule ligne de code du solveur, et sans qu’aucune expertise en programmation par contrainte ne soit requise. De plus, notre solveur a été pensé de manière à ce qu’il y ait le moins possible de paramètres, afin d’éviter la tâche fastidieuse et chronophage d’étalonnage des paramètres avant d’obtenir de bons résultats.

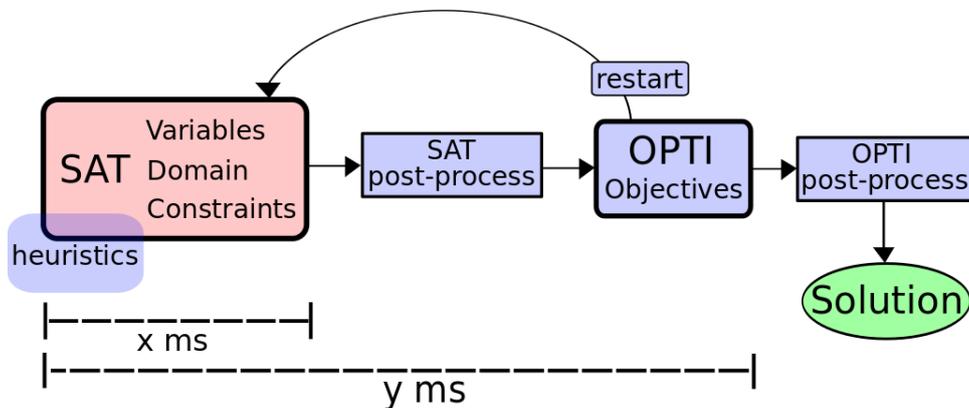


FIGURE 1 – L’architecture du solveur : en rouge, la boucle interne de satisfaction, avec un *timeout* de x millisecondes. En bleu, les mécanismes d’optimisation. Le processus dans son ensemble, en excluant le post-processus d’optimisation, s’exécute en moins de y millisecondes.

Le solveur repose sur une méta-heuristique de recherche locale, implémentant l’algorithme ADAPTIVE SEARCH de Codognet & Diaz (2001). L’utilisateur exécute le solveur via l’appel

1. Code source disponible sur github.com/richoux/GHOST.

dans la figure 1. Elle est composée de deux principales boucles : la boucle externe d’optimisation, contenant elle-même la boucle interne de satisfaction. Cette dernière, en rouge dans la figure, tente seulement de trouver une solution acceptable, c’est-à-dire vérifiant chacune des contraintes. Il est possible d’appeler `Solver::solve` sans définir de fonction objectif, auquel cas le solveur retourne la première solution acceptable qu’il trouve avant l’expiration de x millisecondes, ou une valeur “erreur” le cas échéant.

Si une fonction objectif est donnée, alors la boucle d’optimisation, en bleu dans la figure, est activée. La fonction objectif peut influencer la boucle de satisfaction si des heuristiques facultatives de choix de variables et/ou de valeurs sont implémentées. La boucle d’optimisation ne fait que comparer la qualité d’une solution trouvée par la boucle de satisfaction par rapport aux solutions précédentes, conserve la meilleure et relance la boucle de satisfaction pendant x millisecondes au plus, le tout avec une échéance de $y > x$ millisecondes. Dans un certain sens, la boucle d’optimisation applique un échantillonnage de Monte Carlo.

La partie optimisation applique également deux post-processus optionnels, le premier à la sortie de la boucle de satisfaction, permettant d’épurer une solution trouvée en écartant toutes variables ne contribuant pas à la satisfaction des contraintes, le second à la sortie de la boucle d’optimisation, permettant d’appliquer des dernières tentatives d’optimisation pour encore améliorer la qualité de la meilleure solution trouvée. Le temps d’exécution de cette dernière est fixé à $(y/100)$ ms, ce qui la rend négligeable face au temps total d’exécution.

Concernant les paramètres, il n’en existe que trois : les valeurs x et y du temps de calcul accordé respectivement aux boucles de satisfaction et d’optimisation, ainsi que la longueur de la liste tabou. En effet, l’algorithme ADAPTIVE SEARCH utilise une liste tabou des variables explorées de manière à ne pas revisiter la même variable trop tôt durant la phase de recherche. Le troisième paramètre correspond ainsi au nombre d’itérations que le solveur doit attendre avant de pouvoir revisiter une variable déjà explorée. En réalité, nous avons également implémenté un mécanisme échappatoire afin de ne pas rendre la liste tabou stricte, en autorisant le solveur à tout de même sélectionner une variable tabou s’il n’y a vraiment aucun autre meilleur choix. La liste tabou telle qu’elle est implémentée dans GHOST est plutôt à considérer comme étant une liste de priorité.

Pendant les expérimentations, nous avons réalisé que la même valeur de ce paramètre menait quasi systématiquement aux performances optimales, lorsqu’il est fixé à la valeur $|V| - 1$ où $|V|$ est le nombre de variables dans le modèle. Ce paramètre prend alors la valeur $|V| - 1$ par défaut et ne nécessite pas a priori d’étalonnage, cependant il reste toujours possible à l’utilisateur de la modifier s’il le souhaite.

L’implémentation de GHOST est composée de cinq classes C++ principales : `Variable`, `Domain`, `Constraint`, `Objective` et `Solver`². Un utilisateur avancé doit implémenter ses propres classes en héritant de `Variable`, `Domain`, `Constraint` et `Objective`. Par exemple, pour créer une classe de variables représentant des unités, ceci peut être fait par `class Unit : public Variable`. Certaines classes sont paramétrées par d’autres classes, comme par exemple `Domain` qui doit connaître la classe des variables. Ainsi, déclarer le domaine d’un problème de sélection de cibles peut être fait par `class TargetDomain : public Domain<Unit>`.

2. Voir le manuel de GHOST (richoux.github.io/GHOST/) pour plus de détails

Dans cette section, nous détaillons les modèles CSP/COP et les résultats obtenus par GHOST sur trois problèmes à différents niveaux d’abstraction, du moins abstrait au plus abstrait : les contrôles réactifs (également appelés la “micro-gestion”), la tactique et la stratégie. Nous tenons à souligner qu’aucune optimisation du solveur n’a été faite pour traiter ces trois problèmes.

3.1 Contrôles réactifs : la sélection de cible

Le problème de sélection de cible consiste à assigner à chaque unité d’un groupe une cible sur laquelle tirer dans la *frame* courante (c’est-à-dire l’unité de temps du jeu, 1/24ième de seconde pour StarCraft). Autant la satisfaction de ce problème est facile, puisqu’il s’agit simplement d’affecter à une unité en mesure de tirer une cible vivante qui soit à sa portée, autant l’optimisation de ce problème peut être fortement combinatoire : dans beaucoup de RTS comme StarCraft, des pénalités s’appliquent en fonction de la nature de l’unité attaquante et de sa cible. À cela, il faut ajouter des dégâts de zone qu’inflige certaines unités (c’est-à-dire, sa cible et le voisinage de sa cible subiront des dégâts). Si l’objectif est de maximiser le nombre de cibles éliminées par exemple, l’affectation d’une cible à une unité a un impact sur le choix des autres unités.

Nous avons modélisé ce problème ainsi : une variable correspond à une unité du groupe que nous contrôlons. Leur domaine est l’ensemble des unités d’un groupe adverse. L’unique contrainte assure que chacune de nos unités vivantes et prêtes à tirer vise une cible vivante à sa portée, si une telle cible existe. Deux fonctions objectif ont été écrites pour ce problème : *Max damage*, où notre groupe tente d’infliger autant de dommage que possible, et *Max kill*, où notre groupe cherche à détruire le plus de cibles possibles.

Les expérimentations ont été conduites en simulant une bataille entre un groupe de 14 unités provenant de 6 types d’unité de StarCraft en miroir (les unités adverses sont les mêmes que les nôtres et dans une position symétrique). L’adversaire applique une stratégie de sélection aléatoire de cible. Les mouvements n’étant pas pertinent pour le problème de sélection de cible où l’on choisit sa cible pour la *frame* courante, les unités restent statiques dans nos expériences.

Nous avons lancé 100 simulations de bataille avec comme paramètres $x = 2\text{ms}$ et $y = 5\text{ms}$ pour les *timeouts*. Nous obtenons les taux de victoires avec *Max damage* et 82% avec *Max kill*, avec respectivement en moyenne 2,8 et 3 de nos unités survivant à la bataille, sur 14. Pour un temps de calcul extrêmement court, GHOST permet d’avoir une stratégie significativement meilleure qu’une sélection aléatoire.

3.2 Tactique : le wall-in

La tactique n’a pas donné lieu à d’intensives études dans les RTS. On peut noter les travaux de Čertický (2013) sur le problème du *wall-in* et amélioré par Richoux *et al.* (2014).

La technique du mur, ou *wall-in*, est une technique classique de joueurs de RTS afin de protéger l’accès de sa base en créant un goulot d’étranglement à l’entrée de celle-ci afin d’en faciliter sa défense en cas d’invasion. Ce mur se bâtit en jouxtant bout à bout ses propres bâtiments.

Le problème de satisfaction consiste seulement à placer des bâtiments (nos variables) à des positions dans l’espace (leur domaine) afin de créer un mur, à savoir une ligne continue de bâtiments, d’un point A à un point B (nos contraintes). Encore une fois, l’optimisation peut être de multiples natures. Nous en avons considéré trois : 1) composer le mur avec le moins

plus rapides à obtenir), et 3) avec le moins de trous possibles. En effet dans certains RTS comme StarCraft, deux bâtiments placés côte à côte peuvent en réalité être séparés par un trou de quelques pixels, selon la nature et la position de ces bâtiments. Certaines combinaisons peuvent entraîner des trous assez larges pour laisser passer de petites unités, typiquement la plus petite unité du jeu, le *zergling*, qui est de 16×16 pixels. Nous comptons donc comme étant un trou un écartement de 16 pixels ou plus entre deux bâtiments accolés.

Nous avons extrait 48 entrées de base venant de 7 cartes de StarCraft, et lancé quatre expérimentations de 4800 exécutions de GHOST chacune (100 fois chaque entrée) : la première sans optimisation, ne cherchant qu'à résoudre le problème de satisfaction laissant 160ms au solveur. Puis, nous avons lancé une expérimentation par fonction objectif, avec comme paramètres $x = 20\text{ms}$ et $y = 150\text{ms}$. L'optimisation par GHOST montre des gains significatifs, en particulier pour limiter le nombre de trous : pour 4,05 bâtiments en moyenne dans un mur sans optimisation, on se retrouve avec 2,56 bâtiments avec optimisation, avec 98,04% de solutions trouvées en moins de 150ms. Un indice de 1,99 en technologie est optimisé à 1,35 avec 97,54% de taux de résolution. Enfin, là où l'on avait 1,32 trou de 16 pixels ou plus en moyenne dans un mur, on se retrouve avec seulement 0,03 trou avec la fonction objectif associée, avec un taux de résolution de 97,5%. Pour 4800 entrées de base, 4680 murs ont été trouvés dont 4527 parfaits, c'est-à-dire sans aucun trou de plus de 16 pixels.

3.3 Stratégie : le build order

Le problème de *build order* est l'un des rares problèmes déjà traité via la programmation par contrainte dans Churchill & Buro (2011). Il s'agit d'un problème d'ordonnement d'actions, afin d'atteindre un but fixé en un minimum de temps. Ces actions peuvent être la production d'unités, la construction de bâtiments, ainsi que la recherche de technologies et d'améliorations.

Ce problème peut être vu comme un problème de permutation où les variables sont toutes les actions nécessaires pour atteindre le but donné, le domaine est l'ordre de ces actions et l'unique contrainte est de vérifier la dépendance des actions : souvent, une action A ne peut être effectuée qu'après des actions B et C.

Nous avons extrait le *build order* de 3647 parties jouées par des joueurs de haut niveau et avons comparé en combien de temps GHOST arrive à atteindre le même état de jeu obtenu au bout de 7 minutes de jeu en mode rapide (soit 10.000 *frames*). Ces 7 minutes de jeu en mode rapide correspondent à environ 11 minutes en mode normal, soit 660 secondes.

En moyenne sur 3647 *build orders*, GHOST ordonne les actions afin d'arriver au même état de jeu qu'un joueur humain de haut niveau atteint au bout de 656,02 secondes en mode normal (ne se passant rien d'intéressant en moyenne durant les 3,98 dernières secondes) en seulement 619,62 secondes, le tout avec des *timeouts* de $x = 20\text{ms}$ et $y = 30\text{ms}$. Ceci donne un ratio temps CPU / temps du *build order* de 0,007%. À noter que le taux de résolution en moins de 30ms est en moyenne de 94,4%.

En comparaison, Churchill & Buro (2011) arrivent 90% des fois à des *build order* de même durée que des joueurs de haut niveau, donc de qualité moindre que GHOST, après 3,735 secondes de calcul, donnant un ratio temps CPU / temps du *build order* de 1,5%.

En plus des 3647 parties jouées par des joueurs de haut niveau, nous avons extrait 8 *build orders* venant des meilleurs joueurs professionnels au niveau mondial, tous joueurs sud-coréens. Contrairement aux 3647 *build orders* précédents, ces parties-là ont été choisies "manuellement", expliquant leur faible nombre. Il en résulte que GHOST arrive en l'espace de 30ms à de meilleurs *build order* que les meilleurs joueurs au monde, avec en moyenne des temps de

bout de 643,38 secondes en moyenne. Ce temps est bien plus court que 660 secondes du fait que nous avons arrêté d’enregistrer le *build order* dès que les premières attaques commencent, situation qui arrive plus vite chez les joueurs de très haut niveau. GHOST a trouvé une solution en 30ms dans 96,3% des cas.

4 Conclusion et discussion

Nous avons introduit dans cet article GHOST, une bibliothèque d’optimisation combinatoire visant à résoudre les problèmes liés aux RTS modélisés par un problème de satisfaction/optimisation de contraintes. Les résultats obtenus par GHOST sur trois problèmes classiques dans les RTS à différents niveaux d’abstraction sont très encourageant pour poursuivre le développement de cette bibliothèque et y intégrer de nouveaux problèmes. En effet, le solveur intégré à GHOST trouve des solutions de très bonne qualité en l’espace de quelques dizaines de millisecondes seulement, sans qu’aucune optimisation du solveur ni étalonnage de paramètres (en dehors des *timeouts*) ne soit nécessaire. Les résultats ainsi obtenus sont bien souvent meilleurs que ceux que l’on trouve dans la littérature.

Le lecteur attentif aura réalisé que GHOST est en réalité une bibliothèque qui peut être utilisée pour n’importe quel problème CSP/COP, pas seulement ceux liés aux RTS ni même aux jeux. L’un de nos points de départ était de faire de GHOST une bibliothèque C++ très facile à intégrer dans les IA (dits *bots*) pour le jeu StarCraft. GHOST devrait être bientôt couplé avec la bibliothèque BWAPI 4, nous permettant d’avoir (presque) toutes les informations dont nous avons besoin sur les éléments du jeu, les événements, etc. De plus, une version de GHOST en C# est en cours, ce qui permettrait d’en faire un *plugin* pour le célèbre moteur de jeu *Unity*. À noter aussi que GHOST est en cours de transfert technologique, via un procédé de double licence (une GNU GPL et une propriétaire), auprès de la société de développement de jeu *Insane Unity*³ pour le développement de leur prochain jeu RTS. En d’autres mots, ce n’est pas le cœur de GHOST mais ses futures dépendances et intégrations qui en feront une bibliothèque d’optimisation combinatoire pour les RTS, et de manière plus générale, pour les jeux vidéos.

Références

- CHURCHILL D. & BURO M. (2011). Build order optimization in starcraft. In V. BULITKO & M. RIEDL, Eds., *AIIDE*, p. 14–19 : The AAAI Press.
- CODOGNET P. & DIAZ D. (2001). Yet another local search method for constraint solving. In *proceedings of SAGA’01*, p. 73–90 : Springer Verlag.
- LARA-CABRERA R., COTTA C. & FERNÁNDEZ-LEIVA A. J. (2014). A self-adaptive evolutionary approach to the evolution of aesthetic maps for a RTS game. In *IEEE World Congress on Computational Intelligence (WCCI)*.
- ONTAÑÓN S., SYNNAEVE G., URIARTE A., RICHOUX F., CHURCHILL D. & PREUSS M. (2013). A survey of real-time strategy game AI research and competition in starcraft. *IEEE Transactions on Computational Intelligence and AI in games*, 5(4), 1–19.
- RICHOUX F., URIARTE A. & ONTAÑÓN S. (2014). Walling in strategy games via constraint optimization. In *AIIDE : The AAAI Press*.
- ROBERTSON G. & WATSON I. (2014). A review of real-time strategy game AI. *AI Magazine*.
- ČERTICKÝ M. (2013). Implementing a wall-in building placement in starcraft with declarative programming. *arXiv*.

3. www.insaneunity.com

Simulation et Evaluation Multi-Agents dans les Jeux Sérieux collaboratifs : le projet SIMFOR

Erwan Tranvouez¹, Ali Oulhaci¹, Bernard Espinasse¹ et Sébastien Fournier¹

¹ Aix Marseille Université, CNRS, LSIS UMR 7296,13397, Marseille, France
{prenom.nom}@lsis.org

Résumé : La place de l'Intelligence Artificielle dans la conception des jeux sérieux tend à s'accroître pour apporter des capacités d'analyse et d'interactivité au cours d'une session de jeu. Les Systèmes Multi-Agents contribuent notamment à améliorer l'immersion des joueurs, via des agents joueurs autonomes et coopératifs, ainsi qu'en aidant au rapprochement des jeux sérieux des Environnements Informatique pour l'Apprentissage Humain. Ce papier illustre ces contributions en ajoutant des capacités de simulation orienté-agent et d'évaluation de l'apprentissage de procédures collectives au jeu sérieux SIMFOR dans la gestion de crise.

Mots-clés : Jeux Sérieux, évaluation des apprenants, Systèmes Multi-Agents, Simulation Orientée Agent, Gestion de crise.

1 Introduction

Les jeux sérieux ont pour but d'apprendre une ou plusieurs compétences à un ou plusieurs acteurs/joueurs humains. Un jeu sérieux répond en général à un objectif de formation précis, ciblant un domaine professionnel donné comme un corps de métier (ex. pompiers/sécurité (Querrec et al. 2003)(Amokrane 2010), pilotage (Lourdeaux et al. 2002)...), un corpus de connaissances (type scolaire ou universitaire) (Baudouin et al. 2007), ou en encore de qualités sociales (gestion des conflits, coopération...) (Wendel et al. 2012).

Les enjeux des jeux sérieux sont nombreux. En fonction des besoins des domaines d'applications, ils posent des questions en termes de méthodologie de conception (incluant les aspect de scénarisation ou de conception participative), de mode d'intégration des joueurs (environnement 3D, virtuel, ...), d'interaction, de nombre et type de joueurs (jeu collectif, joueur virtuel ou PNJ¹), ou encore d'ingénierie logicielle supportant ces différentes approches (architectures distribuées, architecture d'intégration, plateforme cible...). A ces enjeux s'ajoutent des considérations pédagogiques déjà abordées dans le domaine des Systèmes Tutoriaux Intelligents - STI - et des Environnements Informatique pour l'Apprentissage Humain - EIAH - (modèle de l'apprenant, modèle d'erreur, évaluation de l'apprentissage...).

Ce papier présente rapidement l'intérêt de l'Intelligence Artificielle pour les jeux sérieux et plus précisément au niveau de la simulation de l'environnement dans lequel évolue le ou les joueurs ainsi que les modalités d'évaluation des joueurs en interaction. Nous illustrerons ces enjeux sur un cas réel de jeu sérieux destiné à l'apprentissage de procédures collectives dans le cadre de la gestion de crise. Après avoir introduit le contexte d'apprentissage dans le projet SIMFOR et les enjeux qui y sont liés, nous montrerons dans un premier temps comment une approche multi-agents peut contribuer à la simulation des PNJ, ainsi qu'à l'évaluation des joueurs d'un jeu sérieux pour enfin décrire brièvement l'architecture logicielle retenue et réalisée.

¹ Personnage Non-Joueur (Non-played Character) : rôle d'un joueur humain endossé par le système.

2 Contexte d'apprentissage d'un jeu sérieux multi-joueurs : le projet SIMFOR

Le projet SIMFOR (cf. Fig. 1) est un jeu sérieux développé par la société SII² en partenariat avec la société Pixxim². Implémenté en C++, il permet une approche ludique de l'apprentissage de la gestion de crise en intégrant l'aspect multi-intervenants. L'objectif est de proposer aux utilisateurs un contexte de gestion de crise en temps réel proche de la réalité en termes d'environnement, de scénarios auto-évolutifs et d'acteurs (rôles). Reposant initialement sur une évaluation humaine des compétences et des PNJ (Personnages Non Joueurs) aux comportements simplifiés, le recours à l'IA est apparu pertinent à plusieurs niveaux : i) la simulation de comportement humain des acteurs non joués; ii) l'accompagnement et l'évaluation des apprenants dans leur formation et cela à l'aide d'une approche multi-agents.



FIG. 1 – Interfaces de jeux de SIMFOR.

SIMFOR est un jeu multi-acteurs permettant à plusieurs personnes (spécialistes du domaine et non professionnels) d'acquérir une compétence (différente ou identique). La gestion d'une crise majeure peut mobiliser plusieurs centaines d'intervenants, du Préfet dans son bureau au pompier sur le terrain, devant communiquer et collaborer pour rétablir la situation à un état normal. Le remplacement de joueurs humains par des joueurs virtuels permet de répondre à des objectifs de formation ciblés ou de palier à l'absence de joueurs. Si une approche de simulation orientée agent s'est rapidement imposée pour développer ces joueurs virtuels, l'approche agent est apparue également pertinente pour réaliser l'évaluation des apprenants.

L'enjeu de l'évaluation de l'apprentissage des joueurs humains dans un jeu sérieux prend de l'importance (Thomas et al. 2012). Elle reste cependant, pour de nombreux jeux, manuelle, prise en charge par des observateurs humains voire par le joueur lui-même. L'évaluation collective est dans ce contexte moins développée comme le reconnaît Wendel et ses collègues (Wendel et al., 2012), qui s'attachent à l'apprentissage de capacités d'interactions sociales. De fait l'apprentissage collaboratif cible majoritairement l'apprentissage de compétences homogènes, à l'image d'une classe où les élèves apprennent la même leçon (Giordani et al. 2005). A l'opposé, la gestion de crise est un processus collaboratif qui passe par la réalisation de plusieurs tâches différentes (dépendant du rôle de l'acteur ainsi que du sinistre), chaque joueur ayant ainsi un objectif pédagogique propre mais restent liés toutefois entre eux par l'objectif commun de gérer une crise.

Ainsi, SIMFOR vise un apprentissage de procédures collectives hétérogènes, réalisées par des acteurs aux rôles différents et présente en conséquence un besoin d'évaluation individualisé et collectif. La résolution d'une crise (incendie, pollution...) passe par le respect de toutes les procédures attribuées aux intervenants : dès lors l'évaluation individuelle peut avoir une influence sur l'évaluation collective, et réciproquement. Considérons, par exemple, un apprenant qui a bien effectué ses procédures, mais dont le but global n'a pas été atteint (e.g. pertes matérielles et humaines). Cet apprenant doit être évalué sur sa performance individuelle et collective pour déduire la raison de l'échec (manque de communication, procédure manquante d'un autre apprenant, ...).

² <http://www.groupe-sii.com> - <http://www.pixxim.fr>

3 Simulation Orientée Agent pour des PNJ adaptatifs réalistes

Le recours à un SMA pour mettre en place des avatars logiciels simulant le comportement de joueurs humain n'est pas nouveau en soi (Chang et al., 2005). L'intérêt des SMA pour ce type de problème est reconnu (Mathieu et al. 2013), aussi nous nous attacherons à présenter les particularités propres au jeu sérieux SIMFOR, illustrant ainsi concrètement l'apport de l'Intelligence Artificielle Distribuée à l'amélioration des capacités de simulation d'un jeu sérieux.

Pour concevoir nos agents joueurs, nous avons adopté une architecture BDI (*Beliefs, Desires, Intentions*), approche classique pour développer des agents aux comportements délibératifs, leur donnant ainsi une certaine capacité d'adaptation au travers de comportements complexes (Rao et Georgeff, 1995). Un outil de conception de comportements d'agents a également été développé, afin de faciliter la conception d'un scénario de jeu en termes de comportements et de types d'agents.. Les PNJ doivent également être capable d'interagir avec d'autres joueurs (réels ou non) et d'agir, au cours d'une session de jeu, en fonction de l'état de l'environnement physique (représenté dans SIMFOR par l'environnement 3D).

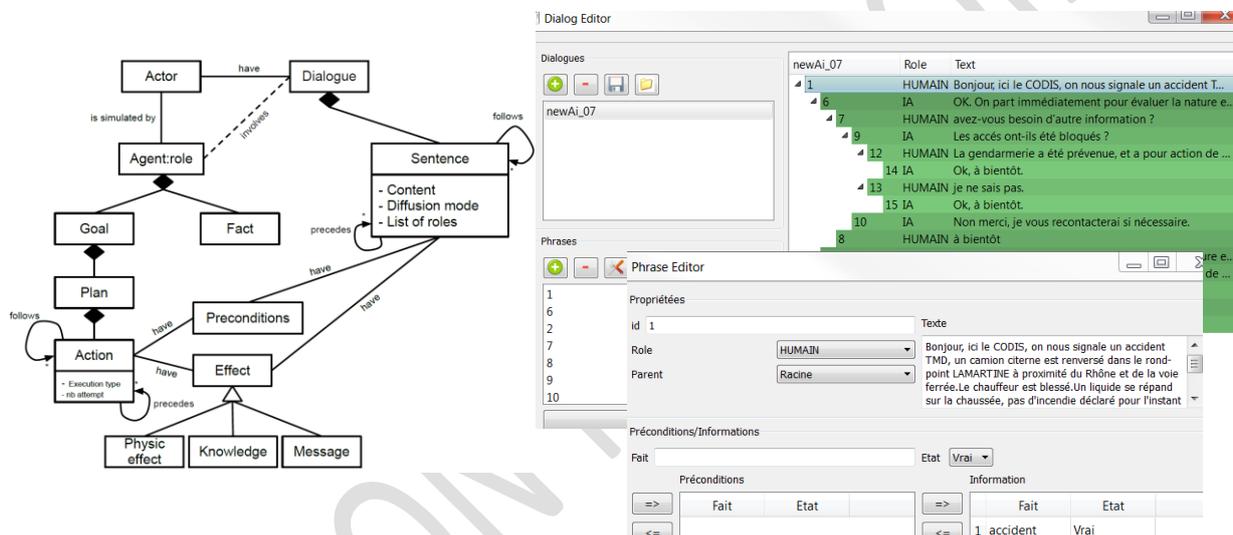


FIG. 2 – Metamodel UML décrivant le modèle d'agent et interfaces graphiques permettant la constructions de dialogues adaptatifs.

Le modèle BDI proposé comprend un ensemble d'agents, d'actions et de faits. Un scénario de jeu SIMFOR consiste à sélectionner les agents PNJ requis pour la session. Le modèle d'agent est le suivant :

$$\text{Modele}_{(\text{role})} = \{ \text{Buts, Plans, Faits, Dialogues} \} \quad (2)$$

Un agent de jeu (GA) cherchera à atteindre ses buts (actifs c'est-à-dire dont les pré-conditions sont validées) en activant les plans adéquats en fonction de ses connaissances (définies comme une liste de faits déclaratifs). Chaque plan est composé d'actions dirigées vers l'environnement ou vers les autres acteurs produisant ainsi des *Effets* sur cet environnement ou l'envoi de message (cf. metamodelle à gauche de la fig. 2). Cette notion traduit les actions sociales et l'influence physique des agents dans l'Environnement. Les raisonnements des agents incluent donc (de manière implicite) des décisions et des actions. Dans le cas d'interactions entre joueurs (humain ou non), les *Dialogues* possibles sont codifiés par un ensemble possible de *Phrases* (cf. exemple de conception de dialogue à droite dans la fig. 2).

4 Evaluation Multi-critères et distribuée

Intégrer l'évaluation dans un jeu sérieux implique l'utilisation de connaissances, d'informations ou de données, produites ou transformées en permanence jusqu'à la fin du déroulement du scénario de jeu. Chaque élément d'information nécessite une manipulation spécifique (voire un raisonnement sur ces connaissances) afin d'en extraire une évaluation. Une façon naturelle de faire face à la complexité de la gestion de ces informations est de diviser et d'organiser ces informations en groupes homogènes dans lesquels des primitives dédiées seront utilisées pour produire une évaluation. La notion d'*Espace d'Evaluation* s'inscrit dans cette démarche en englobant tous les éléments nécessaires pour produire des évaluations, en considérant un scénario de jeu à travers des vues différentes, chacune correspondant à un objectif d'évaluation particulier. Cette section présente brièvement la mise en œuvre de ce concept, on pourra se référer à (Oulhaci et al. 2013) pour une présentation plus détaillée.

Un espace d'évaluation se définit comme un quadruplet:

$$\text{Espace} = \{Kw, I, M, AM\} \quad (1)$$

Avec :

Kw = un ensemble de modèles de représentation des connaissances : différents types de connaissances (déclaratives ou procédurales) sont impliqués et chaque type est basé sur un paradigme de modélisation spécifique (modélisation de données, base de règles, réseau Bayésien...). Pour assurer l'homogénéité de l'évaluation, un espace est associé à un ensemble de modèles de représentation de connaissances similaires.

I = un ensemble d'indicateurs : un indicateur représente une donnée quantitative qui caractérise une situation en évolution (une action ou les conséquences d'une action).

M = un ensemble de métriques : les métriques représentent les méthodes de calcul et les unités de mesure utilisées pour exploiter les connaissances. Il peut s'agir de comparer les comportements attendus et réels des acteurs (actions, décisions) ou de la manière d'analyser un graphe d'interaction. Ainsi, une métrique permettra de quantifier un indicateur en vue d'une évaluation.

AM = un ensemble de modèle d'évaluation: les modèles d'évaluation diffèrent selon l'espace et sa représentation des connaissances. L'évaluation peut porter sur une action, sur une procédure ou bien globalement sur une session de formation. Le calcul d'indicateurs repose sur un modèle d'évaluation spécifique ainsi que sur sa métrique. Selon le modèle de représentation des connaissances, la production d'indicateurs peut requérir des méthodes issues de l'IA (heuristiques, appariement d'ontologies, inférences...). Ainsi, un modèle d'évaluation (AM) peut être vu comme une fonction utilitaire qui produit un indicateur I à partir d'un sous ensemble de connaissances (exprimé selon un mode de représentation Kw) et sa métrique associée (M). Un modèle d'évaluation, par le biais des indicateurs qu'il permet de produire, intervient dans l'évaluation individuelle et collective des joueurs.

Ce cadre de formalisation des besoins et modalités d'évaluation a été appliqué au projet SIMFOR et a abouti à la définition de trois différents espaces d'évaluation : ***Comportemental, Physique et Social***.

L'espace Comportemental comprend les actions et les connaissances des acteurs, ainsi que les différentes informations sur les compétences et procédures à apprendre. Ces informations correspondent au "modèle du domaine" et au "modèle de l'apprenant" dans un STI. Le modèle du domaine est statique (il n'évolue pas au cours du jeu). Il est défini par un expert du domaine sous la forme par exemple d'une ontologie et de plans³. Le modèle de l'apprenant est dynamique (il évolue durant l'expérience de jeu de l'apprenant), et il est alimenté par les actions réalisées par l'apprenant ainsi que par les connaissances acquises. L'évaluation nécessite des capacités de raisonnement sur les modèles de connaissance composant le modèle du domaine. Dans cette espace nous avons à évaluer des actions ainsi que des missions, via des indicateurs dédiés. Par exemple pour l'action *Téléphoner*, les indicateurs sont i) "la durée

³ Eux mêmes décrits par des réseaux de Pétri ou des diagrammes d'activité UML.

de communication", mesurée en secondes (métrique); ii) les informations échangées, ainsi que iii) l'acteur appelé. L'évaluation se basera sur ces indicateurs pour attribuer un score à l'action entreprise. Le tableau 1 ci-dessous donne un exemple des indicateurs définis.

TABLE 1 – Exemple d'indicateurs utilisés pour mesurer le bon déroulement d'une mission

Indicateur	Objectif	Formule	Description
Pré-conditions	Évalue le respect des pré-conditions de la mission	$S_{prec} = \frac{NbRespectedPrec}{NbPrec}$	$NbRespectedPrec$: pré-conditions respectées, $NbPrec$: pré-conditions de la mission
Ordre des actions	Évalue le respect de l'ordre des actions	$S_{order} = \frac{NbOrderedAction}{NbAction}$	$NbOrderedAction$: nombre d'actions faits dans l'ordre, $NbAction$: nombre total des actions de la mission
Nombre d'actions	Évalue le nombre d'action effectuées	$S_{count} = \frac{NbOrderedAction}{NbAction}$	$NbActionPerformed$: nombre d'actions effectuées par l'apprenant $NbAction$: nombre d'action de la mission
Temps écoulé	Évalue la vitesse à effectuer la mission	$S_{Time} = \frac{1}{1 + \exp(-\frac{plannedTime}{takenTime})}$	$plannedTime$: temps estimé par l'expert, $takenTime$: temps mis par l'apprenant pour effectuer la mission.
Temps d'inactivité	Évalue le temps perdu par l'apprenant	$S_{idleTime} = 2 * (1 - \frac{1}{1 + \exp(-idleTime)})$	$idleTime$: temps d'inactivité de l'apprenant.
Efficacité des actions	Évalue l'efficacité des actions effectuées par l'apprenant	$S_{Efficiency} = \frac{\sum_{i=0}^{nbAction} S_{Action_i}}{NbAction}$	$S_{action(i)}$: efficacité de l'action i, $NbAction$: nombre d'action de la mission

L'espace Physique représente le monde virtuel, avec les avatars des acteurs, les moyens mis à disposition des joueurs (voitures, téléphone, fax...) et l'environnement (bâtiments, routes, arbres...). Ces données peuvent évoluer dans le temps comme par exemple l'état d'un sinistre ou la position d'un acteur. Dans SIMFOR, le monde virtuel est représenté au niveau de l'interface 3D du jeu sérieux. Le traitement et la manipulation de ces données sont effectués durant la partie par la simulation du jeu ainsi que par les différents mécanismes du moteur de jeu (interaction, animation, ...). L'évaluation consiste en l'agrégation de ces données via des expressions mathématiques. Par exemple, l'indicateur pertes matérielles et humaines va permettre une évaluation de la performance collective des joueurs. Cet indicateur peut avoir comme métrique le montant en euros des pertes ainsi que le nombre de blessés et de décès.

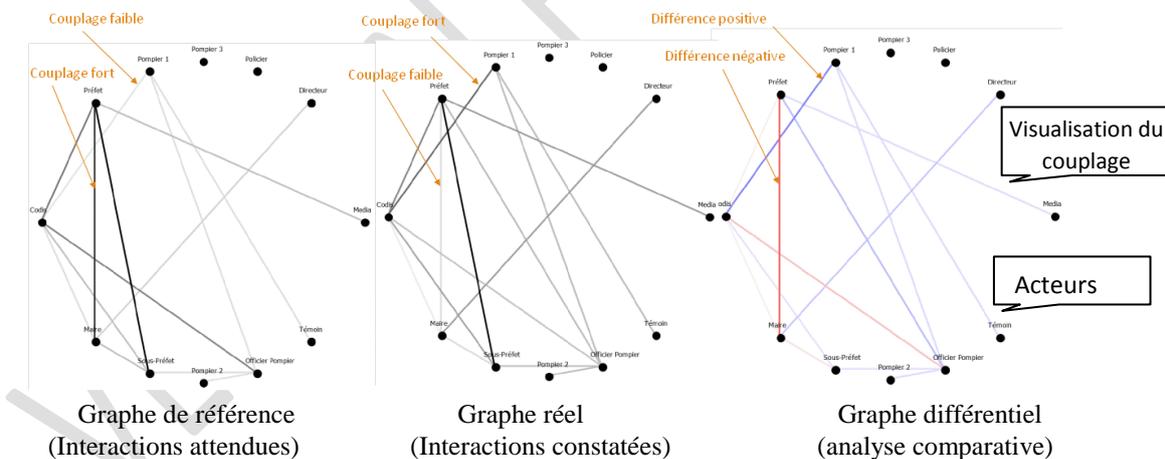


FIG. 3 – Exemple de graphes d'interaction.

L'espace Social représente l'interaction (communication, coordination, coopération,...) entre les différents acteurs intégrant ainsi la dimension collective de l'apprentissage. L'espace social est représenté par un graphe qui reprend chaque interaction entre les acteurs et permet un calcul d'une force d'interaction entre chaque acteur ainsi que le degré d'intégration des acteurs du réseau. L'évaluation sera basée sur ces mesures. Les indicateurs représentés dans cet espace sont le couplage du réseau et la force d'interaction entre les acteurs. Ils sont calculés comme dans (Miller 1978). Les échanges (entre les acteurs) courts et fréquents

produisent un indice de couplage fort, alors que les échanges longs et rares traduisent un couplage faible. Cet indicateur va déterminer si les acteurs ont bien communiqué avec les bonnes personnes en comparant le graphe d'interaction nominal avec celui réalisé (cf. fig. 3).

Cette catégorisation des besoins d'évaluation en 3 espaces d'évaluation permet de rendre compte du caractère hétérogène des traces laissées par les joueurs au cours d'une session de jeu. Elle conserve ainsi, à la manière de méthodes multicritères de surclassement, ce caractère hétérogène au travers de fonction d'évaluation dédiée, tout en les intégrant *in fine* dans un bilan de compétence global. La section suivante décrit l'architecture générale du jeu sérieux SIMFOR une fois les améliorations précitées apportées.

5 Architecture générale

L'aspect distribué et hétérogène du problème nous a conduit à opter pour une architecture multi-agents pour la mise en œuvre de la solution. Les Systèmes Multi-Agents (SMA) permettent une architecture logicielle décentralisée et distribuée. Les agents possèdent des capacités sociales (communication entre les agents) (Wooldridge and Jennings 1995) et ils peuvent également intervenir à un niveau d'intégration logicielle (Serment et al. 2008). Notre objectif est de combiner les différentes technologies des jeux sérieux (3D, mécanisme de jeu, animation, ...) avec celles associées à un STI (représentation des connaissances, pédagogie, ...) pour obtenir un outil d'apprentissage optimal.

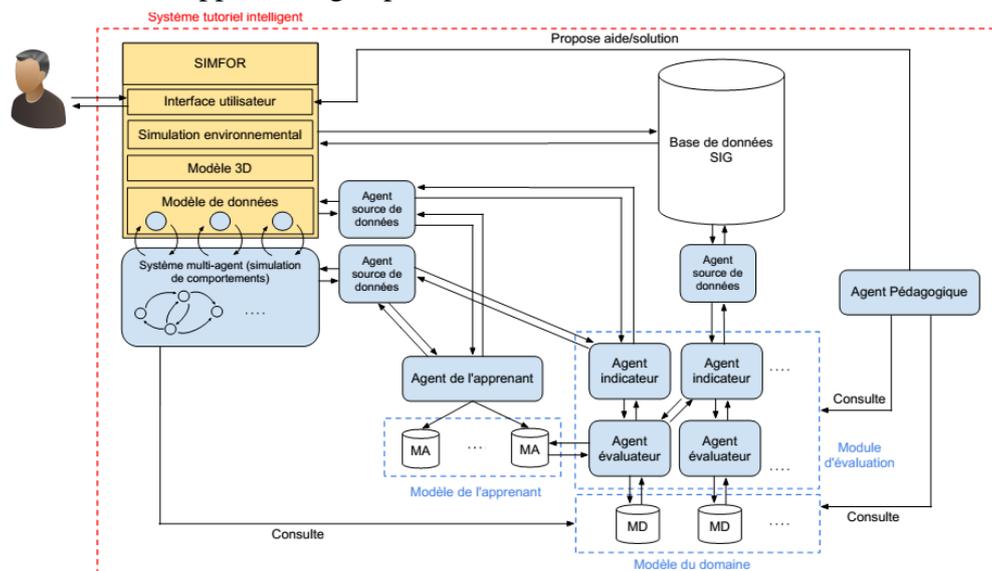


FIG. 4 – Architecture générale du nouveau jeu sérieux

Les agents composant le SMA ont pour mission de collecter les données liées aux apprenants, de traiter et d'évaluer ces données et de fournir un soutien aux apprenants :

- **Agent Source de Données** : il est responsable de la collecte des données (couche logicielle). Chaque agent voit son architecture interne (objet, agent réactif ou cognitif) adaptée au type de données récoltée. Un langage commun abstrait ces différences.
- **Agent Indicateur** : il agrège/transforme les données en indicateurs pour l'évaluation des apprenants en mettant en œuvre des méthodes de calcul adaptée.
- **Agent Apprenant** : il est responsable de la mise à jour du modèle de l'apprenant (et peut informer les *Agents Indicateurs* de changements notables).
- **Agent Evalueur** : il calcule et maintient à jour l'évaluation d'un apprenant en exploitant les informations fournies par les *Agents Indicateurs*.

- **Agent Pédagogique** : cet agent analyse la situation pédagogique (retours des *Agents Évaluateurs*), et sélectionne la stratégie adaptée pour aider l'apprenant.

La figure 4, décrit comment ces agents s'inscrivent dans la nouvelle architecture ainsi que leurs interactions. La partie SIMFOR représente le jeu sérieux préexistant, les autres éléments ont été rajoutés pour répondre aux limitations de SIMFOR.

L'évaluation globale et collective est effectuée par un processus collaboratif (ou processus de négociation comme dans (Chadli et al. 2015)) entre les différents agents évaluateurs. Le processus d'évaluation est basé sur une transformation graduelle des données de jeu jusqu'à la production d'une évaluation des apprenants. Chaque étape est assurée par un agent dédié, de manière asynchrone (i.e. au fur et à mesure du déroulement du jeu). Il s'agit de conserver le caractère hétérogène de ces informations (connaissances, procédure, données brutes) en proposant des méthodes d'évaluation adaptées (techniques relevant de l'IA ou simples agrégations mathématiques). La fig. 5 illustre cette intention en montrant le type de rôle de chaque agent mis en relation avec le type d'information manipulée et ce en fonction des Espaces d'Information concernée. On retrouve l'hétérogénéité des rôles joués (Pompier, CODIS⁴, ...), des comportements associés spécifiques (décrits par des diagrammes d'activité auxquels seront comparés les actions réalisées par les joueurs), des indicateurs et des méthodes d'évaluation différentes selon l'Espace d'Évaluation concernés (analyse différentielle des graphes d'interaction...).

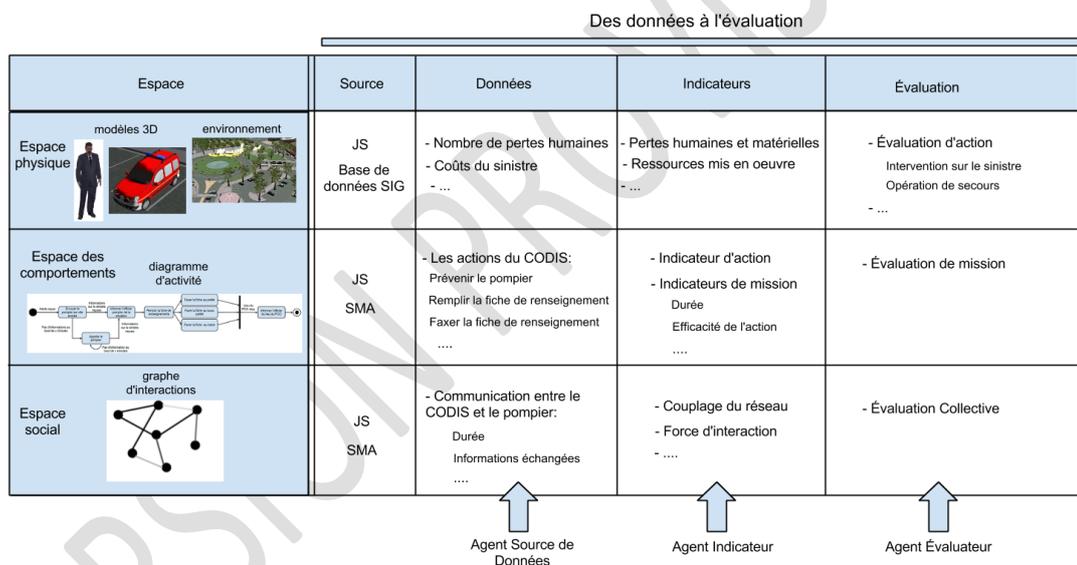


FIG. 5 – Illustration du rôle de chaque agent sur les 3 Espaces d'Évaluation

La spécialisation des agents ainsi que leurs interactions à un niveau de langage indépendant des composants logiciels "bas niveau", permet d'assurer une certaine génériqueité et ainsi de faciliter l'évolution du système en cas de modification des processus d'évaluation ou d'ajout de nouveaux indicateurs et/ou sources d'information (Serment et al. 2008).

6 Conclusion

Cet article a présenté un exemple d'application de l'Intelligence Artificielle Distribuée au domaine des jeux sérieux, en vue d'améliorer leurs capacités de simulation et d'évaluation de

⁴ Centre Départemental d'Incendie et de Secours, en charge de la coordination des secours.

l'apprentissage des joueurs humains. L'ensemble des propositions évoquées dans cet article ont été implémentées et testées mais requièrent encore une validation extensive sur le terrain.

En terme de perspectives de développement, les comportements d'agents commencent à intégrer des facteurs émotionnels nécessaires notamment pour simuler des mouvements de foule dans le cadre de la gestion de crise. En ce qui concerne la dimension évaluation, elle reste peu développée (et ce notamment dans sa dimension collective), mais présente des enjeux d'amélioration en terme d'analyse de comportements de joueurs humains. Sur la contribution d'analyse des interactions, si le cadre permet d'étudier le caractère quantitatif de ces interactions, il est possible de l'étendre en intégrant une dimension qualitative traduisant ainsi la pertinence des messages échangés.

Références

- AMOKRANE K. (2010). Suivi de l'apprenant en environnement virtuel pour la formation à la prévention des risques sur des sites SEVESO, PhD thesis, Université de Technologie Compiègne (December 2010).
- BAUDOIN C., BENEY M., CHEVAILLIER P., LE A. (2007). Recueil de traces pour le suivi de l'activité d'apprenants en travaux pratiques dans un environnement de réalité virtuelle. *Revue Sticef*, vol.14, p. 16-33.
- CHADLI, A., BENDELLA, F., TRANVOUEZ, E. (2015). A Two-Stage Multi-Agent Based Assessment Approach to Enhance Students' Learning Motivation through Negotiated Skills Assessment. *Educational Technology & Society*, 18 (2), 124–137.
- CHANG P. H.-M., CHEN K. T., CHIEN Y.-H., KAO E. & SOO V.-W. (2004) From Reality to Mind: A Cognitive Middle Layer of Environment Concepts for Believable Agents. *Environments for Multi-Agent Systems, First International Workshop, E4MAS 2004*, New York, NY, USA, July 19, 2004.
- GIORDANI A., GEROSA L., SOLLER A., STEVENS R. (2005). Extending an Online Individual Scientific Problem-solving Environment to Support and Mediate Collaborative Learning. 12th International Conference on Artificial Intelligence in Education.
- LOURDEAUX, D., BURKHARDT, J.M., BERNARD, F. AND FUCHS, P. (2002). Relevance of an intelligent agent for virtual reality training, *International Journal of Continuous Engineering and Life-long Learning*, vol. 12, num. 1/2/3/4, pp. 131-143, 2002.
- MILLER J.-G. (1978). *Living Systems*. McGraw-Hill.
- OULHACI A. (2014). Évaluation individuelle et collective dans les jeux sérieux collaboratifs, Manuscrit de thèse de Doctorat, Aix Marseille Université, 06/2014.
- OULHACI A., TRANVOUEZ E., FOURNIER S. & ESPINASSE B. (2013). Évaluation A Multi-Agents System for Learner Assessment in Serious Game: Application to Learning processes in Crisis Management, *IEEE RCIS 05/2013*.
- QUERREC R., BUCHE C., MAFFRE E., & CHEVAILLIER (2003) P., Sécurévi : virtual environments for fire-fighting training. in *Proceedings of the 5th Virtual Reality International Conference, VRIC 2003*, 05/2003.
- RAO A., GEORGEFF M.-P. (1995). *BDI Agents: From Theory to Practice*. ICMAS.
- THOMAS P., LABAT J.-M., MURATET M., YESSAD A. (2012). How to evaluate competencies in game-based learning systems automatically? *Intelligent Tutoring Systems*, Springer.
- WOOLDRIDGE M., JENNINGS N.-R. (1995). Intelligent agents: Theory and practice. *Knowledge engineering review*, vol. 10 n°2, p. 115-152.
- SERMENET J., ESPINASSE B., TRANVOUEZ E. (2008). Systèmes d'aide à la décision environnementale. *Journal of decision systems*, vol. 17 n°2, p. 269-300.
- WENDEL V., GUTJAHR, M., GÖBEL S., STEINMETZ R. (2012). Designing Collaborative Multiplayer Serious Games for Collaborative Learning. *Proceedings of the CSEDU 2012*.
- MATHIEU P., PANZOLI D., PICAULT S. (2013), Virtual Customers in a Multiagent Training Application", in: Z. Pan, A. D. Cheok, W. Müller and F. Liarokapis (eds.), *Transactions on Edutainment IX*, Lecture Notes in Computer Science vol. 7544, p. 97-114 (ch. 6), Springer. ISBN: 978-3-642-37041-0.

Towards efficient data integration and knowledge management in the Agronomic domain

Aravind Venkatesan¹, Nordine El Hassouni², Florian Phillippe³, Cyril Pommier³, Hadi Quesneville³, Manuel Ruiz¹², Pierre Larmande¹⁴⁵

¹ Institut de biologie Computationelle, Montpellier, France
Aravind.Venkatesan@lirmm.fr

² UMR AGAP, CIRAD, Montpellier, France
{nordine.el_hassouni,manuel.ruiz}@cirad.fr

³ URGI, INRA, Versailles, France
{fphilippe,Cyril.pommier,hadi.quesneville}@versailles.inra.fr

⁴ UMR DIADE, IRD, Montpellier, France

⁵ Equipe Zenith, INRIA et LIRMM, Montpellier, France
pierre.larmande@ird.fr

Résumé : Today, the revolution in empirical technologies has generated vast amounts of data. This data deluge has created an urgent need to assimilate it with a panoramic view. To this end, information systems play a central role in managing and integrating these data, aiding the biologists in exploiting this integrated information for the extraction of new knowledge. The plant bioinformatics node of the Institut Français de Bioinformatique (IFB) maintains public information systems where a variety of domain specific data are integrated. Currently, efforts are being taken to expose the IFB plant bioinformatics resources as RDF, utilising domain specific ontologies and metadata. Here, we present the overview and the progress of the project.

Mots-clés: Data integration, data interoperability, knowledge management, Semantic Web, RDF, Bioinformatic application, Agronomic research

1 Introduction

Agronomy is an overarching field constituting various research areas such as genetics, plant molecular biology, ecology and earth science. The last several decades has seen the successful development of high-throughput technologies that have revolutionised and transformed agronomic research. The application of these technologies have generated large quantities of data. These technological advancements have resulted in a number of initiatives been taken to systematically store and share information over the web, such as, Gramene (Monaco et al., 2014), TAIR (Lamesch et al., 2012), OryzaBase (Kurata et al., 2006), Plant Reactome (Croft et al., 2014), GnpIS (Steinbach et al., 2013) and the South Green bioinformatics platform (<http://www.southgreen.fr>), to name a few.

The definitive aim of agronomic research being the improvement of crop production through sustainable methods. It is important to efficiently overlay research findings from the allied fields, by automating data analyses for hypotheses generation, ultimately reducing the knowledge discovery cycle. However, using these resources comprehensively, taking advantage of the associated cross-disciplinary research opportunities poses a major challenge to both domain scientists and information technologists. Effective data integration and management allows a broader perspective across many disciplines, than is possible from one

or a series of individual studies. In the long run, this allows information to be used for purposes other than those for which they were originally intended, to address questions that were unapproachable at the time the data were collected. To this end, the need for an umbrella approach for providing uniform data is a much discussed topic. For instance, the Research Data Alliance (RDA, <https://rd-alliance.org/>) through its interest group namely, the Agriculture Data Interoperability Interest group (<https://rd-alliance.org/groups/agriculture-data-interest-group-igad>), has created a platform to discuss the need to improve data exchange enabling data integration in this domain.

A solution for the data integration challenges is offered by the Semantic Web (SW) technologies (Berners-Lee & Hendler 2001). SW was proposed, to remedy the fragmentation of all the potentially useful information dispersed over the web. This is founded on a stack of technologies such as the Resource Description Framework (RDF, www.w3.org/RDF/), RDF Schema (RDFS, <http://www.w3.org/TR/rdf-schema/>), Web Ontology Language (OWL, www.w3.org/TR/owl-features/) and the SPARQL Query Language (SPARQL, www.w3.org/TR/rdf-sparql-query/). With ontologies providing the knowledge scaffold, a successfully implemented SW application can be used by scientists to pose complex questions that would then assess and return highly relevant answers to those questions. This is useful, as the assessment of biological findings against prior knowledge, after which the best supported hypotheses can be selected for further testing.

We are currently witnessing a growing acceptance of SW (RDF in particular) for the management of disparate biological databases in the bio-medical field. Several projects have been undertaken to demonstrate the potential of SW, some notable initiatives include Bio2RDF (Belleau et al., 2008), BioGateway (Antezana et al., 2009), Linked Life Data (Momtchev et al., 2009), KUPKB (Jupp et al., 2011) and OpenPHACTS (Williams et al., 2012). These initiatives have demonstrated the advantages of SW including rich knowledge representation, streamlined data integration and optimised querying. Moreover, primary data providers such as EMBL-EBI (<https://www.ebi.ac.uk/rdf/platform>), UniprotKB (<http://beta.sparql.uniprot.org/>) and NCBI (Anguita et al., 2013) are also making their data available as RDF.

Presently, in the agronomic domain plant centric ontologies such as Plant Ontology (PO), Plant Trait Ontology (TO) and Plant Environment Ontology (EO) (<http://planteome.org/>), are being used by various databases as a method to provide cross-domain common entry points. Nevertheless, efficient knowledge management additionally requires information to be represented in a machine-readable form. Unlike the bio-medical domain, the agronomic sciences is yet to exploit the full potential of SW. Therefore, initiatives to build on previous efforts to expose agronomy data on the SW is essential. Furthermore, attempts have to be made to pursue various joint collaborations with the intended stakeholders (for instance, plant biologists and breeders) to bridge the gap between SW and the opportunities that come along with it (Venkatesan et al. 2014).

2 Semantification of the IFB plant bioinformatics nodes

Institut Français de Bioinformatique (IFB) is a French national node (<http://www.elixir-europe.org/about/elixir-france>) that is focused on providing integrated services for the life science community. The IFB platform provides access to databases, tools and services that covers three main domains namely, microbial, plant and health sciences. The IFB IT infrastructure is linked to six regional bioinformatics centers, the ReNaBi (French Bioinformatics Platforms Network), representing various regions of the French territory (ReNaBi-NE, North-East; PRABI, Rhône-Alpes region; ReNaBi-GS, Great South; ReNaBi-SO, South-West; ReNaBi-GO, Great West and APLIBIO, Paris area). These six regional centers are consists of regional bioinformatics platforms (PFs). Taken together, IFB will

represent France in the ELIXIR European infrastructure initiative (see Figure 1). To this end, the plant bioinformatics PFs maintain public data repositories that ranges from ‘omics’ to genetic data (genetic markers, maps and phenotypes) for various crop species.

Currently, the plant-centric PFs are working towards exposing their resources as linked data. The objective of the current effort is to develop RDF knowledge base that integrates existing domain specific ontologies and data from the respective PFs. This will promote interoperability between the databases. In the initial phase, two representative PFs are involved in this semantification process, namely:

- a) The *Unité de Recherche Génomique-Info* (URGI) platform (<https://urgi.versailles.inra.fr/>) associated with the *Institut National de la Recherche Agronomique* (INRA), dedicated to maintain curated information on plants and crop parasite. The platform is part of the APLIBIO ReNaBi and plays a key role in the Wheat Initiative (<http://wheatis.org/>).
- b) The South Green Bioinformatics platform (SG) part of the ReNaBi GS mainly associated with *Centre de coopération internationale en recherche agronomique pour le développement* (CIRAD) and *Institut de recherche pour le développement* (IRD) among other regional institutes. SG provides tools and databases dedicated for genomic resource analysis of southern and Mediterranean plants.

Additionally, future efforts will be taken provide RESTful APIs to make the RDF repositories accessible from within workflow environment such as Taverna (Wolstencroft et al., 2013) and Galaxy (Giardine et al., 2005).

3 RDF store integrating South Green resources

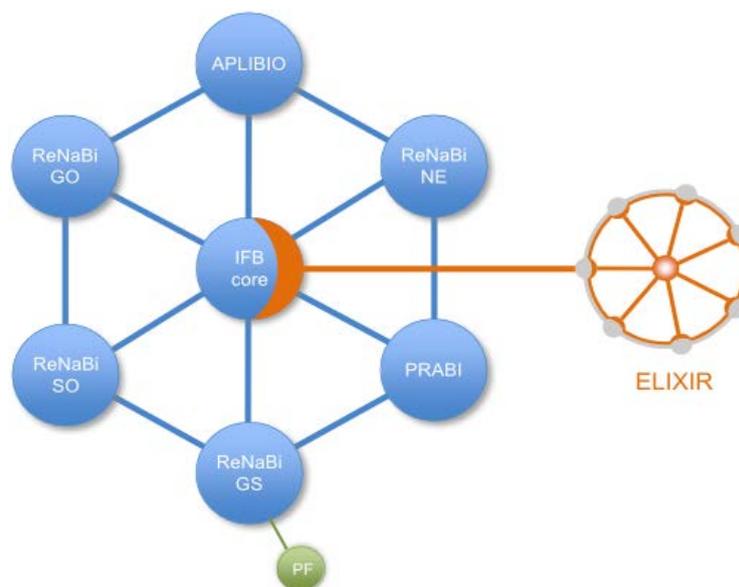
This section provides description of the effort taken to establish the RDF repositories representing the IFB plant PFs. As a primary step, resources hosted under the South Green bioinformatics platform was chosen for modeling data as RDF.

3.1 Design principles

Currently, SG houses 12 databases covering various plant species such as Banana, Cocoa, Maize and Rice. The RDF store is designed to provide plant biologists with a knowledge system that captures knowledge represented in these databases. The following design principles were followed in the RDF store development:

1. Developing the store in various phases.
2. Integrating domain specific data and ontologies to enable comparative analyses.
3. Providing maximum flexibility both for end users and for future extensions.
4. Making the store Linked Data compliant.

Figure 1 - The illustration shows the structure of IFB node with ELIXIR. The blue nodes represent regional platforms (ReNaBi) across France. The green node represent the regional bioinformatics platform of ReNaBi (Figure adapted from Perriere 2012.).



3.2 Data sources

The phase I of the RDF store development was focused on integrating a number of well-established plant centric ontologies, this includes the Gene Ontology (three variants) (Ashburner et al., 2000), PO, TO and EO. For phase I, the SG databases that were included are:

1. TropGeneDB (Hamelin et al., 2012), a database that hosts genetic, molecular and phenotypic information on tropical crop species.
2. OryGenesDB (Droc et al., 2006), a database that serves as a repository on functional genomics for rice.
3. Oryza Tag Line (Larmande et al., 2008), a database that contains sequence information (Flanking Sequence Tags) that are based on molecular categorisation of mutagen insertion sites for rice.
4. GreenPhyl (Conte et al., 2008), provides sequence homology information for the members of kingdom *plantae*.

Furthermore, ontology annotations, proteomics and genomics information from a variety of publically available data sources were integrated, this includes, UniprotKB (Magrane & Uniprot Consortium 2011), GOA (Barrell et al., 2009), Gramene (ontology annotation, gene and Quantitative Trait Loci (QTL) information), AraCyc (Mueller et al., 2003), RiceCyc, SorghumCyc, and MaizeCyc (Dharmawardhana et al., 2013) and OryzaBase. The integration of additional data resources provides the critical mass required for implementing real world use cases. Currently, the RDF store is limited to a selected species namely, *Oryza* species (*O.sativa*, *O.barthii*, *O.brachyantha*, *O.glaberrima* and *O.meridionalis*), *Arabidopsis thaliana*, *Sorghum bicolor*, *Zea mays* and *Triticum* species (*T.aestivum* and *T.uraruta*). Table 1 provides a detailed list of data resources integrated in SG RDF store.

TABLE 1 – The table shows the breakdown of the data sources integrated into SG RDF store with the corresponding species.

Data Sources		Species				
		Oryza spp.	A.thaliana	S.bicolor	Z.mays	Triticum spp
SG platform	TropGeneDB	✓			✓	✓
	OryGenesDB	✓	✓	✓		
	Oryza Tag Line	✓				
	GreenPhyl	✓	✓	✓	✓	
Ontology associations	GOA	✓	✓	✓	✓	✓
	Gramene-PO	✓	✓	✓	✓	✓
	Gramene-TO	✓				
	Gramene-EO	✓				
Gramene genes		✓	✓	✓	✓	✓
Gramene QTL		✓				
UniProtKB		✓	✓	✓	✓	✓
AraCyc			✓			
RiceCyc		✓				
SorghumCyc				✓		
MaizeCyc					✓	

3.3 South Green RDF store construction

The SG RDF store is built using a semi-automated pipeline implemented in Python ver.2.7. The pipeline integrates information from SG resources as well as other resources (refer section: 3.2 Data sources) as RDF. The conversion of these resources was performed with newly developed parsers, for GOA, BioPython (www.biopython.org/) module (Bio.Uniprot.GOA) was used. To avoid duplication, information that were available as RDF was directly utilized such as, the candidate ontologies. The RDF graphs have been loaded into Open Link Virtuoso (<http://virtuoso.openlinksw.com>). The SPARQL endpoint is currently being tested and can be accessed at: <http://volvestre.cirad.fr:8890/sparql>.

In order to make SG RDF store Linked Data compliant, dereferenceable stable URIs provided by Identifiers.org (<http://identifiers.org/>) and Ontobee (www.ontobee.org/) were used. The conversion new datasets that are not included in these registries, new URIs were minted with a common name-space: <http://www.southgreen.fr/agrold/>. The identifiers for these datasets take the form [http://www.southgreen.fr/agrold/\[resource_namespace\]/\[identifier\]](http://www.southgreen.fr/agrold/[resource_namespace]/[identifier]), for example, AraCyc describes metabolic pathway information associated with *A.thaliana* genes. Thus, the URI for AraCyc pathway identifier would be: <http://www.southgreen.fr/agrold/aracyc.pathway/PWYQT-4482>. Similarly, for resources that requiring new properties, would be of the form [http://www.southgreen.fr/agrold/\[vocabulary\]/\[property\]](http://www.southgreen.fr/agrold/[vocabulary]/[property]). For example, Gramene provides protein-EO annotations, the property *expressed_in* is used link the two resources. The absolute URI for this property would be: http://www.southgreen.fr/agrold/vocabulary/expressed_in.

3.4 Querying

In this section we demonstrate the utility of the knowledge base with the help of a few example SPARQL queries. These queries will be made available as a part of a list of sample queries provided on dedicated query page (under construction).

Q1. Retrieve the local neighbourhood of *Oriza sativa japonica* protein: IAA16 - Auxin-responsive protein (UniProt accession: P0C127).

SPARQL query:

```

BASE <http://www.southgreen.fr/agrold/>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX obo:<http://purl.obolibrary.org/obo/>
PREFIX uniprot:<http://purl.uniprot.org/uniprot/>
PREFIX vocab:<vocabulary/>
PREFIX graph:<protein.ontology.associations>

SELECT distinct ?predicate ?object ?object_label
WHERE {
  GRAPH graph: {
    uniprot:P0C127 ?predicate ?object.
    OPTIONAL {
      GRAPH ?g {
        ?object rdfs:label ?object_label.}}}}
    
```

Q2. Retrieve genes that participate in a pathway: Calvin cycle.

SPARQL query:

```

BASE <http://www.southgreen.fr/agrold/>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX obo:<http://purl.obolibrary.org/obo/>
PREFIX uniprot:<http://purl.uniprot.org/uniprot/>
PREFIX vocab:<vocabulary/>
PREFIX graph:<ricecyc>
PREFIX pathway:<http://www.southgreen.fr/agrold/ricecyc.pathway/CALVIN-
PWY>

SELECT DISTINCT ?gene ?name
WHERE {
  GRAPH graph: {
    pathway: vocab:has_agent ?gene.
    ?gene rdfs:label ?name.}}
    
```

These queries offer a glimpse of the range of biological questions that can be addressed to the knowledge base. Since the data is represented as linked data users could also query the

knowledge base in combination with complimentary RDF knowledge bases using query federation features offered by SPARQL ver.1.1.

4 Future Directions

SG RDF store currently hosts information for select species (refer section: 3.2 Data sources), in the subsequent phases information pertaining to other species such as Banana and Cocoa will be integrated. Additionally, the integration of other databases under SG that includes SNIPlayDB (Dereeper et al., 2011), CocoaGenDB (Argout et al., 2007) and EuriGen (Courtois 2012) will be considered. Presently, semantic web knowledge bases are accessed via a SPARQL endpoint. However, these endpoints are more machine-friendly and users are required at the minimum a moderate knowledge of SPARQL to exploit the integrated information. Obviously, this is not an optimal solution for users not acquainted with SPARQL. Hence efforts will be made to develop user friendly query interface that is optimized to aid non-technical users. Furthermore, collaborations with domain experts will be pursued to develop real world use cases to demonstrate the advantages of SW. To this end, the RDF store will be augmented with additional resources to suit the use cases. Furthermore, this effort will be extended to the information hosted at URGI.

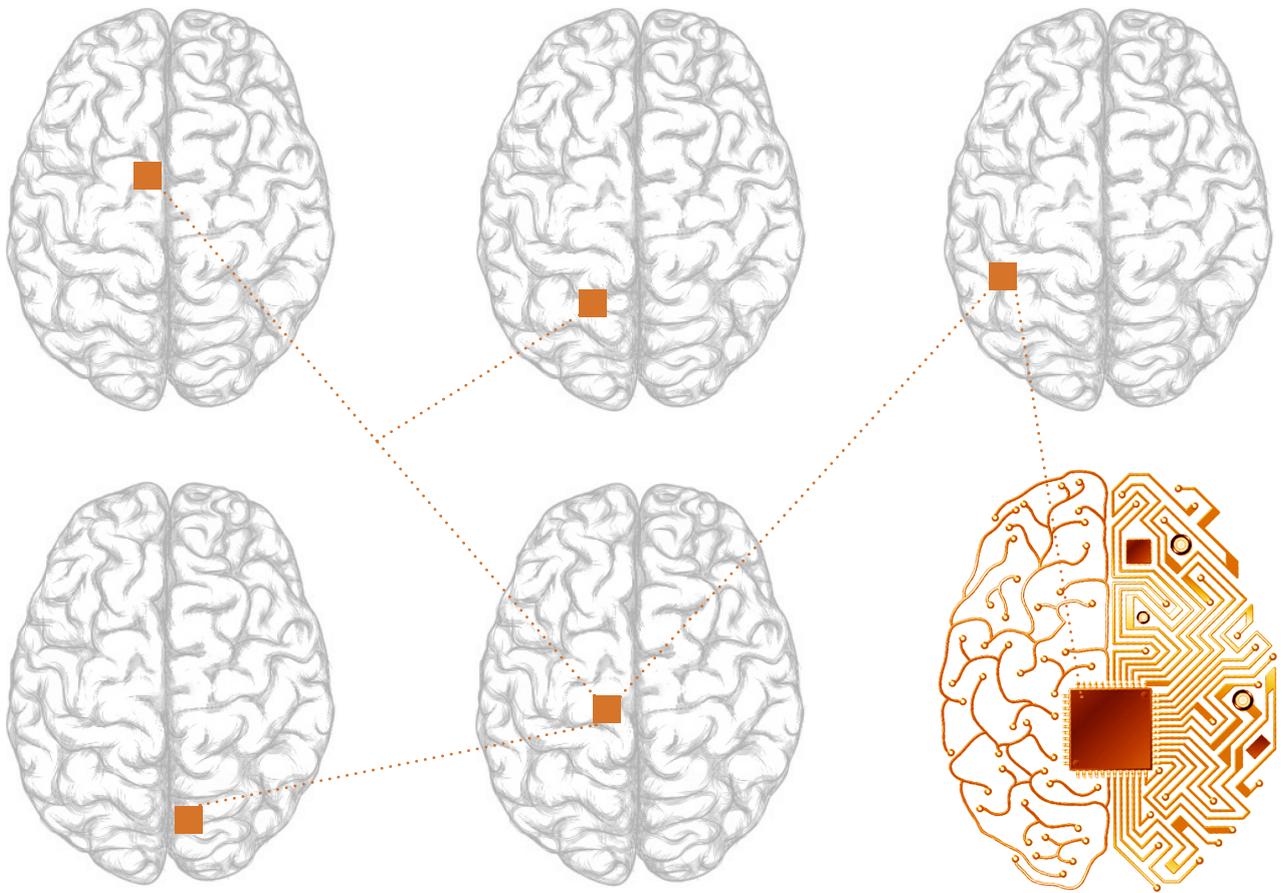
5 Conclusion

The drastic increase in the amount of data generated in the agronomic domain requires efficient knowledge management practices. SW certainly provides a robust method to integrate information representing domain specific knowledge. Exposing the IFB plant PFs as RDF will aid both domain experts and bioinformaticians to take advantage of the integrated information. With the development of SG RDF store, we have taken the initial steps towards this goal.

Références

- ANGUITA, A., GARCIA-REMESAL, M., DE LA IGLESIA, D., & MAOJO, V. (2013). NCBI2RDF: enabling full RDF-based access to NCBI databases. *BioMed Research International*. 983805.
- ANTEZANA E., BLONDÉ W., EGAÑA M., RUTHERFORD A., STEVENS, R. DE BAETS B., MIRONOV V. & KUIPER M. (2009). BioGateway: a semantic systems biology tool for the life sciences. *BMC bioinformatics*, 10(Suppl 10), S11. BOUAUD J. (1978a). De la façon d'écrire un article. *Le Journal LaTeX*. 2, p. 101-105.
- Application Delivery Strategies published by META Group Inc.
- ARGOUT X. RUIZ M. ROUARD M. TURNBULL C. LANAUD C. ROSENOUIST E. ET AL. CocoaGen DB: a Web portal for crossing cocoa phenotypic, genetic and genomic data from ICGD and TropGeneDB databases. *CocoaGen DB: a Web portal for crossing cocoa phenotypic, genetic and genomic data from ICGD and TropGeneDB databases*. In: 15th International Cocoa Research Conference. Vol 1. San Jose, Costa Rica; 2007. p. 515-8
- ASHBURNER ET AL. GENE ONTOLOGY: TOOL FOR THE UNIFICATION OF BIOLOGY (2000) *NAT GENET* **25**(1):25-9.
- BARRELL, D., DIMMER, E., HUNTLEY, R. P., BINNS, D., O'DONOVAN, C., & APWEILER, R. (2009). The GOA database in 2009—an integrated Gene Ontology Annotation resource. *Nucleic acids research*, 37(suppl 1), D396-D403.
- BELLEAU F., NOLIN M. A., TOURIGNY N., RIGAULT P., & MORISSETTE J. (2008). Bio2RDF: towards a mashup to build bioinformatics knowledge systems. *Journal of biomedical informatics*, 41(5), 706-716.
- BERNERS-LEE T. & HENDLER J. (2001). Publishing on the semantic web. *Nature*, 410, 1023-4.
- CONTE, M. G., GAILLARD, S., LANAU, N., ROUARD, M., & PERIN, C. (2008). GreenPhyIDB: a database for plant comparative genomics. *Nucleic acids research*, 36(suppl 1), D991-D998.
- COURTOIS, B., FROUIN, J., GRECO, R., BRUSCHI, G., DROC, G., HAMELIN, C. & AHMADI, N. (2012). Genetic diversity and population structure in a European collection of rice. *Crop science*, 52(4), 1663-1675.

- CROFT D., MUNDO A. F., HAW R., MILACIC M., WEISER J., WU G., CAUDY M., GARAPATI P., GILLESPIE M., KAMDAR M R., JASSAL B., JUPE S., MATTHEWS L., MAY B., PALATNIK S., ROTHFELS K., SHAMOYSKY V., SONG H., WILLIAMS M., BIRNEY E., HERMIAKOB H., STEIN L. & D'EUSTACHIO P. (2014). The Reactome pathway knowledgebase. *Nucleic Acids Research*, 42(D1), D472-D477.
- DEREEPER, A., NICOLAS, S., LE CUNFF, L., BACILIERI, R., DOLIGEZ, A., PEROS, J. P. & THIS, P. (2011). SNIPlav: a web-based tool for detection, management and analysis of SNPs. Application to grapevine diversity projects. *BMC bioinformatics*, 12(1), 134.
- DHARMAWARDHANA, P., REN, L., AMARASINGHE, V., MONACO, M., THOMASON, J., RAVENSCROFT, D. & JAISWAL, P. (2013). A genome scale metabolic network for rice and accompanying analysis of tryptophan, auxin and serotonin biosynthesis regulation under biotic stress. *Rice*, 6(1), 15.
- DROC, G., RUIZ, M., LARMANDE, P., PEREIRA, A., PIFFANELI, J. P., MOREL, J. B. & PERIN, C. (2006). OryGenesDB: a database for rice reverse genetics. *Nucleic acids research*, 34(suppl 1), D736-D740.
- GIARDINE, B., RIEMER, C., HARDISON, R. C., BURHANS, R., ELNITSKI, L., SHAH, P. & NEKRUTENKO, A. (2005). Galaxy: a platform for interactive large-scale genome analysis. *Genome research*, 15(10), 1451-1455.
- HAMELIN, C., SEMPERE, G., JOUFFE, V. & RUIZ, M. (2012). TropGeneDB, the multi-tropical crop information system updated and extended. *Nucleic acids research*, gks1105.
- JUPP, S., KLEIN, J., SCHANSTRA, J., & STEVENS, R. (2011). Developing a kidney and urinary pathway knowledge base. *J. Biomedical Semantics*, 2(S-2), S7.
- KURATA N. & YAMAZAKI Y. (2006). Oryzabase. An integrated biological and genome information database for rice. *Plant physiology*, 140(1), 12-17.
- LAMESCH P., BERARDINI T Z., LI D., SWARBRECK D., WILKS C., SASIDHARAN R. & HUJALA E. (2012). The Arabidopsis Information Resource (TAIR): improved gene annotation and new tools. *Nucleic acids research*, 40(D1), D1202-D1210.
- LARMANDE, P., GAY, C., LORIEUX, M., PÉRIN, C., BOUNIOL, M., DROC, G. & GUIDERDONI, E. (2008). Orvza Tag Line, a phenotypic mutant database for the Genoplante rice insertion line library. *Nucleic acids research*, 36(suppl 1), D1022-D1027.
- MAGRANE, M., & UNIPROT CONSORTIUM. (2011). UniProt Knowledgebase: a hub of integrated protein data. *Database*, 2011, bar009.
- MOMTCHEV, V., PEYCHEV, D., PRIMOV, T., & GEORGIEV, G. (2009). Expanding the pathway and interaction knowledge in linked life data. *Proc. of International Semantic Web Challenge*.
- MONACO K M., STEIN J., NAITHANI S., Wei, DHARMAWARDHANA P., KUMARI S., AMARASINGHE V., CLARK K J., THOMASON J., PREECE J., PASTERNAK S., OLSON A., JIAO Y, LU Z., BOLSER D., KKERHORNOU A., STAINES D., WALTERS B., WU G., D'EUSTACHIO P., HAW R., CROFT D., KERSEY J P., STEIN L., JAISWAL P. & WARE D. (2014). Gramene 2013: comparative plant genomics resources. *Nucleic Acids Reseach*, 42 (D1): D1193-D1199.
- MUELLER, I. A., ZHANG, P., & RHEE, S. Y. (2003). AraCyc: a biochemical pathway database for Arabidopsis. *Plant Physiology*, 132(2), 453-460.
- PERRIERE, G. (2012). ReNaBi-IFB: The French Bioinformatics Infrastructure. *EMBnet journal*, North America, 18, jun. 2012
- STEINBACH D., ALAUX M., AMSELEM J., CHOISNE N., DURAND S., FLORES R., KELIET A., KIMMEL E., LAPALU N., LUYTEN I., MICHOTÉY C., MOHELLIBI N., POMMIER C., REBOUX S., VALDENAIRE D., VERDELET D. & OUESNEVILLE, H. (2013). GnpIS: an information system to integrate genetic and genomic data from plants and fungi. *Database*, 2013. bat058.
- VENKATESAN, A., TRIPATHI, S., DE GALDEANO, A. S., BLONDÉ, W., LÆGREID, A., MIRONOV, V., & KUIJPER, M. (2014). Finding gene regulatory network candidates using the gene expression knowledge base. *BMC bioinformatics*, 15(1), 386.
- WILLIAMS, A. J., HARIAND, I., GROTH, P., PETTIFER, S., CHICHESTER, C., WILLIGHAGEN, E. I., & MONS, B. (2012). Open PHACTS: semantic interoperability for drug discovery. *Drug discovery today*, 17(21), 1188-1198.
- WOLSTENCROFT, K., HAINES, R., FELLOWS, D., WILLIAMS, A., WITHERS, D., OWEN, S. & GOBLE, C. (2013). The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. *Nucleic acids research*, gkt328.



PFIA 2015
<http://pfia2015.inria.fr>

Plate-forme Intelligence Artificielle
Rennes du 29 juin au 3 juillet 2015