13èmes Rencontres des Jeunes Chercheurs en Intelligence Artificielle

Rennes, 29-30 juin 2015

Dans le cadre de la plate-forme AFIA 2015

Table des matières

Classification à base de clustering : ou comment décrire et prédire simultanément

O. Alaoui Ismaili[1,2] V. Lemaire[1] A. Cornuéjols[2]

[1] Orange Labs, 2 av. Pierre Marzin 22307 Lannion, France
[2] AgroParisTech, 16, rue Claude Bernard 75005 Paris, France
oumaima.alaouiismaili@orange.com

Résumé

Dans certains domaines applicatifs, la compréhension (la description) des résultats issus d'un classifieur est une condition aussi importante que sa performance prédictive. De ce fait, la qualité du classifieur réside donc dans sa capacité à fournir des résultats ayant de bonnes performances en prédiction tout en produisant simultanément des résultats compréhensibles par l'utilisateur. On parle ici du compromis interprétation vs. performance des modèles d'apprentissage automatique. Dans cette thèse, on s'intéresse à traiter cette problématique. L'objectif est donc de proposer un classifieur capable de décrire les instances d'un problème de classification supervisée tout en prédisant leur classe d'appartenance (simultanément).

Mots Clef

Classification, Clustering, Interprétation, Performance, Prédiction, Description

Abstract

In some application areas, the ability to understand (describe) the results given by a classifier is as an important condition as its predictive performance is. In this case, the classifier is considered as important if it can produce comprehensible results with a good predictive performance. This is referred as a trade-off "interpretation vs. performance". In our study, we are interested to deal with this problem. The objectif is then to find out a model which is able to describe the instances from a supervised classification problem and to simultaneously predict their class membership.

Keywords

Classification, Clustering, Interpretation, Performance, Prediction, Description

1 Introduction

De nos jours, les données récoltées par ou pour les entreprises sont devenues un atout important. Les informations présentes, mais à découvrir au sein des grands volumes de données, sont devenues pour ces entreprises un facteur de compétitivité et d'innovation. Par exemple, les grandes entreprises comme *Orange* et *Amazon* peuvent avoir un aperçu des attentes et des besoins de leurs clients à travers la connaissance de leurs comportements. Ces données permettent aussi de découvrir et d'expliquer certains phénomènes existants ou bien d'extrapoler des nouvelles informations à partir des informations présentes.

Pour pouvoir exploiter ces données et en extraire des connaissances, de nombreuses techniques ont été développées. Par exemple, l'analyse multivariée regroupe les méthodes statistiques qui s'attachent à l'observation et au traitement simultané de plusieurs variables statistiques en vue d'en dégager une information synthétique pertinente. Les deux grandes catégories de méthodes d'analyse statistique multivariées sont, d'une part, les méthodes dites descriptives et, d'autre part, les méthodes dites explicatives.

Les méthodes descriptives ont pour objectif d'organiser, de simplifier et d'aider à comprendre les phénomènes existant dans un ensemble important de données. Cet ensemble de données $X = \{x_i\}_1^N$ est composé de N instances sans étiquette (ou classe), chacune décrite par plusieurs variables. On notera $x_i = \{x_i^1, \ldots, x_i^d\}$, l'ensemble de d variables décrivant l'instance $i \in \{1, \ldots, N\}$. Dans d, aucune des variables n'a d'importance particulière par rapport aux autres. Toutes les variables sont donc prises en compte au même niveau. Les méthodes descriptives d'analyse multivariée les plus utilisées sont l'analyse en composantes principales (ACP) [1], l'analyse factorielle des correspondances (AFC) [1], l'analyse des correspondances multiples (ACM) [1] et les méthodes de clustering [2] qui visent à trouver une typologie ou une répartition des instances en K groupes distincts où les instances dans chaque groupe $C_k (k \in \{1, \ldots, K\})$ (ou cluster) doivent être les plus homogènes possibles (*i.e.,* partagent les mêmes caractéristiques).

Les méthodes prédictive ont, quant à elles, pour objectif d'expliquer l'une des variables (dite dépendante) à l'aide d'une ou plusieurs variables explicatives (dites indépendantes). Les principales méthodes explicatives utilisées [3] sont la régression multiple , la régression logistique, l'analyse de variance, l'analyse discriminante, les arbres de décision ([4], [5]), les SVM [6], *etc*.

Si on se place dans le cadre de l'apprentissage automatique

[3], on peut placer les méthodes descriptives dans le domaine de l'apprentissage non supervisé et les méthodes explicatives dans le cas de l'apprentissage supervisé.

Dans cet article, on se place dans le cadre de la classification supervisée où il s'agit d'apprendre un concept cible ($X \rightarrow Y$) dans le but de le prédire ultérieurement pour des nouvelles instances. Dans ce cas, le concept cible Y est une variable *'nominale'*, composée de J classes. Afin d'atteindre cet objectif, de nombreux algorithmes ont vu le jour [7]. Certains entre eux fournissent des résultats difficilement compréhensibles de manière immédiate par l'utilisateur : c'est le cas des modèles appelés "boîtes noires" (*e.g.,* les réseaux de neurones [8] (ANN) et les séparateurs à vaste marge [6] (SVM)). D'autres sont naturellement plus interprétables : c'est le cas des modèles boîtes blanches (*e.g.,* les arbres de décision [4], [5]).

Dans certains domaines appelés "domaines critiques" (*e.g.,* la médecine, les service bancaires, ...), la compréhension (la description) des résultats issus d'un modèle d'apprentissage est une condition aussi importante que sa performance prédictive. De ce fait, la qualité de ces modèles réside donc dans leurs capacités à fournir des résultats ayant de bonnes performances en prédiction tout en produisant simultanément des résultats compréhensibles par l'utilisateur. Dans ce genre de situation, il semble logique de se diriger vers les modèles boîtes blanches. Cependant, certaines études comparatives (*e.g.,* [9]) ont montré que les modèles boîtes noires sont plus performants que les modèles boîtes blanches. De ce fait, il existe un grand intérêt à étudier le compromis *"interprétation - performance"*.

Pour tenter de résoudre cette problématique, deux grandes voies existent (voir Figure 1). La première voie consiste à rendre les algorithmes les plus performants (*e.g.,* les SVM, les ANN, les forêts aléatoires, *etc*) plus interprétables. Cette voie a fait l'objet de nombreuses recherches dans les années passées et cela dans de nombreuses disciplines ([10],[11],[12]). La deuxième voie a, quant à elle, pour but d'améliorer la performance des modèles "naturellement" interprétables". Plusieurs techniques ont été proposées dans ce cadre ([13],[14]). Néanmoins, les améliorations apportées se font souvent en détériorant la qualité d'interprétation du modèle.

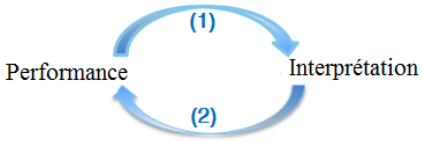


FIGURE 1 – La problématique d'interprétation - performance

Dans cet article, nous nous intéressons à l'étude de la deuxième voie (*i.e.,* de l'interprétation vers la performance). Pour résoudre cette problématique, on part du principe que, d'une part, la performance du modèle dépend de son pouvoir prédictif (*i.e.,* sa capacité à bien prédire la classe des nouvelles instances), et que d'autre part, l'interprétation des résultats du modèle dépend de sa capacité à décrire les données (*i.e.,* sa capacité à fournir des règles compréhensibles par les utilisateurs). De ce fait, notre objectif devient alors : *partir de la description vers la prédiction*.

2 Objectif

L'objectif de cet article est de traiter la problématique "de l'interprétation vers la performance" mais dans un cadre bien précis : *"lorsque chaque classe (ou quelques-unes) dispose d'une structure qui la caractérise"*. La figure 2 présente un exemple illustratif d'un jeu de données caractérisé par la présence de deux classes 'Setosa' et 'Virginica' où chacune dispose d'une structure spécifique : la classe 'Setosa' contient deux sous-groupes distincts ayant chacun des instances homogènes (*i.e.,* partagent les mêmes caractéristique). Tandis que la classe 'Virginica' contient trois sous-groupes distincts.

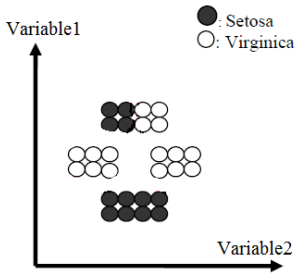


FIGURE 2 – Jeu de données caractérisé par la présence de deux classes ayant chacune une structure qui la caractérise

Dans ce cadre d'étude, on cherche à découvrir les différentes voies qui peuvent mener à une même prédiction. Il s'agit alors de découvrir la structure du concept cible. A titre d'exemple, deux patients x_1 et x_2 ayant comme prédiction un test positif pour l'AVC (*i.e.,* une grande probabilité d'avoir un Accident Vasculaire Cérébral) n'ont pas forcément les mêmes motifs et\ou les mêmes symptômes : il se peut que le patient x_1 soit une personne âgée, qui souffrait de la fibrillation auriculaire et par conséquent, elle a eu des maux de têtes et des difficultés à comprendre. Tandis que le patient x_2, pourrait être une jeune personne qui consommait de l'alcool d'une manière excessive et par conséquent, il a perdu l'équilibre. Généralement, ces différentes raisons que l'on souhaite détecter.

Cette étude a pour but donc, de découvrir, si elle existe, la structure du concept cible à apprendre puis muni de cette structure de pouvoir prédire l'appartenance au concept cible. L'objectif est donc d'essayer de ***décrire et de pré-***

dire d'une manière simultanée. Si le but est atteint, on aura alors à la fois : (1) la prédiction du concept cible et (2) le pourquoi de la prédiction grâce à la découverte de la structure de ce dernier.

Pour atteindre l'objectif désiré, au moins pour l'axe "performance-prédiction", on pense alors immédiatement aux arbres de décisions [4, 5] qui figurent parmi les algorithmes de classification les plus interprétables. Ils ont la capacité de fournir à l'utilisateur des résultats compréhensibles (sous formes de règles) et qui semblent donner une structure du concept cible appris. En effet, l'arbre de décision modélise une hiérarchie de tests sur les valeurs d'un ensemble de variables discriminantes. De ce fait, les instances obtenues suivant un certain chemin (de la racine vers les feuilles terminales) ont normalement la même classe et partagent ainsi les mêmes caractéristiques. Cependant, selon la distribution des données dans l'espace d'entré, l'arbre de décision crée naturellement des polytopes (fermés et ouverts) à l'aide des règles. La présence des polytopes ouverts empêche l'algorithme de découvrir la structure *'complète'* du concept cible Y. La figure 3 présente un exemple illustratif du fonctionnement de l'arbre de décision sur un jeu de données caractérisé par la présence de deux classes ('rouge' et 'noire'), 350 instances et deux variables descriptives $x1$ et $x2$. A partir de ce résultat, on constate que cet algorithme fusionne les deux sous-groupes de classe 'rouge' (situés à la droite de la figure), malgré le fait que les exemples du premier sous-groupe ont des caractéristiques différentes de celles du deuxième sous-groupe. Dans le cas extrême, ces deux sous-groupes peuvent même être très éloignés et donc être de caractéristiques assez différentes.

Pour ce qui est de l'axe "interprétation - description", la découverte de la structure globale d'une base de données (non étiquetées) peut être réalisée à l'aide d'un algorithme de clustering. Les résultats fournit par les algorithmes de clustering sont souvent interprétables. L'utilisateur peut facilement identifier les profils moyens des individus appartenant à un groupe mais aussi les variables qui ont un grand impact sur la formation de chaque groupe. Ceci peut être réalisé à l'aide des outils statistiques simples mais variés (*e.g.,* le tableau des distances, les représentations graphiques et les indices de qualité). Dans notre cas d'étude, l'utilisation d'un algorithme de clustering s'avère insuffisante. En effet, le concept cible à apprendre n'est pas pris en compte et les prédictions ne peuvent pas être réalisées avec une bonne performance : deux instances qui partagent les mêmes caractéristiques n'ont pas forcément la même classe (voir le cluster A de la figure 5 a)).

Partant de ce constat, nous nous sommes fixés comme objectif d'essayer de réaliser au sein du même algorithme d'apprentissage la *description* et la *prédiction* du concept cible à apprendre. Pour cela nous proposons d'adapter un algorithme de clustering aux problèmes de classification supervisée. Autrement dit, l'idée est de modifier un algorithme de clustering afin qu'il soit un bon classifieur (en

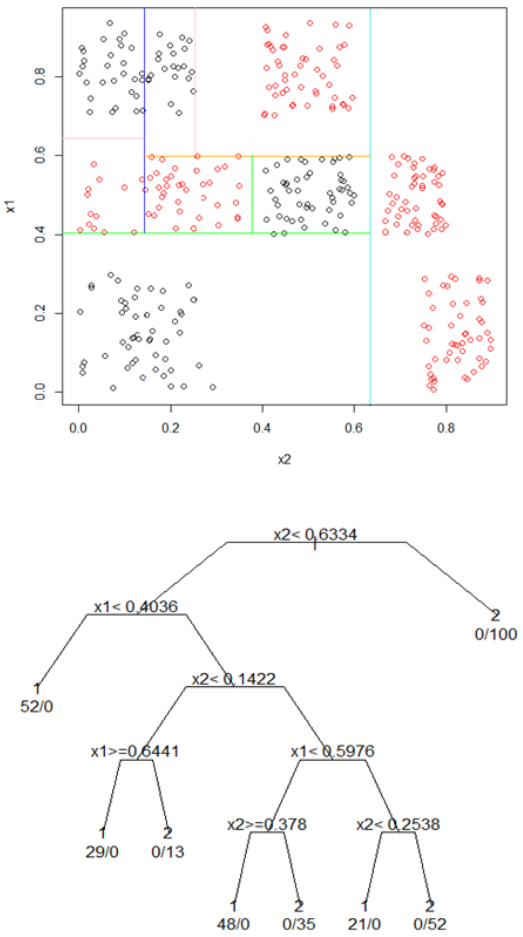


FIGURE 3 – Résolution d'un problème de classification binaire par un arbre de décision. Les feuilles étant pures en termes de classes, l'arbre ne se développe plus.

termes de prédiction) tout en gardant sa faculté à décrire les données et donc le concept cible à apprendre. On parle alors de la ***classification à base de clustering*** (ou décrire et prédire simultanément).
Notre objectif est donc de chercher un modèle qui prend en considération les points suivants :

1. La découverte de la structure interne de la variable cible (la proximité entre les individus).
2. La maximisation de la performance prédictive du modèle.
3. L'interprétation des résultats.
4. La minimisation des connaissances *a priori* requises de la part de l'utilisateur (*i.e.,* pas ou peu de paramètres utilisateur).
5. La minimisation de la taille (complexité) du modèle prédictif - descriptif

3 Classification à base de clustering

Dans notre contexte, la classification à base de clustering est définie comme étant un problème de classification supervisée où un algorithme de clustering standard est soumis à des modifications afin qu'il soit capable à prédire correctement la classe des nouvelles instances. Plus formellement, L'objectif des algorithmes de classification à base de clustering est le même que celui de la classification supervisée, c'est-à-dire prédire la classe des nouvelles instances à partir d'un ensemble de données étiquetées. Dans la phase d'apprentissage (voir Figure 4), ces algorithmes visent à découvrir la structure complète du concept cible à partir la formation des clusters à la fois pures en termes de classe et distincts (*i.e.,* les instances dans chaque cluster doivent être le plus homogènes possibles et différentes de celles appartenant aux autres clusters). A la fin du processus d'apprentissage, chaque groupe appris prend j comme étiquette si la majorité des instances qui le forme sont de la classe j (*i.e.,* l'utilisation du vote majoritaire). Au final, la prédiction d'une nouvelle instance se fait selon son appartenance à un des groupes appris. Autrement dit, l'instance reçoit j comme prédiction si elle est plus proche du centre de gravité du groupe de classe j (*i.e.,* utilisation du 1 plus proche voisin).

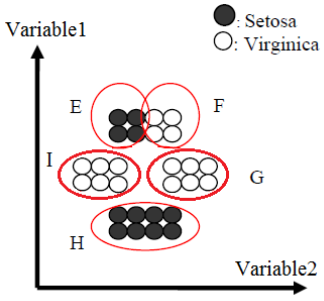


FIGURE 4 – Principe de la classification à base de clustering

Dans la littérature, plusieurs algorithmes de clustering standard ont soumis à des modification afin qu'il soient adapté au problème supervisé ([15],[20],[19]). Ces algorithmes sont connus sous le nom de *"clustering supervisé"* (ou en anglais *"supervised clustering"*). La différence entre le clustering standard (non supervisé) et le clustering supervisé est donnée par la figure 5. Les algorithmes de clustering supervisé visent à former des clusters pures en termes de classe tout en minimisant le nombre de clusters K. Cette contrainte sur K va empêcher les algorithmes de clustering supervisé à découvrir la structure complète du concept cible. De ce fait, un seul cluster peut donc contenir un certain nombre de sous-groupes distincts (voir le cluster G de la figure 5 b)).

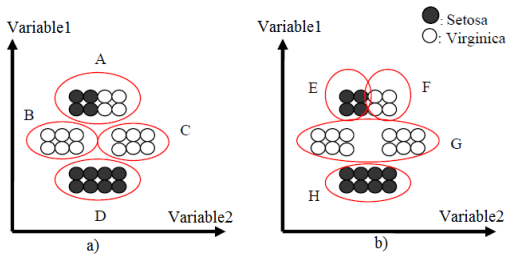


FIGURE 5 – La différence entre le clustering standard a) et le clustering supervisé b)

3.1 État de l'art

Les algorithmes de clustering supervisé les plus répandus dans la littérature sont :

• Al-Harbi *et al.* [15] proposent des modifications au niveau de l'algorithme des K-moyennes. Ils remplacent la distance euclidienne usuelle par une distance euclidienne pondérée. Le vecteur de poids est choisi de telle sorte que la confiance des partitions générées par l'algorithme des K-moyennes soit maximisée. Cette confiance est définie comme étant le pourcentage d'objets classés correctement par rapport au nombre total d'objets dans le jeu de données. Dans cet algorithme, le nombre de clusters est une entrée.

• Aguilar *et al.* [16] et Slonim *et al.* [17] ont proposé des méthodes basées sur l'approche agglomérative ascendante. Dans [16], les auteurs ont proposé un nouvel algorithme de clustering hiérarchique (S-NN) basé sur les techniques du plus proches voisins. Cet algorithme commence par N clusters où N est le nombre d'objets du jeu de données. Ensuite, il fusionne successivement les clusters ayant des voisins identiques (*i.e.,* objets proches ayant la même étiquette). Par conséquent, tous les voisins ayant les distances plus courtes que le premier ennemi (*i.e.,* l'objet qui n'a pas la même étiquette) seront collectés. Tishby *et al.* ont introduit dans [18] la méthode 'information bottleneck'. Basée sur cette méthode, ils ont proposé une nouvelle méthode de clustering (agglomérative) [17] qui maximise d'une manière explicite, l'information mutuelle entre les données et la variable cible par cluster.

• Dans [19], Cevikalp *et al.* ont proposé une méthode qui crée des clusters homogènes, nommée HC. Ces travaux sont effectués dans le but de trouver le nombre et l'emplacement initial des couches cachées pour un réseau RBF. Cevikalp *et al.* supposent que les classes sont séparables puisqu'ils cherchent des clusters purs en classe. Le nombre et l'emplacement des clusters sont déterminés en fonction de la répartition des clusters ayant des chevauchements entre les classes. L'idée centrale de l'algorithme HC est de partir d'un nombre de clusters égal au nombre de classes puis de diviser les clusters qui se chevauchent en tenant compte de l'information supplémentaire donnée par la variable cible.

• Eick *et al.* [20] proposent quatre algorithmes de clustering supervisés, basés sur des exemples représentatifs. Ce

genre d'algorithme a pour but de trouver un sous-ensemble de représentants dans l'ensemble d'entrées de telle sorte que le clustering généré en utilisant ce dernier minimise une certaine fonction de pertinence. Dans [20], les auteurs utilisent une nouvelle fonction pour mesurer la qualité de ces algorithmes. Cette fonction remplit les deux critères suivants : i) Minimisation de l'impureté de classe dans chaque cluster ii) Minimisation du nombre de clusters.

• Vilalta *et al.* [12, 21] mais aussi Wu *et al.* [22] utilisent eux la technique appelée "décomposition des classes". Cette technique repose sur deux étapes principales : (1) réalisation d'un clustering de type k-moyennes (où k est selon les auteurs une entrée ou une sortie de l'algorithme) par groupe d'exemples qui appartiennent à la même classe j. Le nombre de clusters peut différer par classe. On obtient alors P clusters au total ($P = \sum_j k_j$). (2) Entraînement d'un classifieur sur les P classes résultantes et interprétation des résultats. Cette technique permet aussi l'amélioration des classifieurs linéaires (simples).

4 Travaux passés et futurs

L'objectif de cet article est de chercher un modèle qui prend en considération les points énumérés en fin de Section 2 et que ne permettent pas totalement les algorithmes décrits dans la section précédente. Comme une première piste de travail, nous nous intéressons à modifier l'algorithme des K-moyennes. Cet algorithme figure parmi les algorithmes de clustering les plus répandus et le plus efficace en rapport temps de calcul et qualité [23]. Les différentes étapes de l'algorithme des K-moyennes modifié sont présentées dans l'algorithme 1.

1) Prétraitement des données
2) Initialisation des centres
3) Répéter un certain nombre de fois (R) jusqu'à convergence
   3.1 Coeur de l'algorithme
4) Choix de la meilleure convergence
5) Mesure d'importance des variables (après la convergence et sans réapprendre le modéle)
6) Prédiction de la classe des nouveaux exemples.

Algorithme 1 : K-moyennes.

4.1 Travaux réalisés

Etape 1 des k-moyennes (voir Algorithme 1) : Généralement, la tâche de clustering nécessite une étape de prétraitement non supervisé afin de fournir des clusters intéressants (pour l'algorithmes des K-moyennes voir par exemple [24] et [25]). Cette étape de prétraitement peut empêcher certaines variables de dominer lors du calcul des distances. En s'inspirant de ce résultat, nous avons montré dans [26] (à travers l'évaluation de deux approches : conditional Info et Binarization) que l'utilisation d'un prétraitement supervisé peut aider l'algorithme des K-moyennes standard à atteindre une bonne performance prédictive. La prédiction de la classe étant basée sur l'appartenance au cluster le plus proche après la fin de convergence puis sur un vote majoritaire dans le cluster concerné.

Etape 5 des k-moyennes (voir Algorithme 1) : Dans [27] nous avons proposé une méthode supervisée pour mesurer l'importance des variables après la convergence du modèle. L'importance d'une variable est mesurée par son pouvoir prédictif à prédire l'appartenance des instances aux clusters (*i.e.,* la partition générée par le modèle). Autrement dit, une variable est importante si elle est capable de générer une partition proche de celle obtenue en utilisant toute les variables. Dans cette étude, nous n'avons pas considéré les interactions qui peuvent exister entre les variables (*e.g.,* une variable n'est importante qu'en présence des autres). Ceci, fait l'objet des futurs travaux (parmi d'autres).

4.2 Travaux en cours

Etape 2 des k-moyennes (voir Algorithme 1) : Dans [28] nous avons proposé une nouvelle méthode d'initialisation des k-moyennes dans le cas supervisé : le cas où la valeur de K correspond au nombre de classes à prédire (C). Cette méthode est basée sur l'idée de la décomposition des classes après avoir prétraité les données à l'aide de la méthode proposée dans [26]. A l'avenir nous comptons étendre cette méthode lorsque $K \neq C$.

Etape 3.1 des k-moyennes (voir Algorithme 1) : La fonction objectif utilisée dans l'algorithme des k-moyennes standard consiste à minimiser l'inertie intra et par conséquent maximiser l'inertie inter cluster. L'élaboration d'un nouveau critère à optimiser pour pouvoir atteindre l'objectif d'un algorithme à base de clustering est à l'étude.

4.3 Travaux à venir

Pour la deuxième partie de la thèse en cours, on s'intéressera à traiter :

Etape 4 des k-moyennes (voir Algorithme 1) : L'algorithme des k-moyennes n'assure pas de trouver un minimum global. De ce fait, il est souvent exécuté plusieurs fois (on parle de "réplicates") et la meilleure solution en terme d'erreur est alors choisie. Dans le cadre de la classification à base de clustering, le critère utilisé pour choisir la meilleure réplicate doit prendre en considération le compromis homogénéité - pureté des clusters. Dans les travaux à venir, on cherchera à définir un nouveau critère pour le choix de la meilleure "réplicates" dans le cas supervisé.

Etape 6 des k-moyennes (voir Algorithme 1) : par défaut la classe prédite est la classe majoritaire présente dans le cluster. On cherchera à améliorer ce point comme cela existe déjà pour les arbres de décisions.

5 Conclusion

L'ensemble des travaux décrits dans cet article permettra d'obtenir un algorithme complet de "classification à base de clustering" basé sur un algorithme de k-moyennes qui aura été "supervisé" à chaque étape.

Références

[1] Pages, J.P., Cailliez, F., Escoufier, Y. Analyse factorielle : un peu d'histoire et de géométrie. *Revue de Statistique Appliquée, Vol XXVII*, pp. 5-28, 1979.

[2] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering : A review. *ACM* , pp. :264-323, 1999.

[3] Antoine Cornuéjols, Laurent Miclet. Apprentissage artificiel : concepts et algorithmes, Eyrolles 2010.

[4] John Ross Quinlan. C4. 5 : programs for machine learning, volume 1. Morgan kaufmann, 1993.

[5] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. Classification and regression trees. *CRC press*, 1984.

[6] V. N. Vapnik. The Nature of Statistical Learning Theory. *Springer-Verlag New York, Inc.*, 1995.

[7] S. B. Kotsiantis. Supervised machine learning : A review of classification techniques. *In Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering*, pp. 3-24, 2007.

[8] S. Haykin. Neural Networks : A Comprehensive Foundation. *Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition*, 1998.

[9] R. Caruana and A. Niculescu-Mizil. An Empirical Comparison of Supervised Learning Algorithms. *Proceedings of the 23rd International Conference on Machine Learning*, pp. 161-168, 2006

[10] Monirul Kabir, Md Monirul Islam, and Kazuyuki Murase. A new wrapper feature selection approach using neural network. *Neurocomputing*, 73(16) pp. 3273-3283, 2010.

[11] Glenn Fung, Sathyakama Sandilya, and R. Bharat Rao. Rule extraction from linear support vector machines. *In Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 32-40. ACM, 2005.

[12] Ricardo Vilalta, and Irina Rish. A Decomposition of Classes via Clustering to Explain and Improve Naive Bayes. *ECML, volume 2837 of Lecture Notes in Computer Science*, pp. 444-455. Springer, 2003

[13] A. A. Gill, G. D. Smith, and A. J. Bagnall. Improving Decision Tree Performance Through Induction- and Cluster-Based Stratified Sampling. *IDEAL, volume 3177 of Lecture Notes in Computer Science*, pp. 339-344. Springer, 2004

[14] Hassan, Md R and Kotagiri, R. A new approach to enhance the performance of decision tree for classifying gene expression data. *BMC proceedings*, pp. S3, 2013

[15] Sami H Al-Harbi and Victor J Rayward-Smith. Adapting k-means for supervised clustering. *Applied Intelligence*, 24(3), pp. 219-226, 2006.

[16] Aguilar J., Roberto Ruiz, José C Riquelme, and Raúl Giràldez. Snn : A supervised clustering algorithm. *In Engineering of Intelligent Systems*, pp. 207-216. Springer, 2001.

[17] Noam Slonim and Naftali Tishby. Agglomerative information bottleneck. *MIT Press*, pp 617-623, 1999.

[18] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 1999.

[19] Hakan Cevikalp, Diane Larlus, and Frederic Jurie. A supervised clustering algorithm for the initialization of rbf neural network classifiers. *In Signal Processing and Communications Applications, 2007. SIU 2007. IEEE 15th, pages 1âĂŞ4. IEEE*, 2007.

[20] Christoph F Eick, Nidal Zeidat, and Zhenghong Zhao. Supervised clustering algorithms and benefits. *In Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*, pp. 774-776. IEEE, 2004.

[21] Francisco Ocegueda-Hernandez and Ricardo Vilalta. An Empirical Study of the Suitability of Class Decomposition for Linear Models : When Does It Work Well ? *In SIAM* 2013.

[22] Junjie Wu, Hui Xiong and Jian Chen. COG : local decomposition for rare class analysis *In Data Min Knowl Disc (DMKD)*, pp. 191-220. Springer, 2009.

[23] Jain, Anil K. Data Clustering : 50 Years Beyond K-means. *Pattern Recogn. Lett, Elsevier Science Inc*, pp. 651-666, 2009

[24] Milligan, G., Cooper, M. A study of standardization of variables in cluster analysis. *In : Journal of Classi cation, Springer-Verlag.*, 5(2) pp.181-204,1988

[25] Celebi E. M., Hassan A. Kingravi, Patricio A. Vela. A Comparative Study of Efficient Initialization Methods for the K-Means Clustering Algorithm. *Journal of Expert Systems with Applications*, 40(1) pp.200-210, 2013

[26] Alaoui Ismaili O., Lemaire V., Cornuéjols A. Supervised pre-processings are useful for supervised clustering. *Springer Series Studies in Classification, Data Analysis, and Knowledge Organization*, 2015

[27] Alaoui Ismaili O., Lemaire V., and Cornuéjols A. A supervised methodology to measure the variables contribution to a clustering. *In International Conference on Neural Information Processing (ICONIP), Kuching, Sarawak, Malaysia*, 2014

[28] Vincent Lemaire, Oumaima Alaoui Ismaili, Antoine Cornuéjols "An Initialization Scheme for Supervized K-means", *to appear in International Joint Conference on Neural Networks (IJCNN), IEEE, Ireland*, 2015

N-gram as an alternative to stemming in the automated HONcode detection for English and French

C. Boyer[1] L. Dolamic[1]

[1]Heath On the Net Foundation

Chemin du Petit-Bel-Air 2, 1225 Chêne-Bourg, Switzerland
celia.boyer@healthonnet.org

Abstract

The HONcode of conduct is composed of eight ethical and quality criteria (www.hon.ch/Conduct.html) This study evaluates supervised automatic classification algorithms capability to determinate whether a health related web page is in compliance with any of those criteria. Various length character n-gram vectors were used to represent health web page documents. Classification performance of the 5-grams was compared to that obtained by words or stems. The study attempts to determine whether the language-independent approach might result in similar classification performance as word-based classification for both English and French languages. The training/testing collection for both languages were created from web page fragments extracted by HONcode experts during the manual certification process as the basis for individual HONcode compliance. Naive Bayes classifier and DF (document frequency) dimensionality reduction metrics were used. The overall results of this study indicate that the n-gram tokenization provides a potentially viable alternative to document word stemming.

Keywords:

Supervised machine learning, Character n-gram, Naive Bayes, SVM, quality criteria, HONcode

1 Introduction

The health related information provided by Internet, to millions of people worldwide includes at the same time reliable, useful information and potentially harmful information [1]. Discerning trustworthy web-based health information from manipulated or biased content is difficult task for end-users [2], since no obligatory regulatory standards exist for ethical and quality content of health-related websites. The Health On the Net Foundation was created in 1996. Its mission is the promotion of transparency and quality for online medical and health-related information. The HON Foundation's Code of Conduct [3], consists of eight procedural guidelines that help to indicate the credibility of online health information. Currently, HONcode expert reviewers manually assess, re-assess, and certify health websites for compliance with the HONcode conduct principles.

The websites respecting the HONcode display the unique HONcode seal. In this manner the certification gives the possibility to the individual user to easily identify whether website respects the HONcode. The presence of such a seal indicates that the website provides the answers to questions such as: "when was the site last updated" or "who is the author of the content", however it does not validate the website's content per se. The detailed HONcode guidelines can be found at http://www.healthonnet.org/HONcode/Conduct.html.

HONcode is the most widely utilized healthcare website Code of Conduct. To date, HON has certified 8'300 health websites worldwide. HONcode certification is multilingual, since identification of quality web-based information in their native languages is crucial to end-users worldwide. The largest number of certified websites is written in English (34%), followed by French (28%) and Spanish (10%). However, the manual initial evaluation and subsequent audits in order to assess HONcode conformity are limited due to the time-and-personnel intensive processes they involve. Furthermore, many high-quality health websites may not have HONcode certification because the review process is voluntary – conducted at the request of a health site's webmaster. Using current methods, large-scale HON reviews for certification of hundreds of thousands of health websites is not feasible.

In this study we are evaluating the possibility to address the HONcode certification as well as its multilingual aspect automatically. In this study we are reporting to what extent the machine learning based automated classification system, in combination to language independent n-gram classification can be of assistance in the process of HONcode certification. This type of tokenization has proven to be effective for different, particularly for the morphologically complex languages [4], especially when linguistic tools such as stemmers [5] were lacking for the given language. The n-gram approach also showed greater robustness in the setting of frequent typographical errors in the source text [6]. This approach is tested on English and French document collections.

This research project is funded by the European Kconnect project (2015-2017, project No. 644753).

Related work

HONcode certification process must become at least partially automated in order to remain relevant, taking into account the quantitative growth of Internet-based health information websites. The research has already been initiated by HON in the direction of exploiting Machine Learning (ML) algorithms for automated text classification [7, 8]. To be used by a machine learning based classifier the content of healthcare document is represented as vectors of weighted tokens [9], since it cannot be directly interpreted by them. Word stems or lemmas is typical approaches for this purpose, with a goal of increasing the system's recall [10, 11]. Linguistic treatment of such stemming has proven to be a powerful tool especially with morphologically complex languages [12]. For the word stem or

lemma to be created the linguistic tools such as stemmers or morphological analyzers are required. Unfortunately, those tools are not always available. Not for all languages, nor for specific language subdomains, such as health. Thus, a language-independent approach to this issue is needed. Linguistic treatment in English does not imply enormous difference in performance, while for the French the performance differences are somewhat more important [12]. Showing the interchangeability between language dependent (stemming, lemmatization) and language independent approach for these two languages, would however give a credible baseline for other languages, for both simple or more complex from the morphological point of view. Employing character n-grams in this purpose is widely present in text classification [13]. The machine learning system represented each document as a vector of mutually independent weighted "terms" (tokens, features). A term itself could be a word, a lemma, or a stem, etc. The lemma or stems used as "terms" have for the goal matching of different words derived from the same root (e.g. derive, derived, derivation). However, accomplishing this goal requires language dependent tools. The current study explores the usability of language independent character n-gram opposing words and stems. Various approaches to creation of character n-grams exist. Some of them take into account the word boundaries where word "privacy" would yield following 5-grams "_priv", "priva", "rivac", "ivacy" and "vacy_" with "_" representing the word boundary. In other cases the word proximity is taken into account, in which case the phrase "privacy policy" would yield 5-grams such as "acy_p". In our approach we consider every word of the document independently and we do not take the word boundaries into consideration as described in [4]. Thus, for the case of n=5, the document phrase "privacy policy" would yield the following n-grams: "priva", "rivac", "ivacy", "polic" and "olicy".

When it comes to dealing with the issue whether information available on the internet can be trusted or not, studies conducted to date have mostly focused on the e-commerce domain. Nevertheless, despite the volume of research in this domain, basic consensus about the meaning of trust remains elusive [14]. The same lack of consensus applies regarding trustworthiness of online health information. Most reported studies have tested one specific aspect of trust as reported in the following articles [15, 16]. The lack of de facto standards regarding online health information makes comparisons among research studies difficult. In the scope of the CLEF initiative eHealth related workshops have been proposed (http://clefehealth2014.dcu.ie/). However, none of the tasks within these workshops focused on the particular issue of the quality of health information. Thus, no collection is available which would allow to have common data in order to appropriately compare results amongst research approaches.

2 Methods

The test system used in this study is based on the machine learning framework described in [17]. Different Machine Learning algorithms such as SVM, Naive Bayes (NB) or KNN have been implemented and tested. Experience in an earlier study [18] in addition to experimental results obtained for this work led authors to retain a Naive Bayes machine learning algorithm (with tf-idf weighting instead of word frequency, due to better precision), as the most appropriate for automated determination whether a webpage complied with each HONcode principles [19]. In addition, the comparison to SVM for certain criteria is performed in order to give support to such a decision. The document frequency algorithm was used to perform the feature selection in this study [10].

There are eight HONcode principles: *Authority, Complementarity, Privacy, Attribution, Justifiability, Authorship, Sponsorship* and *Advertising.* The principle *Attribution* was divided into two separate criteria, namely *Reference* and *Date* due to disparate fulfilment requirements for these two elements. Authors developed separate classifiers for each of the HONcode criteria, because a document's belonging to one HONcode compliance class is independent, under all circumstances, from its conformance with one or more other HONcode compliance classes (*any-of classification*) [10]. All experiments performed used ten-fold cross validation.

The collections used for training of the machine learning algorithms were created from actual webpage excerpts. As the HONcode team previously reviewed candidate websites for HONcode certification, they stored the part of the site's webpage that indicated compliance with one of the HONcode principles. Another challenge the authors faced with was defining the document (classification unit) within these collections. One recent study using machine learning techniques used the sentence as a classification unit [8]. Closer inspection of such an approach revealed that it generats numerous errors, as individual sentences in a document may not conform to criteria even though the document as a whole may [8]. Each document within training/test collection consists of one previously described extract from one website for one specific HONcode criterion. HONcode certification process includes websites in any of a large set of languages. However, the current study was limited to pages written in English and French languages. These languages have been chosen for two main reasons: 1. the collected sets of compliance justification extracts in English and French were the most exhaustive and 2. these languages represent two different languages types from the morphological complexity point of view. Table 1 gives the number of documents (extracts) available in these two collections. This collection is not for the moment publicly available.

Table 1: Number of extracts per criteria (English and French HONcode compliance extracts collections)

		No. extracts	
Criterion		**English**	**French**
Authority	HC1	2812	2338
Complementarity	HC2	2835	2005
Privacy policy	HC3	2683	2055
Reference (*Attribution*)	HC4	2349	1888
Justifiability	HC5	872	827
Contact details	HC6	2861	2349
Financial disclosure	HC7	2700	2098
Advertising policy	HC8	1412	627
Date (*Attribution*)	HC9	2794	2158

In the current study the words (W1) used as terms is taken as the "baseline". The results obtained by the baseline are compared to those of 5-grams (C5) and stems (W1s). For the English the porter stemmer was used, while for the French we use the snowball stemmer[1]. Various length of character n-grams have been tested in the scope of this evaluation. The 5-gram was retained since it has shown the behaviour closest to the behaviour of word or stems for both languages tested. The goal was to determine the extent to which words might be replaced by 5-grams or stem tokens while not sacrificing system classification performance.

In conducting the study, prior to the tokenization, authors removed stop words such as «le», «la». «du» etc.. from the stud-

1 http://snowball.tartarus.org/algorithms/french/stemmer.html

ied documents. These lists contain 174 words for English and 126 for French.

We have chosen the precision (P), recall (R) and F_1-measure to present the quality of the classification for each of the HONcode criteria. The F_α-measure [19] combines the precision and recall measures, allowing us to give relative importance to each of them. In this study α is set to 1. In this way this measure gives equal importance to both recall and precision.

3 Results

The values for precision (P), recall (R) and F_1-measure for each criterion with NB classifier are given in the Table 2. Those values represent the averages of each respective measure over 10 runs.

Table 2: Precision (P), recall(R) and F_1-measure, NB

HON code		Tokenization					
		English			French		
		W1	W1s	C5	W1	W1s	C5
HC1	P	**0.64**	0.63	0.60	**0.69**	0.66	0.63
	R	**0.75**	0.71	0.65	**0.72**	0.72	0.71
	F_1	**0.69**	0.67	0.62	**0.70**	0.69	0.67
HC2	P	**0.83**	0.82	0.77	**0.95**	0.94	0.91
	R	0.96	0.96	**0.96**	0.88	**0.92**	0.90
	F_1	**0.89**	0.88	0.85	0.92	**0.93**	0.91
HC3	P	**0.91**	0.90	0.89	**0.94**	0.92	0.92
	R	**0.98**	0.98	0.91	0.98	0.98	**0.99**
	F_1	**0.94**	0.94	0.94	**0.96**	0.95	0.95
HC4	P	0.56	0.58	**0.60**	**0.82**	0.77	0.76
	R	**0.61**	0.58	0.53	0.41	0.43	**0.44**
	F_1	**0.59**	0.58	0.56	0.55	0.55	**0.56**
HC5	P	**0.74**	0.69	0.64	**0.97**	0.94	0.76
	R	**0.31**	0.27	0.25	0.38	**0.42**	**0.42**
	F_1	**0.43**	0.37	0.36	0.55	**0.58**	0.54
HC6	P	0.92	**0.93**	0.92	**0.96**	0.94	0.91
	R	**0.94**	0.93	0.86	**0.86**	0.86	0.83
	F_1	**0.93**	0.93	0.89	**0.91**	0.90	0.87
HC7	P	**0.77**	0.76	0.74	**0.92**	0.82	0.75
	R	**0.79**	0.76	0.71	0.43	**0.85**	0.82
	F_1	**0.78**	0.76	0.72	0.59	**0.83**	0.78
HC8	P	**0.77**	0.76	0.72	**1.00**	0.95	0.83
	R	0.73	0.70	**0.79**	0.37	0.35	**0.50**
	F_1	**0.75**	0.73	0.75	0.54	0.51	**0.63**
HC9	P	0.97	**0.97**	0.95	**0.99**	0.99	0.96
	R	**0.95**	0.94	0.93	**0.92**	**0.92**	0.89
	F_1	**0.96**	0.93	0.94	**0.96**	0.95	0.93

Three tokenization schema namely word (W1), stems (W1s) and 5-gram (C5) for both English and French were used, with 30% of most frequent terms being retained for each class. In this table, the highest value for each measure, tokenization and criteria combination is marked in bold.

The results presented in Table 2 show that the tokenization resulting in the highest values in terms of precision varies between HONcode criteria but also between languages. For the English, the W1 tokenization results in highest value of precision and F1 for all the principles except for "Reference (*Attribution*) HC4" and "Contact details HC6" respectively obtained with 5-gram tokenization C5 (60%) and W1s (93%). The same tendency is noticeable for the French between word and 5-gram tokenization. However the differences between the highest and other values never exceed 10%, using highest value as a baseline.

Even thought the NB classifiers tend to be outperformed by more sophisticated algorithms such as SVM, this is not the case for the collection on-hand. Tables 3 and 4 give the comparison of the classification performance for "Reference (HC4)" and "Privacy policy (HC3)" criteria respectively, between NB and SVM algorithms. These algorithms were compared for both stemming and 5-gram tokenization.

Table 3: NB vs SVM, for stems (W1s) and 5-grams (C5), Reference criterion (HC4)

HC4	English				French			
	W1s		C5		W1s		C5	
	NB	SVM	NB	SVM	NB	SVM	NB	SVM
P	0.58	**0.62**	**0.60**	0.59	**0.77**	0.57	**0.76**	0.59
R	0.58	**0.76**	0.53	**0.77**	0.43	**0.68**	0.44	**0.66**
F_1	0.58	**0.68**	0.56	**0.67**	0.55	**0.62**	0.56	**0.62**

The results presented in Tables 3 and 4 show that, besides or the stem tokenization for the Reference criteria in English, the NB algorithm results in higher precision when compared to SVM for this collection. The most important difference can be noticed for the HC3 criterion (Table 4), where for the French language and W1s tokenization NB achieves precision of 0.92 compared to 0.46 achieved by SVM. In return, SVM results in higher recall, although the relative difference is not as important.

Table 4: NB vs SVM, for stems (W1s) and 5-grams (C5), Privacy policy criterion (HC3)

HC3	English				French			
	W1s		C5		W1s		C5	
	NB	SVM	NB	SVM	NB	SVM	NB	SVM
P	**0.90**	0.70	**0.89**	0.63	**0.92**	0.46	**0.92**	0.48
R	0.98	**0.99**	0.99	**0.99**	0.98	**0.99**	0.99	**0.99**
F_1	**0.94**	0.82	**0.94**	0.77	**0.95**	0.62	**0.95**	0.65

To demonstrate the similar behaviour of different tokenization in relation with dimensionality reduction, we have compared in Tables 5 and 6 the relative precision loss and relative recall gain respectively, between the cases where 80% and 30% of all features are kept.

Table 5: Average precision loss for English and French

	Token.	English			French		
		Kept %		Diff %	Kept %		Diff %
		80	30		80	30	
P	W1	0.89	0.79	-11.08	0.93	0.91	-1.76
	W1s	0.87	0.78	-10.28	0.92	0.88	-4.44
	C5	0.85	0.76	-10.41	0.90	0.83	-7.92

The loss in precision for the English language, shown by the results in Table 5 is the smallest in the case of W1s tokenization (10.28%). The gain in the recall is however the most important in the case of C5 being used (38.86%, Table 6) for this language. For the French the smallest loss in precision is noticed for W1 tokenization as well as the highest gain in the recall.

Table 6: Average recall gain for English and French

	Token.	English			French		
		Kept %		Diff %	Kept %		Diff %
		80	30		80	30	
	W1	0.59	0.78	31.14	0.48	0.71	47.53
R	W1s	0.57	0.76	33.13	0.51	0.72	41.37
	C5	0.53	0.74	38.86	0.52	0.72	40.16

4 Conclusion

The character n-gram tokenization has been evaluated in this publication with a goal to determine whether it could be used as an alternative to bag of words or stems. Based on the results presented in the Table 2, we can conclude that even though the word tokenization results in best performance in terms of precision. When the recall is the measure one desires to augment both the n-gram and stems impose themselves as the better solutions. This tendency is more pronounced for the French than English language, which could be explained by the fact that the French is a morphologically richer language, thus benefits more from linguistic treatment.

We have also shown that Naive Bayes algorithm is capable of outperforming the algorithms such as SVM. For the HC3 criterion, using W1s tokenization for the French the SVM results in precision not less than 50% smaller than that of NB, using NB as baseline. On the other hand the SVM results in somewhat higher recall for all language, tokenization, criterion combinations. The relative difference between two algorithms is however less important raising up to 37% (French, W1s, HC4). These results as well as time/resource consumption difference between the two algorithms are in favour of NB as a solution for the task presented.

The results presented in the tables 5 and 6 demonstrate that different tokenization techniques evaluated in this publication, namely word (W1), stem (W1s) or five gram (C5), show the same behaviour in precision/recall evolution towards dimensionality reduction by feature selection. This also indicates interchangeability of these tokenization technique. The baseline established here for the English and confirmed for the French, show that the language independent approach would be a viable alternative to word-based tokenization for wide variety of languages, especially for the morphologically complex ones.

Acknowledgements

The research is conducted in the scope of the European project Kconnect and funded by this project (2015-2018, project No. 644753).

References

[1] Fahy E, Hardikar R, Fox A, Mackay S. Quality of patient health information on the Internet: reviewing a complex and evolving landscape. Australasian Med J. 2014; 7(1) 24–28. PMID: 24567763

[2] White R, Horvitz E. Cyberchondria: Studies of the Escalation of Medical Concerns in Web Search. http://research.microsoft.com/pubs/76529/TR-2008-178.pdf Archived at: http://www.webcitation.org/6RoyoFIYT

[3] Boyer C, Baujard V, Scherrer J. HONcode: a standard to improve the quality of medical/health information on the internet and HON's 5th survey on the use of internet for medical and health purposes. In 6th Internet World Congress for Biomedical Sciences (INABIS 2000), 1999.

[4] Mc Namee P, Mayfield J. Character n-gram tokenization for european language text retrieval. Information Retrieval, 2004;7(1-2):73–97

[5] Mc Namee P, Mayfield J, Nicholas CK. Don't have a stemmer?: be un+concern+ed. In Sung-Hyon Myaeng, Douglas W. Oard, Fabrizio Sebastiani, Tat-Seng Chua, and Mun-Kew Leong, editors, SIGIR, ACM, 2008; 813–814.

[6] Naji N, Savoy J, Dolamic L. Recherche d'information dans un corpus bruité(ocr). In Gabriella Pasiand, Patrice Bellot, editors, CORIA, 6Editions Universitaires d'Avignon, 2011;271–28.

[7] Gaudinat A, Grabar N, Boyer C. Automatic Retrieval of Web Pages with Standards of Ethics and Trustworthiness Within a Medical Portal: What a Page Name Tells Us. Artificial Intelligence in Medicine [Internet]. Springer; 2007;185–189.

[8] Gaudinat A, Grabar N, Boyer C. Machine learning approach for automatic quality criteria detection of health web pages. In Klaus A. Kuhn, James R. Warren, and Tze-Yun Leong, editors, MedInfo, Studies in Health Technology and Informatics, IOS Press, 2007;129:705–709.

[9] Sebastiani F. Machine Learning in Automated Text Categorization, ACM Computing Surveys,2002; 34: 1-47.

[10] Manning CD, Raghavan P, Schutze H. Introduction to information retrieval. Cambridge University Press, 2008.

[11] Junker M, Hoch R. An experimental evaluation of OCR text representations for learning document classifiers. International Journal on Document Analysis and Recognition, 1998, 1(2):116-122

[12] Tomlinson, S. (2004). Lexical and algorithmic stemming compared for 9 European languages with Hummingbird SearchServerTM at CLEF 2003. In Comparative evaluation of multilingual information access systems. LNCS #3237 (pp. 286–300). Berlin: Springer-Verlag.

[13] Cavnar W.B., Trenkle J.M. N-Gram-Based Text Categorization, In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, 1994, 161—175

[14] Beldad A, de Jong M, Steehouder M. How shall I trust the faceless and the intangible? A literature review on the antecedents of online trust. Computers in Human Behavior 2010;26(5):857-869

[15] Eysenbach G, Powell J, Kuss O, Sa ER. Empirical Studies Assessing the Quality of Health Information for Consumers on the World Wide Web – A Systematic Review Journal of the American Medical Association JAMA 2002;287(20):2691—2700

[16] Sillence E, Briggs P, Harris PR, Fishwick L. How do patients evaluate and make use of online health information? Social Science and Medicine, 2007;64 (9):1853-1862.

[17] Williams K, Calvo RA. A framework for document categorization. 7th Australasian Document Computing Symposium. December 2002. Sydney, Australia. 13-19.

[18] Boyer C, Dolamic L. Feasibility of automated detection of honcode conformity for health related websites. IJACSA, 2014 Mar; 5(3):69-74, doi: 10.14569/IJACSA.2014.050309

[19] Baeza-Yates RA, Ribeiro-Neto B. Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.1999

Address for correspondence

Célia Boyer
Health on The Net Foundation
Chemin du Petit-Bel-Air 2
1225 Chêne-Bourg, Switzerland
celia.boyer@healthonnet.org
+41 (0) 22 37 26 250

De l'intérêt de l'éthique collective pour les systèmes multi-agents *

Nicolas Cointe[1] Grégory Bonnet[2] Olivier Boissier[1]

[1] Institut Henri Fayol – Mines Saint-Étienne, Laboratoire Hubert Curien, UMR CNRS 5516
[2] Normandie Université, GREYC, CNRS UMR 6072, F-14032 Caen, France

Mines Saint-Étienne, 158 cours Fauriel 42023 Saint-Étienne Cedex 2
nicolas.cointe@emse.fr gregory.bonnet@unicaen.fr olivier.boissier@emse.fr

Résumé

L'utilisation croissante des technologies multi-agents pour le développement de systèmes socio-techniques révèle de nombreux verrous, dont ceux liés à l'éthique des décisions autonomes que de tels systèmes peuvent être amenés à prendre. Ce problème, souvent considéré dans une perspective centrée sur l'agent, est ici abordé d'un point de vue du collectif soulevant ainsi les questions éthiques découlant des interactions entre agents autonomes ou de leur participation à des coalitions ou à des structures plus pérennes telles que des organisations. Cet article propose un panorama de ces différentes questions en vue de travaux à venir.

Mots Clef

Systèmes Multi-Agents, Éthique Collective, Dilemmes.

Abstract

The increasingly current and future presence of multi-agent systems in various areas leads us to ask many questions about the ethics of their decisions. Indeed, decisions that these systems must be made sometimes require, consideration of ethical concepts, both in as an individual entity and a member of an organization. This problem, often considered from an agent centered perspective, is addressed in this paper from a collective point of view. This position paper discusses the concepts to be modeled within an agent, a set of issues and a roadmap for addressing this question.

Keywords

Multi-Agents Systems, Collective Ethics, Dilemmas.

1 Introduction

L'introduction d'agents autonomes artificiels dans des domaines tels que le milieu hospitalier, le trading haute fréquence ou encore les transports pourrait soulever de nombreux problèmes si ces agents ne sont pas en mesure de comprendre et suivre certaines règles morales. Par exemple, des agents capables de comprendre et utiliser le code de déontologie médicale pourraient s'appuyer sur des motivations éthiques afin de choisir quelles informations diffuser, à qui et sous quelles conditions, conformément au principe du secret médical. L'intérêt pour le comportement éthique des agents autonomes semble être récemment apparu dans la communauté [9, 15], comme en témoignent les nombreux articles [5, 20, 22, 23] et conférences [1]. Cependant, ces travaux s'intéressent uniquement à l'éthique à l'échelle individuelle de l'agent.

Or, dans un système multi-agent (ou SMA), cette représentation individuelle permet à un agent de se comporter individuellement de manière éthique dans un collectif, mais le laisse démuni lorsqu'il doit tenir compte de l'éthique des autres agents. Par exemple, un agent gestionnaire d'investissements financiers pourrait être tout à fait capable de se comporter selon des principes de gestion responsable sans pour autant être capable de constater si ses partenaires ont les mêmes scrupules. Prendre en considération la dimension multi-agent de ce problème nécessite l'exploration de nouvelles pistes telles que la création d'une ou plusieurs éthiques collectives ou la prise de décisions en coopération face à des problèmes d'éthique. Ces questions ont d'autant plus d'importance dans le contexte actuel de déploiement d'un nombre croissant d'agents dans notre environnement, collaborant entre eux ou avec des humains. Cet article a pour but de proposer des définitions et des questions mettant en évidence la problématique des éthiques collectives dans les systèmes multi-agents. Elles permettront dans des travaux futurs de proposer une formalisation de notions de philosophie pour la représentation explicite de concepts éthiques au sein d'agents autonomes.

Cet article est structuré comme suit. La section 2 présente les concepts philosophiques et techniques employés dans cet article, puis en section 3 nous considérons un modèle abstrait d'éthique individuelle au sein d'un agent que nous utilisons ensuite pour présenter les problématiques spécifiques touchant aux questions d'éthiques collectives dans des systèmes multi-agents en sections 4 et 5.

*Les auteurs remercient le support de l'Agence Nationale de la Recherche (ANR), projet ETHICAA ANR-13-CORD-0006.

1. Symposium on Roboethics, International Conference on Computer Ethics and Philosophical Enquiry, Workshop on AI and Ethics, International Conference on AI and Ethics.

2 Préliminaires

Afin de tracer les contours des concepts employés dans cet article, nous commençons par donner quelques définitions de notions philosophiques d'éthique. Il ne s'agit nullement d'en donner une vision exhaustive. Nous invitons le lecteur intéressé à se reporter à la bibliographie pour approfondir ces concepts. La deuxième partie de cette section donnera également quelques définitions issues du domaine SMA.

2.1 Ethique et philosophie

L'éthique est une discipline philosophique qui traite depuis l'Antiquité de questions concernant le *bien-fondé d'un acte* et son *jugement par autrui* [7]. L'un des problèmes majeurs rencontrés en éthique est la définition des notions qui l'entourent. En effet, la littérature concernant l'éthique est riche d'une grande diversité de points de vues. Certains ouvrages tels que [28] sont le fruit d'exercices philosophiques qui tentent de démontrer que l'éthique repose sur des bases logiques et rationnelles. Les neurologues ont également menés de nombreuses expériences [4, 12] visant à montrer les phénomènes biologiques et psychologiques à l'origine de nos aptitudes morales. Enfin, les sciences cognitives ont brouillé les frontières entre disciplines et cherchent à proposer des modèles formels de raisonnement éthique [11]. Si de nombreuses théories ont été présentées, développées et confrontées depuis la philosophie antique jusqu'aux recherches actuelles sur les bonnes conduites à adopter dans la démarche scientifique et l'usage de la technique ou du pouvoir [14], nous admettons dans cet article la définition suivante de l'éthique, inspirée de travaux de philosophes tels que Paul Ricoeur [25] :

> L'**éthique** est une discipline philosophique pratique (traitant des actions) et normative (qui formule des règles) visant à indiquer comment les êtres humains doivent se comporter, agir et être envers ce et ceux qui les entourent. L'éthique propose souvent des compromis afin de concilier règles morales, désirs et capacités.

L'éthique peut ainsi être appliquée à divers domaines, en tenant compte de *règles morales* qui leur sont associées, ainsi que des *désirs* et *capacités* des individus qui y agissent. Bien que les termes *morale* et *éthique* désignent initialement une même idée, certains auteurs proposent diverses nuances [10]. Nous choisissons ici de considérer la morale comme un ensemble de règles théoriques que l'éthique prendra soin de concilier avec les désirs et capacités de l'agent [2] pour acquérir son caractère pratique.

> La **morale** désigne l'ensemble de règles déterminant la conformité des pensées ou actions d'un individu avec les mœurs, us et coutumes d'une société, d'un groupe (communauté religieuse) ou d'un individu pour évaluer son propre comportement. Ces règles reposent sur les valeurs normatives de bien et de mal. Elles peuvent être universelles ou relatives, c'est-à-dire liées ou non à une époque, un peuple, un lieu, etc.

Les philosophes ont proposé de nombreuses conceptions structurées de l'éthique sous forme de *principes éthiques*. Nous pouvons les catégoriser en trois approches majeures selon lesquelles le bien-fondé d'un acte ou d'une pensée est :

- jugé par sa conformité à des valeurs telles que la sagesse, le courage ou la justice, entre autres [18], appelée **éthique des vertus** ;
- jugé par son adéquation avec les obligations et permissions associées à la situation [1], appelée **éthique déontologique** ;
- évalué à l'aune de ses conséquences [27], appelée **éthique conséquentialiste**.

Ainsi l'éthique des vertus juge essentiellement l'acte sur l'adéquation entre les désirs qui le motivent et les vertus reconnues par les règles morales, tandis que l'éthique déontologique cherche davantage à vérifier une correspondance entre les devoirs mentionnés par les règles morales et les capacités de l'agent lorsqu'un choix lui est proposé. Enfin, l'éthique conséquentialiste cherche à concilier les désirs et règles morales dans les conséquences d'un acte réalisable dans la mesure des capacités de l'agent. Ces approches majeures capturent des principes éthiques plus spécifiques tels que l'Impératif Catégorique d'Emmanuel Kant pour l'éthique déontologique ou la doctrine du double effet de Thomas d'Aquin pour l'éthique conséquentialiste. Il arrive cependant qu'une situation amène un principe éthique à ses limites en devenant incapable de répondre à la question du choix de la meilleure action. Considérés comme le point problématique central de l'éthique, les dilemmes constituent une catégorie de problèmes jugés difficiles [21] :

> Un **dilemme** éthique est un choix donné à un agent entre deux options connues, avec du point de vue de l'agent une raison éthique de choisir individuellement chacune de ces options. L'agent n'a pas la possibilité d'opter pour les deux options. L'intérêt de chaque choix pour l'agent est équivalent. Tout choix entraînera un regret.

2.2 Agent autonome et système multi-agent

Dans cet article, nous considérons les modèles et technologies issues des SMA. Dans ce domaine, un *agent autonome* est un système informatique ou robotique situé dans un environnement, espace partagé avec d'autres agents avec lesquels il interagit. Nous nous intéressons plus particulièrement ici à des agents cognitifs, i.e. des agents dotés d'une représentation symbolique permettant de manipuler explicitement les concepts d'éthique que nous avons décrits dans la section précédente. Cette représentation explicite permet de mettre en place des raisonnements et des explications sur ces concepts en tenant compte de leur sémantique. Par le terme de *système multi-agent*, nous entendons

2. Le terme d'agent est utilisé dans cette sous-section au sens premier de l'individu auteur d'une action, et non d'agent autonome artificiel.

un ensemble d'agents autonomes plongés dans un environnement dans lequel ils évoluent de manière indépendante, peuvent communiquer entre eux et poursuivre leurs propres objectifs [13]. Ces agents sont souvent intégrés à des organisations et peuvent adopter des rôles décrivant leur fonction au sein du système. La conception d'un SMA touche ainsi à des problématiques d'intelligence artificielle, à des questions de relations et d'organisation sociale, et à la prise en compte de cet aspect dans les processus de décision.

3 Éthique individuelle

Afin de montrer comment les concepts éthiques présentés dans la section précédente peuvent être représentés au sein des SMA, nous considérons un modèle simplifié de ce que pourrait être un modèle éthique individuel et montrons comment les travaux de la littérature proposent de le mettre en oeuvre. Ce modèle nous permettra de proposer en section 5 une définition d'éthique collective et d'évoquer une série de problématiques liées à cette notion dans un cadre multi-agent.

3.1 Modèle éthique individuel

Nous considérons un modèle *Belief-Desire-Intentions* (BDI) [24] couramment utilisé pour modéliser des agents dotés de représentations symboliques dans le domaine des SMA. Il consiste en un modèle de raisonnement comprenant la représentation des informations sur l'état du monde (*beliefs*) et des objectifs de l'agent (*desires*) afin d'en déduire l'action possible (*intention*) qui dans un état de croyances devrait permettre au mieux de satisfaire les objectifs de l'agent. Nous définissons alors l'éthique individuelle comme suit :

> L'**éthique individuelle** est la combinaison de principes éthiques et de règles morales permettant à un processus de décider d'une action satisfaisant au mieux les règles morales, les désirs et les croyances de l'agent, étant donné les capacités de ce dernier.

Le modèle éthique individuel présenté en Fig. 1 enrichit ainsi le modèle BDI par un ensemble de *règles morales* et de *principes éthiques* intervenant dans le choix de l'action à effectuer. Les règles morales associent une valeur de bien ou de mal à des combinaisons d'actions, de croyances ou de désirs. Les principes éthiques, comme les règles d'Aristote ou l'Impératif Catégorique de Kant [16], se présentent comme des contraintes sur la conciliation des désirs et des règles morales. Le raisonnement éthique exploite alors ces principes éthiques pour identifier l'intention à maintenir.
Le choix des principes éthiques permet de mettre l'accent sur tel ou tel composant. Ainsi, par exemple, une conception vertueuse propose de se conformer à des règles morales portant sur des valeurs, tandis qu'un raisonnement déontologique se préoccupe de l'adéquation entre l'état supposé par les croyances de l'agent et les actions obligatoires ou interdites dans ces circonstances. Une éthique conséquentialiste, enfin, chercherait à optimiser ses chances d'atteindre des états définis comme désirables moralement. Ce modèle abstrait pourrait être mis en oeuvre par différentes approches issues de la littérature. Ainsi, par exemple, les travaux menés par Ronald Arkin [2] proposent une implémentation directe des règles qu'un humain devrait suivre pour adopter un comportement conforme à un code d'éthique pré-établi (dans son contexte : les règles d'engagement militaire). D'autres ont au contraire proposé une représentation explicite des règles morales par l'emploi de programmation logique [26], de logique non-monotone [19, 17], de techniques d'argumentation formelle [3] ou de modèles BDI normatifs [29].

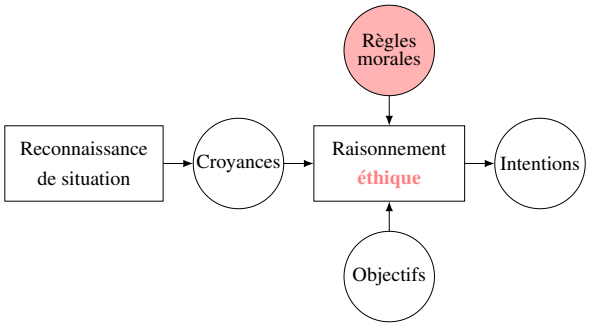


FIGURE 1 – Modèle éthique individuel

3.2 Exemple

Afin d'illustrer notre propos, considérons un exemple impliquant des agents propriétaires de sommes d'argent. Cet exemple est proposé dans un formalisme volontairement simplifié dont nous sommes conscient des limites. Chaque agent peut être dans l'un des trois états `PAUVRE`, `RICHE` ou (exclusif) `NEUTRE` et est doté du modèle d'action suivant :

- `VOLER(A)` pour prendre à un agent `A` une part des richesses en sa possession ;
- `DONNER(A)` pour donner de l'argent à l'agent `A` ;
- `TAXER(A)` pour réclamer à l'agent `A` le versement d'une partie de ses richesses ;
- `COURTISER(A)` pour tenter de s'attirer les faveurs de l'agent `A`.

Supposons un agent `Robin_des_Bois` (ou `RdB`) doté des règles morales suivantes :

$M1$. `MAL(TAXER(A)` $\wedge$ `PAUVRE(A))` ;
$M2$. `MAL(VOLER(A)` $\wedge$ `PAUVRE(A))` ;
$M3$. `BIEN(DONNER(A)` $\wedge$ `PAUVRE(A))` ;
$M4$. `MAL(RICHE(A))`.

Les trois premières règles définissent des interdits moraux ($M1$ et $M2$) et des devoirs moraux ($M3$) par association d'une action (`TAXER, DONNER, VOLER`), d'attributs d'agents (`PAUVRE(A)`) et d'une valuation morale (`MAL(X)` ou `BIEN(X)`). L'immoralité de la fortune est formulée par l'association d'un attribut d'agent (`RICHE(A)`) à une valuation morale négative.
`RdB` est également motivé par des désirs :

$D1$. `COURTISER(Marianne)` ;

*D*2. DONNER(A) si A est pauvre.

Supposons que le principe éthique de RdB est le suivant : *une action est éthique si elle est réalisable, désirée par l'agent et considérée comme bonne par une règle morale*. Considérons le cas où RdB n'aurait le choix qu'entre les deux actions possibles suivantes, étant donnée la situation courante :

1. DONNER(Paysan) sachant Paysan pauvre ;
2. COURTISER(Marianne).

Le choix le plus éthique est l'action 1 puisqu'elle se conforme aux capacités de l'agent, qu'elle est motivée par le désir *D*2 et constitue un devoir moral selon la règle *M*3. L'action 2, bien que n'allant à l'encontre d'aucun désir ou règle morale, n'est pas directement motivée par un devoir moral. Cette action est simplement amorale et pourrait être envisagée si l'action 1 devient impossible par exemple.

Comme l'action 1 ne contrevient pas aux autres règles morales ou désirs, RdB ne rencontre pas de dilemme. Ce modèle de raisonnement éthique individuel prend en compte l'existence d'autres agents par les croyances que RdB possède sur les autres agents. En effet, si PAUVRE(Paysan) n'est pas perçu par RdB alors le bien-fondé de l'action 1 n'est plus justifié. Notons enfin que si RdB n'a pas le désir *D*2, il se trouve alors face à un dilemme puisque l'action 1 va à l'encontre de ses désirs et l'action 2 ne satisfait pas sa morale. Bien que le choix de la bonne action dans un cas aussi idéal puisse paraître trivial, de nombreuses situations plus problématiques peuvent se présenter : choix entre plusieurs actions parfaitement éthiques, conflits dans les règles morales ou les désirs, choix entre plusieurs actions conformes aux désirs mais pas à la morale, et vice-versa, etc.

4 Ethiques individuelles en interaction

La prise en compte de la dimension multi-agent ne peut se satisfaire d'un agent doté d'une éthique individuelle, telle que décrite ci-dessus. En effet, ce dernier doit être capable, par exemple, de conduire des raisonnements sur l'éthique des autres agents et donc de pouvoir représenter l'éthique d'un autre agent selon son point de vue, la juger et l'utiliser dans ses interactions avec les autres agents.

4.1 Exemple

Afin d'illustrer notre propos, reprenons l'exemple précédent où, en sus d'être riches ou pauvres, les agents peuvent être nobles à l'instar de l'agent Prince_Jean (ou PceJ), ce qui leur permet l'usage exclusif de l'action TAXER. Considérons l'agent Petit_Jean (ou PJ) doté de désirs et de règles morales identiques à RdB à l'exception de *D*1 et l'agent Frère_Tuck (ou FT) identique à PJ mis à part *M*2 qui est remplacée par une règle plus générale :

*M*5. MAL(VOLER(A)).

Considérons enfin Sheriff_de_Nottingham (ou SdN), autre agent de ce système, qui dispose uniquement de la règle *M*5, tout comme PceJ.

Lors de la rencontre entre SdN et RdB, un différend éthique entre eux devrait pouvoir apparaître lorsque PceJ demandera à percevoir des taxes sur l'agent Paysan. Il en est de même pour PJ et RdB. Dans ces deux cas se posent les questions de représentation de l'éthique de l'autre, de son jugement et de la coopération étant donné ce jugement.

4.2 Description externe d'une éthique

La capacité à prendre en compte l'existence d'autres agents dans son raisonnement, comme illustrée dans la section précédente, doit pouvoir être enrichie au niveau de la reconnaissance de situation. En effet, celle-ci doit pouvoir être capable **de construire et de représenter le modèle de raisonnement éthique individuel transmis par un autre agent**. Cette reconstruction pourrait être faite de manière simple et directe par échange d'information ou, indirectement, par inférence et analyse du comportement observé.

Au-delà de cette capacité, un agent devrait être également capable de faire évoluer la description réalisée et donc de **pouvoir vérifier l'adéquation entre le comportement d'un autre agent et la description éthique qu'il en a construite**. Ainsi, par exemple, RdB doit pouvoir vérifier l'adéquation de la description éthique de PJ à partir des actes de celui-ci qu'il a observés.

4.3 Jugement de l'éthique d'un autre

Découlant de la capacité précédente, on peut s'attendre à ce qu'un agent puisse **juger l'éthique d'un autre**. Le sens du jugement est ici l'affectation d'une valuation morale au comportement d'un agent [8]. Le jugement peut être construit à partir de l'action de l'agent, de la prise d'un risque. Comme nous le verrons également dans la section 5 il peut être aussi construit en prenant en compte l'adéquation entre le comportement de l'agent et le rôle qu'il joue, par exemple.

De manière centrale, se pose alors la question de la définition d'une **métrique pour représenter et comparer des similarités entre éthiques**. Calculer un degré de proximité entre deux éthiques permettrait par exemple d'évaluer la possibilité d'entente entre des agents et la difficulté d'un rapprochement.

4.4 Coopération fondée sur l'éthique

Doté d'une telle capacité de jugement, nous pouvons ainsi imaginer que l'agent l'utilise pour décider d'une collaboration, pour partager des données sensibles ou pour constituer un collectif. **Il peut donc tenir compte de ce jugement dans le choix éthique d'une action.** Si RdB constate une forte similarité entre son éthique et celle de PJ, cela pourrait influencer l'évaluation de ses actions à l'égard de cet agent. Par exemple, si PJ vole une importante somme d'argent à un riche, il faudrait peut-être ne pas le considérer immédiatement comme un riche ordinaire en supposant que son éthique le poussera à distribuer cette somme aux pauvres.

5 Éthique collective

Dans un SMA, les interactions entre agents peuvent déboucher sur la formation de coalitions et peuvent également prendre place au sein d'organisations. Coalitions et organisations constituent des structures de plus haut niveau que la notion d'agent et peuvent également se voir dotées, explicitement ou non, d'une *éthique collective* :

> L'**éthique collective** est une combinaison de règles morales et de principes éthiques qui guident le choix d'actions satisfaisant les règles morales du collectif étant donné les désirs du collectif, les croyances individuelles des agents et leurs capacités.

5.1 Exemple

Reprenons l'exemple précédent. Après avoir constaté la similarité de leurs éthiques [3] et l'utilité d'une collaboration pour voler les agents fortunés, `RdB` et `PJ` peuvent décider de bâtir une organisation appelée `Joyeux compagnons (JC)`. De son côté, `SdN`, ayant trouvé des agents partageant son point de vue sur le vol, peut avoir créé un second collectif, les `Soldats (Sol)`, chargé de faire respecter $M5$. L'exemple des `JC` et des `Sol` illustre les questions majeures soulevées par les éthiques collectives à travers deux grandes catégories : celles qui concernent les interactions entre agents et organisations d'agents et celles qui concernent les interactions entre organisations.

5.2 Ethique collective et éthiques individuelles en interaction

Transposant à l'échelle du collectif l'éthique individuelle, il s'agit **pour les agents de construire une éthique collective**. Cette éthique peut découler directement de l'organisation ou être issue de l'interaction des agents qui composent le collectif. La mise en oeuvre *via* l'organisation peut par exemple découler de représentations normatives. La construction peut s'appuyer sur des techniques d'aggrégation spécifiques, comme en théorie du choix social ou en formation de coalitions.

A partir de la représentation d'une telle éthique, se pose la question de la manière dont un agent, externe ou interne au collectif, peut **identifier l'éthique de ce collectif**. Par exemple, `FT` devrait pouvoir identifier l'éthique des `JC`. Une piste pourrait être, de la même manière que pour un agent isolé, de la déduire à partir du comportement du collectif. Une fois identifiée, l'agent doit pouvoir **juger l'éthique du collectif.** Une fois que `FT` dispose d'une représentation de l'éthique des `JC`, il veut l'évaluer par rapport à la sienne et en mesurer la proximité afin de pouvoir décider par exemple d'entrer au sein de ce collectif.

De manière naturelle, il faut considérer **la cohabitation entre éthique(s) individuelle(s) et éthique collective.** Supposons que `FT` ait constaté une proximité entre son éthique individuelle et celle des `JC` et ait décidé de rejoindre ce collectif. Il apparaît comme nécessaire de définir comment l'agent va intégrer l'éthique collective au sein de son raisonnement et sous quelles conditions éventuelles. Une fois intégré à ce collectif, sa règle $M5$ ne pourrait-elle pas constituer une exception au sein de l'organisation ?

Découlant de cette cohabitation, peuvent s'ensuivre des changements, des modifications. Se pose alors la question de **l'évolution de l'éthique collective** à partir, par exemple, des éthiques individuelles. Le collectif des `JC` pourrait obtenir les règles du nouvel arrivant et décider de faire évoluer l'éthique collective. De son côté, l'agent pourrait laisser de côté son éthique individuelle pour se conformer pleinement à l'éthique du collectif. Si ni le collectif, ni l'agent ne peuvent se résoudre à réviser leurs éthiques respectives, `FT` pourrait également se contenter de n'effectuer que des actes conformes aux deux éthiques, c'est-à-dire *donner aux pauvres* dans notre exemple.

`SdN` ayant constaté l'entrée de `FT` chez les `JC`, des modifications de son jugement à son encontre sont envisageables, même s'il ne l'a jamais vu commettre de vols. Nous voyons ainsi un autre aspect, celui du **jugement de l'éthique de membres d'un collectif à partir de l'éthique collective**.

Il s'agit, enfin, de pouvoir **faire respecter l'éthique collective au sein d'une organisation.** Il faut alors considérer les réactions possibles d'un collectif face à un écart de l'un de ses membres vis-à-vis de l'éthique collective. Dans la mesure où les agents peuvent être contraints à choisir entre leur éthique individuelle et celle du collectif, il peut être intéressant de tenir compte de ces individualités lors de l'attribution des rôles par exemple.

5.3 Ethiques collectives en interaction

Passant au niveau supérieur, nous pouvons soulever quelques points découlant de la possible existence de plusieurs éthiques collectives au sein d'un SMA. Nous pouvons ainsi nous interroger sur les **relations entre organisations en fonction de la proximité de leurs éthiques collectives.** Ces relations peuvent varier en fonction de la situation (par exemple face à une menace extérieure à laquelle aucune de ces deux organisations ne pourrait survivre sans collaboration avec l'autre).

Naturellement se pose la question de **soumettre un agent à plusieurs éthiques collectives**. Sous réserve d'un ensemble de conditions, un agent proche d'un ensemble de collectifs pourrait être tenté d'adhérer à plusieurs de ces organisations, ce que l'on retrouve dans le domaine de la formation de coalitions recouvrantes. À l'inverse, un agent contraint à des appartenances multiples devrait chercher à concilier des éthiques collectives. De plus, cela pourrait également introduire une modification du comportement de ces organisations l'une envers l'autre.

Enfin se pose la question de **la fusion ou fission d'organisations pour des raisons éthiques.** L'éthique peut être envisagée comme quelque chose de dynamique et source de changement dans les organisations. Par exemple, une fis-

3. Au sens d'une métrique comme évoqué précédemment.

sion d'une organisation pourrait être envisagée si l'éthique collective perd sa cohérence, afin d'établir de nouvelles éthiques distinctes proposant des alternatives cohérentes. À l'inverse, si la proximité entre deux organisations devient importante, il pourrait être envisageable de chercher un consensus éthique en vue d'une fusion.

6 Conclusion et perspectives

En nous appuyant sur un modèle abstrait d'éthique individuelle, nous avons présenté dans cet article un vaste ensemble de questions structurées autour des interactions entre agents, interactions entre agents et organisations et interactions entre plusieurs organisations. Nous avons en particulier identifié trois questions clés pour un agent : comment représenter l'éthique des autres agents, les juger et prendre en compte ce jugement dans les mécanismes de décision ? Nous avons aussi identifié trois questions clés pour une organisation : comment construire, fusionner ou fissionner des éthiques collectives, comment les faire respecter, comment les faire cohabiter avec des éthiques individuelles ? Après nous être dotés d'un premier modèle plus précis de l'éthique individuelle, nous envisageons de définir des mesures de similarité et de mécanismes de jugement qui sont les éléments nécessaires et fondamentaux pour envisager des éthiques collectives.

Références

[1] L. Alexander and M. Moore. *Deontological ethics*. In Edward N. Zalta, editor, The Stanford Encyclopedia of Philosophy. 2012.

[2] R. Arkin. *Governing Lethal Behavior in Autonomous Robots*. Chapman an Hall, 2009.

[3] K. Atkinson and T.J.M. Bench-Capon. *Addressing moral problems through practical reasoning ?* Journal of Applied Logic, 6(2) : 135-151,2008.

[4] B. Baertschi. *Neurosciences et Éthique : Que nous apprend le dilemme du wagon fou ?* IGITUR, 3(3) : 1-17,2011.

[5] A.F Beaver. *Robot ethics : the ethical and social implication of robotics*. In Moral machines and the threat of ethical nihilism, pages 333-386, 2008.

[6] J.-C. Brustoloni, *Autonomous Agents : Characterization and Requirements*, Carnegie Mellon University, 1991.

[7] M. Canto-Sperber *Dictionnaire d'éthique et de philosophie morale*, PUF, 2004.

[8] H. Coelho and A. C. da Rocha Costa, *On the Intelligence of Moral Agency*, 2009.

[9] Commission de réflexion sur l'Éthique de la Recherche en science et technologies du Numérique d'Allistene (CERNA), Rapport num. 1, *Éthique de la recherche en robotique*, 2014.

[10] A. Comte Sponville, PUF, *La philosophie*, 2012.

[11] F. Cova, *L'Architecture de la Cognition Morale*, EHESS, 2014.

[12] A. R. Damasio, *L'Erreur de Descartes : la raison des émotions*, Paris, Odile Jacob, 1995.

[13] J. Ferber, *Les systèmes multi-agents : Vers une intelligence collective*, Paris, Inter Editions, 1995.

[14] J. Fieser. Ethics. In *The Internet Encyclopedia of Philosophy*, ISSN 2161-0002. 2015.

[15] Future of Life Institute, *Research Priorities for Robust and Beneficial Artificial Intelligence* , 2015.

[16] J.-G. Ganascia, *Ethical System Formalization using Non-Monotonic Logics*, Cognitive Science conference, 2007.

[17] J.-G. Ganascia, *Modelling ethical rules of lying with Answer Set Programming*, Ethics and Information Technology, vol. 9, pp. 39-47, 2007.

[18] R. Hursthouse. *Virtue ethics*. In Edward N. Zalta, editor, The Stanford Encyclopedia of Philosophy. 2013.

[19] S. Larroque. *Simulation des raisonnements éthiques par logiques non-monotones*. RJCIA, 2014.

[20] D. McDermott. *Why Ethics is a High Hurdle for AI*. North American Conference on Computers and Philosophy, 2008.

[21] T. McConnell. *Moral dilemmas*. In Edward N. Zalta, editor, The Stanford Encyclopedia of Philosophy. 2014.

[22] B.M McLaren. *Computational models of ethical reasoning : challenges, initial steps, and future directions*. IEEE Intelligent Systems, 21(4) : 29-37,2006.

[23] J.H Moor. *The Nature, Importance, and Difficulty of Machine Ethics*. IEEE Intelligent Systems, 21(4) : 29-37,2006.

[24] A. S. Rao and M. P. Georgeff, *BDI Agents : From Theory to Practice*, ICMAS, 1995.

[25] P. Ricoeur. *Soi-même comme un autre*, Points Essais 330, 1990.

[26] A. Saptawijaya and L.M. Pereira. *Towards Modeling Morality Computationally with Logic Programming*. 16th International Symposium on Practical Aspects of Declarative Languages, pages 104-119, 2014.

[27] W. Sinnot-Armstrong. *Consequentialism*. In Edward N. Zalta, editor, The Stanford Encyclopedia of Philosophy. 2014.

[28] B. Spinoza. *L'Éthique*, 1677, Folio Essais 235.

[29] M. Tufis and J.-G. Ganascia. *Normative Rational Agents : A BDI Approach*. 1st Workshop on Rights and Duties of Autonomous Agents, CEUR Proceedings Vol. 885, pages 37-43, 2012.

Authentification d'un utilisateur à partir de ses traces d'interaction

Fatma Derbel[1,2] Pierre-Antoine Champin[1] Amélie Cordier[1] Damien Munch[2]

[1] Université de Lyon, CNRS Université Lyon 1, LIRIS, UMR5205, F-69622, France
[2] Ignition Factory

fatma.derbel@liris.cnrs.fr

Résumé

Cet article [1] présente une démarche pour concevoir et implémenter un système d'authentification des utilisateurs dans les plates-formes web de formations certifiantes. Le système que nous proposons s'appuie sur deux modules : l'un combinant intelligemment différentes méthodes d'authentification connues, en fonction de leur disponibilité, et l'autre proposant une nouvelle approche d'authentification des utilisateurs à partir de leurs traces d'interaction. Ce système sera intégré dans la plate-forme de formation en ligne conçue et développée par la société Ignition Factory [2]. Dans cet article, nous présentons nos travaux préliminaires pour la définition de l'architecture de ce système, ainsi que l'état de l'art des méthodes d'authentification que nous avons réalisé.

Mots Clef

Authentification des utilisateurs, traces modélisées, environnements informatiques pour l'apprentissage humain, plates-formes de formation en ligne.

1 Introduction

Les plates-formes de formation en ligne sont de plus en plus répandues, notamment du fait du succès actuel des MOOCs (Massive Open Online Courses). On distingue deux types d'usage de ces plates-formes : d'une part la diffusion ouverte, massive et libre de contenus pédagogiques sur des sujets variés, et d'autre part, l'usage pour les formations certifiantes. Dans ce dernier cas, l'authentification des utilisateurs soulève des problèmes qui n'apparaissent pas souvent dans le cas général. Par exemple, lorsqu'il s'agit de leur compte en banque ou de leur compte sur un réseau social, les utilisateurs ont rarement intérêt à partager leurs éléments d'authentification (mot de passe, réponse à une question secrète, etc.). Quand bien même ils le feraient, ils seraient les seuls à devoir en assumer les conséquences. En revanche, dans le cas d'une formation certifiante, les utilisateurs pourraient être amenés à partager leurs éléments d'authentification pour frauder en vue de l'obtention de la certification. Du point de vue des organismes utilisant ces plateformes pour délivrer des certifications, la question de la fiabilité de l'authentification est donc cruciale car elle garantit la crédibilité des certificats délivrés. Cet article s'intéresse à la problématique de l'authentification d'apprenants dans le contexte des formations certifiantes en ligne. Plus précisément, nous cherchons à garantir avec un certain degré de confiance, et ce, tout au long de la formation, que c'est bien la même personne qui l'a suivie et validée. Nous cherchons donc à bâtir un système qui a les propriétés suivantes :

Le système est **indépendant du contexte matériel de l'apprenant** : nous ne faisons pas d'hypothèse forte sur la possibilité de l'apprenant à avoir accès à des dispositifs tels qu'une webcam, un lecteur d'empreintes digitales ou encore un micro. Le système permet **une authentification continue** : nous proposons une solution qui vérifie régulièrement l'identité de l'utilisateur, et ce, durant tout le processus de formation. Ce processus garantit un minimum d'interruptions de l'apprenant durant son activité.

Le système analyse en **temps réel** les comportements de l'utilisateur et déclenche **dynamiquement** des alertes en cas de comportements suspects. À leur tour, ces alertes déclenchent un mécanisme complémentaire d'authentification.

Le système permet une **combinaison de différente méthode d'authentification** (voir 2.1) selon leur disponibilité à la phase inscription.

Le système pourra utiliser **des mesures comportementales** pour déterminer l'identité de l'utilisateur en fonction d'une analyse de son comportement, capturé via ses interactions avec le système et via ses activités toute au long de la session d'apprentissage. Ces interactions peuvent en effet être capturer d'une manière continue, difficilement falsifiable et non intrusive. L'ensemble des informations de ces interactions peuvent donc créer un modèle de référence et utiliser dans des algorithmes pour vérifier l'identité de l'apprenant.

Le système doit être **adapté au contexte des formations en ligne**, c'est-à-dire qu'il doit prendre en compte des valeurs et des mesures générées par la plate-forme d'apprentissage (note d'un examen ou d'un qcm, taux de partici-

1. Ce travail a été réalisé dans le cadre du projet OFS - Open Food System, programme investissements d'avenir

2. `http://ignition-factory.com/fr/`

pation au forum, durée de suivre une session, etc.) pour authentifier les apprenants.
Le système que nous décrivons dans cet article, repose d'une part sur la combinaison intelligente de mécanismes d'authentification existants, et d'autre part sur un module d'authentification à partir de traces d'interaction. Les différentes sources d'authentification disponibles sont combinées et pondérées de sorte à déterminer un taux de certitude associé à l'authentification établie.
Les problématiques scientifiques liées à la création de ce système, sont diverses et de complexité croissante : identifier les moyens d'authentification disponibles chez les apprenants ; savoir comment combiner les différentes informations d'authentification ; savoir à quel moment déclencher une alerte et vérifier l'identité de l'utilisateur effectuant l'activité pour confirmer ou infirmer son identité ; définir des mesures de confiance et des métriques pour garantir l'identité d'un utilisateur connecté ; être capable de s'adapter au contexte de l'activité ; et enfin, construire un mécanisme d'authentification à partir des traces d'interaction d'un utilisateur avec un environnement informatique.
Dans la section suivante, après un rappel du contexte et des contraintes qui lui sont liées, nous présentons un état de l'art des méthodes d'authentification existantes. Ensuite, dans la section 3, nous présentons l'architecture générale du système que nous proposons, ainsi que les composants qui ont déjà été réalisés. Enfin, en section 4, nous discutons des questions de recherche que nous avons identifiées. Nous mettons en particulier l'accent sur la problématique de l'authentification à partir des traces d'interaction qui est le principal enjeu scientifique de cette thèse.
L'ensemble de ce travail est réalisé dans le cadre d'une thèse qui vient de débuter, en collaboration entre le LIRIS et la société Ignition Factory.

2 Contexte et état des lieux quant à l'authentification

Nos travaux s'inscrivent dans le cadre du développement d'une nouvelle plate-forme de formation en ligne (LMS, Learning Management System) à destination des entreprises (TPE/PME) et des professionnels indépendants. Cette plate-forme développée par la société Ignition Factory vise à combler certains points faibles de la formation en ligne, à savoir le contrle de l'identité et des connaissances, ainsi que les lourdeurs administratives pour la formation professionnelle. Ignition Factory propose pour cela deux axes d'innovation principaux : le premier vise à automatiser les démarches administratives et à faciliter l'accès aux formations et à leur remboursement ; le deuxième cherche à résoudre le problème de l'authentification des utilisateurs afin de délivrer les certifications, avec comme objectif idéal de donner une garantie de l'identité des utilisateurs équivalente à celle des formations traditionnelles en présence. C'est ce second axe qui nous concerne tout particulièrement et que nous allons développer dans la suite de l'article. Nous présentons dans la suite une étude des différentes méthodes d'authentification des utilisateurs, dans un contexte général, puis dans le contexte spécifique de l'apprentissage en ligne.

2.1 Approches classiques de l'authentification

L'authentification est le processus de vérification de l'identité des utilisateurs. La littérature [1] organise les approches d'authentification des utilisateurs en trois catégories : authentification à base de connaissances (i.e. en fonction de questions / réponses), authentification à base d'objets et authentification biométrique.
L'authentification à base de connaissances est la méthode qui permet de vérifier l'identité des utilisateurs en se basant sur des informations mémorisées par le système lors de la phase d'inscription, par exemple un mot de passe ou une réponse à une question de sécurité. Ce sont les méthode d'authentification les plus populaires et classiques puisque les données d'authentifications sont faciles à mémoriser par les utilisateurs. En raison de la facilité à contourner ce type de système lorsque l'utilisateur a la volonté de partager ses identifiants, nous ne réaliserons pas une analyse détaillée des différentes approches.
L'authentification à base d'objets est une approche selon laquelle les utilisateurs sont authentifiés par les informations contenues dans un objet physique en leur possession, par exemple une clé USB ou une carte électronique. Dans les examens en ligne, la présence d'un tel objet pour vérifier l'identité des utilisateurs peut augmenter le degré de sécurité des informations, mais ces objets peuvent également être perdus ou prêtés, ce qui nous ramène au problème précédent. De plus, de telles méthodes ajoutent de fortes contraintes matérielles (nécessité d'avoir des lecteurs de cartes par exemple).
L'authentification biométrique est l'ensemble des techniques et méthodes qui permettent la vérification automatique de l'identité des personnes sur la base des caractéristiques physiologiques et/ou comportementales [2]. Nous allons maintenant détailler les méthodes d'authentification biométriques. Ces méthodes peuvent être classées en deux catégories : les méthodes basées sur l'analyse des traits physiques qui sont particuliers, distincts et permanents pour chacun, tel que les empreintes digitales, la forme de la main, le fond de l'oeil, etc. ; et les autres méthodes fondées sur l'analyse du comportement et des actions de l'utilisateur tels que l'analyse de la signature, la dynamique de frappe clavier et la façon d'utiliser la souris. Les approches d'authentification physiques permettent d'avoir un pourcentage élevé de certitude mais ils exigent en contrepartie la disponibilité d'appareils de mesure coteux et parfois peu répandus (caméra, capteur d'empreintes digitales, etc.) alors que les systèmes d'authentification comportementaux sont moins coteux puisque l'on peut capter les informations nécessaires au moyen de dispositifs courants (clavier et souris).
Parmi ces méthodes nous citons celles les plus répandues

et utilisées dans le contexte de la formation en ligne [3].

La reconnaissance des empreintes digitales : c'est une technique biométrique qui se base sur l'identification de l'utilisateur par ses empreintes digitales. Ces techniques d'authentification sont les plus utilisées et répandues dans les applications de sécurité, par exemple pour le contrôle d'accès. Différentes techniques permettent de capturer les images d'une empreinte digitale telle que les capteurs optiques, capteurs en silicium, capteurs ultra sonique, etc.

La reconnaissance faciale [4] : c'est la méthode de d'authentification des utilisateurs par les traits du visage. Le principal avantage de cette méthode est de ne pas être intrusive. De plus, les algorithmes appliqués dans ces méthodes d'authentification ont connu des progrès très importants ces dernières années. Les approches de reconnaissance du visage sont fondées sur des algorithmes qui permettent de mesurer des attributs faciaux comme la distance entre les yeux, la position du menton, etc.

La reconnaissance vocale : c'est une technique qui se base sur les caractéristiques de la parole, uniques chaque personne. Les techniques de la reconnaissance vocale se basent sur l'analyse des caractéristiques quantitatives, par exemple la fréquence, les harmoniques, la puissance sonore, etc.

La dynamique de frappe au clavier [5] : la dynamique de frappe est une modalité biométrique comportementale qui permet d'identifier les personnes en mesurant leur façon de taper au clavier. Plus précisément, la méthode génère un comportement de frappe comme modèle de référence (une signature). Ce modèle est obtenu par la combinaison des attributs tels que la durée d'une frappe, la vitesse de frappe, le temps de latence entre l'appui des touches, le temps entre le relâchement d'une touche et la pression d'une autre, le temps entre deux relâchements de touches, etc.

La dynamique du mouvement de la souris [6] : la dynamique du mouvement de la souris est une technologie de la biométrie comportementale récemment proposée [7] qui permet d'identifier les utilisateurs en se basant sur l'analyse des clics et des mouvements de la souris. Les évènements de la souris sont généralement insuffisants (clic double, clic simple, molette) pour l'analyse d'un comportement, mais des algorithmes de regroupement ont été proposés pour obtenir des informations de plus haut niveau (trajet d'un mouvement puis clic, glisser-déposer, etc.) à partir desquels des modèles de référence significatifs peuvent être détectés. Des études ont démontré que la dynamique de la souris a un fort potentiel en tant que méthode d'identification des utilisateurs [8].

Combinaison des méthodes biométriques : D'autres études ont analysé la question de combiner plusieurs méthodes biométriques afin de créer un système d'authentification plus performant appelé «système d'authentification multi-modèles» (Mutimodal Biometric System MBS) [9]. Toutes les études présentées par Sahoo et al. [9] sur les MBS montrent une amélioration des performances suite à cette combinaison d'informations issues de méthodes biométriques distinctes. Un MBS intégrante les méthodes d'authentification par empreintes digitales et par caractéristiques du visage et de la voix est implémenté dans [10]. Les résultats de cette expérience montrent que la combinaison de différentes méthodes d'authentification est efficace pour surmonter les limitations de l'implémentation d'une seule méthode. D'autres études, comme [11], ont proposé un système d'authentification continue qui est fondé sur l'analyse du comportement des utilisateurs. Les méthodes utilisées dans cette approche sont la dynamique de frappe et la dynamique de mouvement de la souris. Les résultats obtenus par cette recherche sont prometteurs mais ils ne sont pas encore suffisamment fiables pour être déployés en conditions réelles.

2.2 L'authentification dans les plates-formes d'apprentissage

De nombreuses études ont appliqué les différents méthodes d'authentification citées ci-dessus dans différents domaines de la formation en ligne, tel que les MOOCs, les examens en ligne et les systèmes d'apprentissage en ligne. Une solution d'authentification par les connaissances dans les formations en ligne a été proposée dans les travaux [[12], [13]] d'A. Ullah. Ces derniers proposent un système basé d'une part sur le profil de l'utilisateur (Profile Based Authentication Framework, PBAF) et d'autre part sur l'identifiant et le mot de passe. L'apprenant se connecte à la plate-forme par son identifiant et mot de passe, puis est invité à répondre à un ensemble de questions de sécurité durant la phase d'apprentissage afin de générer son profil. Lors de l'accès aux examens en ligne, il doit répondre correctement à une liste de questions sélectionnées aléatoirement.

L'authentification biométrique est la plus répandue dans le domaine de la formation en ligne. En effet, plusieurs recherches ont été effectuées récemment pour l'application d'une ou d'un ensemble des méthodes d'authentification biométriques citées ci-dessus au sein des plates-formes de formation en ligne.

Des études comme [14], [15] ont proposé une intégration de la méthode d'authentification par la reconnaissance des empreintes digitales dans les plates-formes d'apprentissage, y compris dans la plate-forme de formation en ligne MOODLE [15]. Ce dernier (Fingerprint Identification System) implique un traitement d'image et d'extraction des informations à partir de l'image de l'empreinte digitale.Il est vrai que ce système permet de garantir la présence de l'apprenant au moment de l'identification, mais au prix d'un scanner d'empreintes digitales.

D'autres recherches [16], [17] ont proposé l'implémentation d'un système de reconnaissance faciale dans les plates-formes d'apprentissages. L'authentification faciale est généralement une méthode fiable pour l'identification, mais la variation des qualités des webcams en plus des conditions d'éclairage chez les apprenants ne permettent pas toujours d'avoir une image d'une qualité suffisante pour les

authentifier.

D'autres études se basent sur le comportement des apprenants via les frappes clavier pour avoir une authentification continue dans les examens en ligne [18]. L'utilisation de l'analyse de frappe clavier pour caractériser un comportement de l'utilisateur parane méthode intéressante, mais ne peut pas être le seul critère. En effet, sur certaines activités l'utilisateur n'utilise pas le clavier.

L'authentification par une seule méthode étant facilement sujette à échecs, certains travaux ont donc opté pour des MBS dans les formations en ligne. Dans [19], le système est basé sur l'authentification via les empreintes digitales et sur les mouvements de souris. La dynamique des mouvements de souris est utilisée pour gérer le comportement de l'utilisateur et la reconnaissance des empreintes permet d'augmenter la sécurité de l'authentification.

Dans le contexte des MOOCs, actuellement, seule la plateforme Coursera applique l'authentification biométrique pour la certification des apprenants [20] via son système nommé « signature Track » [21]. Ce système est basé sur deux approches d'authentification biométriques : la reconnaissance faciale et la reconnaissance du rythme de frappe clavier. Lors de l'inscription, le système demande de prendre une photo via une webcam, de fournir un scan de la carte d'identité et enfin de saisir une courte phrase. Une vérification de l'image de la webcam et de la photo des pièces d'identité est effectuée par Coursera, ensuite le scan de La pièce d'identité est supprimée. Pendant le suivi du cours, et à chaque évaluation (test, quiz, etc.) la plateforme demande de faire la même chose (photo webcam et saisir une phrase). La combinaison de ces deux caractéristiques biométriques constitue le processus de vérification de l'identité.

2.3 Synthèse et discussion

Nous avons présenté les principales méthodes d'authentification, et en particulier les approches biométriques, indépendamment de leur contexte d'application, puis nous avons présenté une sélection de systèmes d'authentifications spécifiquement liés aux environnements de la formation en ligne. Nous vérifions dans cette partie si ces systèmes répondent aux contraintes que nous avons identifiées et que nous reprenons ici et résumons dans le tableau 1 :

-**Être indépendant du contexte matériel (ICM)** de l'apprenant : en fait on cherche une solution non invasive (n'oblige pas les apprenants pour être filmé, photographié ou parlé) et destiné à toute personne équipée d'un ordinateur d'autent plus que l'on s'adresse au public professionnel.

-**Permettre une authentification continue (AC)** en authentifiant l'utilisateur tout au long de la formation, et pas uniquement au début ou à sa clôture.

-**Avoir une signature de l'utilisateur dynamique (D)** qui permet de prendre en compte le changement de comportement des apprenants au cours du temps.

-Permettre la **combinaison de différentes solutions d'authentification** (CSA).

-**Avoir un composant comportemental** (C) : Cette piste mérite d'être explorer puisqu'elle répond à certain limitations des approches existants.

-**S'intégrer au contexte de la formation en ligne** (CFL), ce qui signifie que le système doit s'adapter aux différentes activités pédagogiques (contenue du rédactionnel, vidéos, quiz, etc.).

TABLE 1 – Validation des contraintes par les travaux existants

Approche	ICM	CFL	AC	D	CSA	C
PBAF [12]	x		x	x		
FIS [15]						
Face recognition [17]						
Keystroke [18]	x		x			x
Multimodal [19]					x	x
Signature Track [21]					x	x

Nous constatons qu'aucune des méthodes proposées ne répond à tous nos critères. Dans le cas de l'adaptation au contexte de la formation en ligne, nous n'avons trouvé aucune méthode qui remplisse ce critère. Une première piste est de combiner ces approches. En supposant que cela soit possible, la combinaison de PBAF et Signature Track validerait cinq des six critères que nous avons fixés, ce qui permettrait de couvrir un bien plus large panel de situations, et renforcerait peut-être la certitude accordée à l'authentification finale. Néanmoins, le critère de l'adaptation au contexte de la formation en ligne, pourtant critique à notre avis, n'est toujours pas traité. Il nous faut donc explorer une nouvelle approche, jamais abordée dans ce contexte à notre connaissance.

3 Une approche d'authentification à base de traces

Le système d'authentification que nous proposons dans cet article repose sur une nouvelle approche : l'authentification des utilisateurs à partir de leurs traces d'interaction. Une trace d'interaction est un enregistrement de l'ensemble des actions effectuées par l'utilisateur dans un système. Ces traces sont un bon moyen pour capturer les activités des apprenants dans les plates-formes de formation en ligne sous la forme des inscriptions numériques d'une manière homogène, difficilement falsifiable et indépendant du contexte dans lequel l'utilisateur se situe. De plus on peut obtenir ces informations sans avoir de dispositif supplémentaire. Cependant, pour que ces traces capturées soient réutilisables et exploitables au sein d'un processus d'authentification, il faut s'appliquer à choisir la manière de les représenter et de les collecter. En effet, pour que ces traces d'interaction puissent devenir une source de connaissance pour l'identification des utilisateurs, elles doivent être correctement formalisées et nous devons disposer d'outils pour les exploiter. Pour cela, nous nous appuyons sur une représen-

tation de l'activité d'apprentissage sous forme des traces modélisées.
Le concept des traces modélisées (M-Trace) a été proposé par l'équipe SILEX du LIRIS [22]. Une trace modélisée est un objet informatique qui permet de décrire d'une manière détaillée les actions effectuées par les utilisateurs sur un système d'information pendant une durée déterminée. Une trace modélisée est toujours associée à un modèle contenant la définition des types des éléments qui la composent ainsi que leurs relations. Nous avons déjà une bonne expérience en matière d'exploitation des traces modélisées, et en particulier dans le contexte des plates-formes d'enseignement. En effet, nous avons proposé une architecture pour observer des activités d'apprentissage et les collecter sous forme des traces modélisées [23]. Un assistant Samo-TraceMe [3] a été conçu pour fournir des outils d'exploitation de ces traces collectées. Nous souhaitons nous appuyer sur notre expérience et étendre les outils que nous avons développés pour traiter la problématique de l'authentification.
Une architecture d'authentification par les traces modélisées est représentée dans la figure 1. L'acheminement des traces passe par les trois composants principaux :
-Le premier c'est l'outil responsable de la collecte de l'activité d'apprentissage. Le processus de collecte que nous proposons se divise en deux phases : la phase de spécification du traçage au cours de laquelle les concepteurs de la formation spécifient les événements qu'il souhaitent collecter sur des éléments identifiés par leur sélecteurs. La deuxième phase est la phase d'exécution de la collecte. Cette phase est inspirée du système de collecte TraceMe [4] [23] qui permet le traçage d'une activité d'apprentissage dans les MOOCs en prenant en compte les actions provenant du coté serveur de l'application et les interactions produites par le navigateur côté client. La limitation de ce système est que la collecte côté client doit se faire par l'installation volontaire par les utilisateurs d'une extension pour les navigateurs. Il existe donc un risque plus important de perdre les traces côté client si l'utilisateur choisit de ne pas l'installer, ou qu'il oublie de l'activer. Nous envisageons donc plutôt un script persistant qui s'intègre dans les pages webs de la plate-forme afin de collecter directement les traces de navigation de l'utilisateur selon une configuration définie dans la phase précédente.
-Le deuxième composant permet le stockage des traces collectées. Nous nous appuyons dans ces travaux sur le système KTBS [5] (Kernel Trace Base System). Il s'agit d'un logiciel open source qui permet de stocker des traces et leur modèle et permet également de calculer des traces transformées à l'aide d'un ensemble d'opérateurs (filtre, fusion, requête-Sparql). Le KTBS est développé comme un service web RESTful qui permet une communication via le protocole HTTP avec n'importe quelle application (quel que soit le langage de programmation et le système d'exploitation). Il utilise la représentation RDF pour le stockage des données afin de pouvoir capturer la sémantique de ces informations. Il permet aussi d'échanger ces données au format JSON.
-Le troisième est un ensemble de plusieurs modules qui permettent d'effectuer des transformations et des calculs de mesures sur les traces stockées. Ce composant représente le noyau de nos prochains travaux de recherches.

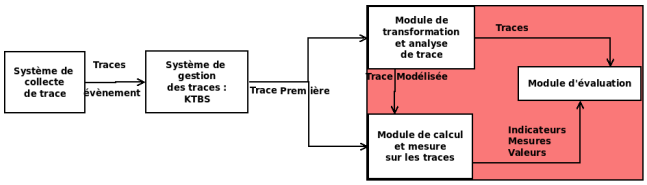


FIGURE 1 – Architecture du système d'authentification se de trace

4 Conclusion et questions de recherche

La problématique centrale de notre thèse est : est-il possible d'authentifier un utilisateur à partir de ses traces d'interaction ? Cette question majeure soulève d'autres problématiques et se confronte à des enjeux techniques significatifs. Nos premières réflexions nous ont amenés à identifier certains de ces éléments, que nous prévoyons d'approfondir dans le cadre de cette thèse.
Tout d'abord, se posent les questions éthiques. Dans quelle mesure une telle approche peut-elle être mise en place ? Présente-t-elle des risques ? Sera-t-elle acceptée par les utilisateurs dans le contexte de l'obtention de certificats dans un milieu professionnel ? Quelles garanties doit-on offrir, notamment en matière de sécurité et de visibilité des données collectées ? Comment bâtir une solution efficace tout en respectant la vie privée et le droit d'accès aux données des utilisateurs ?
Ensuite, se posent les questions liées à la démarche que nous souhaitons employer pour ce premier volet de notre travail. En effet, nous souhaitons mettre en place une approche inspirée d'autres méthodes d'authentification comportementales, et nous nous posons donc des questions similaires. Existe-t-il des éléments et des invariants dans les traces qui permettent d'identifier des utilisateurs ? Si oui comment peut-on les définir et les identifier ? Qu'en est-il de la fiabilité de l'approche ? Comment peut-on mesurer le degré de certitude et évaluer l'efficacité de nos mesures ? Comment utiliser ces valeurs dans un contexte pratique ?
Notre travail contient également un second volet qui consiste à proposer un algorithme permettant la combinaison intelligente de méthodes d'authentification selon leur disponibilité. Pour cela, il nous faudra être capable de ré-

3. https://github.com/fderbel/Assistant-Samo-Trace-Me
4. https://github.com/fderbel/Trace-Me
5. https://kernel-for-trace-based-systems.readthedocs.org/en/latest/

pertorier et caractériser les différentes méthodes d'authentification, de vérifier si elles sont disponibles chez les apprenants et de les solliciter au bon moment. Ensuite, il nous faudra proposer un modèle permettant de les combiner et de les pondérer en fonction de la confiance que l'on pourra leur accorder. L'ensemble de ces travaux fera l'objet d'une implémentation dans la plate-forme d'enseignement développée par la société Ignition Factory. Cette plate-forme nous permettra de tester nos outils et de mettre à l'épreuve notre hypothèse principale.

Références

[1] L. O'Gorman, "Comparing passwords, tokens, and biometrics for user authentication," vol. 91, no. 12, pp. 2021–2040.

[2] Z. Riha *et al.*, "Toward reliable user authentication through biometrics," *IEEE Security & Privacy*, vol. 1, no. 3, pp. 45–49, 2003.

[3] K. Delac and M. Grgic, "A survey of biometric recognition methods," in *Electronics in Marine, 2004. Proceedings Elmar 2004. 46th International Symposium*. IEEE, 2004, pp. 184–193.

[4] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition : A literature survey," *Acm Computing Surveys (CSUR)*, vol. 35, no. 4, pp. 399–458, 2003.

[5] P. S. Teh, A. B. J. Teoh, and S. Yue, "A survey of keystroke dynamics biometrics," *The Scientific World Journal*, vol. 2013, 2013.

[6] Z. Jorgensen and T. Yu, "On mouse dynamics as a behavioral biometric for authentication," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*. ACM, 2011, pp. 476–482.

[7] A. A. E. Ahmed and I. Traore, "A new biometric technology based on mouse dynamics," *Dependable and Secure Computing, IEEE Transactions on*, vol. 4, no. 3, pp. 165–179, 2007.

[8] C. Shen, Z. Cai, X. Guan, Y. Du, and R. A. Maxion, "User authentication through mouse dynamics," *Information Forensics and Security, IEEE Transactions on*, vol. 8, no. 1, pp. 16–30, 2013.

[9] S. K. Sahoo, T. Choubisa, and S. M. Prasanna, "Multimodal biometric person authentication : A review," *IETE Technical Review*, vol. 29, no. 1, pp. 54–75, 2012.

[10] A. K. Jain, L. Hong, and Y. Kulkarni, "A multimodal biometric system using fingerprint, face and speech," in *Proceedings of 2nd Int'l Conference on Audio-and Video-based Biometric Person Authentication, Washington DC*, 1999, pp. 182–187.

[11] S. Mondal and P. Bours, "Continuous authentication in a real world settings," in *Advances in Pattern Recognition (ICAPR), 2015 Eighth International Conference on*. IEEE, 2015, pp. 1–6.

[12] A. Ullah, H. Xiao, and M. Lilley, "Profile based student authentication in online examination," in *Information Society (i-Society), 2012 International Conference on*. IEEE, 2012, pp. 109–113.

[13] A. Ullah, H. Xiao, T. Barker, and M. Lilley, "Evaluating security and usability of profile based challenge questions authentication in online examinations," *Journal of Internet Services and Applications*, vol. 5, no. 1, p. 2, 2014.

[14] C. Gil, M. Castro, and M. Wyne, "Identification in web evaluation in learning management system by fingerprint identification system," in *Frontiers in Education Conference (FIE)*, 2010.

[15] S. Alotaibi, "Using biometrics authentication via fingerprint recognition in e-exams in e-learning environment," 2010.

[16] E. G. Agulla, L. A. Rifón, J. L. A. Castro, and C. G. Mateo, "Is my student at the other side ? applying biometric web authentication to e-learning environments," in *Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on*. IEEE, 2008, pp. 551–553.

[17] Q. Zhao and M. Ye, "The application and implementation of face recognition in authentication system for distance education," in *Networking and Digital Society (ICNDS), 2010 2nd International Conference on*, vol. 1. IEEE, 2010, pp. 487–489.

[18] E. Flior and K. Kowalski, "Continuous biometric user authentication in online examinations," in *Information Technology : New Generations (ITNG), 2010 Seventh International Conference on*. IEEE, 2010, pp. 488–492.

[19] S. Asha and C. Chellappan, "Authentication of e-learners using multimodal biometric technology," in *Biometrics and Security Technologies, 2008. ISBAST 2008. International Symposium on*. IEEE, 2008, pp. 1–6.

[20] P. Bond, "Biometric authentication in moocs," 2013.

[21] A. Maas, C. Heather, C. T. Do, R. Brandman, D. Koller, and A. Ng, "Offering verified credentials in massive open online courses : Moocs and technology to advance learning and learning research (ubiquity symposium)," *Ubiquity*, vol. 2014, no. May, p. 2, 2014.

[22] P.-A. Champin, A. Mille, and Y. Prié, "Vers des traces numériques comme objets informatiques de premier niveau : une approche par les traces modélisées," *Intellectica*, vol. 59, pp. 171–204, 2013.

[23] A. Cordier, F. Derbel, and A. Mille, "Observing a web based learning activity : a knowledge oriented approach," Ph.D. dissertation, LIRIS UMR CNRS 5205, 2015.

Influence of Selection Pressure in Online, Distributed Evolutionary Robotics

Iñaki Fernández Pérez Amine Boumaza François Charpillet

Inria, Villers-lès-Nancy, F-54600, France
CNRS, Loria, UMR 7503, Vandœuvre-lès-Nancy, F-54500, France
Université de Lorraine, Loria, UMR 7503, Vandœuvre-lès-Nancy, F-54500, France
firstname.lastname@loria.fr

Résumé

L'effet de la pression à la sélection sur l'évolution dans des algorithmes évolutionnaires (AEs) centralisés est relativement bien compris. La pression à la sélection pousse l'évolution vers des individus plus performants. Cependant, les AEs distribués en robotique évolutionnaire diffèrent du fait que la population est distribuée sur tous les agents et qu'il n'existe pas de vision globale de tous les individus.

Dans cet article, nous analysons l'influence de la pression à la sélection dans un tel cadre distribué. Nous proposons une version de mEDEA qui introduit une pression à la sélection, et nous évaluons son effet sur deux tâches multi-robot : navigation avec évitement d'obstacles et collecte d'objets. Nos expériences montrent que même des légères pressions à la sélection mènent à des bonnes performances, et que la performance augmente avec la pression à la sélection. Ceci s'oppose aux approches centralisées, où une plus faible pression à la sélection est en général préférable afin d'éviter de stagner dans des optima locaux.

Mots Clefs

Evolutionary Robotics, Artificial Neural Networks, Selection pressures, Multi-robot learning.

Abstract

The effect of selection pressure on evolution in centralized evolutionary algorithms (EA's) is relatively well understood. Selection pressure pushes evolution toward better performing individuals. However, distributed EA's in an Evolutionary Robotics (ER) context differ w.r.t. selection in that the population is distributed across the agents, and a global vision of all the individuals is not available.

In this paper, we analyze the influence of selection pressure in such a distributed context. We propose a version of mEDEA that adds a selection pressure, and evaluate its effect on two multi-robot tasks: navigation and obstacle-avoidance, and collective foraging. Experiments show that even small intensities of selection pressure lead to good performances, and that performance increases with selection pressure. This is opposed to the lower selection pressure that is usually preferred in centralized approaches to avoid stagnating in local optima.

1 Introduction

One of the goals of evolutionary robotics (ER) [Nolfi and Floreano, 2000] is to automatically build robotic agents' controllers using evolutionary algorithms (EA) [Eiben and Smith, 2003]. In ER contexts, EA's are usually seen as a tool to optimize the controller of one or more agents regarding an objective function that measures agent performance (fitness). Once the required behavior is learned and the controller optimized, the agents are deployed and their behavior exploited, while their controllers remain fixed, *i.e.* there is no further evolution. This is known as offline evolution.

On the other hand, in online evolution [Watson et al., 2002] behavior learning takes place at the same time as the execution of the task at hand. As such, optimization is a continuous process, *i.e.* agents are constantly exploring new behaviors and learning to react to new environmental conditions, which is usually referred to as adaptation.

In our work, we focus on on-line distributed evolution of agent controllers. Our research concerns learning agent behaviors in a distributed context where multiple agents adapt their controllers to environmental conditions independently. In this sense, this approach finds many ties with Artificial Life, where the objective is to design autonomous organisms that adapt to their environment. Agents can locally communicate with each other, and no agent has a view of the entire population. Online distributed evolution can be thought as distributing an EA on the team of agents. Standard evolutionary operators (mutation, crossover etc.) are implemented on the agents, and local communication allows for the spread of genomes in the team of agents.

In EA's, selection operators drive evolution toward better performing individuals by regulating the intensity of selection pressure to learn to solve the given task. Selection operators and their influence on evolutionary dynamics have been extensively studied in offline contexts [Eiben and Smith, 2003]. In this work, we analyze their impact in an online distributed setup, where evolutionary dynamics are different from the offline case: selection is local, it acts over partial populations, and fitness values on which selection is performed are not reliable, due to different evaluation conditions. As our experiments show, in this context a strong selection pressure produces the best

results. This is opposed to a lower selection pressure that is preferred in offline centralized contexts to maintain diversity in the population and avoid premature convergence. Our results suggest that, in distributed ER algorithms, diversity is implicitly maintained by the fact that there exist disjoint subpopulations across the agents of the team.

Online evolution of agent controllers has been addressed by several authors in different contexts: adaptation to varying conditions [Dinu et al., 2013], automatic parameter configuration [Eiben et al., 2010], light-following and navigation [Karafotias et al., 2011, Silva et al., 2014]. Some of these works are described in the next section. The authors use selection operators that induce different degrees of selection pressure to drive evolution. Here, we evaluate the influence of selection pressure on the performances obtained when learning two multi-agent tasks: navigation and foraging.

2 Selection in Distributed ER

A common characteristic of on-line distributed ER algorithms is that each agent has one controller at a time, that it executes (the active controller), and locally spreads altered copies of this controller to other agents. In this sense, agents have only a partial view of the population in the swarm (a local repository). Fitness assignment or evaluation of individual genomes is performed by the agents themselves and is thus noisy, as different agents evaluate their active controllers in different conditions. Selection takes place when the active controller is to be replaced by a new one from the repository.

PGTA (Probabilistic Gene Transfer Algorithm), introduced by [Watson et al., 2002], is usually cited as the first algorithm to evolve agent controllers in an online distributed manner. It evolves the weights of neural controllers, and agents locally exchange parts (genes) of their respective genomes when they cross each other. The rate at which an agent spreads its genes is proportional to its performance, and the rate at which an agent accepts received genes is inversely proportional to its performance. In this sense, selection pressure is induced in that fit agents transfer their genes to unfit ones.

In mEDEA (minimal Environment-driven Distributed Evolutionary Algorithm, [Bredeche and Montanier, 2010]), the authors study evolutionary adaptation with an implicit fitness, *i.e.* without a task-driven fitness function. Local selection is performed randomly by a given agent upon the genomes gathered during the execution of the agent's controller. As such, successful genomes in mEDEA are those that spread the most over the agents. The spread of a genome is maximized by increasing mating opportunities and minimizing the risk for the agent carrying it.

The authors show that agents learn to navigate and avoid obstacles, which allows them to better spread their genomes when local selection is random. This work shows that environmental selection pressure alone can maintain a certain level of adaptation in a team of robotic agents. A modified version of this algorithm is used in this work and is detailed in the next section.

[Noskov et al., 2013] proposed MONEE (Multi-Objective aNd open-Ended Evolution), an extension to mEDEA adding a task-driven pressure. The algorithm was designed for dealing with multiple objectives, and implements a mechanism (called market) for balancing the distribution of different tasks among the population of agents, so as not to disregard harder tasks w.r.t. simpler ones. In this sense, a selection pressure toward task diversity is induced. Their experiments show that MONEE is capable of improving mEDEA's performances in a collective foraging task, in which agents have to collect items of several kinds.

The authors show that the agents are able to adapt to the environment (as in mEDEA), and to forage different kinds of items, *i.e.* optimize the task-solving behavior. The algorithm uses an explicit fitness function in order to guide the search toward better performing solutions. In their paper, the agent's next controller is selected using rank-based selection from the agent's list of genomes. The authors argue that when a specific task is to be addressed, a task-driven selection pressure is necessary. This idea is discussed in the remainder of this paper.

In other related algorithms, selection is applied differently. For instance, odNEAT [Silva et al., 2014] (a distributed version of NEAT [Stanley and Miikkulainen, 2002] that evolves both the topology and weights of neural networks) maintains local populations structured in niches w.r.t. topology similarity. The controllers share the fitness of their respective niches, which consists in the average fitness divided by the number of individuals in the niche. By doing this, a diversity of topologies is maintained, since a smaller niche whose individuals have a lower fitness have a chance to survive and potentially improve, given that its shared fitness is divided by a lower factor.

In EDEA (Embodied Distributed Evolutionary Algorithm) [Karafotias et al., 2011], selection pressure is applied by selecting and recombining a received genome, x', with the current active one, x, with a probability $\frac{f(x')}{s_c \times f(x)}$, where s_c is a factor regulating selection pressure. The higher s_c is, the lower the probability of selecting a received genome.

In these works, authors used standard selection operators from evolutionary computation in distributed ER contexts. Nevertheless, it is unclear if similar evolutionary dynamics can be expected as when they are used in an offline centralized ER setup, given that online distributed contexts have different properties: selection is performed locally at the agent level, and over the genomes of the other agents it had the opportunity to meet. Furthermore, as it is common in many ER setups, in online distributed evolution, fitness evaluation is intrinsically noisy, since agents evaluate their controllers in different conditions, and this may strongly influence their performance and resulting behaviors. In this sense, a question we study here is: does it still make sense to use selection in distributed contexts? And if yes, what intensity of selection pressure is adequate?

In this paper, we study the influence of four different se-

lection methods inducing different intensities of selection pressure. We apply these methods in a version of mEDEA adding task-driven selection, and evaluate their impact on two different multi-robot tasks.

3 Algorithm and selection operators

In this section, we describe the algorithm used in our experiments (Alg. 1), a variant of mEDEA. The algorithm is independently executed by all the agents. Each agent has a single active controller, which is initialized randomly.

The main difference with the original mEDEA is that, in our variant an agent alternates between two phases, as proposed in [Noskov et al., 2013]: an *evaluation phase* lasting T_e, in which the agent runs, evaluates and locally broadcasts its controller to listening agents, and a *listening phase* lasting T_l, in which the agent stops and listens to incoming genomes sent by nearby agents. For different agents, phases are desynchronized, so agents in the evaluation phase are able to spread their genomes to other agents that are in the listening phase.

The agent's controller is executed and evaluated during the evaluation phase. At each time-step the agent reads its sensors, passes them as input to its controller and computes the motors' outputs. It also updates the fitness value of the controller depending on the result of its actions, and broadcasts the genome corresponding to its controller and its current fitness value to listening robots in the vicinity.

At the end of the T_e evaluation steps, the listening phase begins. At this point, the agent stops for T_l time-steps and listens for genomes from nearby passing agents (agents that are evaluating their controllers). Since the genomes are broadcast along with their respective fitness values, at the end of this phase, an agent has a local list of genomes and fitnesses, or local population. In our variant, unlike original mEDEA, an agent's current genome is added to its local population, to ensure that all agents always have at least one genome in their respective populations. This can occur if an agent is isolated during its listening phase and thus does not receive any genome. In the original mEDEA, agents stay inactive until they receive a new genome.

Once the listening phase is finished, the agent loads a new controller for the next evaluation phase. This is done by selecting a genome from the local population based on its fitness, and using one of the selection operators presented below. The selected genome is mutated and becomes the agent's active controller. In our experiments, mutation is performed by adding a normal random variable with mean 0 and variance σ^2 to each connection weight.

At this point, before the new controller's evaluation phase begins, the local population is emptied. As such, selection is performed on a list of genomes gathered by the agent during the previous listening phase. One complete iteration of the algorithm (evaluation and listening phase) is referred to as one generation.

Here, we study four selection operators, each inducing a different intensity of task-driven selection pressure, from the strongest to the lowest: *Best*, *Rank-based*, *Binary Tournament* and *Random* selection. *Best* always selects the genome with the highest fitness, while *Rank-based* assigns probabilities of selection proportional to the rank of each genome in the population once sorted w.r.t. fitness. *Binary Tournament* selection consists in drawing two genomes at random and selecting the best between them. Finally, *Random* selection picks a random genome from the local population, thus completely disregarding fitness values, as in mEDEA. The choice of these selection methods aims at giving a large span of intensities of selection pressure.

Algorithm 1 mEDEA variant

```
g_a := random()
while true do
  l := ∅
  // Evaluation phase
  for t = 1 to T_e do
    exec(g_a)
    broadcast(g_a)
  end for
  // Listening phase
  for t = 1 to T_l do
    l := l ⋃ listen()
  end for
  l := l ⋃ {g_a}
  selected := select(l)
  g_a := mutate(selected)
end while
```

4 Experiments on selection pressure

Here, we compare the four presented selection methods on a set of experiments in simulation for two tasks, navigation with obstacle avoidance and collective foraging, which are two well-studied benchmark tasks in multi-robot setups. Our experiments were performed on the RoboRobo simulator [Bredeche et al., 2013].

4.1 Robots and tasks description

In all experiments, a team of 50 robotic agents is deployed in a square bounded environment containing static obstacles. Other agents are perceived as mobile obstacles.

All the agents in the team have the same physical properties, sensors and motors, and the only difference between them lays in the weights of their neural controllers. Agents have 8 obstacle proximity sensors, evenly spaced around the agent. 8 item sensors are added for the foraging task. An item sensor measures the distance to the closest item in the direction and range of the sensor.

The agents' neural controllers are recurrent neural networks, where the inputs of the network are the activation values of all sensors, and the 2 outputs correspond to the translational and rotational velocities of the agent. The activation function of the output neurons is a hyperbolic tangent, taking values in $[-1, +1]$. Two bias connections (one

Table 1: Experimental settings.

Number of food items	150
Number of runs	30
Evolution length	~ 250 generations
T_e	$2000 - rand(0, 500)$ sim. steps
T_l	200 sim. steps
Mutation step-size	$\sigma = 0.5$

for each output neuron) and 4 recurrent connections (previous speed and rotation to both outputs) are added. In total, 22 connection weights need to be optimized for the navigation task, and 38 for the foraging task. The genome of the controller is the vector of these weights. Table 1 summarizes the parameters we used in the experiments.

The navigation task consists in learning to move rapidly and minimizing turns, while avoiding static and mobile obstacles. Foraging requires agents to gather food items in the environment. Items are gathered when agents pass over them, and when an item is collected, it is replaced by another one at a random position.

The fitness function for navigation is defined after the one introduced in [Nolfi and Floreano, 2000]. An agent r computes its fitness at generation g as:

$$f_r^g = \sum_{t=1}^{T_e} v_t(t) \cdot (1 - |v_r(t)|) \cdot min(a_s(t)) \qquad (1)$$

where $v_t(t)$, $v_r(t)$ and $a_s(t)$ are respectively the translational and rotational velocities at time t, and the vector of activations of the obstacle sensors of the agent at time-step t of its evaluation phase. As for the foraging task, the fitness is computed as the number of items collected by the agent during its evaluation phase.

Since in our work we want to study the performance of the entire team of agents, we define swarm fitness as the sum of the fitness of all agents at the end of each generation:

$$F_s(g) = \sum_{r \in swarm} f_r^g \qquad (2)$$

4.2 Performance measures in online ER

Online evolving agents learn in an open-ended manner at the same time as they are performing the actual task. Consequently they are always exploring new solutions, and the best fitness ever reached may not be a reliable estimator of the quality of the algorithm, since a high best fitness only reflects a good performance at one point of evolution. Additionally, online fitness evaluation is noisy in essence, given the different conditions in which agents evaluate their controllers. Taking in account these issues, in [Fernández Pérez et al., 2014] we introduced four measures, that we use to analyze the influence of the intensity of selection pressure in our experiments. These measures integrate swarm fitness information spanning over several generations, and are computed after evolution has finished in order to compare the selection operators. A pictorial description of these four measures is shown in Figure 1.

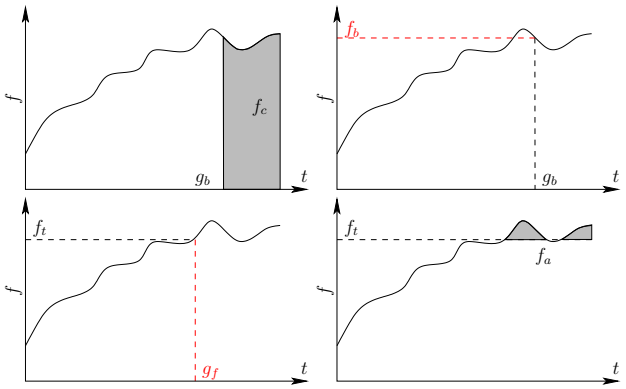


Figure 1: From left to right, top to bottom: f_c, f_b, g_f, f_a.

The aforementioned measures are the following. The *average accumulated swarm fitness* (f_c) is the average swarm fitness in the last generations (here, the last 8% generations). The *fixed budget swarm fitness* (f_b) is the swarm fitness reached at a certain generation (computational budget, here 92% of the evolution, *i.e.* the first generation considered in f_c). The *time to reach target* (g_f) is the first generation at which a predefined target fitness is reached, or the last generation if this level is never reached. Here, we fixed the target at 80% of the maximum fitness reached over all runs and all selection operators. Finally, the *accumulated fitness above target* (f_a) is the sum of all swarm fitness values above the same target value as for g_f.

These measures need to be considered in combination when comparing two experiments. For instance, f_c and f_b provide information of the level and stability of the performance reached by the agent team at the end of evolution. If they are close, the performance is stable. Also, g_f and f_a combined reflect how rapidly a target fitness value is attained, and by how much that level is exceeded.

4.3 Results and discussion

We launched 30 independent runs and measured F_s at each generation for each variant of the experiment (each selection operator in both tasks). The median F_s per generation over all runs is presented in Figures 2 (navigation) and 3 (foraging). The four performance measures are shown in Figure 4 (navigation), and in Figure 5 (foraging). For both tasks, we performed 99% confidence Mann-Whitney tests on the four measures, for all pairwise combinations between the four selection operators.

When observing Fig. 2 and Fig. 3, we notice that, in both tasks, a high fitness level is rapidly reached whenever there is a task-driven selection pressure, *i.e.* with *Best*, *Rank-based*, or *Binary tournament* selection. Furthermore, the algorithm reaches similar levels of swarm fitness (median values). An exception can be noted for *Best* selection in the foraging task, which outperforms all other selection operators. However, if no selection pressure is induced at

the agent level (*i.e. Random* selection), learning is much slower, and thus reaches lower levels in the allotted time.

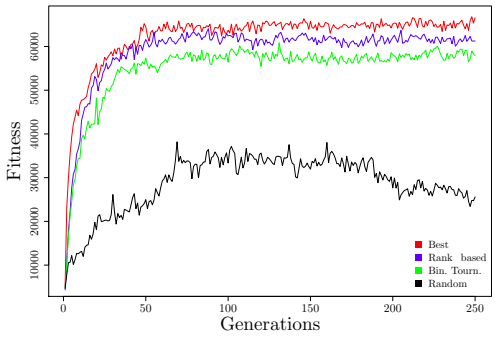


Figure 2: Median swarm fitness for the navigation task.

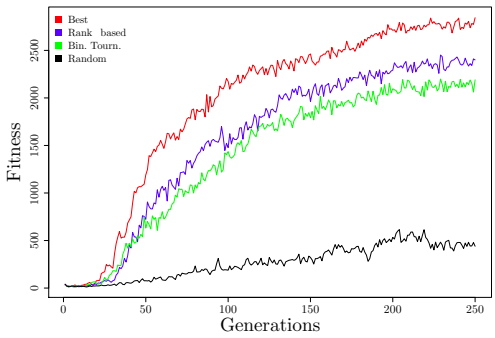


Figure 3: Same as in Figure 2 for the foraging task.

Even if the results achieved by *Random* are worse, agents still improve their performances in both tasks. This can be observed in the increasing trend in Fig. 2 and Fig. 3. This is expected for the navigation task, since environmental pressure leads to behaviors that increase mating opportunities by exploring the environment, thus improving the swarm fitness, as in mEDEA [Bredeche and Montanier, 2010].

As for the foraging task, when *Random* selection is applied the improvement is slower but still present. The explanation is that collecting items can be a byproduct of exploring the environment. Items are gathered by chance in this case, when agents move while trying to mate. Upon inspection of the evolved behaviors in the simulator, we noticed that, when selection pressure is present, agents move toward the food items, which means that evolution drove the agents to exploit the item sensors. However, without selection pressure (*Random*), there can not be a similar drive, which we confirmed by inspecting the simulator: agents were not attracted by food items for *Random* selection.

When analyzing the comparison measures we introduced above, the same trends are observed. Figure 4 (respectively Figure 5) presents the box and whiskers plots of the four measures for each selection method over the 30 runs for the navigation task (respectively the foraging task).

For the navigation task, pairwise comparisons of the four measures result in a significant difference between all selection operators, except for the time to reach target (g_f)

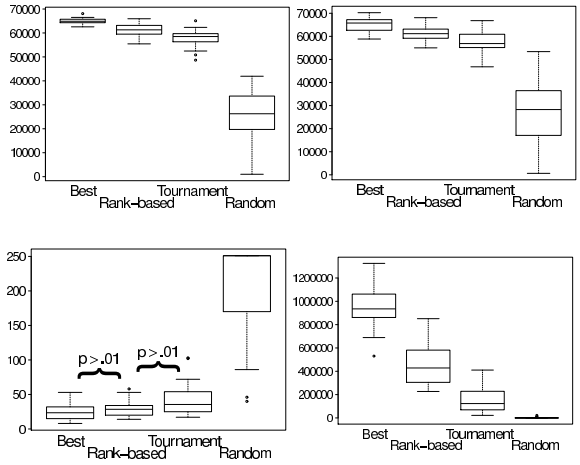


Figure 4: Performance measures on the navigation task. From left to right and top to bottom: f_c, f_b, g_f and f_a. The label $p > 0.01$ indicates no statistical difference.

between *Best* and *Rank-based* ($p{-}value = 0.07$) and between *Rank-based* and *Binary tournament* ($p{-}value = 0.012$). We observe that *Best* reaches a higher swarm fitness than the other selection operators, and this level is maintained at the end of evolution, as indicated by f_c and f_b (upper left and right in the figure). The target fitness level is rapidly attained for the three operators with selection pressure, and there is not significant difference between *Best* and *Rank-based*, nor between *Rank-based* and *Binary Tournament* regarding g_f (lower left). Moreover, in the case of *Best*, the target level is not only reached but surpassed during the entire evolution, yielding much higher values of f_a than the rest of selection operators (lower right). However, this is not the case for *Random* selection that leads to lower f_b and f_c (top), and does not reach the target fitness level on more than half of the runs (bottom).

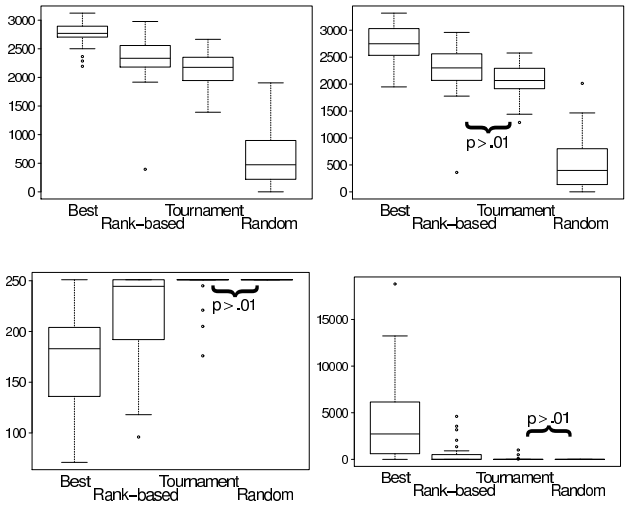


Figure 5: Same as in Figure 4 for the foraging task.

Concerning foraging, differences are significant for all pairwise comparisons, except between *Binary Tournament* and *Random* for the time to reach target, g_f, and the accumulated fitness above target, f_a ($p{-}value = 0.042$ in both cases). The reason behind this is that very few runs reached the required level, and thus g_f is the last generation and f_a is almost zero, for both selection operators. There is also no statistical difference between *Rank-based* and *Binary Tournament* on the fixed budget swarm fitness, f_b ($p{-}value = 0.011$). This means that *Binary Tournament* reaches a fitness at the given budget that is comparable to the one of *Rank-based*, but it is not able to maintain the level so effectively, since the difference on f_c between these two operators is significant.

On the foraging task, *Best* also leads to better results: a high swarm fitness is reached and maintained (f_b and f_c, upper left and right). It surpasses the required fitness level in almost all runs much faster and to a larger extent than *Rank-based*, that also manages to reach the target level for most runs (g_f, lower left), although by a lesser extent (f_a, lower right). The picture is different regarding *Tournament* and *Random*, which do not achieve the target fitness level for most runs (lower left and right).

To summarize, we can confirm that all task-driven selection pressures lead to much better results on both tasks compared to *Random* selection. Consequently, we may conclude that selection pressure has a positive impact on performances when solving a given task, *i.e.* when the goal is not only to achieve environmental adaptation as it was the original motivation of mEDEA. Moreover, statistical tests show a direct correlation between selection pressure and the performances achieved on the two considered tasks. In other words, the stronger the selection pressure is, the better the performances reached by the team of agents.

It has been argued that in general, elitist approaches are not desirable in traditional EA's, and this also applies to traditional ER. The reason behind this is that elitist strategies can result in a premature convergence at local optima. This has been extensively studied, especially in non-convex optimization, where it is preferable to explicitly force a certain level of diversity in the population to allow evolution to escape local optima and deal with the exploration versus exploitation dilemma. As our experiments show, this requirement is perhaps not as strong in distributed ER. Since selection is performed at the agent level and over a fraction of the population, we might argue that these algorithms already maintain a certain diversity, given that subpopulations are distributed on the different agents. Investigating possible ties with other approaches in which separated subpopulations are evolved, *e.g.* spatially structured EA's [Tomassini, 2005] and island models [Alba and Tomassini, 2002], could provide more information on the dynamics of distributed evolution.

5 Conclusions and future work

This paper has studied the influence of different degrees of selection pressures in an online distributed context for agent behavior evolution. In distributed ER, the impact of selection pressures on evolution is unclear, given that selection is applied over partial populations and fitness values are noisy. Four selection operators inducing different degrees of selection pressure were compared on two tasks: navigation with obstacle avoidance and collective foraging. In our experiments, we show that even a small degree of selection pressure largely improves performances, and that the intensity of the selection operator positively correlates with the performances of the team of agents.
Foraging and navigation are arguably relatively simple tasks, and we deem interesting to study selection pressures on more complex and challenging ones, involving deceptive fitness landscapes. This could further clarify the impact of selection on evolution dynamics in distributed approaches to the evolution of agent behavior.

References

[Alba and Tomassini, 2002] Alba, E. and Tomassini, M. (2002). Parallelism and evolutionary algorithms. *Evolutionary Computation, IEEE Transactions on*, 6(5):443–462.

[Bredeche and Montanier, 2010] Bredeche, N. and Montanier, J.-M. (2010). Environment-driven Embodied Evolution in a Population of Autonomous Agents. In *PPSN 2010*, pages 290–299, Krakow, Poland.

[Bredeche et al., 2013] Bredeche, N., Montanier, J.-M., Weel, B., and Haasdijk, E. (2013). Roborobo! a fast robot simulator for swarm and collective robotics. *CoRR*, abs/1304.2888.

[Dinu et al., 2013] Dinu, C. M., Dimitrov, P., Weel, B., and Eiben, A. (2013). Self-adapting fitness evaluation times for on-line evolution of simulated robots. In *Proc. of GECCO'13*, pages 191–198. ACM.

[Eiben et al., 2010] Eiben, A., Karafotias, G., and Haasdijk, E. (2010). Self-adaptive mutation in on-line, on-board evolutionary robotics. In *Fourth IEEE Int. Conf. on Self-Adaptive and Self-Organizing Systems Workshop (SASOW), 2010*, pages 147–152. IEEE.

[Eiben and Smith, 2003] Eiben, A. E. and Smith, J. E. (2003). *Introduction to Evolutionary Computing*. Springer.

[Fernández Pérez et al., 2014] Fernández Pérez, I., Boumaza, A., and Charpillet, F. (2014). Comparison of Selection Methods in On-line Distributed Evolutionary Robotics. In *ALIFE 14*, New York, US.

[Karafotias et al., 2011] Karafotias, G., Haasdijk, E., and Eiben, A. E. (2011). An algorithm for distributed on-line, on-board evolutionary robotics. In *Proc. of GECCO '11*, pages 171–178, New York. ACM.

[Nolfi and Floreano, 2000] Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics*. MIT Press.

[Noskov et al., 2013] Noskov, N., Haasdijk, E., Weel, B., and Eiben, A. (2013). Monee: Using parental investment to combine open-ended and task-driven evolution. In Esparcia-Alcázar, editor, *App. of Evol. Comput.*, volume 7835 of *LNCS*. Springer Berlin.

[Silva et al., 2014] Silva, F., Urbano, P., Correia, L., and Christensen, A. L. (2014). odneat: An algorithm for decentralised online evolution of robotic controllers. *Evol. Comput.* MIT Press. Available online.

[Stanley and Miikkulainen, 2002] Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evol. Comput.*, 10(2):99–127.

[Tomassini, 2005] Tomassini, M. (2005). *Spatially structured evolutionary algorithms*. Springer Berlin.

[Watson et al., 2002] Watson, R. A., Ficici, S. G., and Pollack, J. B. (2002). Embodied evolution: Distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Syst.* Elsevier.

Gérer l'incertitude lors de l'extraction de relations et lors de l'inférence de nouvelles connaissances

Pierre-Antoine Jean[1] Sébastien Harispe[1] Patrice Bellot[2] Sylvie Ranwez[1] Jacky Montmain[1]

[1] Laboratoire de Génie Informatique et d'Ingénierie de Production (LGI2P) - Nîmes
[2] Laboratoire des Sciences de l'Information et des Systèmes (LSIS) - Marseille

LGI2P, École des mines d'Alès, 69 rue Georges Besse
F-30035 Nîmes cedex 1
{prenom.nom}@mines-ales.fr
patrice.bellot@lsis.org

Résumé

Malgré leur volume important et leur accessibilité, de nombreuses données numériques ne peuvent être correctement exploitées car elles sont contenues dans des textes sous des formes peu ou pas structurées. L'extraction de relations est un processus qui rassemble des techniques pour extraire des entités et des relations à partir de textes, nous donnant la possibilité d'enrichir des bases de connaissances de façon automatique. Cependant le langage naturel est de façon intrinsèque porteur d'ambiguïté, ce qui constitue un premier niveau d'incertitude auquel on peut rajouter l'imprécision due aux formulations telles que "je crois que", "il semble que", etc. La base de connaissances doit donc tenir compte de cette incertitude par exemple en associant à chaque nouvelle connaissance extraite un score de confiance dépendant du degré de certitude. Cet article est une ***communication de synthèse*** *qui détaille les différentes problématiques liées à l'incertitude et à l'imprécision au cours de la chaîne de traitement allant de l'extraction d'information dans les textes à l'inférence de connaissances. Il y sera notamment question de stratégie d'agrégation des différentes sources d'incertitude et d'imprécision et de leur prise en compte dans les traitements ultérieurs (par exemple la recherche d'information ou l'aide à la décision).*

Mots Clef

TALN, extraction d'information, incertitude, inférence de règles

Abstract

Among the increasing volume of electronic resources available, non-structured texts expressed through natural language are difficult to process automatically. In this context, relation extraction techniques propose to combine various approaches to extract entities and their relations from texts, e.g. to automatically enrich a knowledge base. Nevertheless, the natural language is per se ambiguous, which makes extraction results uncertain. It can also be used to express imprecise or uncertain statements, "It seems to me", "I believe", etc. Therefore, any knowledge base enriched through text analyses must consider these uncertainties, for instance by combining a confidence score to each knowledge extraction according to its associated level of uncertainty. This information will be of major importance to infer additional knowledge from these extractions. However, how to characterize, capture and integrate the uncertainty and imprecision of natural language ? In addition, how to take into account this uncertainty to infer new knowledge ? This paper is a synthesis communication related to the consideration of uncertainty and imprecision in the context of Information Extraction from texts and knowledge inference from these extractions. We propose in particular to define the terminology, to characterize the several sources of uncertainty and to discuss strategies that can be used to capture and consider the uncertainty in knowledge extraction and knowledge inference treatments.

Keywords

NLP, information extraction, uncertainty, mining rules

1 Introduction

De nos jours, les bases de connaissances telles que DBPedia [1], YAGO [2] ou Freebase [3] contiennent des millions d'entités et des centaines de millions de faits [8]. Malgré tout, ces bases de connaissances restent incomplètes. De nombreuses informations complémentaires sont contenues dans des textes non structurés qui peuvent être utilisés pour enrichir ces bases et ainsi améliorer les traitements qui en découlent : recherche d'information, *question answering* ou aide à la décision. Trois principales approches s'offrent à

1. www.dbpedia.org
2. www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago
3. www.freebase.com

nous pour compléter ces bases de connaissances : *(i)* de manière manuelle, coûteuse et chronophage, *(ii)* de manière automatique grâce à **l'extraction d'informations** dans les textes, qui est plus incertaine mais bien plus rapide, ou *(iii)* par des méthodes **d'inférence de connaissances** à partir de faits dans la base de connaissances (*cf.* Figure 1). Dans notre étude, nous nous intéressons à ces deux dernières approches et plus particulièrement à la prise en compte de l'incertitude qui leur est inhérente.
Dans ce qui suit, nous positionnons notre étude par rapport aux approches proposées dans la littérature. Ensuite, nous discuterons plus particulièrement des méthodes d'extraction à partir de textes, puis de l'inférence de nouvelles connaissances.

2 Positionnement

2.1 L'extraction d'information

L'extraction d'information est un vaste domaine de recherche initié en 1992 par la première conférence MUC (*Message Understanding Conference*). Elle se définit comme l'extraction d'informations structurées à partir de textes écrits en langage naturel. Contrairement à la recherche d'information, elle a pour objectif de retourner directement au lecteur l'information souhaitée et non des pointeurs vers des documents. Dans le cadre de notre étude, nous nous focaliserons tout d'abord sur l'extraction de triplets (Sujet - Prédicat - Objet) ; on parle alors **d'extraction de relations binaires**, et aux marqueurs d'incertitude qui peuvent y être associés (contexte de la relation). On distingue traditionnellement trois principales tâches dans l'extraction d'informations : la **Reconnaissance d'Entités Nommées** (REN) dont le but est d'identifier et de désambiguïser des entités dans les textes, **l'extraction de relations** permettant d'extraire des connections sémantiques entre les entités identifiées et **l'extraction d'événements** qui consiste à remplir de manière automatique une structure informationnelle (un formulaire) associant différentes informations à un événement donné (par exemple pour un événement sportif, la structure informationnelle contiendrait : le nom des deux équipes, le score, le lieu, etc.). Ces trois tâches sont fortement dépendantes les unes des autres, l'extraction de relations nécessite une REN, notamment si elle est destinée à l'enrichissement d'une base de connaissances et l'extraction d'événements peut être perçue comme une extraction de relations n-aire. De nombreux travaux de la littérature proposent des approches différentes pour l'extraction d'informations. Ces approches diffèrent selon la disponibilité d'exemples annotés induisant l'utilisation des méthodes supervisées ou dans le cas contraire des méthodes semi ou non supervisées et selon les textes analysés (c'est le cas des textes biomédicaux, par exemple, où le vocabulaire employé peut entraîner certaines difficultés, notamment pour la REN [12] - conventions de nommage non respectées, flux constant de nouvelles entités). Ces approches se distinguent aussi par les techniques employées qui peuvent être linguistiques, basées sur des patrons syntaxiques et lexicaux [5, 6], mais aussi statistiques, comme les modèles graphiques dirigés (chaînes de Markov cachées) ou non dirigés (champs markoviens aléatoires (CRFs)) [6, 11] ou encore basées sur des systèmes hybrides couplant les avantages de plusieurs techniques [15].

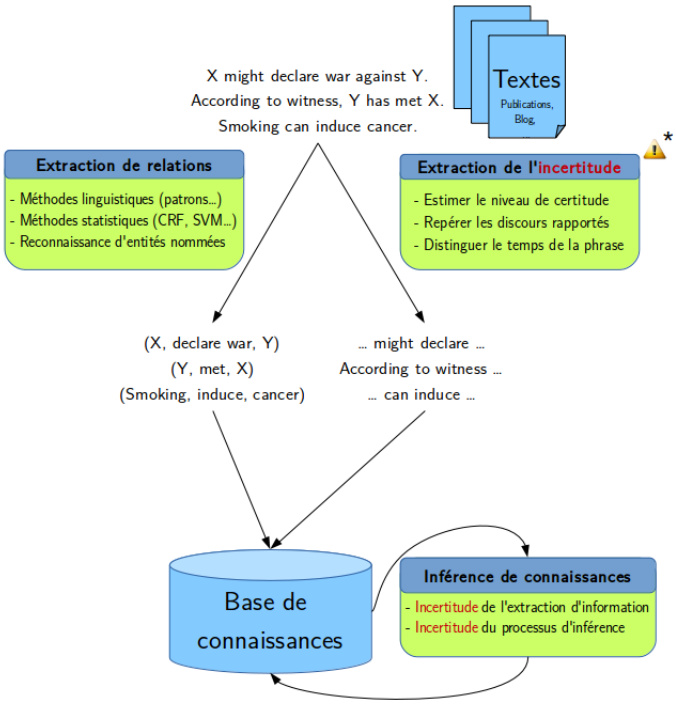


FIGURE 1 – Structure globale de l'approche envisagée. L'extraction d'informations permet de coupler les informations contenues dans les textes (relations, incertitude) avec une base de connaissances qui, elle-même, permet d'inférer de nouvelles connaissances. Le symbole (*) précise que nous nous intéressons à l'incertitude inhérente au contenu des textes. La source et le croisement contradictoire de différentes relations ne sont pas considérés.

2.2 L'inférence de connaissances

Les méthodes d'extraction d'informations permettent d'enrichir de manière automatique une base de connaissances. Cette dernière peut servir ensuite de support à différents traitements (*e.g* aide à la décision) qui seront d'autant plus performants que la base sera complète. Pour la compléter il est possible, à partir des connaissances présentes dans la base et de certaines règles d'inférence, d'inférer de nouvelles connaissances. Ce processus éprouvé depuis longtemps en *Data Mining* peut avoir plusieurs objectifs [9] : trouver de nouvelles connaissances et de nouveaux faits dans la base de connaissances, identifier d'éventuelles erreurs (d'insertion ou d'inférence) et enfin permettre d'avoir une meilleure compréhension des données par l'intermédiaire de règles décrivant des phénomènes généraux. De

plus, [9] précise que les techniques couramment utilisées en *Data Mining* (par exemple l'apprentissage par Programmation Logique Inductive -PLI- qui utilise des exemples positifs et négatifs pour trouver des hypothèses couvrant tous les exemples positifs et aucun exemple négatif) ne sont pas adaptées aux exigences des bases de connaissances de type ontologie dû aux contraintes qui y sont liées : important volume de données (passage à l'échelle) et hypothèse à domaine ouvert. Ce type d'hypothèse signifie qu'une donnée absente ne peut être utilisée comme un contre-exemple. Pour répondre à ce problème, Muggleton [13] a développé un score d'évaluation d'apprentissage basé uniquement sur les exemples positifs pour la PLI, l'approche génère des contres exemples de manière aléatoire. L'article [9] présente une approche (AMIE) qui utilise une autre stratégie pour générer des contres exemples [8] : *the Partial Completeness Assumption* (PCA) qui suppose que si une base de connaissances connaît une certaine relation pour une entité elle connaît alors toutes ses relations.

2.3 Gestion de l'incertitude

Un des objectifs de notre étude réside dans la prise en compte et la gestion de l'incertitude inhérente au langage naturel ou liée à l'inférence de règles et de nouvelles connaissances. L'article [5] décompose l'incertitude associée aux textes en deux sous-parties. Elle distingue : l'ambiguïté et l'imprécision. L'**ambiguïté linguistique** correspond à plusieurs associations possibles entre des symboles (des mots ou des suites de mots) et leurs significations. La polysémie [4] et l'homonymie [5] sont les principales sources d'ambiguïté. La différence entre ces deux notions est que deux mots homonymes partagent une même forme orale et/ou écrite mais n'ont pas la même étymologie. Par exemple, la phrase : « J'aime le rouge. » est ambiguë car « rouge » est un polysème pouvant signifier ici, le vin ou la couleur. L'ambiguïté est principalement traitée dans les méthodes de REN. L'article [3] présente une méthode se basant sur l'environnement lexical d'un terme pour le désambiguïser. La méthodologie qui y est décrite s'appuie sur un graphe de coocurrences des termes apparaissant avec le mot ambigu. **L'imprécision** (le flou) représente un savoir incomplet qui est exprimé sur des faits ou des événements. Par exemple, la phrase « Plusieurs véhicules se dirigent vers l'est » est imprécise de par l'utilisation de l'adjectif indéfini « Plusieurs ». L'article [14] propose une catégorisation des marqueurs de certitude présents dans les phrases. Ces catégories tiennent compte du degré de certitude exprimé (fort ou faible), du temps employé, de la perspective (discours rapporté ou point de vue de l'écrivain) ou de la nature de l'information délivrée. Ces marqueurs peuvent inclure par exemple les adverbes épistémiques « selon moi, à mes yeux, à mon avis » permettant au locuteur de mentionner sa subjectivité et ainsi de relativiser son propos ou bien l'emploi du conditionnel donnant le point de vue du locuteur sur l'énoncé [4].

L'incertitude peut être appréhendée sur d'autres aspects que le texte, notamment au niveau de la source de l'information. Par exemple, si un pneumologue dit « Le tabac peut induire le cancer » et une personne *lambda* dit « Le tabac induit le cancer », malgré le marqueur d'incertitude présent dans la phrase du pneumologue, la source est plus fiable et la phrase plus exacte. L'incertitude peut être également observée lors du croisement de plusieurs documents affirmant une information contradictoire. Cependant pour cette étude, seule la première source d'incertitude liée aux textes est traitée (en ce qui concerne l'incertitude inhérente aux textes).

Cette étude se concentrera sur deux éléments centraux que sont l'**information**, qui est extraite à partir de textes sous la forme de triplets (Sujet/EN1, Prédicat/R, Objet/EN2) et les **connaissances** qui peuvent être inférées à partir de ces informations et structurées dans une base de connaissances. De plus, la gestion de l'incertitude, inhérente aux textes et au processus d'extraction d'information et liée aux approches d'inférence de connaissances et de règles, sera un élément central de notre étude.

3 Exploitation des textes

3.1 L'extraction de relations

La première partie de l'étude se focalisera sur l'extraction de relations entre deux entités nommées dans une phrase. L'objectif est de récupérer un triplet désambiguïsé et de relever les marqueurs d'incertitude associés. L'article [16] caractérise la structure d'une relation en trois principales parties construites autour des entités : la partie *Cmid* qui porte généralement la relation et les parties *Cpre* et *Cpost* qui apportent généralement des précisions sur le contexte (*cf.* Figure 2). L'article [6] précise que sur un corpus de 300 phrases aléatoires du Web, 96% des relations binaires possèdent ce format et que 4% des relations ne le respectent pas (par exemple « ***Discovered by*** *Y,X* ... » ou bien « ... *the Y that X* ***discovered*** » avec X et Y représentant les deux entités). L'extraction de relations peut être à domaine ouvert [7] ou se focaliser sur des relations précises, par exemple la catégorisation de relations de régulation entre deux composants cellulaires dans des textes biomédicaux [2].

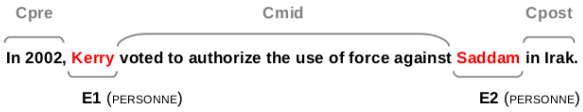


FIGURE 2 – Exemple de la structure type d'une relation extraite à partir de texte [16].

L'article [16] propose trois traitements afin d'extraire des

4. La polysémie est la caractéristique d'un mot ou d'une expression qui a plusieurs sens (au moins deux) ou significations différentes. Par exemple « Opéra » peut signifier la pâtisserie, le lieu ou l'art.

5. L'homonymie est la caractéristique de plusieurs mots partageant une même forme orale et/ou écrite mais qui ont des sens différents, par exemple « Une livre de pain qu'il livre avec un livre de recettes ».

relations (EN1, R, EN2) entre des entités pré-définies (PERSONNE, LIEU, ORGANISATION). Le premier traitement est une analyse linguistique. Cette analyse inclut une homogénéisation du contenu textuel grâce à une lemmatisation, une désambiguïsation morpho-syntaxique et une REN. Le second traitement correspond à l'extraction des relations candidates en filtrant les phrases contenant au moins un verbe entre les deux entités. Enfin, le troisième traitement permet de filtrer les relations obtenues probablement fausses (discours rapporté, relations trop complexes) à l'aide de champs conditionnels aléatoires (CRFs) linéaires entraînés sur 1000 exemples de relations. Leur système d'extraction de relations, qui est basé sur une stratégie de récupération des entités en premier, obtient une précision de 76.2% et un rappel de 78.2%, soit une F-mesure de 77,19%, sur les relations extraites à partir d'une sous-partie du corpus AQUAINT-2 contenant 18 mois d'articles du quotidien *The New York Times*.

L'article [1] propose une méthodologie pour extraire les relations entre les types sémantiques d'entités médicales (par exemple *Diazoxide - typeOf - treatment*) en utilisant des patrons linguistiques et une base de connaissances. Leur méthodologie débute par une REN spécialisée dans le domaine médical à l'aide de MetaMap [6], leur permettant également de récupérer les types sémantiques associés à chaque entité en utilisant UMLS Metathesaurus [7]. La méthode récupère par la suite toutes les relations existantes entre deux types sémantiques dans *UMLS Semantic Network*. Puis, pour chaque type de relation, un patron linguistique est construit et utilisé pour la reconnaissance de nouvelles relations. La construction d'un patron pour une relation donnée débute par la recherche de l'ensemble des paires de termes reliées par cette relation dans UMLS. Par la suite, une collection de textes contenant ces paires est récupérée grâce au système de requêtes avancées de PubMed. Chaque requête est affinée afin d'approximer les textes qui vont potentiellement contenir la relation souhaitée à partir d'une entité donnée (par exemple *Rhinitis, Vasomotor/TH* est une requête décrivant une relation de traitement (/TH) entre un traitement non spécifié et une *Rhinitis*). Leur méthode, basée elle aussi sur l'extraction des entités, obtient un score de précision de 75,72% et un score de rappel de 60,46%, soit une F-mesure de 67,23%.

Contrairement aux deux précédents systèmes présentés, ReVerb, introduit dans [7], débute par la récupération de signature de relations. En effet, l'approche utilise un ensemble de contraintes syntaxiques, sous la forme de patrons basés sur les étiquettes morpho-syntaxiques (*part-of-speech* (POS)) (*cf.* figure 3) et de contraintes lexicales permettant d'éviter l'extraction de relations trop spécifiques. Les patrons lexicaux sont basés sur la construction d'un large dictionnaire d'entités récupérées sur 500 millions de relations à partir du Web ; ainsi le nombre d'apparitions d'une paire d'entités doit être supérieur à une valeur seuil pour qu'elle soit considérée. Cette première phase permet de récupérer 85% des relations verbales binaires. Une fois ces relations extraites, la méthode détermine les limites des entités contenues dans la phrase en utilisant trois classifieurs spécifiquement entraînés pour repérer les limites gauches et droites des différentes entités [6]. Les auteurs utilisent REPTree [8] de Weka et un CRF et obtiennent une précision d'environ 60% et un rappel de 70%, soit une F-mesure de 64,6%.

V | VP | VW*P
V = *verb ? adv ?*
W = (*noun* | adj | adv | pron | det)
P = (prep | *particle* | *inf. marker*)

FIGURE 3 – Expressions rationnelles basées sur le POS des relations verbales. Ces expressions extraient soit un verbe simple (*invented*), soit un verbe suivi par une préposition (*located in*) ou soit un verbe suivi par un syntagme nominal et terminant par une préposition (*has atomic weight of*) [6].

Nous pouvons observer que les trois méthodes présentées considèrent uniquement l'extraction de relations sans tenir compte des marqueurs d'incertitudes qui peuvent être présents dans une phrase. Dans ces applications, la phrase « *According to witness, Y has met X* » est réduite à sa seconde partie [10]. La perte de l'expression de l'incertitude modifie la compréhension et la fiabilité de l'information délivrée par la phrase. La sous-section suivante présente des approches abordant le problème de l'extraction de marqueurs d'incertitude.

3.2 Évaluation et extraction de l'incertitude

Notre objectif initial est d'inférer de la connaissance en prenant en compte les différents niveaux d'incertitude : au niveau des textes et des connaissances inférées. En effet, si nous ajoutons un fait dans une base de connaissances sans tenir compte de l'incertitude, le résultat n'aura pas la même signification que l'information originelle exprimée dans le texte. La prise en compte de l'incertitude au niveau des textes passe par la reconnaissance de différents éléments phrastiques ou expressions qui nuancent l'information délivrée par une relation. Rubin et *al.* propose dans [14] un modèle de catégorisation de la certitude au travers de quatre dimensions :

— **le niveau de certitude** (*Absolute, High, Moderate, Low*). Par exemple, l'expression suivante « *... will almost certainly have to ...* » représente un niveau de certitude *Absolute* tandis que l'utilisation du modal *might* dans « *... might buy ...* » représente un niveau de certitude *Low* ;

— **la perspective** (point de vue de l'écrivain, point de vue rapporté). Par exemple, la certitude dans la phrase « *More evenhanded coverage of the presidential race would help enhance the legitimacy of*

6. www.metamap.nlm.nih.gov/
7. www.nlm.nih.gov/pubs/factsheets/umlsmeta.html
8. www.weka.sourceforge.net/doc.dev/weka/classifiers/trees/REPTree.html

the eventual winner, which ***now appears likely to be*** *Putin.* » est attribuée à l'écrivain selon ses connaissances et son expérience au moment où il écrit la phrase tandis que « *According to witness, Y has met X* » est un discours rapporté dans lequel la fiabilité de l'information est associée à celle de la source secondaire « *witness* » ;

— **le *focus*** de la certitude. Cette dimension est divisée en deux parties en fonction de la nature de l'information délivrée, soit abstraite (jugements, opinions, croyances, émotions) qui reflète une idée qui ne représente pas une réalité mais plutôt un monde hypothétique ou soit factuelle qui rapporte des états ou des événements et des faits connus ;

— **le temps.** Cette dimension prend en compte la pertinence du temps (passé, présent, futur). Le passé inclut des événements complets, le présent des états immédiats et incomplets et le futur est une prédiction ou une action suggérée.

L'article [10] s'inspire et enrichit ce modèle de catégorisation afin d'évaluer la certitude dans les phrases. Leur approche ajoute une nouvelle dimension qui est l'identification de la source (si la phrase est un discours rapporté) et modifie la troisième dimension (*focus*) par la notion de « réalité » qui permet de différencier l'affirmatif et le négatif (le modèle précédent ne traitait pas ces notions). Par la suite, leur approche utilise des patrons linguistiques pour détecter dans les textes l'incertitude selon ces cinq dimensions. Ces patrons se basent sur une association entre des termes ou des expressions et une dimension particulière, par exemple, l'adjectif « présumé » est considéré dans leur modèle comme un niveau d'incertitude *Moderate* ou bien les structures « selon, d'après, de source(s) ... » indique un point de vue rapporté. Enfin, chacune des identifications ajoute une annotation en entête d'un fichier RDF permettant de préciser les cinq dimensions (par exemple : *<onto :Level>Moderate</onto :Level>*). Leur approche a été évaluée sur chacune des dimensions : les dimensions Source et Niveau sont les dimensions obtenant la plus faible F-mesure (64% et 69%) tandis que la dimension temps obtient une F-mesure de 100%.

Ces différentes annotations pourraient permettre une pondération associée aux triplets lorsqu'ils sont insérés dans la base de connaissances. Cette pondération serait une information importante à considérer au sein du processus d'inférence.

4 Inférence de connaissances

Les bases de connaissances sont au carrefour de plusieurs disciplines : la recherche d'information, le traitement automatique des langues avec les systèmes de *question answering* et le raisonnement. La deuxième partie de nos travaux consistera à exploiter les relations extraites afin d'enrichir une base de connaissances et d'y appliquer des mécanismes de raisonnement en considérant l'incertitude liée aux phrases et à la connaissances et/ou règles qui pourront être inférées à partir de cet enrichissement. Actuellement, nous nous concentrons sur l'extraction d'informations, aussi cette seconde partie présente uniquement au travers d'un exemple les résultats que l'on attendrait.

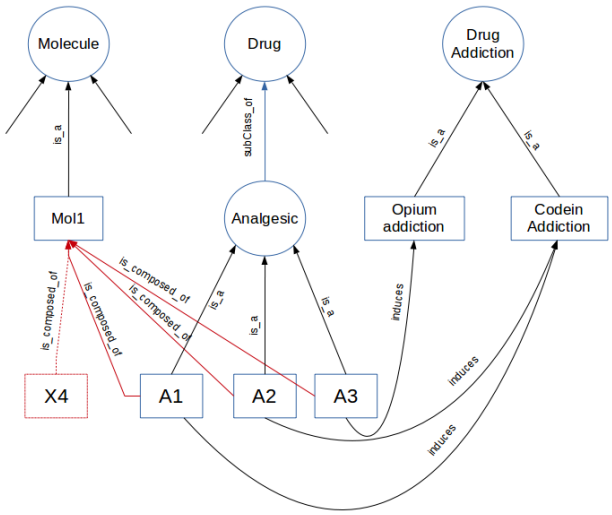


FIGURE 4 – Base de connaissances partielle d'une relation entre un analgésique et une molécule et entre un analgésique et une addiction.

La figure 4 décrit une base de connaissances partielle sur les médicaments, les molécules et les addictions aux médicaments. On peut observer que les trois instances de la classe *Analgesic* contiennent la molécule Mol1. Supposons que cette molécule soit principalement rattachée aux analgésiques dans la base de connaissances complète. Imaginons maintenant que nous avons extrait la phrase suivante : « X4 est un médicament composé de la molécule Mol1. ». Une induction relativement évidente, au vu de la base de connaissances, est d'inférer que X4 est un analgésique, donc inférer de la connaissance. Cette inférence doit être couplée avec une estimation de l'incertitude afin d'apporter à la connaissance une mesure de fiabilité. Cette incertitude doit dépendre du contexte d'incertitude des phrases, par exemple : « X4 est un médicament qui pourrait être composé de Mol1 » ne doit pas être évaluée de le même manière que la phrase précédente car elle nuance la certitude de la relation. De plus, les scores de confiance résultant des phases de désambiguïsation des entités correspond également à une information importante à prendre en compte. L'idée sous-jacente, qui constitue les prémices de nos travaux, est de réaliser une fonction pour agglomérer les différentes formes d'incertitude pouvant être extraites des phrases afin de pondérer une relation entre deux entités dans une base de connaissances.

De plus dans ce schéma de connaissances, on observe que chaque instance de la classe *Analgesic* induit une addiction aux médicaments. Dans le cas présent, inférer la règle suivante : « *Analgesic induce Drug Addiction* » est intui-

tif. L'article de Leaman et al. [9] présente une méthode d'extraction de règles qui explore l'espace de recherche des règles possibles à l'aide d'une extension itérative des règles de Horn en ajoutant des opérateurs d'extraction spécifiques dans le corps de la règle. Ce type de règle possède une tête représentée par une unique relation (*fatherOf(f ;c)*) et un corps constitué d'une conjonction de faits, par exemple :

$$motherOf(m\,;c) \wedge marriedTo(m\,;f)) \Rightarrow fatherOf(f\,;c)$$

Afin de réduire l'espace de recherche leur approche extrait des règles de Horn dites fermées correspondant au fait que chaque variable apparaisse au moins deux fois dans la règle (comme dans l'exemple précédent). De plus, les auteurs proposent une mesure de confiance associée à la règle extraite. Cette mesure est basée sur le nombre d'instanciations de la règle qui apparaissent dans la base de connaissances normalisée avec l'ensemble des faits positifs (en utilisant la *Partial Completeness Assumption*). Cette mesure considère uniquement le processus d'inférence (la base de connaissances est certaine).

5 Conclusion

Nous avons présenté l'état actuel de notre positionnement par rapport aux approches de la littérature concernant l'extraction de relations à partir de textes et la prise en compte de l'incertitude. Cette incertitude peut provenir de différentes sources et impacter le processus d'extraction de connaissances, à différents niveaux. Nous avons observé que les marqueurs d'incertitude présents dans les phrases peuvent être positionnés selon plusieurs dimensions principales : le niveau, la perspective, le *focus* et le temps. Ces marqueurs peuvent être extraits par l'intermédiaire d'approches de Traitement Automatique des Langues, telles que les patrons linguistiques appris automatiquement ou non, et permettent d'ajouter une annotation sur la certitude de la phrase. Par la suite, une base de connaissances construite avec ces données incertaines représente une mine de connaissances dans laquelle nous pourrions inférer des règles pouvant prendre en compte d'une part l'incertitude inhérente aux textes et d'autre part l'incertitude de l'inférence elle-même.

Références

[1] A. B. Abacha and P. Zweigenbaum. Automatic extraction of semantic relations between medical entities : a rule based approach. In *Fourth International Symposium on Semantic Mining in Biomedicine*, page S4, 2011.

[2] S. Ananiadou, P. Thompson, R. Nawaz, J. McNaught, and D. B. Kel. Event-based text mining for biology and functional genomics. In *Briefings in Functional Genomics*, pages 381–390, 2014.

[3] B. Andreopoulos, D. Alexopoulou, and M. Schroeder. Word sense disambiguation in biomedical ontologies with term co-occurrence analysis and document clustering. In *International Journal of Data Mining and Bioinformatics*, pages 193–215, 2008.

[4] A. Borillo. Les « adverbes d'opinion forte » selon moi, à mes yeux, à mon avis,... : point de vue subjectif et effet d'atténuation. In *Langue française*, pages 31–40, 2004.

[5] V. Dragos. An ontological analysis of uncertainty in soft data. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 1566–1573, 2013.

[6] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and Mausam. Open information extraction : the second generation. In *IJCAI*, pages 3–10, 2011.

[7] A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, 2011.

[8] L. Galárraga, N. Preda, and F. Suchanek. Mining rules to align knowledge bases. In *Workshop on Automated knowledge base construction*, pages 43–48, 2013.

[9] L. Galárraga, C. T. F. Suchanek, and K. Hose. Amie : Association rule mining under incomplete evidence in ontological knowledge bases. In *20th International World Wide Web Conference*, pages 413–422, 2013.

[10] B. Gounjon. Uncertainty detection for information extraction. In *International conference RANLP*, pages 118–122, 2009.

[11] Y. Kim, P. Bellot, E. Faath, and M. Dacos. Automatic annotation of bibliographical reference in digital humanities books, articles and blogs. In *CIKM*, pages 41–48, 2011.

[12] R. Leaman and G. Gonzalez. Banner : an executable survey of advances in biomedical named entity recognition. In *Pacific Symposium on Biocomputing*, pages 652–663, 2008.

[13] S. Muggleton. Learning from positive data. In *Inductive Logic Programming Workshop*, pages 358–376, 1996.

[14] V. Rubin, E. Liddy, and N. Kando. Certainty identification in texts : Categorization model and manual tagging results. In *Computing Attitude and Affect in Text : Theory and Applications, The Information Retrieval Series*, pages 61–76, 2005.

[15] C. Wang, A. Kalyanpur, J. Fan, B. K. Boguraev, and D. C. Gondek. Relation extraction and scoring in deepqa. In *IBM Journal of Research and Development*, pages 1–9, 2012.

[16] W. Wang, R. Besançon, O. Ferret, and B. Grau. Regroupement sémantique de relations pour l'extraction d'information non supervisée. In *TALN-Récital*, 2013.

Vers une coopération des collectifs de systèmes multi-agents hétérogènes

A. Khenifar Bessadi[1] J-P. Jamont[2] M. Occello[2] M. Koudil[1]

[1] Laboratoire de Modélisation et de Conception des Systèmes (LMCS),
Ecole Nationale Supérieure d'Informatique (ESI), Alger, Algérie
[2] Université Grenoble Alpes, LCIS, F-26000 Valence, France

BP 68M OUED SMAR, 16309 El Harrach Alger Algérie.
a_khenifar@esi.dz

Résumé

La coopération de différents Systèmes Multi-Agents (SMA) permet d'exploiter les produits collectifs de chacun d'entre eux afin de résoudre des problèmes complexes, améliorer leur résilience ou la qualité des services qu'ils fournissent. L'interaction de ces SMA, vus au niveau macroscopique comme des entités uniques, permet de diminuer la complexité en augmentant l'intelligibilité de la solution globale. Cependant, l'hétérogénéité des systèmes rend difficile l'interaction directe entre ces SMA. Nous exposons dans cet article le problème de la coopération de collectifs produits par différents SMA et notre démarche méthodologique pour permettre la conception de SMA coopérants.

Mots clef

Systèmes Multi-Agents, coopération, interopérabilité, niveau collectif, méthode, conception.

Abstract

The cooperation of different Multi-Agent Systems (MAS) allows exploiting the collective products of each of them in order to solve complex problems, improve their resilience or the quality of services they provide. The interaction of these MAS seen at the macroscopic level, each as a single entity, allows decreasing the complexity by increasing the intelligibility of the overall solution. However, systems heterogeneity makes difficult the direct interaction between the MAS. We expose in this article the problem of MAS collectives' cooperation and our methodological approach to enable the design of cooperating MAS.

Keywords

Multi-Agent Systems, cooperation, interoperability, collective level, method, conception.

1 Introduction

Contexte Les Systèmes multi-agents (SMA) proposent un ensemble d'approches, de modèles et d'architectures pour résoudre des problèmes complexes en mettant en œuvre la coopération de différents agents. La coopération permet aux différentes entités autonomes d'atteindre des objectifs communs en partageant leurs connaissances et/ou leurs savoir-faire. Plus loin, la coopération d'agents appartenant à des SMA différents permet à un SMA d'atteindre ses propres objectifs en exploitant de manière opportuniste les compétences disponibles dans d'autres SMA. Cependant, mettre en œuvre une telle coopération de SMA est un défi identifié depuis longtemps [16] et toujours d'actualité [6]. Ces problèmes d'interopérabilité peuvent être vus aux différents niveaux de description d'un SMA : celui des modèles, des architectures et des implantations.

Dans le cadre des SMA embarqués qui nous intéressent plus particulièrement, un même environnement physique peut être partagé par plusieurs SMA. L'interaction et la coopération de ces SMA peut apporter une valeur ajoutée aux différentes organisations et aux utilisateurs qu'ils servent (augmentation de l'adaptation, de la résilience, de la qualité de service rendue, etc.).

Problématique Les travaux sur l'interopérabilité des SMA reformulent souvent le problème de la coopération de SMA en un problème de coopération d'agents de ces SMA [1, 3, 10, 11]. Nous appelons cette démarche qui fait appel aux composants du plus bas niveau «coopération de niveau micro». Les agents des différents SMA sont amenés à coopérer pour atteindre leurs propres objectifs individuels. Ainsi, cette coopération ne se concentre ni sur l'objectif global du SMA ni sur comment l'atteindre mais sur une sous-tâche du système (qui peut éventuellement lui permettre d'atteindre l'objectif global). Dans ce sens, la plupart des travaux visent à identifier les agents qui sont adéquats pour accomplir la coopération.

L'objet de notre étude est quant à elle dirigée par l'objectif global du SMA et sur les services qu'il peut effectuer en totalité. En effet, plusieurs niveaux de coopération existent entre deux systèmes intelligents [17]. Dans notre étude, nous nous intéressons au niveau de coopération le plus élevé. Ainsi, chaque SMA est vu comme une entité autonome, indépendante éventuellement préexistante au système à concevoir. Un SMA peut décider lui-même de coopérer avec un autre si cela lui permet d'atteindre son objectif global. Nous appellerons cette approche centrée sur la

coopération des produits collectifs fournis par le SMA «coopération de niveau macro». Notre conviction est qu'elle permettra :

- une meilleure résolution collective des problèmes tout en préservant l'indépendance opérationnelle et l'autonomie de gestion des systèmes mis-en-jeux,
- une meilleure dynamique d'adaptation à l'environnement : possibilité d'un développement évolutif basé sur l'ajout/suppression de SMA, meilleure distribution spatiale/temporelle des SMA pour réagir aux dynamiques de l'environnement, meilleure utilisation de ressources partagées etc.
- une meilleure intelligibilité du produit collectif qui répond aux objectifs à atteindre en masquant la complexité interne de chacun des SMA : la décomposition du processus de résolution du problème sera alors plus compréhensible par un observateur externe qu'il soit humain ou artificiel.

Il est difficile pour un concepteur d'application de construire un tel système faisant coopérer des collectifs. Si l'on considère à titre d'exemple les nombreux travaux apparaissant à la frontière des SMA et des services web [8, 12, 20], il est plus facile pour le concepteur d'approcher le système en se plaçant au niveau des individus et donc de faire coopérer des agents qui effectuent des sous-services (un agent *shopbot* qui cherche les profils des malades en urgence et un agent *messagerie* du médecin [14]) que de considérer le problème au niveau des services globaux (un service d'alerte du médecin et un service de gestion patients vu comme des entités indépendantes en faisant abstraction de l'implantation à base d'agents). Toutefois, faire interagir les services rend la solution plus intelligible pour créer un nouveau service car il n'est alors pas indispensable de comprendre le détail d'implantation des solutions afin de pouvoir produire un nouveau service.

Contribution Nous proposons une approche méthodologique qui prend en considération les caractéristiques spécifiques lorsqu'on travaille au niveau macro de la coopération. Les contributions de ce travail sont :

- une approche méthodologique que le concepteur suit pour développer des SMA coopératifs au niveau collectif en éclaircissant les différentes caractéristiques à prendre en considération à un tel niveau de coopération;
- un outil d'analyse qui doit être renseigné par le concepteur désirant mettre en œuvre une coopération au niveau collectif de SMA pour analyser les SMA existants ou développer de nouveaux SMA afin de répondre à l'objectif du SoMAS qu'il veut construire.

Organisation de l'article Dans un premier temps (section 2) nous présentons le contexte et les motivations de la coopération des collectifs des SMA hétérogènes notamment à l'aide d'un scénario d'illustration. Nous présentons ensuite (section 3) la démarche que nous proposons pour concevoir des SMA coopérants au niveau collectif. Nous concluons cet article en présentant les perspectives de ce travail de recherche.

2 De l'interopérabilité des collectifs de SMA

2.1 Définitions préliminaires

Un phénomène collectif est le résultat observable des interactions d'un ensemble d'agents entre eux (et éventuellement avec leur environnement), dans le but d'atteindre un objectif commun.
Un phénomène collectif est a priori pré-planifié par le concepteur bien que les interactions inter-agents qui le font apparaitre soient imprévisibles dans la mesure où on ne peut pas pré-planifier ces interactions qui le font apparaitre. C'est le phénomène global d'un SMA créé pour répondre à certaines intentions ou buts [4] mais qui n'a pas nécessité la description d'un plan global d'exécution.

La coopération des collectifs est nécessaire lorsqu'un SMA ne peut pas complètement ou partiellement accomplir une tâche. Il va alors confier toute ou partie de l'exécution d'une tâche à un autre SMA qu'il juge capable d'accomplir. Chaque SMA peut alors être vu comme une entité autonome et indépendante qui exécute une tâche. Une tâche exécutée par un SMA est un phénomène collectif résultant des interactions entre les différents agents qui le composent. Les motivations d'un SMA pour coopérer proviennent de la nécessité qu'il éprouve à agir sur l'environnement, à maintenir un état (assurer la résilience) ou à se confronter un autre point de vue en acquérant une compétence existante ou en créant une nouvelle compétence (qui peut être qualifiée de collective).

Système de Systèmes Multi-Agents Cette volonté d'assurer la coopération des collectifs nous conduit à la notion de ***Système de Systèmes Multi-Agents (***SoMAS) dont l'objectif global ne pourra être atteint qu'à travers la coopération de chacun des SMA qui le constituent. Un produit collectif global (un comportement, une structure, une propriété) émergeant des interactions entre différents SMA peut être considéré comme un observable d'un Système de Systèmes Multi-agents (SoMAS) [2]. Le phénomène résultant de la coopération des collectifs de SMA sera donc l'accomplissement d'un objectif global du SoMAS. Les SoMAS partagent de nombreuses caractéristiques avec les Systèmes de Systèmes (SoS) [18] :

- *Chacun des SMA composant le SoMAS est indépendant* que ce soit au niveau « opérationnel » (chaque SMA a ses propres objectifs et fonctionne indépendamment pour les atteindre), au niveau dit « gestionnaire » (chaque SMA peut être introduit/retiré indépendamment des autres) ainsi qu'au niveau « évolutionnaire » (introduction, suppression ou modification de fonctionnalités de chacun des SMA).
- *Des phénomènes émergents au niveau du SoMAS* d'une manière similaire à ce qui se produit pour un SMA. L'interaction entre les collectifs des SMA du SoMAS engendre en effet l'apparition de nouveaux produits collectifs qui ne peuvent pas être associés spécifiquement à l'un des SMA. Ces phénomènes émergents ne sont pas nécessairement anticipés pendant la conception du SoMAS.
- *Un environnement différent pour chaque* SMA composant le SoMAS qui rend nécessaire l'existence de moyens d'interaction entre eux. Les SMA d'un SoMAS interagissent principalement par l'échange direct d'informations mais peuvent aussi interagir via l'environnement partagé.
- *Une hétérogénéité des systèmes* qui le compose. Répondre à un besoin au niveau du SoMAS nécessite des compétences et des connaissances de divers systèmes qui interagissent. La diversité des compétences est donc bénéfique pour le SoMAS. La contrepartie est que l'hétérogénéité des SMA complexifie le processus de conception car il faut permettre leur interaction. Le caractère opportuniste évoqué précédemment de ces interactions nécessite de développer des interfaces d'interaction pour faire face aux hétérogénéités et à la dynamique d'introduction/retrait de SMA.
- *Des systèmes vus comme des réseaux* car le SoMAS est composé de SMA imbriqués bien qu'ils ne soient pas nécessairement tous présents lors du déploiement initial (propriété d'ouverture). Ces réseaux émergent au fil du temps suite à l'adaptation des systèmes notamment aux instabilités de l'environnement

En plus des caractéristiques des SoS, la conception des SoMAS fait intervenir des caractéristiques issues des SMA:

- *Une automatisation du processus de coopération des collectifs au niveau de chaque SMA* car un SMA doit être capable d'observer, d'initier et de mettre fin à une phase de coopération des collectifs avec n'importe quel autre composant du SoMAS. Aussi, un SMA doit être observable et sollicitable pour permettre la coopération des collectifs accomplissant une fonction du SoMAS.
- *L'autonomie de chaque SMA*, car comme pour un agent, cette caractéristique est essentielle pour assurer le découplage des systèmes et permettre la mise en œuvre d'une intelligence collective. Chaque SMA doit s'adapter, apprendre et éventuellement évoluer pour accomplir ses tâches. Le degré d'autonomie d'un SMA peut être mesuré en prenant en compte sa capacité à accomplir ses propres tâches par lui-même.
- *Une dynamique de gestion de la coopération des collectifs par chaque SMA* car chaque SMA modifie dynamiquement son degré d'altruisme pour s'adapter aux besoins du SoMAS. Ce degré d'altruisme dépend de divers paramètres que le système contrôle comme le temps moyen passé aux activités consacrées à l'accomplissement des fonctions du SoMAS qui peut être au détriment du temps passé à répondre à ses objectifs individuels.

2.2 Motivations

Au niveau conceptuel, de par l'autonomie des agents, il n'est pas possible de prévoir à l'avance quelles sont les compétences que les agents peuvent nécessiter pour atteindre leurs objectifs individuels. En outre, il est difficile de spécifier a priori les contextes dans lesquels un agent pourrait avoir besoin d'interagir avec un autre pour satisfaire ses besoins en services [11]. Le problème de la coopération au niveau micro réside dans la vue locale de chaque agent. En effet, un agent coopère avec un autre agent pour répondre à ses propres besoins (les besoins collectifs étant souvent exprimés explicitement au sein de chaque agent). Les travaux qui ont traité de ce sujet pour les SMA se sont concentrés sur :

- la façon dont un agent pouvait interagir avec un autre agent hétérogène, en faisant abstraction du SMA auquel il appartenait (comme dans [9]).
- le choix du bon agent qui doit coopérer afin que la coopération soit bénéfique au SMA global comme c'est le cas dans [3] .

Il est difficile d'imaginer pour toutes les applications l'introduction d'une ou plusieurs entités (agents) résumant tout le système (exposition des connaissances et de l'ensemble des compétences). Dans le même contexte, les auteurs dans [21] défendent l'idée que tous les agents de tous les SMA soient égoïstes : ils ne cherchent en fait que la satisfaction de leurs propres objectifs et pas l'objectif global du SMA.. Beaucoup de travaux des SMA se limitent à la recherche par un seul agent d'une compétence requise au nom de son groupe [5, 7, 13, 19, 22, 23] et se reposent donc sur une coopération au niveau micro. Nous favorisons donc le niveau collectif de coopération qui se concentre plutôt sur l'objectif global du SMA.

2.3 Cas d'étude

Pour illustrer l'intérêt et les motivations de la coopération de niveau macro, considérons trois SMA qui œuvrent à la surveillance de feux de forêts (figure 1). Le premier SMA (SMA1 : Surveillance au sol) est

constitué d'agents capteurs qui doivent mesurer (périodiquement ou à la demande) la température au sol et l'envoyer vers une station de base qui collecte et traite l'ensemble des données. Le deuxième SMA (SMA2 : Surveillance aérienne) est constitué de drones qui survolent les espaces boisés à la recherche d'un départ d'incendies (caméra IR) et qui participent à leur extinction le cas échéant via un dispositif de dispersion de poudre extinctrice. Le troisième SMA (SMA3) est constitué d'agents humains instrumentés (les pompiers), des véhicules d'intervention, et d'un centre final de traitement. Les appareils de surveillance détectent le feu et envoient l'alarme au centre final de traitement le plus proche. Ce dernier planifie la mission d'extinction du feu en affectant des rôles aux véhicules de fonction.

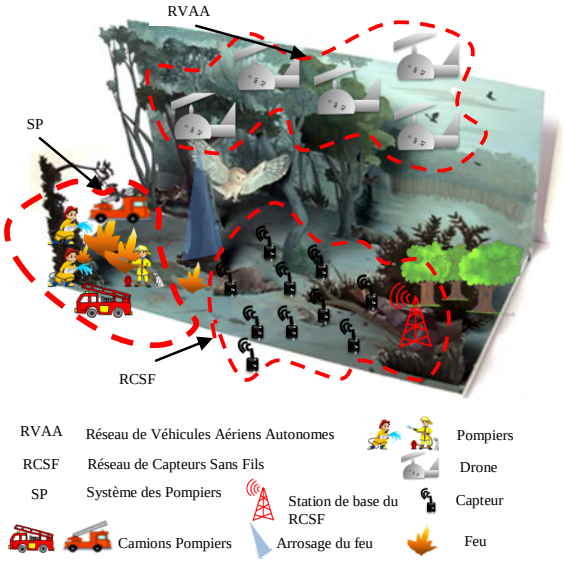


FIG. 1–Cas d'étude de surveillance de feux de forêts.

Chacun de ces SMA est déjà opérationnel et accomplit une tâche de surveillance de feux de forêts d'une façon autonome (en comptant sur ses propres compétences). L'objectif global de chacun des trois SMA est donc « le contrôle d'incendies des forêts ».
Considérons maintenant les services collectifs (i.e. compétences) produits par chacun des SMA et indépendamment des autres :

- Service collectif fourni par le SMA1 : acheminement de mesures des capteurs vers la station de base.
- Services collectifs du SMA2 : acheminement de messages entre les drones, extinction du feu.
- Service collectif du SMA3 : acheminement de données des véhicules au QG, extinction du feu.

Supposons que le SMA1 ne puisse plus acheminer l'information de déclenchement du feu à la station de base en raison d'un capteur défaillant. Cette panne provoque une partition du réseau de communication interne au SMA1. La figure 2 illustre ce partitionnement en 2 régions distinctes du SMA1.
Un agent du SMA1 détecte la coupure de connexion avec les autres capteurs permettant d'atteindre la station de base. Dans le cas de la coopération au niveau micro entre SMA, cet agent sollicite un agent (un drone ou un camion de pompiers) en ignorant s'il appartient ou non à son propre SMA et s'il est le bon agent (celui qui permettra de rétablir le service de communication).

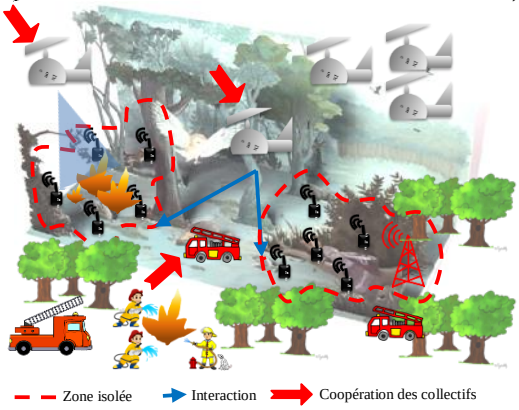


FIG. 2– Scénario de coopération des collectifs.

Du point de vue de la coopération au niveau macro, la requête du rétablissement du lien ou celle d'extinction du feu est remontée par l'agent qui détecte la panne au niveau collectif du SMA1. Il suffit donc que le SMA1 puisse observer un produit collectif d'un autre SMA, qu'il juge compatible avec son besoin, et le solliciter pour que le lien soit rétabli et le feu éteint. Ainsi, le SMA1 peut renforcer une compétence qu'il avait déjà (l'acheminement des données). Il peut acquérir une nouvelle compétence qui est l'extinction du feu et qui répond à son objectif global : le contrôle d'incendies des forêts. De cette façon, nous concevons un SoSMA à partir de SMA déjà fonctionnels afin de répondre à un objectif commun.

3 Approche méthodologique proposée

Démarche Avant de permettre l'utilisation du collectif d'un SMA par un autre, ce dernier doit prendre connaissance de l'existence de ce collectif. Les questions qui se posent sont : Comment se présente un collectif ? Comment l'observer? Où l'observer? Est ce qu'il est observable par tous les autres collectifs? Que produit ce collectif ?

Une fois que le SMA solliciteur arrive à identifier le collectif et à juger de son utilité pour atteindre son propre objectif, il essaye de coopérer avec le SMA correspondant. On s'intéresse alors aux questions suivantes: où coopérer ? Par quel médium? Faut-il remonter une requête du niveau micro au niveau collectif pour déclencher la coopération des collectifs? Sous quelle forme? En utilisant quel sous-ensemble d'agents? Quelles sont les informations échangées durant une phase de coopération des collectifs ?

Après avoir fait coopérer deux SMA au niveau collectif, il est alors important que le SMA qui l'exploite évalue la qualité du service qu'il a rendu. Une réponse aux questions suivantes est nécessaire: Quel a été l'apport/l'impact de la coopération des collectifs sur le fonctionnement du SMA solliciteur/sollicité? Comment le SMA peut-il évaluer la phase de coopération au niveau macro ?

Cette réflexion nous amène naturellement à diviser méthodologiquement la coopération des niveaux collectifs de SMA en trois principales phases comme l'illustre la figure 3 :

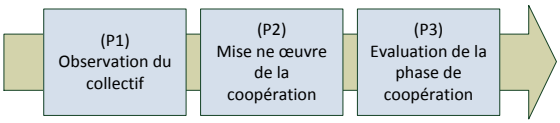


FIG. 3 – Les phases du processus de coopération des collectifs.

P1 : Cette phase consiste à permettre l'identification, le nommage et la compréhension d'un collectif. Nous avons défini une grille d'analyse qui pose différentes questions permettant de fixer les limites de l'étude et de faire la synthèse des collectifs. Elle peut être utilisée à différents niveaux du cycle de vie du SMA :

- L'étape d'analyse par un concepteur : la grille agit comme une check-list des questions à se poser pour chacun des SMA afin de comprendre les collectifs.
- L'étape d'identification automatique des collectifs par d'autres SMA: cette grille est alors exposée par un collectif pour se présenter ou être complétée de manière automatique par un collectif observant en utilisant des techniques de reconnaissance d'intentions, d'activités etc.

P2 : Les collectifs sont identifiés (P1), on sélectionne les collectifs pertinents pour la coopération (il faudra pour cela être sûr de la compatibilité des objectifs des différents collectifs dans P1) et on applique alors la stratégie de coopération.

P3 : La stratégie de coopération et le produit collectif sont évalués (analyse de retour d'expérience) par l'utilisateur, qu'il soit humain ou artificiel, afin de juger de sa pertinence.

Une fiche de conception Afin d'aider les concepteurs à se poser les bonnes questions lorsqu'ils souhaitent faire coopérer des SMA au niveau collectif, nous proposons une fiche de conception. Cette fiche contient les possibilités de conception pour le niveau micro (agents) et pour la formation du SoMAS. Elle permet aussi de mettre à disposition du concepteur les caractéristiques d'un collectif et de la coopération des collectifs classées dans les trois phases déjà proposées. Plus loin, cette fiche pourrait être utilisée par un SMA pour analyser un autre.

Nous exposons à titre d'exemple certaines des questions soulevées qui figurent dans la fiche que nous avons définie, en illustrant chacune d'elle à l'aide du scénario exposé:

- **Q1**. *Pourquoi faire coopérer les SMA au niveau collectif (P2, P3) ? – Pour agir sur l'environnement, maintenir un état ou assurer la résilience, se confronter avec un autre point de vue, renforcer une compétence, créer une nouvelle compétence.*
  Cette question porte sur la motivation des SMA à coopérer (les besoins). Pour le SMA1, soit il peut s'agir de renforcer une compétence déjà existante (l'acheminement des données) soit acquérir une nouvelle compétence (l'extinction du feu). Dans ce cas SMA1 tente de coopérer avec SMA2 et SMA3 respectivement pour maintenir l'interconnexion de l'ensemble des nœuds qui le composent et éteindre un feu qu'il détecte.

- **Q2**. *Quels types de phénomènes collectifs met en jeu la coopération des collectifs des SMA (P1, P2) ? Structures, propriétés, comportements/services.*
  Cette question s'intéresse à la nature des produits collectifs existants. SMA1 propose par exemple un service global d'acheminement de messages mais aussi une structure de groupes issue de l'application d'un processus d'auto-organisation inhérent (utilisation de MWAC [15] par exemple) qui peut être détournée de son utilisation première pour caractériser des zones géographiques.

- **Q3**. *Comment peuvent interagir les deux SMA afin de coopérer au niveau collectif (P1, P2) ?* [10] – *Communication, Action directe, Interaction par l'environnement.*
  A ce stade, le concepteur identifie les capacités d'interaction des SMA. Par exemple si les protocoles réseaux sans fil des capteurs et des drones sont les mêmes, ils pourront interagir directement en communiquant et pallier la partition du réseau sera plus aisé que s'il usurpait l'identité du nœud défaillant. Un drone qui détecterait un départ de feu mais qui ne pourrait pas communiquer, pourrait artificiellement augmenter la température au voisinage d'un nœud pour simuler un départ de feu (action par l'environnement).

- Nous avons défini de nombreuses autres questions que nous ne pouvons détailler ici. Comment un SMA peut observer et inscrire un phénomène collectif d'un autre SMA (P1) ? – Par observation d'échange de messages entre individus, observation de motifs d'interactions physiques, observation d'un réseau social/d'accointance. Comment un SMA peut guider

un collectif d'un autre SMA (P2) ? – En contactant un agent particulier, en déposant des marques dans l'environnement, en rejoignant une organisation. Quelle influence va avoir l'utilisation du collectif (P2, P3) ? Quel est le design du processus de coopération collective (P2) ? [10] – Explicite, par évolution ou par adaptation. Quel est le degré d'altruisme des SMA (P1, P2, P3) ? [10] Quel est le coût d'utilisation du service par l'utilisateur ? Quel est le coût du produit collectif pour le système participant ? Quel est le bénéfice au participant d'un produit collectif? … En tout, 18 questions ont été identifiées.

4 Conclusion et perspectives

Cet article a défini la notion de Système de Systèmes Multi-Agents et motivé le besoin de coopération de SMA au niveau macro. Une approche est proposée, concrétisée notamment par une fiche de conception pour l'analyse de la coopération des collectifs de SMA. Nous visons la proposition d'un modèle qui utilise cette fiche de conception permettant de répondre aux questions suivantes :

- Un SMA non coopératif au niveau collectif pourra t-il le devenir en intégrant différents mécanismes ou devra t-il être conçu pour l'être dès le début de la conception ?
- La coopération des collectifs doit-elle être motivée, a priori, ou peut-elle émerger des interactions des collectifs ?
- Quels sont les mécanismes clés qui donnent lieu à une coopération au niveau collectif et lui permettre de se maintenir ?
- Dans quelle mesure ces mécanismes sont nécessaires ? Est-il possible d'identifier un ensemble de conditions nécessaires pour la coopération des collectifs ?

L'instanciation de ce modèle en une architecture et son implantation permettra de valider globalement cette approche.

Bibliographie

[1] W. Alshabi, S. Ramaswamy, M. Itmi, and H. Abdulrab, "Coordination, cooperation and conflict resolution in multi-agent systems," in *Innovations and Advanced Techniques in Computer and Information Sciences and Engineering*, ed: Springer, 2007, pp. 495-500.

[2] S. Biswas, L. Bai, and Q. Dong, "Multi-objective consensus of interconnected system of multi-agent systems," in *Resilient Control Systems (ISRCS), 2013 6th International Symposium on*, 2013, pp. 42-47.

[3] F. Buccafurri, D. Rosaci, G. M. Sarnè, and L. Palopoli, "Modeling cooperation in multi-agent communities," *Cognitive Systems Research*, vol. 5, pp. 171-190, 2004.

[4] P. Cariani, "Emergence of new signal-primitives in neural systems," *Intellectica*, vol. 2, pp. 95-143, 1997.

[5] J. A. Carlson and R. P. McAfee, "Joint search for several goods," *Journal of Economic Theory*, vol. 32, pp. 337-345, 1984.

[6] J. Chen, Y. Hong, J. Huang, and J. Sun, "Guest editorial special issue on cooperative control of multi-agents systems: Theory and applications," *Automatica Sinica, IEEE/CAA Journal of*, vol. 1, pp. II-II, 2014.

[7] V. Conitzer and T. Sandholm, "Automated mechanism design for a self-interested designer," in *Proceedings of the 4th ACM conference on Electronic commerce*, 2003, pp. 232-233.

[8] J. A. G. Coria, J. A. Castellanos-Garzón, and J. M. Corchado, "Intelligent business processes composition based on multi-agent systems," *Expert Systems with Applications*, vol. 41, pp. 1189-1205, 2014.

[9] L. R. Coutinho, A. A. Brandão, J. S. Sichman, J. F. Hübner, and O. Boissier, "A model-based architecture for organizational interoperability in open multiagent systems," in *Coordination, Organizations, Institutions and Norms in Agent Systems V*, ed: Springer, 2010, pp. 102-113.

[10] J. E. Doran, S. Franklin, N. R. Jennings, and T. J. Norman, "On cooperation in multi-agent systems," *The Knowledge Engineering Review*, vol. 12, pp. 309-314, 1997.

[11] T. Ebadi, M. Purvis, and M. Purvis, "A framework for facilitating cooperation in multi-agent systems," *The Journal of Supercomputing*, vol. 51, pp. 393-417, 2010.

[12] W. El Kholy, J. Bentahar, M. El Menshawy, H. Qu, and R. Dssouli, "Modeling and verifying choreographed multi-agent-based web service compositions regulated by commitment protocols," *Expert Systems with Applications*, vol. 41, pp. 7478-7494, 2014.

[13] J. R. J. Gatti, "Multi-commodity consumer search," *Journal of Economic Theory*, vol. 86, pp. 219-244, 1999.

[14] M. N. Huhns, "Software agents: the future of web services," in *Agent Technologies, Infrastructures, Tools, and Applications for E-Services*, ed: Springer, 2003, pp. 1-18.

[15] J. Jamont and M. Occello, "A multiagent tool to simulate hybrid real/virtual embedded agent societies," in *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on*, 2009, pp. 501-504.

[16] N. R. Jennings, K. Sycara, and M. Wooldridge, "A roadmap of agent research and development," *Autonomous agents and multi-agent systems*, vol. 1, pp. 7-38, 1998.

[17] A. Khenifar, J.-P. Jamont, M. Occello, C.-B. Ben-Yelles, and M. Koudil, "A recursive approach to enable the collective level interaction of the Web of Things applications," in *Proceedings of the 2014 International Workshop on Web Intelligence and Smart Sensing*, 2014, pp. 1-2.

[18] P. N. Lowe and M. W. Chen, "System of systems complexity: modeling and simulation issues," in *Proceedings of the 2008 Summer Computer Simulation Conference*, 2008.

[19] E. Manisterski, D. Sarne, and S. Kraus, "Enhancing cooperative search with concurrent interactions," *Journal of Artificial Intelligence Research*, vol. 32, pp. 1-36, 2008.

[20] D. Mitrović, M. Ivanović, Z. Budimac, and M. Vidaković, "Radigost: Interoperable web-based multi-agent platform," *Journal of Systems and Software*, vol. 90, pp. 167-178, 2014.

[21] I. Rochlin, D. Sarne, and M. Mash, "Joint search with self-interested agents and the failure of cooperation enhancers," *Artificial Intelligence*, vol. 214, pp. 45-65, 2014.

[22] D. Sarne and S. Kraus, "Cooperative exploration in the electronic marketplace," in *AAAI*, 2005, pp. 158-163.

[23] D. Sarne, E. Manisterski, and S. Kraus, "Multi-goal economic search using dynamic search structures," *Autonomous Agents and Multi-Agent Systems*, vol. 21, pp. 204-236, 2010.

Vers un modèle de stratégie basé sur la gestion des ressources pour la création des IA dans les RTS

Juliette LEMAITRE [1,2] Domitile LOURDEAUX [1] Caroline CHOPINAUD [2]

[1] Sorbonne universités, Université de technologie de Compiègne, CNRS, Heudiasyc - UMR CNRS 7253, CS 60 319, 60 203 Compiègne cedex, France
[2] MASA Group, 8 rue de la Michodière, 75002 Paris, France

juliette.lemaitre@hds.utc.fr

Résumé

Dans les jeux de stratégie en temps réel existants à ce jour, les comportements des opposants au joueur sont souvent critiqués. Les stratégies qu'ils suivent ne sont plus suffisamment intéressantes dès lors que l'on dépasse le stade du joueur débutant ou de l'entraînement, car trop simples et prédictibles. Le mode multijoueurs est alors largement préféré par une majorité de joueurs chevronnés. Pourtant de nombreuses recherches ont été menées sur le sujet de l'IA dans les jeux vidéo et en particulier pour les RTS, qui permettent aux entités d'exhiber des comportements de plus haut niveau et très efficaces pour remplir les objectifs du jeu. Mais les solutions proposées sont trop compliquées à comprendre et à utiliser pour intéresser le milieu industriel du jeu vidéo, soumis à de fortes contraintes de temps. Pour permettre le développement d'IA présentant un plus grand challenge, nous proposons dans ce papier un modèle de stratégie qui se veut intelligible et qui permet de créer facilement des comportements plus complexes que ceux actuellement proposés, capable de s'adapter à la dynamique du jeu.

Mots Clef

Stratégie, modélisation, game design.

Abstract

The artificial intelligence used for opponent non-player characters in commercial real-time strategy games is often criticized by players. It is used to discover the game but soon becomes too easy and too predictable. Yet, a lot of research has been done on the subject, and successful complex behaviors have been created, but the systems used are too complicated to be used by the video games industry, as they would need time for the game designer to learn how they function, which ultimately proves prohibitive. Moreover these systems often lack control for the game designer to be adapted to the desired behavior. To address the issue, we propose an accessible strategy model that can adapt itself to the player and can be easily created and modified by the game designer.

Keywords

Strategy, Behavior Modeling, Game Design.

1 Introduction

Les Intelligences Artificielles (IA) contrôlant les opposants dans les jeux vidéo commerciaux ne tirent pas pleinement avantage des résultats de la recherche académique sur le sujet. Certains systèmes comme la planification ou même les algorithmes d'apprentissage ont déjà été utilisés par des jeux[7, 4], mais cela reste des exceptions. La principale raison est l'existence de trop grands écarts entre leurs buts et leurs contraintes, rendant les solutions proposées par la recherche académique peu adaptées à l'industrie du jeu.

Au cours de la création d'un jeu vidéo, la conception des IA opposantes est rarement la priorité, on lui préfère souvent d'autres aspects comme les graphismes ou l'animation, rendant la contrainte de temps d'autant plus critique. Cette forte contrainte temporelle ne permet généralement pas au studio de développer une solution d'IA poussée et réutilisable d'un jeu à l'autre. Ces développements spécifiques obligent alors à systématiquement créer de nouvelles IA à partir de zéro. Nous cherchons donc à proposer une solution rapidement intégrable dans ce processus, qui facilite et accélère la création de stratégies tout en permettant leur réutilisabilité.

Les jeux de stratégie en temps réel (RTS) possèdent des environnements présentant de nombreux challenges pour la recherche en IA qui les utilise comme environnements de test. Pourtant, même si ces environnements nécessitent des IA complexes, les solutions utilisées par l'industrie pour créer les IA intégrées au jeu sont trop simples et maladaptées. En effet, des solutions de type machines à états (FSM), arbres de décisions (BT), ou dans le pire des cas, des scripts sont préférés à des solutions plus complexes car elles apparaissent comme simples d'utilisation et intelligibles de prime abord.

Cette simplicité n'est pas la seule raison pour le choix de ces systèmes d'IA, en effet les BT et FSM offrent non seulement une simplicité de compréhension et d'utilisabi-

lité mais aussi de la fiabilité et du contrôle. Au moins un de ces critères manquent à la plupart des solutions proposées par la recherche académique : certains systèmes ne trouvent pas toujours de solution et les entités contrôlées peuvent se retrouver inanimées ; il est également souvent difficile voire impossible pour les systèmes proposés de personnaliser la prise de décision pour obtenir le comportement voulu. En effet, l'objectif orientant la plupart des recherches académiques est l'optimisation des résultats (plus rapides ou plus efficaces) alors qu'un jeu vidéo a pour objectif de divertir le joueur. Si ces deux buts ne sont pas incompatibles, ils sont bien souvent résolus par des solutions différentes.

Pour résumer, l'industrie du jeu vidéo est soumise à des contraintes de temps qui diffèrent de celles de la recherche académique, et qui nécessitent d'utiliser des systèmes simples de compréhension et d'utilisation, d'autant plus pour la conception d'IA qui est rarement une priorité. De plus l'objectif d'un jeu vidéo étant de produire une expérience divertissante pour le joueur, les IA créées doivent être faciles à contrôler et personnaliser pour qu'elles contribuent à l'expérience souhaitée par le game designer. Dans le but de répondre à l'ensemble de ces attentes, nous proposons une solution pour concevoir et tester de manière efficace des comportements d'opposants dans les RTS, qui prend la forme d'un modèle de "stratégies" adaptatives et réutilisables. A partir d'une présentation des limites des approches actuellement existantes, nous détaillerons notre modèle de conception de stratégie et nous discuterons de ses avantages et de sa place dans une approche plus globale d'aide à la conception de stratégies.

2 Etat de l'art

Un aperçu des travaux réalisés au sujet des IA dans les RTS a été publié par Ontanon [6], il y cite à la fois le travail effectué dans le cadre des compétitions d'IA pour le jeu Starcraft organisées lors des conférences CIG et AAAI, et également les autres travaux de recherche qui ont été conduits sur le sujet.

Plusieurs raisons font des compétitions mentionnées cidessus de mauvais candidats pour fournir un système utilisable par l'industrie. Premièrement, la victoire lors des duels est le seul aspect servant à la comparaison de deux candidats. Le principe général est de mettre le système d'IA de chaque participant face aux systèmes de ses adversaires et de comparer les pourcentages de jeux gagnés, perdus ou nuls pour effectuer un classement. Deuxièmement, les architectures créées pour ces événements sont spécifiques au jeu Starcraft, elles sont découpées à la fois de façon parallèle et hiérarchique en sous-modules[6] correspondant aux spécificités du jeu. La spécialisation des sous-modules les rendent difficiles à réutiliser dans un environnement différent, cependant cela montre la complexité de la tâche de décision et l'importance de la décomposer. Avec notre modèle nous cherchons à fournir une structure de stratégie qui lui permette d'être réutilisée dans des environnements diverses.

Les environnements de RTS sont très utilisés dans la recherche académique comme environnement de test pour les nombreux challenges qu'ils présentent [1]. Plusieurs techniques d'IA y ont été appliquées pour tester leur efficacité mais elles manquent souvent de l'utilisabilité nécessaire pour être utilisées par l'industrie. Par exemple [3] utilise des ensembles de logs de jeux sur Starcraft pour créer des modèles de Markov cachés. Les comportements obtenus s'appuient intégralement sur des probabilités le rendant imprédictible et diminuant grandement le contrôle du game designer sur les comportements en résultant.

La planification à base de cas appliquée dans [5], et étudié plus largement dans [8] manque également de contrôle sur les comportements obtenus, car elle se base exclusivement sur des bibliothèques d'exemples obtenus à partir de jeux joués et commentés par des experts. Le fonctionnement de la planification à base de cas rend également difficile la compréhension des raisons de la génération de comportements non souhaités.

De plus, ces deux systèmes utilisent une méthode d'apprentissage qui peine à s'introduire dans le processus de création de jeux vidéo. En effet l'apprentissage est un processus long et qui nécessite une quantité importante de données en entrée. Si l'apprentissage peut faire économiser du temps lors de l'étape de création des comportements, les données nécessaires au processus peuvent prendre du temps à être générées. De plus, pour que des modifications puissent être apportées aux comportements produits par apprentissage, il est indispensable que les résultats obtenus soient intelligibles par le game designer, ce qui rend le processus d'autant plus complexe.

La planification a également été testée lors de travaux de recherche, [2] l'utilise pour optimiser l'ordre des constructions dans Starcraft. Cette méthode peut donc être efficace sur une sous partie du raisonnement mais à cause du vaste espace de recherche, son utilisation pour le mécanisme entier de prise de décision est impossible car nécessitant un temps trop important de calcul non compatible avec des jeux en temps réel. De plus il peut être compliqué lors de l'observation d'un comportement non voulu, d'en comprendre les raisons.

3 PROPOSITION

L'objectif de notre travail est de fournir un modèle de stratégie accessible qui permet de simplifier la création de comportements complexes. Une stratégie est définie comme le processus de prise de décision pour l'allocation des ressources aux sous-tâches de façon à poursuivre un objectif global. Les comportements produits doivent être facilement modifiables et réutilisables. Cela permettra au système d'être facilement intégrable au processus de création de jeu vidéo durant lequel des modifications fréquentes des mécaniques de jeu nécessitent l'adaptation de tous les éléments, y compris de l'IA. Nous voulons également que le modèle soit appréhendable, autrement dit que la rai-

son de l'apparition d'un comportement non voulu peut être compris et résolu facilement de façon à permettre aux game designers de garder le contrôle sur les comportements obtenus.

3.1 Le modèle de stratégie

Notre modèle a pour objectif de faciliter la création de comportements collectifs dans les RTS, et plus généralement de comportements impliquant plusieurs agents ayant besoin de se coordonner. Pour construire une stratégie, nous utilisons une structure hiérarchique qui permet sa décomposition en sous-comportements plus simples, permettant ainsi au concepteur de se concentrer sur le niveau d'abstraction voulu et d'ajouter facilement un niveau supérieur utilisant les comportements précédemment créés. Un comportement est donc composé d'un ensemble de sous-comportements et la stratégie peut alors être représentée par un graphe orienté acyclique (DAG) avec un unique noeud racine et dans lequel les fils d'un noeud correspondent à ses sous-comportements. Le schéma 1 représente un exemple partiel de stratégie qui correspond à celle décrite dans le paragraphe 4. Tous les noeuds sous la forme d'un losange sont des tâches primitives, c'est-à-dire qu'elles peuvent être directement exécutées dans l'environnement virtuel, elles ne possèdent pas de sous-comportements. Les ronds et les carrés ne possédant pas d'arcs sortants sont incomplets et possèdent en réalité des sous-comportements qui ne sont pas représentés ici pour une question de place.

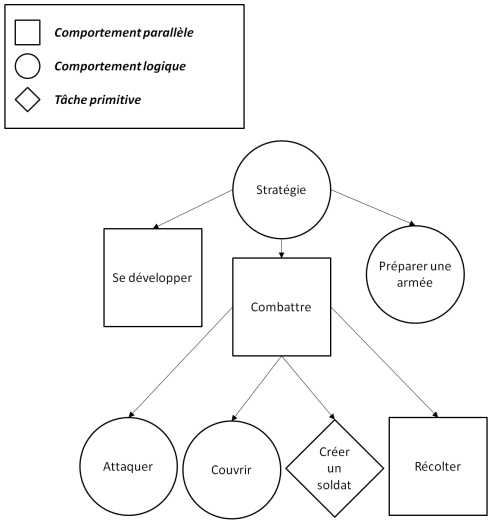


FIGURE 1 – Stratégie.

Un comportement est défini par un ensemble de sous-comportements, par son type, et par les ressources dont il a besoin. Le type d'un comportement peut être soit logique soit parallèle et détermine le mode d'utilisation des sous-comportements. Pour chaque comportement, l'ensemble des sous-comportements associé peut être composé d'un mélange de comportements parallèles, logiques ainsi que de tâches primitives. Le type parallèle aborde la problématique d'allocation des ressources car il va effectuer plusieurs de ses sous-comportements en même temps ; il est représenté dans nos graphes par un noeud carré. Le type logique permet à l'IA de choisir un seul de ses sous-comportements en fonction de l'état courant du jeu ; il est représenté dans nos graphes par un noeud rond. Chaque type a besoin d'informations supplémentaires relatives à leur type ; leur fonctionnement est davantage détaillé dans la suite de cette partie.

Tâche Primitive. Une tâche primitive correspond à la plus petite unité de décomposition d'un comportement. Elle ne peut plus être décomposée, mais elle est liée à une fonction qui lui permet d'être exécutée dans l'environnement virtuel. Elle est définie par un ensemble de ressources qui lui sont nécessaires pour être effectuée sous la forme de tuples *<R, Min, Max>* où :

```
R est le type de ressource
Min est la quantité minimum nécessaire
Max la quantité maximum possible
```

R est défini grâce au modèle de ressources présenté dans le paragraphe 3.2. Max peut être mis à -1 s'il n'y en a pas. Cette représentation permet une grande flexibilité et est adaptée pour des quantités variables de ressources, comme c'est le cas pour les jeux de type RTS. Par exemple, pour une tâche de construction, on peut lui associer les ressources suivantes :

```
<<ouvrier, 1, -1>, <terrain, 1, 1>>
```

De cette manière, plusieurs ouvriers peuvent être affectés à la même tâche de construction qui n'utilisera par contre qu'une unité de terrain. De façon à gérer le résultat d'une tâche, celle-ci peut retourner une valeur de retour qui sera envoyée au comportement parent, qui pourra alors la prendre en compte pour la suite du comportement.

Comportement Logique. Un comportement logique permet de représenter une logique de sélection de l'un des sous-comportements. Son exécution consiste à sélectionner le sous-comportement devant être exécuté, en tenant compte du sous-comportement précédemment sélectionné et de l'état courant du monde. Il est représenté par un tuple *<B, M, SB, CB>* où :

```
B est un ensemble d'états
M est un ensemble des transitions <OB, T,
    DB>
SB est le sous-comportement initial
CB est le sous-comportement couramment
    sélectionné
```

Une transition est composée d'un déclencheur T qui déclenche la sélection d'un état DB si OB est l'état courant. Un état peut être soit un sous-comportement, soit un état retour qui déclenche l'envoi d'un signal renfermant une valeur de retour au comportement parent.
Les déclencheurs peuvent être des signaux internes ou externes, ou des requêtes d'information : les signaux internes

correspondent aux retours du sous-comportement courant, les signaux externes viennent de modules dédiés au jeu, et les requêtes d'information sont représentées par des formules booléennes. Par exemple, un signal externe peut venir d'un module qui gère l'état du monde si la transition doit être activée par un changement dans l'état du monde, ou alors le signal peut venir d'un module de modélisation du joueur si la condition dépend de l'état du joueur. Dans le cas d'une requête d'information, la transition est activée si la formule est vraie. Les signaux sont utilisés pour des événements ponctuels alors que les requêtes d'information sont utilisées pour observer des états du monde plus statiques.

Considérons un exemple consistant de 3 sous-comportements : *Explorer*, *Combattre*, et *Récolter*. *Explorer* est défini comme étant le sous-comportement initial, puis, si des ennemis sont rencontrés ou de la nourriture est trouvée durant l'exploration du monde, on sélectionne respectivement le sous-comportement *Combattre* ou *Récolter*. Quand les sous-comportements *Combattre* ou *Récolter* renvoient un signal de succès, le comportement *Explorer* est de nouveau sélectionné. Dans le cas où des ennemis sont rencontrés alors que le sous-comportement *Récolter* est en cours de réalisation, le sous-comportement *Combattre* est préféré. Cet exemple peut être représenté comme suit, le comportement *Récolter* étant le sous-comportement actuellement sélectionné.

```
B = {Explorer, Combattre, Récolter}
M = {
      <Explorer, Signal(EnnemiRepéré),
          Combattre>,
      <Explorer, Signal(NourritureRepérée),
           Récolter>,
      <Combattre, Signal(Succès), Explorer
          >,
      <Récolter, Signal(Succès), Explorer>,
      <Récolter, Signal(EnnemiRepéré),
          Combattre> }
SB = Explorer
CB = Récolter
```

Un comportement logique peut ainsi être représenté par un graphe orienté qui peut être cyclique, les sous-comportements et les signaux de retour étant les noeuds et M énumérant les arcs avec OB le noeud origine et DB le noeud destination. Un graphe illustrant l'exemple décrit plus haut est représenté sur la figure 2.

Comportement Parallèle. Un comportement parallèle répartit les ressources qui lui ont été allouées entre ses différents sous-comportements. Ceux-ci sont distribués entre plusieurs niveaux de priorité qui sont représentés par des couples *<M, C>*. M correspond au nombre de fois maximum où le niveau peut être exécuté en parallèle, mais il est facultatif : il est notamment inutile lors de la définition des tâches par défaut qui doivent donc être allouées à toutes les ressources restantes qui peuvent l'effectuer, quelque soit leur nombre. Par convention, nous utiliserons la valeur -1

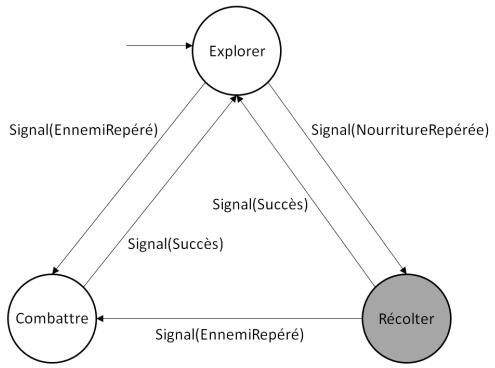


FIGURE 2 – Exemple simple de comportement logique.

pour indiquer qu'il n'y a pas de limite. C représente un ensemble de couples *<B, W>* où B est un sous-comportement et W une quantité, correspondant au nombre de fois où le sous-comportement est effectué à chaque exécution du niveau.

Les ressources disponibles sont d'abord allouées au niveau de plus haute priorité, et cela tant que c'est possible et dans la limite de M fois. Pour qu'une instance d'un niveau de priorité soit ajouté à la liste d'exécution, il faut que les ressources disponibles permettent d'effectuer l'ensemble de ses sous-comportements, autant de fois qu'indiqué par les quantités W. Un sous-comportement peut apparaître dans plusieurs niveaux de priorité, mais il ne peut apparaître qu'une seule fois à l'intérieur d'un niveau. Cette description permet une distribution des ressources qui s'adapte à la quantité des ressources disponibles.

TABLE 1 – Exemple simple de comportement parallèle.

Itérations maximum	Sous-comportement	Quantité
2	Combattre	1
	Couvrir	2
1	Alerter	
-1	Explorer	

Le tableau 1 décrit un exemple de comportement parallèle. Le premier niveau, pour être réalisé, a besoin de déclencher un sous-comportement *Combattre* et deux sous-comportements *Couvrir*, il peut être effectué au maximum 2 fois. Le deuxième niveau a besoin pour être réalisé de déclencher un sous-comportement *Alerter* et ne peut être effectué qu'une seule fois. Le dernier niveau représente le comportement par défaut pour toutes les ressources restantes, il a donc pour nombre maximum -1, et a besoin pour se réaliser de déclencher un sous-comportement *Explorer*.

En considérant que chacun des sous-comportements ne nécessite comme ressource qu'un soldat, et que nous avons à notre disposition 10 soldats, ils se répartiront alors comme suit :

- le premier niveau sera d'abord sélectionné, pour être

réalisé il a besoin de 3 soldats, 1 pour combattre, et 2 pour couvrir. Il nous reste donc 7 soldats.
- le premier niveau n'a été effectué qu'une fois, le nombre maximum d'itérations n'est donc pas encore atteint, il nous reste assez de soldats pour effectuer de nouveau le premier niveau, 3 soldats sont donc de nouveau alloués. Il nous reste 4 soldats.
- le nombre maximum d'itérations du premier niveau est atteint, nous passons donc au deuxième niveau qui a besoin d'un seul soldat, il nous en reste de disponible, un soldat est donc alloué pour le sous-comportement *Alerter*. Il nous reste 3 soldats.
- le nombre maximum d'itérations est atteint pour le deuxième niveau, nous passons donc au troisième qui a besoin lui aussi d'un seul soldat. Le niveau n'ayant pas de nombre maximum d'itérations, il sera effectué autant de fois qu'il reste de soldats, et le sous-comportement *Explorer* sera donc alloué à chacun des trois soldats restants.

3.2 Le modèle de ressources

Une ressource peut être un objet, un agent, mais aussi un lieu, ou bien une compétence. Une hiérarchie de ressources doit être définie et seules les feuilles peuvent être utilisées pour définir le type d'une ressource lors de sa création dans l'environnement virtuel. En revanche, tous les types peuvent être utilisés pour définir les ressources nécessaires à une tâche primitive, autrement dit, quand une ressource est nécessaire pour la réalisation d'une tâche ou d'un comportement, son type peut être plus ou moins spécifique. Par exemple, la figure 3 représente une hiérarchie de types de ressources. Un *Agent* d'un jeu pour lequel cette hiérarchie est utilisée peut être soit un *Soldat* soit un *Ouvrier*, l'environnement inclut également des ressources de type *Epée* et *Pelle*. Cela signifie que si une tâche primitive a pour ressource nécessaire une ressource de type *Agent*, elle pourra être réalisée par un un *Soldat* ou bien par un *Ouvrier*. Des options plus avancées permettent au game designer d'indiquer que si une tâche a besoin de deux ressources de type *Agent*, ils ne doivent pas avoir le même sous-type, ou bien au contraire ils doivent avoir le même, ou bien encore cela n'a pas d'importance.
Une ressource est caractérisée par les propriétés suivantes : elle possède un type qui ne peut pas changer au cours de son existence ; une ressource peut être créée ou détruite par une tâche primitive ; la modification du type d'une ressource peut donc être simulée avec une destruction puis une création.

4 CAS D'USAGE

L'exemple suivant illustre un autre cas d'utilisation de notre modèle pour construire une stratégie dans un environnement utilisant quelques mécanismes des jeux RTS : la recherche et l'extraction de ressources minérales servant à la création de bâtiments et d'unités (ouvriers et soldats) ;

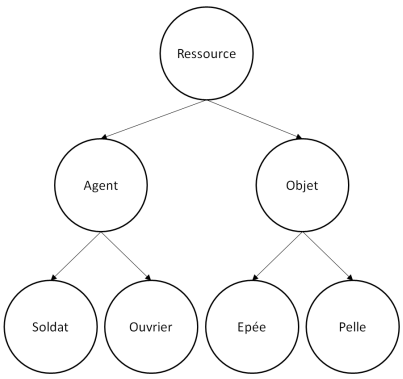


FIGURE 3 – Hiérarchie de ressources.

les bâtiments sont utiles pour augmenter la protection de la ville et l'efficacité des soldats produits. Pour simplifier la compréhension de la stratégie proposée, seules certaines parties seront détaillées.
La stratégie générale est présentée sur la figure 4, on y distingue trois phases du jeu : elle débute par une phase d'expansion durant laquelle des bâtiments sont créés jusqu'à ce qu'il soit considéré nécessaire de passer en phase de préparation au combat avec la construction d'une armée, pour finir inexorablement par une phase de combat avec l'ennemi. Plusieurs raisons peuvent expliquer la nécessité de créer une armée : cela peut être une durée de jeu, le nombre de bâtiments ayant déjà été créés, mais également si par espionnage il a été découvert que l'ennemi préparait une armée. Si l'ennemi attaque par surprise sans que la construction d'une armée est pu être débutée, le combat est directement engagé.

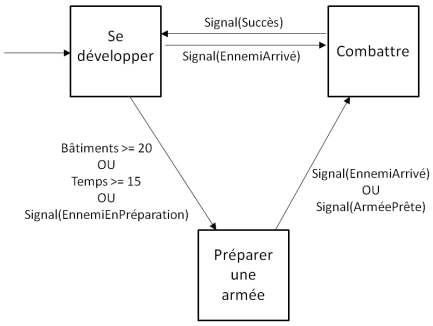


FIGURE 4 – Stratégie globale de l'exemple.

Le sous-comportement *Combattre* peut être décrit par un comportement parallèle, comme celui représenté dans le tableau 2. Les soldats disponibles vont alors commencer le combat pendant que les ouvriers continuent de collecter des ressources pour qu'elles puissent être utilisées pour créer davantage de soldats. Dans cet exemple, chaque soldat qui attaque un ennemi est couvert par un autre soldat.

Le sous-comportement *Récolter* est le comportement par défaut pour les ouvriers et est donc placé en dernier. Si les ressources nécessaires pour construire un soldat ne sont pas disponibles, le sous-comportement *Créer un soldat* ne sera pas effectué, et tous les ouvriers récolteront des ressources. Dès que les ressources nécessaires seront disponibles, un ouvrier sera assigné à cette tâche.

TABLE 2 – Comportement parallèle *Combattre*.

Itérations maximum	Sous-comportement	Quantité
-1	Attaquer	1
	Couvrir	1
-1	Créer un soldat	
-1	Récolter	

Les sous-comportements *Attaquer* et *Couvrir* sont respectivement détaillés sur les figures 5 et 6. Le comportement *Attaquer* est relativement simple : tirer sur l'ennemi à portée de tir le plus faible et si aucun ennemi n'est à portée de tir alors se déplacer vers l'ennemi le plus proche. Le comportement *Couvrir* est un peu plus complexe et nécessite des signaux externes et des requêtes d'information sur l'environnement. Il consiste à trouver une place pour se mettre à couvert, y déplacer le soldat, le mettant accroupi si nécessaire, puis tirer sur les ennemis visibles. Ensuite, si le soldat couvert s'est déplacé et que la place à couvert actuelle n'est alors plus adaptée, le soldat doit être déplacé.

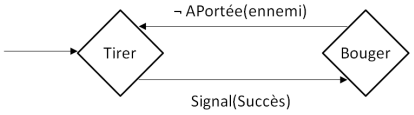


FIGURE 5 – Comportement logique *Attaquer*.

5 CONCLUSION

Dans ce papier nous avons présenté un modèle pour la définition de stratégies pour les jeux de type RTS dont la spécificité est de combiner la facilité d'utilisation avec la capacité à produire des comportements stratégiques performants. Ce modèle utilise les principes de hiérarchie et de parallélisme pour être facilement compréhensible et utilisable. Les agents sont manipulés à la manière de ressources, en utilisant des niveaux de priorité pour les répartir sur différentes tâches, rendant le modèle adaptable pour des ressources aux quantités variables. Une stratégie peut être facilement modifiée et ses sous-comportements peuvent facilement être extraits et réutilisés. Pour la suite il est nécessaire de prévoir une phase de test en faisant appel à des game designers afin d'évaluer la simplicité de compréhension et d'utilisation du modèle ainsi que son expressivité, de façon à étendre celle-ci en ajoutant des fonctionnalités répondant à leurs besoins tout en gardant l'utilisabilité. Ce modèle fait partie d'un projet plus vaste qui a pour ambition de fournir un générateur de stratégies, autrement dit un système qui proposerait au game designer une première stratégie à partir de laquelle travailler, et ce simplement en renseignant les mécanismes du jeu.

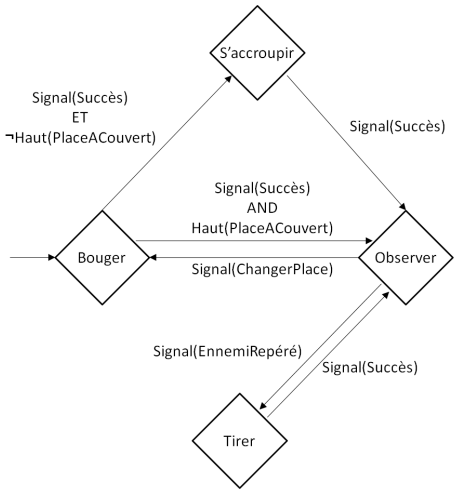


FIGURE 6 – Comportement logique *Couvrir*.

Références

[1] M. Buro. Real-time strategy games : A new AI research challenge. In *IJCAI*, pages 1534–1535, 2003.

[2] D. Churchill and M. Buro. Build order optimization in StarCraft. In *AIIDE*, 2011.

[3] E. W. Dereszynski, J. Hostetler, A. Fern, T. G. Dietterich, T.-T. Hoang, and M. Udarbe. Learning probabilistic behavior models in real-time strategy games. In *AIIDE*, 2011.

[4] R. Evans. Varieties of learning. *AI Game Programming Wisdom*, 2, 2002.

[5] S. Ontanon, K. Mishra, N. Sugandh, and A. Ram. Case-based planning and execution for real-time strategy games. In *Case-Based Reasoning Research and Development*, pages 164–178. Springer, 2007.

[6] S. Ontanon, G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill, and M. Preuss. A survey of real-time strategy game AI research and competition in StarCraft. *Computational Intelligence and AI in Games, IEEE Transactions on*, 5(4) :293–311, 2013.

[7] J. Orkin. Three states and a plan : the ai of fear. In *Game Developers Conference*, volume 2006, page 4, 2006.

[8] R. Palma, P. A. Gonzalez-Calero, M. A. Gomez-Martin, and P. P. Gomez-Martin. Extending case-based planning with behavior trees. In *FLAIRS Conference*, 2011.

Réparation de plans dans les HTNs réactifs en utilisant la planification symbolique

Lydia Ould Ouali[1] Charles Rich[2] Nicolas Sabouret[1]

[1] LIMSI-CNRS, UPR 3251, Orsay, France & Univ. Paris-Sud, Orsay, France
[2] Worcester Polytechnic Institute, Worcester, MA, USA

Résumé

Construire des modèles formels pour planifier les prochaines actions d'un agent est un problème crucial en Intelligence Artificielle. Dans ce travail, nous proposons de combiner deux approches connues, à savoir les réseaux de tâches hiérarchiques (HTNs) et la planification symbolique linéaire. La motivation principale de cette approche hybride est de d'assurer la réparation des cassures durant l'exécution du HTN en invoquant dynamiquement un planificateur symbolique. Ce travail nous a aussi permis de mettre en exergue les problèmes liés à la combinaison de la modélisation procédurale et symbolique. Nous avons implémenté notre approche qui combine une HTN réactif appelé Disco et le planificateur STRIPS implémenté en Prolog, et nous avons conduit des évaluations préliminaires.

Mots Clef

HTN réactif, planification symbolique, réparation de cassures.

Abstract

Building formal models of the world and using them to plan future action is a central problem in artificial intelligence. In this work, we combine two well-known approaches to this problem, namely, reactive hierarchical task networks (HTNs) and symbolic linear planning. The practical motivation for this hybrid approach was to recover from breakdowns in HTN execution by dynamically invoking symbolic planning. This work also reflects, however, on the deeper issue of tradeoffs between procedural and symbolic modeling. We have implemented our approach in a system that combines a reactive HTN engine, called Disco, with a STRIPS planner implemented in Prolog, and conducted a preliminary evaluation.

Keywords

Reactive HTN, symbolic linear planning, breakdowns, recovery.

1 Introduction

Les réseaux de tâches hiérarchiques ou HTN [4] (Hierarchical task networks) sont largement utilisés pour contrôler des agents et des robots évoluant dans des environnements dynamiques. Il existe dans la littérature différentes formalisations et représentations graphiques. Nous utiliserons ici une simple représentation en arbre, avec différents niveaux de nœuds *tâches* et des nœuds de *décompositions*. Un HTN est donc défini avec un nœud tâche au niveau de la racine. Les feuilles constituent des *tâches primitives*, correspondant aux actions de l'agent dans le monde. Les autres nœuds de l'arbre définissent soit des *tâches abstraites* à décomposer (c'est par exemple le cas de la racine de l'arbre), soit des *recettes* décrivant une décomposition possible en sous-tâches (elles-mêmes décomposables à leur tour, jusqu'à atteindre des tâches primitives).
Ces HTNs sont généralement codés à la main par un modélisateur qui définit la décomposition des tâches en recettes et en sous-tâches (éventuellement partiellement ordonnées). En plus de cette décomposition, la plupart des HTNs sont définis avec des conditions pour le contrôle de l'exécution. Ainsi, le modélisateur peut définir pour chaque recette des *conditions d'applicabilité* qui déterminent l'applicabilité de la recette dans l'état courant, et pour chaque tâche des préconditions et des post-conditions. Une tâche ne peut être effectuée que si ses préconditions sont valides et une tâche est considérée comme réussie si ses post-conditions sont satisfaites.
À l'origine, les HTN sont une extension hiérarchique des plans linéaires classiques (e.g. STRIPS [5]), et comme pour les modèles classiques, les conditions associées aux tâches sont dites *symboliques*, *i.e.* elles sont écrites en logique formelle qui permet à un moteur d'inférence de raisonner dessus. La principale difficulté, comme l'a souligné [8], réside dans la modélisation complète et fidèle d'une monde complexe à l'aide de prédicats. En réponse à ces difficultés liées à la modèlisation des connaissances symboliques, une variante appelée HTNs *réactifs* a été developpée dans laquelle les conditions sont *procédurales*, *i.e.* elles sont écrites dans un langage de programmation et évaluée

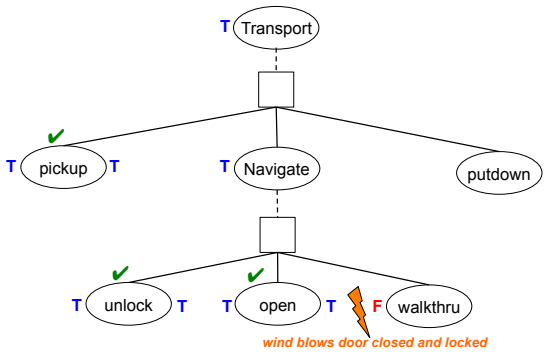


FIGURE 1 – Cassure dans l'exécution du HTN après que le vent a fait claquer la porte et l'a bloquée. Les coches indiquent les tâches dont l'exécution est terminée. "T" représente une condition dont l'évaluation a retourné *vrai* ; "F" représente une condition qui a retourné *faux*.

par un interpréteur de ce langage de programmation. Les HTNs réactifs sont utilisés par exemple dans le domaine des jeux vidéos sous le nom de *behaviour tree* [1]. Notre travail porte sur les HTNs réactifs et, en particulier, sur la réparation des cassures (ou *breakdowns*) qui peuvent survenir durant l'exécution. L'idée principale est d'ajouter une petite proportion de conditions symboliques au HTN réactif afin de permettre à un planificateur linéaire de produire des plans de réparation locaux.

2 Motivations

Afin de mieux introduire et expliquer notre travail, considérons un exemple simple mais intuitif d'une cassure dans l'exécution d'un HTN et sa réparation, illustré dans la FIGURE 1. Il s'agit d'implémenter en HTN le comportement d'un robot pour transporter un objet a travers une porte fermée. Dans le HTN, la tâche but *transport*, est décomposée en trois étapes : ramasser l'objet (*pickup*), manipuler la porte *(navigate)* et poser l'object *(putdown)*. La tâche *navigate* est à son tour décomposée en quatre étapes : déverrouiller la porte *(unlock)*, l'ouvrir *(open)* et franchir la porte *(walkthrough)*. Chacune de ces tâches est représentée par un ovale dans la FIGURE 1. Les boîtes représentent les différentes alternatives de décomposition qui peuvent être ignorées dans cet exemple. Au moment de l'exécution décrit dans la FIGURE 1, le robot a ramassé l'objet, déverrouillé et ouvert la porte avec succès. Cependant, avant que la précondition de la tâche *walkthrough* ne soit évaluée, le vent souffle et la porte de ferme et se verrouille (l'exemple ne vise pas à être réaliste mais à illustrer notre propos). La précondition de la tâche *walkthrough* évalue si la porte est ouverte et retourne faux. À ce moment, aucune tâche ne peut être exécutée

1. See http ://aigamedev.com/open/article/popular-behavior-tree-design

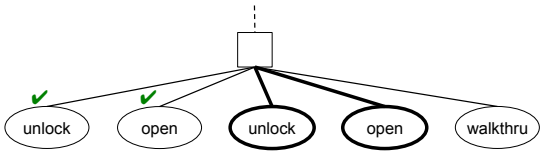
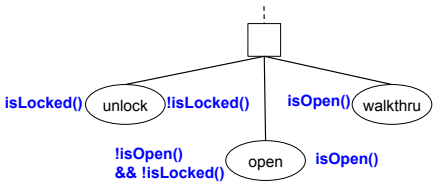


FIGURE 2 – Séquence de deux tâches primitives (en gras) ajoutées au plan hiérarchique de la tâche "navigate" pour réparer la cassure de la FIGURE 1.

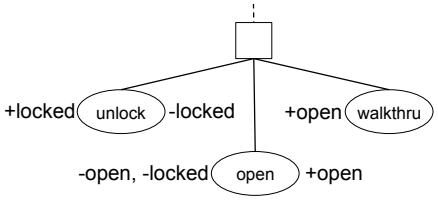
dans le HTN et c'est ce que nous appelons une *cassure*. Il n'est pas rare qu'un HTN réactif rencontre des cassures, spécialement dans le cas d'exécution dans des environnements dynamiques et complexes. Cependant, en analysant la cassure produite dans cet exemple, la solution montrée dans la FIGURE 2 parait évidente : déverrouiller la porte et l'ouvrir. Trouver une solution pour cette cassure est un problème trivial pour un planificateur linéaire symbolique comme STRIPS si les (pré/post)-conditions des tâches primitives correspondantes étaient spécifiées symboliquement.
Or justement, ce qui caractérise un HTN réactif, c'est que *les pré et post-conditions sont définies à l'aide de scripts* (elles sont définies en langage procédural et évaluées par un interpréteur approprié à ce langage) et *ne peuvent donc pas être manipulés comme des connaissances symboliques*. La FIGURE 3a montre les conditions procédurales définies dans le plan *Navigate* comme elles seraient typiquement écrites par exemple en JavaScript. Par exemple, la procédure "isOpen()" appelle un code spécifique dans le système de capteurs du robot pour évaluer si la porte est actuellement ouverte. À titre de comparaison, la FIGURE 3b montre la formalisation des mêmes tâches en STRIPS.
Supposons que lorsque la cassure a été détectée dans l'exemple de la FIGURE 1, le moteur d'exécution du HTN contenait la connaissance symbolique qui est fournie sur la FIGURE 3b. La réparation de la cassure pourrait donc être traitée comme une problème de planification STRIPS (voir FIGURE 4) dans lequel l'état initial est représenté par l'état courant du monde et l'état final est la précondition échouée de la tâche *walkthrough* (la porte est ouverte). Un simple chainage arrière peut rapidement trouver une séquence d'actions (à savoir déverrouiller et ouvrir la porte). Le plan produit est ensuite intégré au HTN comme dans la FIGURE 2 afin de permettre au HTN de reprendre son exécution. Le but de cet article est de généraliser la solution de cet exemple en algorithme de réparation de cassures qui soit indépendant des applications et auquel on associe une méthodologie de modélisation des HTNs réactifs.

3 Travaux connexes

La question que pourrait soulever la section précédente est pourquoi ne pas utiliser seulement la planification symbolique au lieu d'un HTN réactif et ap-



(a) Conditions procédurales



(b) Représentation symbolique des conditions

FIGURE 3 – Conditions procédurales versus symboliques pour le plan Navigate.

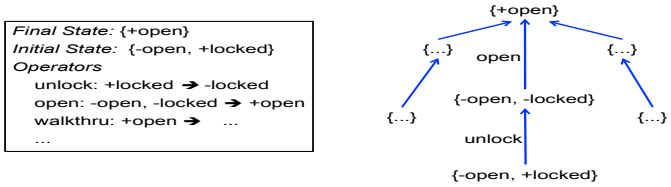


FIGURE 4 – Réparation de la cassure de la FIGURE 1 comme un problème de planification STRIPS.

pliquer les solutions existantes de réparation des cassures ? a problématique de la réparation des cassures durant l'exécution a déjà été étudiée dans le domaine des HTNs symboliques et de nombreux modèles ont été proposés[2, 11, 1, 12]. Ces techniques de réparation de plans sont généralement très efficaces.

La réponse à cette question réside dans le rapport entre la modélisation du problème et l'exécution du plan produit. En effet, les planificateurs symboliques basées sur des connaissances symboliques linéaires (en langage STRIPS ou PDDL [7]) ou hiérarchiques (à l'aide de HTNs) requièrent une description *complète* et *correcte* de toutes les tâches dans le domaine du problème. Si la description symbolique est incomplète ou incorrecte, les plans générés vont subir des cassures à l'exécution.

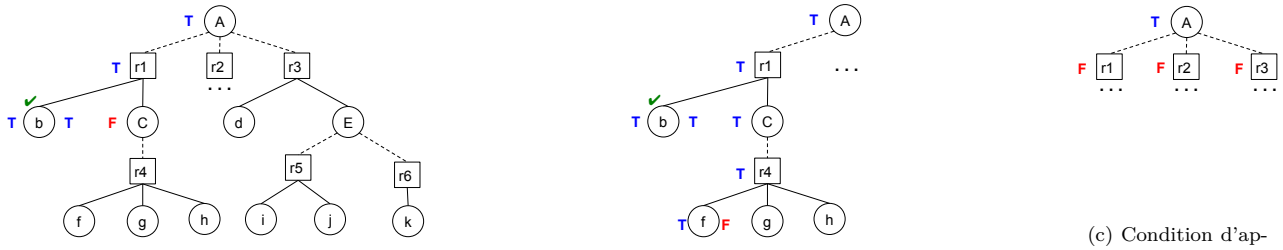
Malheureusement, la recherche en IA [8] a montré que la modélisation d'une description logique du monde réel qui soit complète et correcte peut se révéler extrêmement difficile voir impossible en pratique. C'est pour cela que les HTNs réactifs ont été inventés : la connaissance est insérée dans les HTNs réactifs principalement à deux endroits : dans la structure de décomposition de l'arbre et dans le code pour définir les conditions procédurales (en particulier dans les conditions d'applicabilité pour les choix de décompositions). L'intérêt des HTNs réactifs provient de la facilité de modélisation sur des problèmes dont le domaine de connaissance est complexe pour être modélisé de manière symbolique. De plus, les conditions procédurales ne sont évaluées que dans l'état *courant* du monde (alors que les descriptions symboliques doivent être vrais dans tous les mondes possibles). Ceci n'empêche pas les HTNs réactifs de rencontrer des cassures, comme dans notre exemple, et c'est ce qui nous a conduit à proposer une approche hybride dans laquelle un HTN réactif est augmenté avec de la connaissance symbolique afin d'aider à la réparation des cassures.
D'autre travaux avant nous ont proposé d'ajouter un module de raisonnement symbolique pour étendre les HTNs réactifs. Par exemple, Firby [6] propose d'ordonner les tâches dans la file d'exécution du HTN et de choisir d'autres alternatives de décompositions ce qui permet au HTN de détecter les situations problématiques avant qu'elles ne se produisent. Brom [3] propose d'utiliser la planification afin d'assurer l'exécution des tâches avec des contraintes de temps. Cependant, aucune de ces deux approches ne propose d'étudier la possibilité de réparer des HTN réactifs à l'aide de planification symbolique. Notre originalité est de proposer un algorithme hybride combinant le procédural et le symbolique.

4 L'approche hybride

Dans cette section, nous proposons une généralisation sur deux niveaux de l'exemple de la section 2 en considérant : (1) les différents types de cassures, (2) l'ensemble des états finaux possibles. Nous présenterons d'abord l'algorithme général de réparation de cassures et discuterons ensuite la méthodologie de modélisation associée.

4.1 Les HTNs réactifs

Les HTNs réactifs peuvent être représentés comme un arbre et/ou, où les tâches sont des "et", et les nœuds de décomposition sont des "ou". Contrairement aux HTNs symboliques, les HTNs réactifs ne visent pas à anticiper le futur et construire un plan complet afin de l'exécuter par la suite. Ils se content seulement de calculer la prochaine tâche a exécuter à partir de l'état courant du monde. L'exécution est en profondeur d'abord, avec un parcours de gauche à droite à partir du nœuds racine, durant lequel les conditions sont évaluées, comme décrit ci-après.
Si l'exécution du nœud courant est une tâche alors ses éventuelles préconditions représentés en procédures booléennes sont évaluées. Si elles retournent faux, alors l'exécution est interrompue *(cassure)* ; sinon l'exécution continue. Si la tâche est primitive, elle est directement exécutée pour changer l'état du monde. Sinon, si elle est abstraite alors les conditions d'applicabilités des nœuds fils (décompositions) sont évalués dans l'ordre jusqu'à l'obtention d'un nœud qui retourne vrai. L'exécution continue alors avec cette décomposition. Si toutes les conditions d'applicabilités sont



(a) Précondition échouée (b) post-condition échouée (c) Condition d'applicabilité échouée

FIGURE 5 – Exemples des trois types de cassures dans l'exécution d'un HTN.

fausses alors l'exécution est arrêtée *(cassure)*. Une fois l'exécution de la tâche terminée, ses éventuelles post-conditions sont évaluées, si elles retournent faux, alors l'exécution est interrompue *(cassure)*. Sinon l'exécution continue. Si le nœud courant est un nœud de décomposition, alors les nœuds fils (tâches) sont exécutées dans l'ordre.

La FIGURE 5 résume les trois types de cassures qu'un HTN peut rencontrer durant son exécution. La cassure dans l'exemple de la section 2 est causé par une précondition échouée. Cependant, cette taxonomie n'est pas apte à distinguer les raisons sous-jacentes qui peuvent provoquer des cassures. En effet, une cassure peut être causée par un changement inattendu dans le monde (*e.g.* le vent dans la section 2) ou à cause d'une erreur de programmation (condition mal codée, structure d'arbre erronée). L'impossibilité de détecter les causes des cassures est une limitation intrinsèque des HTN réactifs.

4.2 Algorithme de réparation de plans

La généralisation la plus significative de l'algorithme sur l'exemple de la section 2, concerne le choix de l'état final pour le planificateur linéaire. En effet, pour que l'algorithme de réparation fonctionne il a besoin d'avoir en entrée un état but à atteindre. Cependant, pour ce même exemple, différents état buts peuvent être trouvés. Par exemple, supposons que *walkthrough* ait une post-condition qui spécifie que le robot est dans la pièce de l'autre côté de la porte. Après la cassure, le robot peut trouver une autre méthode de réparation : trouver un plan pour satisfaire cette post-condition (par exemple, sortir de la pièce actuelle par une autre porte...). Cette procédure consiste à définir la post-condition comme un *état but* candidat pour la réparation. De la même façon, supposons que *walkthrough* n'a pas de représentation symbolique pour sa post-condition, mais que la post-condition symbolique de *navigate* spécifie l'emplacement but du robot. Dans ce cas la post-condition de *navigate* est considérée comme un état but de réparation. De manière similaire, si on suppose que la cassure dans notre exemple a été provoquée par l'échec de la post-condition de *unlock*, la représentation symbolique de la précondition de *walkthrough* ainsi que celles des post-conditions de *walkthrough* et de *navigate* sont de bons candidats pour la réparation.

Sur la base de ce raisonnement, l'algorithme que nous présentons ci-dessous repose sur l'idée de construire les états but candidats pour la procédure de réparation à partir de l'ensemble des conditions du HTN affectées par la cassures (dont l'évaluation retourne faux) et qui n'ont pas encore été validées durant l'exécution. Nous pensons que cette première approximation est trop grossière (trop de candidats sont considérés) mais nous avons besoin d'étudier notre approche sur des exemples plus concrêts pour concevoir une meilleure heuristique de recherche. Il est a noter qu'une condition d'applicabilité est considérée comme candidat dans l'unique cas où toutes ses sœurs dans l'arbre ont été évaluées comme fausses.

La FIGURE 6 présente le pseudo-code du système hybride ainsi conçu. La procédure principale, EXECUTE, execute un HTN jusqu'à ce qu'il se termine (succès) ou jusqu'à ce qu'une cassure soit détectée. Le code de réparation des cassures commence à la ligne 5. La sous procédure, FINDCANDIDATES va récursivement parcourir le HTN afin de calculer les candidats cibles pour la procédure de réparation. La procédure SYMBOLICPLANNER n'est pas décrite car n'importe quel planificateur linéaire peut être utilisé. De plus, comme l'ensemble des opérateurs symboliques ne change pas durant l'exécution, il n'est pas défini comme un argument explicite du planificateur symbolique. On peut voir à ligne 6 que notre approche requiert une méthode pour calculer, à partir de l'état courant, l'état initial du planificateur symbolique dans un formalisme symbolique que le planificateur puisse exploiter. Par exemple, pour l'exemple de la section 2, chaque fonction symbolique comme "open" doit être associée à une procédure comme "isOpen()" donnant sa valeur dans l'état courant. C'est là que réside la difficulté de la modélisation hybride (discutée dans la section 4.3). La ligne 8 montre que les candidats sont triés par rapport à leurs distance du nœud courant dans l'arbre, en utilisant une simple métrique (par exemple le plus court chemin dans un arbre non-orienté). Notre but est ainsi de favoriser des réparations locales qui préservent au maximum la structure originale du HTN. Cependant, nous n'avons pas encore testé les perfor-

```
1: procedure EXECUTE(htn)
2:     while htn is not completed do
3:         current ← next executable node in htn
4:         if current ≠ null then execute current
5:         else [breakdown occurred]
6:             initial ← symbolic description of current world state
7:             candidates ← FindCandidates(htn)
8:             sort candidates by distance from current
9:             for final ∈ candidates do
10:                plan← SymbolicPlanner(initial,final)
11:                if plan ≠ null then
12:                    splice plan into htn between current and final
13:                    continue while loop above
14:            Recovery failed !

15: procedure FINDCANDIDATES(task)
16:    conditions ← ∅
17:    pre ← symbolic precondition of task
18:    if pre ≠ null ∧ procedural prec of task has not evaluated to true then
19:        add pre to conditions
20:    post ← symbolic post-condition of task
21:    if post ≠ null ∧ procedural postc of task has not evaluated to true then
22:        add post to conditions
23:    applicables ← ∅
24:    allFalse ← true
25:    for decomp ∈ children of task do
26:        for task ∈ children of decomp do FindCandidates(task)
27:        if allFalse then
28:            if procedural appl condition of decomp has evaluated to false then
29:                app ← symbolic applicability condition of decomp
30:                if app ≠ null then add app to applicables
31:            else allFalse ← false
32:    if allFalse then add applicables to conditions
33:    return conditions
```

FIGURE 6 – Pseudocode de l'exécution et réparation du système HTN réactif hybride

mances de cette heuristique. Enfin, à la ligne 12 de la procédure EXECUTE, quand un plan est trouvé, il doit être intégré dans le HTN pour être exécuté. Une fois le plan de réparation exécuté, le HTN récupéré la tâche de la condition réparée pour poursuivre son l'exécution. Dans l'exemple de la figure FIGURE 4 la condition réparée est "isOpen()". Par conséquent, la prochaine tâche à exécuter après le plan de réparation est la tâche où la condition est utilisée, à savoir *walkthrough*.

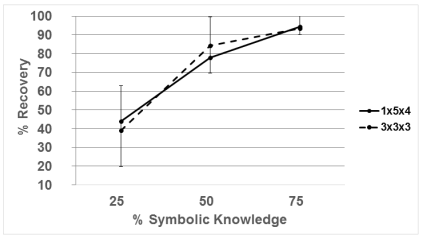
4.3 Méthodologie de modélisation

Le système hybride prend avantage de la connaissance symbolique construite par l'auteur du HTN, sachant que toute condition dans le HTN peut, ou non, avoir une représentation symbolique. Comme nous l'avons annoncé précédemment, la difficulté liée à la réalisation d'une modélisation symbolique a conduit les auteurs des HTNs à utiliser la modélisation procédurale. Il existe donc deux problèmes méthodologiques liées à la conception d'un HTN hybride. D'une part, il faut déterminer où il est préférable d'investir l'effort nécessaire de modélisation symbolique. D'une autre part, il faut définir une stratégie qui facilite à l'auteur le processus de mixage entre le procédural et le symbolique. Notre intuition première est qu'il faut réduire la modélisation symbolique aux tâches primitives, car nous espérons que l'utilisation d'une stratégie locale pour la réparation des cassures sera plus efficace. Le choix des opérateurs à introduire au planificateur linéaire a une implication directe sur ses performances. En effet, seules les tâches primitives disposant d'une représentation symbolique de ses pré et post-conditions peuvent être incluses dans l'ensemble des opérateurs. Cependant, si une tâche abstraite dispose de ces caractéristiques, elle peut alors être utilisée pour la réparation des cassures, en utilisant durant son exécution une de ses décompositions valables dans l'état courant.
Nous envisageons de développer un outil qui aiderait à simplifier la modélisation hybride en détectant les liens entre les deux modélisations comme dans la FIGURE 3a et générer automatiquement la représentation symbolique comme dans la FIGURE 3b. L'outil pourrait aussi garder l'historique des cassures et les utiliser pour guider l'auteur dans l'ajout de connaissances symboliques dans le HTN.

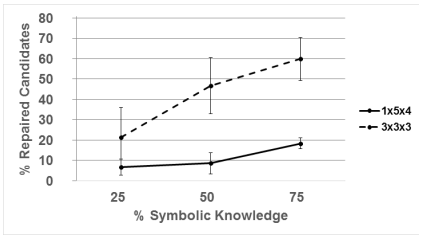
5 Implémentation et évaluation

Nous avons implémenté notre système hybride en Java en utilisant le standard ANSI/CEA-2012 [9] pour la modélisation du HTN et Disco [10] comme le moteur d'exécution de ce HTN. Concernant le planificateur symbolique, nous avons utilisé une simple implémentation de STRIPS exécutée dans un environnement Java de Prolog [2]. L'utilisation de Prolog facilite l'ajout de règles de raisonnement symboliques durant le processus de planification.
Une véritable évaluation de notre approche consisterait à faire évoluer plusieurs agents dans des environnements dynamiques du monde réel et évaluer leurs performances ainsi que leur robustesse face aux cassures. En attendant, nous avons validé notre système avec une expérimentation préliminaire sur des HTNs générés synthétiquement en variant le niveau de connaissance symbolique. Nous sommes partis d'une théorie simple qui préconise que plus il y'a des connaissances symboliques dans le HTN, meilleures seront ses performances pour la réparation des cassures.

2. See http ://tuprolog.apice.unibo.it



(a) Ration des réparation/cassure



(b) Proportion des candidats réparés

FIGURE 7 – Résultats des expérimentations sur des HTNs générés synthétiquement avec différents niveaux de connaissances symbolique.

La FIGURE 7 montre les résultats obtenus qui confirment notre hypothèse. Nos HTN synthétiques sont caractérisés par trois dimensions $R \times S \times D$, où R est le facteur de ramification d'une décomposition (recipe), S le facteur de ramification d'une tâche, D étant la profondeur du HTN. Nous avons testé deux configurations de HTNs de dimensions respectivement $3 \times 3 \times 3$ et $1 \times 5 \times 4$. Pour chaque test, nous avons aléatoirement extrait des combinaisons possibles de connaissance symbolique sur trois différent niveaux ; 25% , 50%, 75% (le pourcentage des conditions qui ont une représentation symbolique). Nous nous sommes restreints à ces modélisations pour des questions de temps d'exécution. La FIGURE 7a montre l'évolution des performances de réparation des cassures en fonction de l'augmentation du niveaux de connaissance symbolique. Dans la FIGURE 7b, nous nous sommes intéressés à la proportion des problèmes de planification soumis au planificateur (buts candidats) qui ont été correctement résolus. Nous pouvons remarquer que cette dernière augmente aussi en fonction du niveaux de connaissance symbolique (pour ces tests, nous avons modifié notre système pour qu'il génère un plan pour tous les candidats).

6 Conclusion

Nous avons présenté dans cet article une première contribution pour la réparation des cassure qui consiste à augmenter un HTN réactif avec un planificateur linéaire symbolique qui propose des plans de réparation. Nous avons ensuite soulevé la question sous-jacente à la modélisation hybride de notre système et nous avons expliqué le compromis existant entre la modélisation symbolique et la modélisation procédurale. Une implémentation en Java a été proposé qui combine un HTN réactif appelé Disco et le planificateur STRIPS implémenté en Prolog. Des tests sur des HTNs synthétiques ont été menés pour valider le système et les résultats obtenus confirment nos théories de départ. Notre proposition reste préliminaire, bien qu'elle est implémentée et testée sur des données générées synthétiquement, car la question de ses performances en pratique reste une question ouverte. Nous poursuivons actuellement ce travail dans une thèse dont l'objectif est d'intégrer notre modèle dans un système de dialogue social afin de gérer de manière opportuniste les éventuelles cassures.

Références

[1] Ayan et al. Hotride : Hierarchical ordered task replanning in dynamic environments. In *Planning and Plan Execution for Real-World Systems–Principles and Practices for Planning in Execution : Papers from the ICAPS Workshop. Providence, RI*, 2007.

[2] G. Boella and R. Damiano. A replanning algorithm for a reactive agent architecture. In *Artificial Intelligence : Methodology, Systems, and Applications*, pages 183–192. Springer, 2002.

[3] C. Brom. Hierarchical reactive planning : Where is its limit. *Proceedings of MNAS : Modelling Natural Action Selection. Edinburgh, Scotland*, 2005.

[4] K. Erol, J. Hendler, and D. S. Nau. Htn planning : Complexity and expressivity. In *AAAI*, volume 94, pages 1123–1128, 1994.

[5] R. E. Fikes and N. J. Nilsson. Strips : A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3) :189–208, 1972.

[6] R. J. Firby. An investigation into reactive planning in complex domains. In *AAAI*, volume 87, pages 202–206, 1987.

[7] D. E. Ghallab, Malik et al. Pddl-the planning domain definition language. 1998.

[8] Y. Gil. Acquiring domain knowledge for planning by experimentation. Technical report, DTIC Document, 1992.

[9] C. Rich. Building task-based user interfaces with ansi/cea-2018. *Computer*, 42(8) :20–27, 2009.

[10] C. Rich and C. L. Sidner. Using collaborative discourse theory to partially automate dialogue tree authoring. In *Intelligent Virtual Agents*, pages 327–340. Springer, 2012.

[11] R. Van Der Krogt and M. De Weerdt. Plan repair as an extension of planning. In *ICAPS*, volume 5, pages 161–170, 2005.

[12] I. Warfield, C. Hogg, S. Lee-Urban, and H. Munoz-Avila. Adaptation of hierarchical task network plans. In *FLAIRS Conference*, pages 429–434, 2007.

Représentation symbolique de séries temporelles cycliques basée sur les propriétés des cycles : application à la biomécanique

Vanel Steve Siyou Fotso[1] Engelbert Mephu-Nguifo[1] Philippe Vaslin[1]

[1] Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes (LIMOS)

Clermont Université, Université Blaise Pascal, LIMOS,
BP 10448, F-63000 CLERMONT-FERRAND CNRS, UMR 6158, LIMOS, F-63173 AUBIERE
siyou@isima.fr, mephu@isima.fr, vaslin@isima.fr

Résumé

L'analyse des séries temporelles cycliques issues de la biomécanique repose sur la comparaison des propriétés de leurs cycles. Les algorithmes usuels de classification de séries temporelles ignorant cette particularité, nous proposons une représentation symbolique de séries temporelles cycliques basée sur les propriétés de leurs cycles et nommée SAX-P. Les chaînes de caractères ainsi obtenues peuvent ensuite être comparées à l'aide de la distance Dynamic Time Warping. L'application de SAX-P aux moments propulsifs de trois sujets (S1, S2, S3) se déplaçant en Fauteuil Roulant Manuel a permis de mettre en évidence la dissymétrie de leur propulsion en ligne droite. La représentation symbolique SAX-P facilite la lecture des séries temporelles cycliques et l'interprétation des résultats de leur classification par les cliniciens.

Mots Clef

Séries temporelles cycliques, Représentation symbolique, SAX, Biomécanique, Fauteuil Roulant Manuel.

Abstract

The analysis of cyclic time series from biomechanics is based on the comparison of the properties of their cycles. As usual algorithms of time series classification ignore this particularity, we propose a symbolic representation of cyclic time series based on the properties of cycles, named SAX-P. The resulting character strings can be compared using the Dynamic Time Warping distance. The application of SAX-P to propulsive moments of three subjects (S1, S2, S3) moving in Manual Wheelchair highlight the asymmetry of their propulsion. The symbolic representation SAX-P facilitates the reading of the cyclic time series and the clinical interpretation of the classification results.

Keywords

cyclic time series, symbolic representation, SAX, Biomechanics, Manual Wheelchair

1 Introduction

D'une manière générale, pour se déplacer, l'être humain effectue des mouvements cycliques (ex : marche, course, natation, cyclisme). L'analyse biomécanique de ces mouvements est réalisée avec différents instruments de mesure (ex : capteurs de force et d'accélération, systèmes d'analyse cinématique) qui permettent l'enregistrement en continu, sur de longues périodes, de nombreuses grandeurs mécaniques (cinématiques et dynamiques). Ces enregistrements produisent de longues séries temporelles composées de nombreux cycles, ou patrons, représentatifs des mouvements réalisés et des efforts produits par l'individu au cours de ses déplacements (Figure 2).

Ces cycles constituent les unités d'analyse des séries temporelles et possèdent plusieurs propriétés caractéristiques telles que : la valeur minimale, la surface sous le cycle, la durée du cycle [1]. Plusieurs études antérieures ont comparé des séries temporelles en les décomposant en petits segments puis en comparant les propriétés de ces segments.Un segment d'une série temporelle est une suite de valeurs consécutives appartenant à celle-ci [2].

Keogh et Chakrabarti [3] ont proposé de remplacer chaque segment $x_{\frac{n}{N}(i-1)+1}, x_{\frac{n}{N}(i-1)+2}, \cdots, x_{\frac{n}{N}i}$ d'une série temporelle $X = x_1, x_2, \cdots, x_n$ par la moyenne $\bar{x}_i$ de ses valeurs ($\bar{x}_i = \frac{N}{n} \sum_{j=\frac{n}{N}(i-1)+1}^{(\frac{n}{N})i} x_j$) transformant ainsi la série temporelle, qui est une suite de valeurs, en la suite des moyennes de ses N segments $\bar{X} = \bar{x}_1, \bar{x}_2, \cdots, \bar{x}_N$. Cette méthode, connue sous le nom Piecewise Aggragate Approximation (PAA), avait pour objectif premier de réduire la taille des séries temporelles. Cependant, elle permet également de comparer deux séries temporelles C et Q en calculant la distance DR entre les suites $\bar{C}$ et $\bar{Q}$ des moyennes leurs segments (équation 1).

$$DR(\bar{C}, \bar{Q}) = \sqrt{\frac{n}{N} \sum_{i=1}^{N} (\bar{c}_i - \bar{q}_i)^2} \qquad (1)$$

Lin et al. [4] se sont basés sur la méthode PAA pour pro-

poser une représentation symbolique des séries temporelles baptisée Symbolic Aggregate approXimation (SAX). L'objectif de SAX est d'assigner de manière équiprobable une lettre à chaque segment. Pour ce faire, le domaine des valeurs de la série temporelle est divisé en intervalles de sorte que chaque point de la série temporelle ait approximativement la même probabilité d'appartenir à un intervalle et une lettre est associée à chacun de ces intervalles. Ensuite, chaque segment de la série temporelle est associé à la lettre de l'intervalle auquel appartient sa moyenne. Avec SAX, la différence entre deux chaînes de caractères est calculée par la distance entre les bornes des intervalles que représente chaque caractère de la chaîne (équation 2).

$$MINDIST(\hat{Q},\hat{C}) = \sqrt{\frac{n}{N}\sum_{i=1}^{N}(dist(\hat{q}_i,\hat{c}_i))^2} \qquad (2)$$

$\hat{q}_i \, et \, \hat{c}_i$ sont des caractères et $dist()$ est la distance entre les bornes des intervalles que représentent ces caractères [4].
Cependant, la moyenne seule n'est pas suffisante pour associer un segment à une lettre. En effet, des segments peuvent avoir la même moyenne et être associés à la même lettre alors qu'ils ont des allures très différentes.
Pour y remédier, Lkhagva et Kawagoe [5] ont proposé le modèle ESAX qui prend en compte trois propriétés de chaque segment : sa moyenne, son minimum et son maximum. Par la suite, Sun et al. [6] ont proposé le modèle SAX-TD qui prend en compte deux propriétés pour chaque segment : la moyenne et la tendance. Ils adaptent ensuite la distance utilisée par la méthode SAX pour qu'elle prenne en compte la tendance.
Ces deux dernières méthodes fournissent de meilleurs résultats que la méthode SAX [6]. Elles ont toutefois l'inconvénient d'augmenter le nombre de symboles nécessaires à la représentation des séries temporelles. En effet, la méthode ESAX triple la taille de la représentation d'une série temporelle fournie par la méthode SAX, tandis que la méthode SAX-TD la double. De plus, les quatre méthodes précédentes ont deux inconvénients majeurs : elles considèrent des segments de taille fixe, alors que les cycles sont des segments de taille variable, et elles ne prennent pas en compte les propriétés caractéristiques des cycles telles que la durée et la surface sous un cycle (Figure 2). Notre objectif est de proposer une représentation symbolique qui prend en compte plusieurs propriétés pour chaque cycle, mais sans accroître le nombre de symboles utilisés pour la représentation symbolique.
Ces nouvelles représentations symboliques permettrons en outre l'utilisation d'un grand nombre d'algorithmes d'analyse de séquences tels que les algorithmes de détection d'anomalie (rechercher une sous-séquence inhabituelle dans une séquence), les algorithmes de découverte de motifs (chercher des sous-séquences qui se répètent)[7], les algorithmes de classification supervisée ou non supervisée, et aussi des algorithmes provenant de l'analyse de texte ou de la communauté de bio-informatique [8] [9].

2 SAX-P : Nouvelle représentation symbolique de séries temporelles cyclique

Définition 1 : Une série temporelle $TS = t_{s0}, t_{s1}, \cdots, t_{sm}$ est un ensemble ordonné de variables à valeurs réelles où t_{sm} est la valeur la plus récente.

Définition 2 : Une sous-séquence d'une série temporelle TS de taille m est un segment $T_{si,l}$ de taille $l < m$ constitué de variables consécutives commençant à la position i. Formellement, $T_{si,l} = t_{si}, t_{si+1}, ..., t_{si+l-1}$

Problème : Étant donnée une série temporelle cyclique T_S et une fonction F qui la divise en k segments consécutifs et disjoints (ici les segments sont des cycles) telle que $F(T_S) = T_{s0,l1}, T_{sl1,l2}, \cdots, T_{sm-lk,lk}$. Chaque segment $T_{si,li}$ est caractérisé par n propriétés dont l'ensemble est $P(T_{si,li})$. Notre problème est de comparer deux séries temporelles T_S et T_{S0} en se basant sur les propriétés des segments de $F(T_S)$ et de $F(T_{S0})$.
Le principe de SAX-P est le suivant (Tableau 1) :

1. **Découper la série temporelle en segments correspondant aux cycles successifs :** Pour cela, nous définissons un seuil tel que lorsqu'une valeur de la série temporelle est inférieure ou égale au seuil, alors nous revenons en arrière pour trouver un point proche de zéro correspondant à l'instant de début du cycle. Un cycle est constitué de tous les points situés entre deux débuts de cycles consécutifs ou de tous les points situés après le dernier début de cycle.
2. **Calculer un ensemble de propriétés pour chaque segment.** Pour l'analyse des séries temporelles cycliques, nous calculons les propriétés suivantes : la moyenne des valeurs et leur écart type, la valeur minimale, la durée du cycle et la surface sous un cycle. D'autres propriétés pourront toutefois être ajoutées en fonction de l'application visée.
3. **Regrouper dans les mêmes classes les segments ayant des propriétés similaires.** Comme les propriétés ont des unités différentes, le vecteur des propriétés associé à chaque cycle de propulsion a été centré et réduit. A partir de ce vecteur, les cycles de propulsion ont été regroupés en classes grâce à un algorithme de classification non supervisée. Nous avons choisi l'algorithme k-moyennes car il peut traiter efficacement de grands jeux de données [10].
4. **Remplacer chaque cycle de propulsion par la lettre de la classe à laquelle il appartient.** Nous avons ensuite associé une lettre à chaque classe ainsi obtenue et remplacé chaque cycle par la lettre de la classe à laquelle il appartient. Cette dernière étape nous a permis d'obtenir une chaîne de caractères représentant une série temporelle.

La figure 1 illustre le fonctionnement de SAX-P.

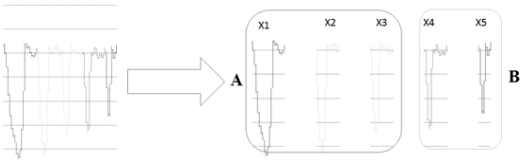


FIGURE 1 – Une série temporelle cyclique est segmentée en 5 cycles de propulsion (X1, X2, X3, X4, X5). Pour chaque cycle, un ensemble de propriétés a été calculé et les cycles ayant des propriétés similaires ont été regroupés dans une même classe (A ou B). La série temporelle initiale est ainsi transformée en une chaîne de caractères (ici : AAABB).

Algorithme 1. SAX-P(D, S, K)

Données :
D : ensemble de séries temporelles cycliques
S : Seuil de détection des cycles
K : Le nombre de lettres ou de classes à former
Résultats : CHAINES ; Représentations symboliques des séries temporelles cycliques de D

1. cycles = F(D, S) ;
2. proprietesCycles = centrerReduire(P(cycles)) ;
3. classes = kMoyennes(K, proprietesCycles) ;
4. CHAINES = representationSymbolique(cycles, classes) ;
5. retourner CHAINES ;

TABLE 1 – Pseudo-code de SAX-P

Après avoir appliqué la méthode SAX-P à des séries temporelles cycliques, nous avons obtenu des représentations symboliques qu'il restait à comparer. La comparaison de deux chaînes de caractères consiste à comparer les caractères qui les constituent, ce qui revient à comparer les classes entre elles. Cette comparaison est effectuée à l'aide de la distance euclidienne entre les vecteurs moyens des propriétés de chaque classe. La comparaison entre les chaînes de caractères est faite à partir d'une distance basée sur la déformation temporelle dynamique, ou Dynamic Time Warping (DTW) [11]. En effet, l'utilisation de DTW associée à l'algorithme des K plus proches voisins est une méthode bien connue et efficace de classification de séries temporelles [12] [13].

3 Application à la locomotion en Fauteuil Roulant Manuel

Le modèle SAX-P a été appliqué aux séries temporelles obtenues par l'enregistrement de moments axiaux exercés sur les roues arrière droite (RD) et gauche (RG) d'un Fauteuil Roulant Manuel (FRM) instrumenté par trois sujets (S1, S2, S3) ayant effectué chacun cinq déplacements rectilignes. Dans un premier temps, les séries temporelles ont été segmentées en cycles de propulsion. Puis, pour chaque cycle de propulsion, neuf propriétés ont été calculées :

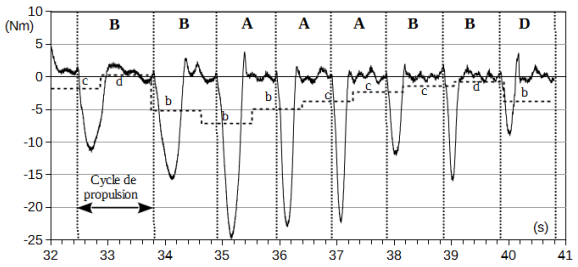


FIGURE 2 – Exemple de série temporelle cyclique : Moment propulsif appliqué par un sujet à la roue arrière d'un Fauteuil Roulant Manuel pendant un déplacement rectiligne. Dans ce cas, le patron est appelé cycle de propulsion. Les barres verticales délimitent les cycles de propulsion (segments) identifiés pendant les déplacements, et les lettres majuscules au-dessus de chaque cycle indiquent les classes auxquelles ils appartiennent. L'application de SAX à la série temporelle la divise en dix intervalles (lettres minuscules) de taille égale, mais sans prendre en compte les cycles de propulsion, l'aire sous le cycle, le temps de cycle et le temps de poussée ne sont pas pris en compte par SAX.

— Des propriétés directement utiles à l'analyse : les valeurs minimale et maximale du moment propulsif, la durée d'un cycle, la durée d'une poussée, la surface sous un cycle ;
— Des propriétés statistiques : la moyenne, la médiane, l'écart type, l'écart interquartile. Puis nous avons regroupé les cycles de propulsion dans 5 classes (A, B, C, D, E), dont les vecteurs moyens des propriétés sont présentés dans le Tableau 2.

En remplaçant chaque cycle de propulsion par la lettre de la classe à laquelle il appartient (Figure 2), on obtient les chaînes de caractères présentées dans le Tableau 3.

L'étude biomécanique des mouvements humains au cours d'activités telles que la marche, la course, le cyclisme ou encore la locomotion en FRM produit un grand nombre de séries temporelles cycliques qui restent encore généralement inexploitables par les personnes directement concernées en priorité, à savoir les pratiquants et les patients eux-mêmes, mais aussi et surtout les médecins spécialistes en rééducation fonctionnelle et les kinésithérapeutes. Il est donc apparu nécessaire de transformer ces multiples signaux afin de les représenter, sans les dénaturer, sous une forme simplifiée facilitant leur analyse et leur interprétation.

L'analyse des séries temporelles est un domaine de recherche où ont été développées de nombreuses méthodes répondant à cet objectif, telles que : PAA, SAX, ESAX et SAX-TD. Cependant, ces méthodes ne prennent en compte qu'un nombre limité des propriétés des segments qui composent les séries temporelles analysées (moyenne, minimum, maximum, tendance). En outre, ces méthodes sont basées sur des segments de longueur fixe : elles ne sont donc pas adaptées aux séries temporelles cycliques pro-

Classes	A	B	C	D	E
Nb de cycles	18	36	59	18	104
Temps cycle (s)	1.2	1.0	1.0	1.7	0.8
Temps poussée (s)	0.6	0.3	0.4	1.0	0.3
Mz Min (Nm)	-22.3	-17.4	-11.4	-8.7	-6.4
Mz Max (Nm)	0.1	0.1	0.1	0.7	0.1
Moyenne (Nm)	13.6	-8.1	-6.2	-3.0	-3.3
Médianne (Nm)	-16.1	-10.8	-7.4	-4.2	-3.9
IRQ (Nm)	12.4	10.7	6.0	4.3	3.1
Ecart-Type (Nm)	7.1	5.6	3.4	2.6	1.8
Aire (Nm.s)	-7.1	-2.3	-2.2	-1.8	-1.0

TABLE 2 – Vecteurs moyens des propriétés des classes (A, B, C, D, E) utilisées pour la représentation symbolique des moments propulsifs (Mz). SAX-P prend en compte le temps de poussée et le temps de cycle, la surface sous la poussée.

Suject	S1		S2		S3	
Poussée	Drte	Gche	Drte	Gche	Drte	Gche
1	C	A	C	D	E	D
2	B	B	E	E	D	E
3	B	B	C	E	C	E
4	B	B	C	E	E	E
5	B	B	C	D	E	E
6	C	B	E	C		D
7	B	C	E	E		E
8	E		E	E		
9			C	C		
10			E	E		
11			C	E		
12			C	E		
13				E		
DTW	268		354		44	

TABLE 3 – Chaînes de caractères obtenues par la méthode SAX-P appliquée aux séries temporelles des moments propulsifs enregistrés par les capteurs des roues droite (RD) et gauche (RG) d'un fauteuil roulant manuel instrumenté lors de déplacements rectilignes effectués par trois sujets (S1, S2, S3). La distance DTW mesure l'alignement des séquences des roues droite et gauche d'un même sujet pendant un même déplacement : lorsqu'elle est non nulle, elle est une mesure de la dissymétrie de la propulsion du sujet concerné.

duites par l'étude biomécanique de la locomotion humaine et constituées de cycles successifs de durée variable. La méthode SAX-P, présentée ici, permet de segmenter une série temporelle cyclique en cycles successifs et prend en compte simultanément plusieurs propriétés d'un même cycle. Elle se traduit par la représentation symbolique d'une série temporelle cyclique sous la forme d'une chaîne de caractères qui, à la fois, réduit la série temporelle et facilite son interprétation.

Dans le cas de la locomotion en FRM, les séries temporelles cycliques ont été représentées sous la forme de chaînes de caractères qui ont ensuite pu être comparées à l'aide de la méthode DTW. Ces comparaisons ont permis de mettre en évidence que les trois sujets dont les déplacements ont été étudiés ici n'avaient pas un mode de propulsion symétrique (comparaison des cycles de propulsion droits et gauches) et qu'ils avaient des modes de propulsion différents les uns des autres (comparaisons interindividuelles). Les représentations symboliques ainsi obtenues ont un autre avantage ; elles permettent d'utiliser le grand nombre d'algorithmes de traitement de séquences disponibles dans la littérature pour analyser les séries temporelles.

Bien que l'application de SAX-P effectuée ici ne concernent que trois sujets, les résultats de cette analyse sont encourageants car ils démontrent à la fois la pertinence de la méthode proposée et les perspectives qu'elle offre dans le domaine clinique pour évaluer, par exemple, les progrès réalisés par un patient au cours de sa rééducation et tout au long de sa vie, mais aussi pour évaluer les méthodes de rééducation elles-mêmes.

4 Conclusion

Dans ce travail, nous avons proposé une méthode de représentation symbolique des séries temporelles cycliques nommée SAX-P. Cette méthode permet de représenter une série temporelle cyclique sous la forme d'une chaîne de caractères, chaque caractère représentant une classe définie par les propriétés des cycles de la série considérée. Les chaînes de caractères ainsi tenues ont ensuite été comparées à l'aide de la distance Dynamic Time Warping utilisée pour l'alignement des séquences. Le modèle SAX-P a été appliqué aux moments propulsifs mesurés pendant les déplacements en ligne droite de trois personnes en FRM. Les résultats préliminaires obtenus ont notamment permis de démontrer que ces sujets avaient des modes de propulsion différents et que les cycles de propulsion d'un même sujet n'étaient pas toujours symétriques. La suite de se travail va consister à valider le modèle SAX-P. Pour ce faire, nous envisageons de comparer les méthodes SAX, ESAX, SAX-TD et SAX-P sur une tâche de classification supervisée. Dans des travaux ultérieurs, nous nous intéresserons aux différents paramètres de la méthode SAX-P notemmant le choix de l'algorithme de classification des cycles de propulsion à utiliser.

Remerciements

Ce travail a été partiellement financé par le Labex IMOBS3.

Références

[1] R. J. K. Vegter, C. J. Lamoth, S. de Groot, D. H. E. J. Veeger, and L. H. V. van der Woude, "Inter-individual differences in the initial 80 minutes of motor learning of handrim wheelchair propulsion." *PloS one*, vol. 9, no. 2, p. e89729, Jan. 2014.

[2] J. Abonyi, B. Feil, S. Nemeth, and P. Arva, "Fuzzy clustering based segmentation of time-series," *Advances in Intelligent Data Analysis V*, pp. 275–285, 2003.

[3] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases," *Knowledge and Information Systems*, vol. 3, no. January, pp. 263–286, 2001.

[4] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery - DMKD '03*, p. 2, 2003.

[5] B. Lkhagva and K. Kawagoe, "New Time Series Data Representation ESAX for Financial Applications," in *22nd International Conference on Data Engineering Workshops (ICDEW'06)*. IEEE, Apr. 2006, pp. x115–x115.

[6] Y. Sun, J. Li, J. Liu, B. Sun, and C. Chow, "An improvement of symbolic aggregate approximation distance measure for time series," *Neurocomputing*, vol. 138, pp. 189–198, Aug. 2014.

[7] N. Begum and E. Keogh, "Rare time series motif discovery from unbounded streams," *Proceedings of the VLDB Endowment*, vol. 8, no. 2, pp. 149–160, Oct. 2014.

[8] J. Aach and G. M. Church, "Aligning gene expression time series with time warping algorithms," *Bioinformatics*, vol. 17, no. 6, pp. 495–508, Jun. 2001.

[9] P. Papapetrou, V. Athitsos, M. Potamias, G. Kollios, and D. Gunopulos, "Embedding-based subsequence matching in time-series databases," *ACM Transactions on Database Systems*, vol. 36, no. 3, pp. 1–39, Aug. 2011.

[10] P. Esling and C. Agon, "Time-series data mining," *ACM Computing Surveys (CSUR)*, vol. 45, no. 1, pp. 1–34, 2012.

[11] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, Feb. 1978.

[12] F. Petitjean, A. Ketterlin, and P. Gançarski, "A global averaging method for dynamic time warping, with applications to clustering," *Pattern Recognition*, vol. 44, no. 3, pp. 678–693, Mar. 2011.

[13] F. F. Petitjean, G. Forestier, G. I. Webb, A. E. Nicholson, Y. Chen, and E. Keogh, "Dynamic Time Warping Averaging of Time Series Allows Faster and More Accurate Classification," in *2014 IEEE International Conference on Data Mining*. IEEE, Dec. 2014, pp. 470–479.

[14] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data," *Proceedings of the VLDB Endowment*, vol. 1, no. 2, pp. 1542–1552, Aug. 2008.

[15] T. G. Dietterich, "Machine Learning for Sequential Data : A Review," *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pp. 16–30, Aug. 2002.

Modèle d'articulation entre les règles définissant un système d'assistance aLDEAS

Le Vinh Thai Blandine Ginon Stéphanie Jean-Daubias Marie Lefèvre Pierre-Antoine Champin

Université de Lyon, CNRS, France
Université Lyon 1, LIRIS, UMR5205, F-69622, France
Bât. Nautibus - 23-25 av. Pierre de Coubertin, 69 100 Villeurbanne - France
le-vinh.thai@liris.cnrs.fr

Résumé

Le projet AGATE a donné naissance au système SEPIA qui permet à un concepteur d'assistance de définir des systèmes d'assistance greffés dans des applications-cibles (Windows, Java, Web…), sous forme d'un ensemble de règles aLDEAS. L'articulation entre ces règles était jusqu'à maintenant exprimée implicitement dans le langage aLDEAS. L'étude de systèmes d'assistance existants montre que cette articulation entre les règles d'assistance peut prendre des formes variées. Nous proposons dans cet article d'une part un modèle d'articulation entre règles comportant les cinq modes d'articulation que nous avons identifiés, et d'autre part l'implémentation de ce modèle au sein du système SEPIA. Ce modèle explicite et facilite la définition d'articulations entre les règles d'un système d'assistance.

Mots Clef

Système d'assistance, assistance à l'utilisateur, système à base de règles, langage.

1 Introduction

De nos jours, les applications informatiques sont très nombreuses, variées et indispensables. Cependant, beaucoup d'utilisateurs renoncent à utiliser certaines applications en raison de difficultés de prise en main et d'utilisation [1]. De même, les utilisateurs peuvent sous-exploiter l'application et ignorer certaines fonctionnalités à cause de leur complexité réelle ou supposée. L'assistance aux utilisateurs est l'une des solutions pour pallier ces difficultés. Cette assistance comporte différents aspects, chacun porteur d'une grande variété. Gapenne [2] a classifié l'assistance en fonction du degré d'intervention du système d'assistance : substitution, suppléance, assistance et aide. Ginon et al. [5] ont classifié les objectifs de l'assistance (compréhension du système, découverte d'une fonctionnalité, suivi de la tâche…), les techniques d'assistance (message, exemple, modification de l'interface, création automatisée…), ainsi que les approches d'assistance (manuel d'aide, aide contextualisée…). Dans cet article, nous nous intéressons à un autre aspect de l'assistance : l'articulation entre les différents éléments qui constituent un système d'assistance. Si faciliter la compréhension d'une application est l'objectif de l'assistance, et si la technique d'assistance utilisée pour atteindre cet objectif est l'affichage de messages, alors l'articulation dans l'assistance concerne l'ordre d'affichage de ces messages. Par exemple, les messages d'aide peuvent être affichés en même temps, l'un après l'autre selon un ordre préétabli ou en fonction des actions de l'utilisateur.

Pour présenter nos travaux sur la notion d'articulation au sein des systèmes d'assistance, nous présentons tout d'abord le projet AGATE, ainsi que le langage aLDEAS et le système SEPIA qui constituent le contexte de notre travail. Pour savoir comment décrire l'articulation dans l'assistance avec aLDEAS, nous présentons ensuite notre proposition de modèle d'articulation entre règles d'assistance qui s'appuie sur une étude de l'existant concernant les systèmes d'assistance d'applications variées. Nous décrivons notamment les cinq modes d'articulation que nous avons identifiés. Puis, nous présentons l'implémentation de ce modèle dans le système SEPIA, ainsi que l'évaluation que nous avons effectuée, avant de conclure.

2 Le projet AGATE

Le projet AGATE (Approche Générique d'Assistance aux Tâches complexEs) vise à proposer des modèles génériques et des outils unifiés pour permettre la mise en place de systèmes d'assistance dans des applications existantes, que nous appelons applications-cibles. Les modèles et outils proposés dans le cadre du projet AGATE ne sont spécifiques ni à une application ni à un domaine, mais peuvent au contraire être exploités pour ajouter une assistance à des applications les plus variées, sans que celles-ci aient été spécifiquement conçues pour permettre l'intégration d'une assistance. Pour permettre la mise en place de telles assistances, nous avons précédemment proposé un processus d'adjonction d'un système d'assistance épiphyte sur une application-cible [3]. Un système d'assistance épiphyte est un système qui peut être greffé sur une application-cible sans perturber son fonctionnement [7]. Ce processus comporte deux phases : la spécification de l'assistance, puis l'exécution de cette assistance de manière épiphyte.

La première phase est effectuée par un expert de l'application-cible, appelé par la suite concepteur d'assistance. Cette phase préparatoire permet au concepteur de spécifier l'assistance qu'il souhaite pour

une application-cible. La seconde phase concerne les utilisateurs finaux de l'application-cible. Elle consiste en l'exécution de l'assistance spécifiée par le concepteur. Cette phase a lieu à chaque utilisation de l'application-cible par son utilisateur. Des épi-détecteurs rendent possible la surveillance de l'application-cible. Ils exploitent des bibliothèques d'accessibilité, chaque bibliothèque étant compatible avec un type d'application (applications Windows natives, applications Java, applications Web…) pour pouvoir détecter les événements liés aux interactions entre l'utilisateur et leurs interfaces. Enfin, des épi-assistants rendent possible l'élaboration de la réponse qui permet de fournir de l'assistance à l'utilisateur final, sous la forme d'une action d'assistance. Cette multiplicité des assistants épiphytes permet de diversifier l'assistance fournie. Ainsi, un message d'aide peut notamment être transmis à l'utilisateur final *via* une fenêtre pop-up, une synthèse vocale ou un agent animé. De même, la mise en valeur d'un composant, qui permet de guider l'utilisateur, peut prendre la forme d'un symbole affiché à côté du composant, d'un entourage de ce composant ou encore d'une modification de sa couleur.

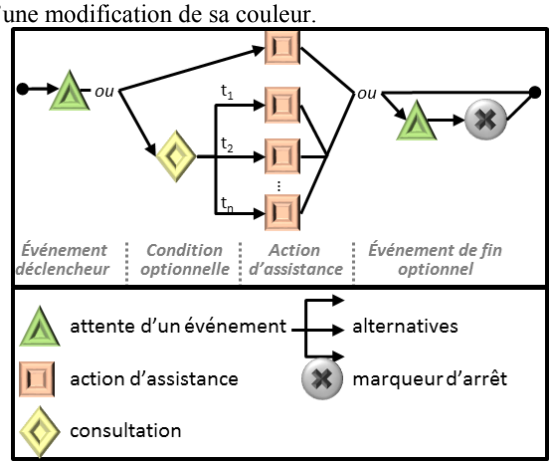


Figure 1 – Patron de règles aLDEAS

Le lien entre les deux phases du processus d'adjonction se fait grâce au langage pivot aLDEAS (a Language to Define Epi-Assistance Systems) [3]. Ce langage graphique se compose de 3 principaux éléments : attente d'événements (un clic sur un bouton…), consultation (du profil, de l'état de l'application…) et actions d'assistance (message, mise en valeur d'un élément de l'interface…). Ce langage est complété d'une part par un patron de règles (cf. Figure 1) qui permet de spécifier un système d'assistance par un ensemble de règles, et d'autre part par des patrons qui facilitent la définition d'actions d'assistance (patron pas-à-pas, présentation guidée…). Une règle débute par l'attente d'un événement nommé événement déclencheur. Dès que cet événement a lieu, le lancement de la ou des actions d'assistance est soit immédiat (chemin du haut sur la Figure 1), soit contraint par une condition (chemin du bas). Cette condition prend la forme d'une consultation avec des alternatives associées chacune à une de ces actions. Enfin, la règle peut se terminer par un événement de fin qui met fin à toutes les actions élémentaires déclenchées par cette règle.

aLDEAS et les patrons qui le complètent sont implémentés dans le système SEPIA [4] qui se compose de deux principaux outils : l'éditeur d'assistance et le moteur d'assistance. L'éditeur opérationnalise la phrase de spécification d'assistance. Il fournit une interface qui permet au concepteur de l'assistance de définir le système d'assistance en créant un ensemble de règles d'assistance en aLDEAS. Le moteur d'assistance opérationnalise la phase d'exécution d'assistance. Il permet d'exécuter le système d'assistance créé lors de la phase précédente en exécutant l'ensemble des règles d'assistance en aLDEAS définies avec l'éditeur d'assistance.

3 Modèle d'articulation entre les règles d'un système d'assistance

Si aLDEAS et sa mise en œuvre dans SEPIA permettent déjà de décrire des systèmes d'assistance comportant une articulation entre les éléments d'assistance, l'expression de l'articulation entre les règles aLDEAS est implicite et peut être complexe à définir pour le concepteur de l'assistance. Pour opérationnaliser plus efficacement ces modes d'articulation dans nos propositions, il est nécessaire de permettre l'explicitation de l'articulation entre les règles d'un système d'assistance aLDEAS.

Ainsi, un système d'assistance était précédemment défini dans le projet AGATE par un ensemble de règles aLDEAS de même niveau. Dans le patron de règles aLDEAS (cf. Figure 1), l'événement déclencheur, l'événement de fin et la condition de déclenchement sont des éléments centraux pour former l'articulation entre règles. Par exemple, pour définir l'articulation entre deux règles R_1 et R_2, telle que R_2 soit déclenchée lors de la fin de R_1, l'événement déclencheur de R_2 doit être l'événement « fin de R_1». Cette articulation entre règles est exprimée implicitement. En plus de se concentrer sur le contenu de chaque règle d'assistance, le concepteur d'assistance doit définir rigoureusement les événements et les conditions afin qu'ils assurent une articulation correcte entre les règles. Ces différents points entraînent une complexité dans la définition d'un système d'assistance.

Pour ces raisons, nous proposons de compléter notre langage par un modèle explicite d'articulation entre les règles d'assistance. Pour simplifier ici la représentation du modèle, nous ne notons que les règles entre lesquelles nous voulons étudier l'articulation. Ces règles sont nommées R_i, avec $i \in [1, n]$ ($n \geq 2$). La représentation globale de notre modèle est donnée en Figure 2. Elle donne une vue d'ensemble des cinq modes d'articulation que nous avons identifiés dans les différents travaux que nous avons étudiés: *indépendant*, *successif*, *simultané*, *progressif* et *interactif*. Le premier mode d'articulation,

indépendant (cf. partie haute de la Figure 2) exprime une absence d'articulation explicite entre règles, même s'il est toujours possible de trouver un lien, même faible entre elles. Ainsi, le mode d'articulation *indépendant* reflète la définition par défaut d'un système d'assistance en règles aLDEAS : les règles sont au même niveau et l'articulation entre elles est implicite.

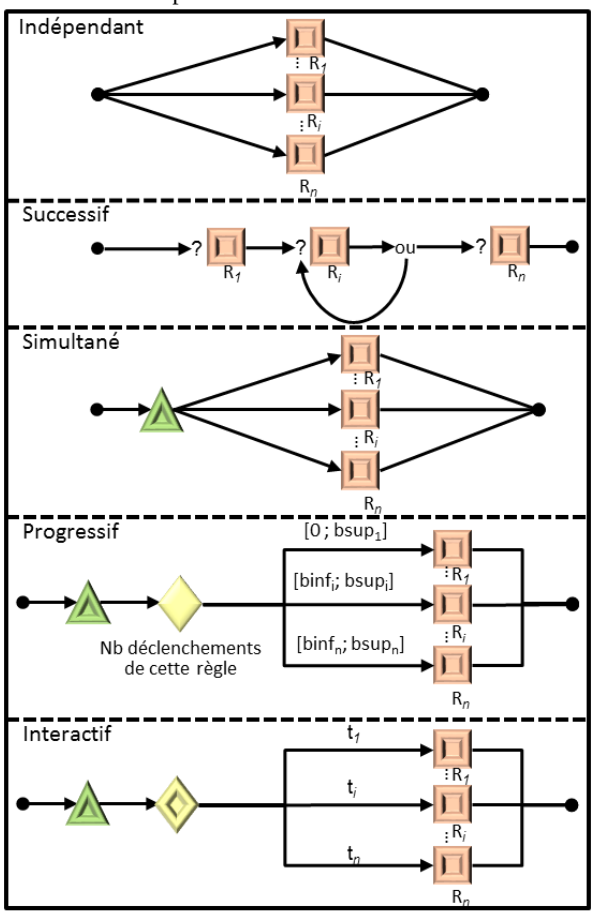


Figure 2 – Modèle d'articulation entre des règles d'assistance aLDEAS.

Dans chaque mode d'articulation, il existe des contraintes que des règles doivent respecter afin d'assurer l'articulation entre elles (par exemple, en mode *successif*, chaque règle doit être déclenchée par la fin de la règle précédente). Les contraintes propres à chaque mode d'articulation sont présentées dans la section suivante. De plus, ces modes peuvent être combinés pour fournir une assistance complète. La notion de combinaison de modes d'articulation est ensuite abordée.

3.1 Mode d'articulation successif

Dans le mode d'articulation *successif* (cf. Figure 2), les règles sont déclenchées les unes après les autres, c'est-à-dire qu'à la fin d'une règle R_i, la règle R_{i+1} est lancée. Ce mode d'articulation correspond à des assistances « étape par étape » telle que le tutoriel de Connectify[1].

Dans la définition détaillé de ce mode d'articulation (cf. ① Figure 3), nous pouvons observer que la règle R_i est contrainte à posséder un événement de fin et la règle R_{i+1} à posséder comme événement déclencheur « fin de R_i ». Cette contrainte est valable pour toutes les règles excepté la première et la dernière : la première règle R_1 peut débuter par n'importe quel(s) événement(s) déclencheur(s) et la dernière règle R_n peut se terminer par aucun, un ou plusieurs événements de fin. Dans ce mode d'articulation, la règle R_1 est un point d'entrée pour lancer l'ensemble des règles R_i. Les événements déclencheurs de cette règle R_1 sont donc également ceux qui lancent cet ensemble de règles.

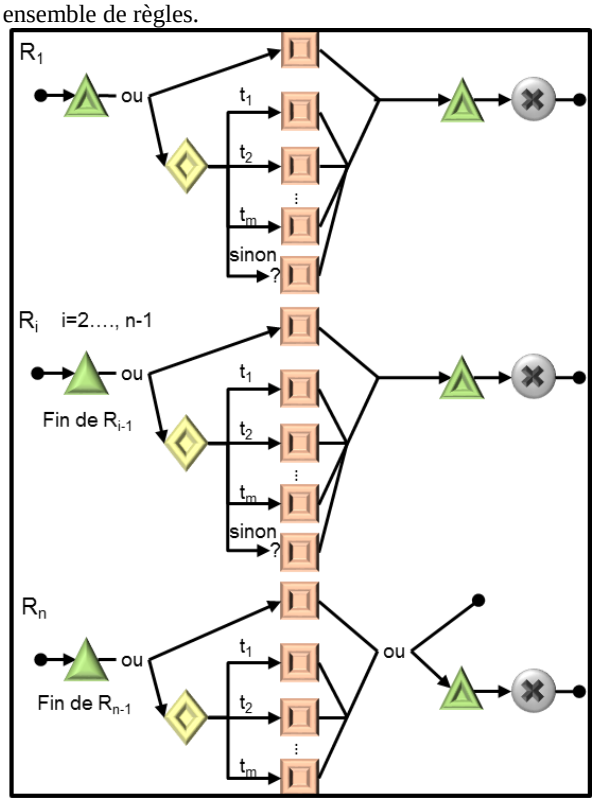


Figure 3 – Représentation en aLDEAS de n règles en mode d'articulation successif.

De plus, une règle R_{i+1} ne peut être lancée qu'après la fin de la règle R_i qui la précède. En conséquence, si R_i possède une condition de déclenchement qui n'est pas validée au moment de l'exécution de l'assistance, la règle ne sera pas exécutée jusqu'à son événement de fin, et la règle suivante ne sera donc pas déclenchée. Ainsi, en mode *successif*, si une condition de déclenchement n'est pas validée, toute la succession des règles est interrompue. Cela contraint une règle R_i contenant une condition à

[1]http://www.connectify.me/

posséder une alternative « sinon ». Cette alternative assure que la condition est toujours validée.

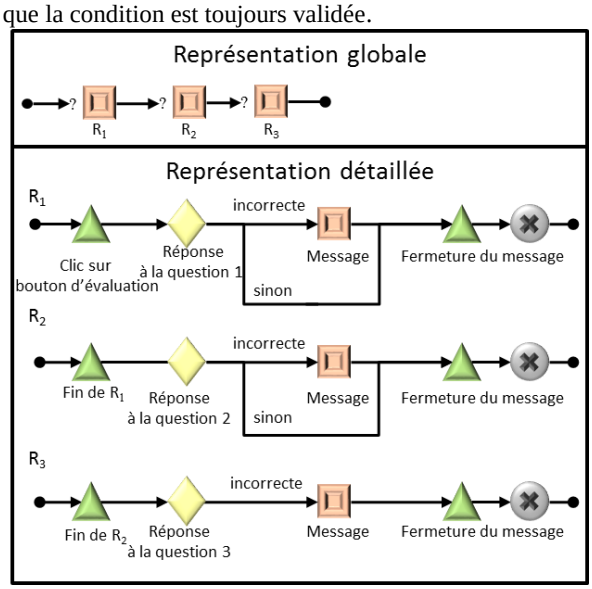


Figure 4 – Exemple de trois règles aLDEAS articulées

Prenons l'exemple d'une assistance pédagogique dans un logiciel éducatif (cf. Figure 4). Lorsque de l'utilisateur valide un exercice, l'assistance vérifie successivement les réponses de l'utilisateur pour les trois questions de l'exercice : en cas de réponse incorrecte, l'assistance fournit à l'utilisateur une information sur cette erreur, puis poursuit son exécution avec la question suivante jusqu'à la dernière question de l'exercice. Dans cet exemple, l'assistance est définie par un ensemble de trois règles articulées en mode *successif*, dans lequel chaque règle correspond à une question de l'exercice. Chaque règle contient une condition de déclenchement qui vérifie si la réponse donnée par l'utilisateur à la question est correcte et dans le cas contraire une explication est affichée à l'utilisateur. Dans ce cas, il peut être important de permettre à l'assistance de ne pas s'interrompre à la première réponse correcte de l'utilisateur, c'est-à-dire à la première règle dont la condition de déclenchement n'est pas validée. Pour cette raison, il faut ajouter une branche « sinon » dans toutes les règles (sauf la dernière). Lorsque la condition de déclenchement n'est pas validée, c'est-à-dire si aucun des tests des alternatives n'est vérifié, alors la règle prend fin dans la branche « sinon ». Les règles R_1, R_2 doivent obligatoirement posséder une branche « sinon », et des événements de fin. Les règles R_2 et R_3 doivent obligatoirement posséder un événement déclencheur, respectivement « fin de R_1 » et « fin de R_2 ».

3.2 Mode d'articulation simultané

Le mode d'articulation *simultané* (cf. Figure 2) permet de lancer plusieurs règles d'assistance en même temps, comme c'est le cas avec l'assistance du formulaire d'inscription de Google[2] qui met en valeur simultanément toutes les erreurs de saisies de l'utilisateur.

Dans le mode d'articulation *successif* (cf. section 3.1), la règle R_1 était le point d'entrée du système d'assistance pour lancer les règles R_i successivement, alors que dans ce mode d'articulation *simultané*, il faut une règle supplémentaire comme point d'entrée, afin de lancer toutes les règles R_i simultanément. Nous appelons cette règle R'. R' doit comporter un ou plusieurs événements déclencheurs quelconques, et les règles R_i doivent débuter par l'événement déclencheur « lancement par une règle (R') ».

3.3 Mode d'articulation progressif

Le mode d'articulation *progressif* (cf. Figure 2) permet de déclencher des règles d'assistance qui diffèrent selon le nombre de fois que l'utilisateur se retrouve dans une même situation problématique. En particulier, ce mode d'articulation permet de proposer progressivement à l'utilisateur une assistance de plus en plus importante pour répondre à un besoin d'assistance qui se répète. C'est le cas avec l'assistance pédagogique d'ActiveMaths [6] qui donne dans un premier temps un indice, puis la solution lorsque l'utilisateur donne plusieurs fois de suite une mauvaise réponse à une même question.

Dans ce mode d'articulation *progressif*, comme dans le mode d'articulation *simultané*, une règle R' sert de point d'entrée pour le lancement de l'assistance. Cette règle lance une seule règle parmi les règles R_i, en fonction du nombre de lancements de la règle R'. Le nombre de déclenchements d'une règle R_i dépend de l'intervalle $[binf_i, bsup_i]$: R_i sera déclenchée un nombre de fois compris entre la borne inférieure $binf_i$ et la borne supérieure $bsup_i$. Lors de ses $bsup_1$ premiers déclenchements, R' déclenche R_1, puis lors de ses déclenchements suivants, R' déclenchera R_2, puis R_3, etc. Dans ce mode d'articulation, les règles R_i doivent débuter par l'événement déclencheur « lancement par une règle (R') ».

3.4 Mode d'articulation interactif

Dans le mode d'articulation *interactif* (cf. Figure 2), une règle parmi les règles R_i est lancée en fonction d'une consultation du profil, de l'état de l'application, de l'historique de l'assistance, des traces et/ou de l'utilisateur. Par exemple, ce mode d'articulation permet de consulter l'utilisateur pour lui demander de choisir entre plusieurs assistances correspondant à différentes fonctionnalités du logiciel comme l'assistance dans le centre d'aide de Google[3]. Dans d'autres systèmes, l'assistance pour un utilisateur peut différer de celle proposée à d'autres utilisateurs [8] grâce à la consultation

[2] https://accounts.google.com/SignUp?
[3] https://support.google.com/chrome/

du profil de l'utilisateur contenant des informations relatives à ses préférences ou à son expérience.

À nouveau, une règle R' sert de point d'entrée pour le lancement de l'assistance. Chaque règle R_i est associée à une alternative de la condition de déclenchement de la règle R'. Les règles R_i doivent débuter par l'événement déclencheur « lancement par une règle (R') ». Le mode d'articulation *progressif* (cf. section 3.3) est un cas particulier du mode d'articulation *interactif*, fréquemment rencontré dans les systèmes d'assistance existants et dans lequel la condition de déclenchement de R' porte exclusivement sur un nombre de déclenchement.

3.5 Combinaison de modes d'articulation

Un système d'assistance donné n'utilise pas forcément un seul mode d'articulation entre règles. La combinaison de modes d'articulation peut concerner tous les modes que nous avons identifiés. Par exemple, les tutoriels fournissent une assistance étape par étape, mais dans une étape, ils peuvent simultanément afficher un message textuel et effectuer la mise en valeur d'un bouton.

4 Mise en œuvre des modes d'articulation entre règles d'un système d'assistance

Nous avons implémenté le modèle d'articulation que nous avons proposé (cf. section 3) dans SEPIA. Jusqu'à présent, SEPIA n'intégrait pas ce modèle d'articulation et un système d'assistance était composé d'un ensemble de règles aLDEAS sans articulation explicite entre elles, ce qui correspond au mode d'articulation que nous appelons *indépendant*. Nous avons donc enrichi l'éditeur d'assistance de SEPIA pour permettre au concepteur de l'assistance de définir explicitement des modes d'articulation entre les règles de son système d'assistance.

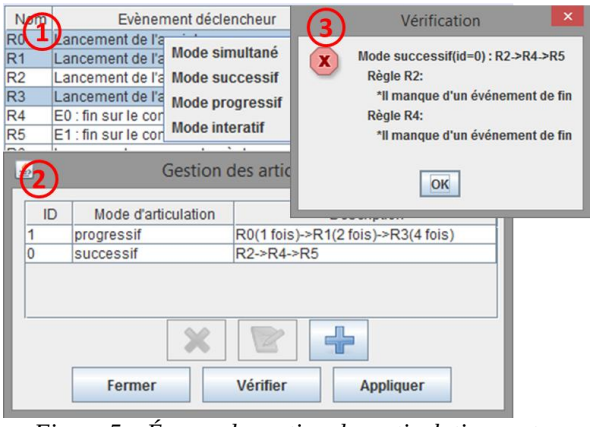


Figure 5 – Écrans de gestion des articulations entre règles dans le système SEPIA

Dans un premier temps, le concepteur d'assistance peut définir (cf. ① dans la Figure 5) et visualiser (cf. ② dans la Figure 5) les modes d'articulation entre les règles de son système d'assistance, qui sont par défaut articulées en mode indépendant. L'éditeur d'assistance complète alors si besoin les règles d'assistance concernées pour qu'elles respectent les contraintes imposées par le mode d'articulation choisi. L'éditeur génère automatiquement les événements manquants, qu'ils soient déclencheurs ou de fin. Par exemple, pour une articulation en mode *successif*, chaque règle doit avoir pour événement déclencheur la fin de la règle précédente. En cas de conflit avec la définition d'une règle d'assistance pour laquelle le concepteur de l'assistance avait précédemment défini un événement déclencheur ou de fin qui ne respecte pas les contraintes imposées par le mode d'articulation choisi, l'éditeur d'assistance informe le concepteur du problème (cf. ③ dans la Figure 5) afin qu'il choisisse entre accepter les modifications nécessaires et conserver la règle telle qu'elle en la retirant de l'ensemble des règles concernées par le mode d'articulation. La vérification du respect des contraintes imposées par un mode d'articulation peut être faite à tout moment.

5 Évaluation de nos propositions

Notre modèle d'articulation a été implémenté dans le système SEPIA de façon opérationnelle, ce qui montre la faisabilité de notre approche. Afin de tester cette mise en œuvre, nous avons utilisé cette nouvelle version de notre outil pour créer un système d'assistance assez riche pour un exercice de l'application web Mathématiques Faciles[4]. Cet exercice permet aux élèves de pratiquer l'addition en colonne sur des nombres à trois chiffres. L'élève doit saisir le résultat de l'addition en complétant les trois champs correspondant aux chiffres des unités, dizaines et centaines (cf. Figure 6).

Le système d'assistance, que nous avons créé avec SEPIA, affiche d'abord un bouton d'aide dès le lancement de l'assistance (cf. ① dans la Figure 6). Pendant la réalisation du premier exercice, l'élève peut cliquer sur ce bouton d'aide. Si sa réponse est mauvaise, le système d'assistance fournit des explications sur les étapes lors d'une première intervention, et un diagnostic lors des interventions suivantes. Le mode d'articulation est ici *progressif*. Toujours dans ce système d'assistance, les explications sur les différentes étapes de résolution de l'exercice sont affichées les unes après les autres (cf. ② dans la Figure 6). Le mode d'articulation est donc *successif* pour cette partie. Enfin, le diagnostic sur les différentes valeurs saisies est effectué simultanément, afin de mettre un symbole « erreur » à côté de chaque champ contenant une saisie incorrecte (cf. ③ dans la Figure 6). Le mode d'articulation de cette partie est donc *simultané*.

[4] http://www.mathematiquesfaciles.com/additions-a-trous-1-nombres-entiers_2_48182.htm

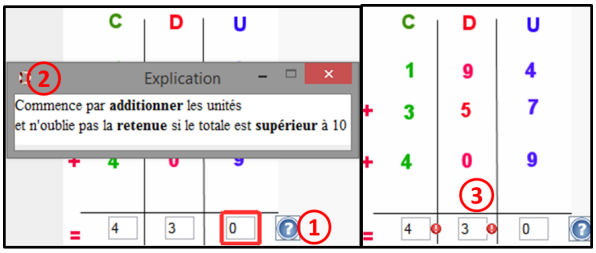


Figure 6 – Capture d'écrans des étapes d'assistance pour l'application Mathématiques Faciles.

Ce premier test a montré la pertinence de nos propositions et mis en lumière trois points forts. Tout d'abord, la version de SEPIA qui intègre notre modèle d'articulation entre règles permet de créer et visualiser explicitement des déroulements variés d'assistance. Ensuite, cette version assure que les règles respectent les contraintes liées aux articulations, ce qui réduit les erreurs pendant la définition de systèmes d'assistance par le concepteur d'assistance. Enfin, la nouvelle version de SEPIA décharge le concepteur d'une partie de ses tâches, grâce à la génération automatique de certains éléments aLDEAS nécessaires à la définition d'une assistance. En plus de la mise en œuvre des cinq modes d'articulation que nous avons identifiés, la combinaison de modes d'articulation est également possible. Toutefois, le concepteur doit actuellement établir manuellement cette combinaison, car l'interface du système n'intègre pas pour l'instant de définition graphique de cette combinaison.

Cette première évaluation de nos propositions nécessite bien sûr d'être complétée. Nous allons en effet créer d'autres systèmes d'assistance dans des contextes variés (reproduction de systèmes d'assistance existants ou créations) pour tester les différents modes d'articulation entre règles et leur combinaison. Nous prévoyons également de conduire des expérimentations confrontant un groupe de concepteurs d'assistance au modèle d'articulation et à la nouvelle version de SEPIA afin d'évaluer l'utilisabilité et l'utilité du modèle et de sa mise en œuvre. Nous demanderons pour cela à des concepteurs d'assistance de définir des systèmes d'assistance sur papier avec le langage aLDEAS sans connaissance du modèle d'articulation, et en ayant connaissance de ce modèle. Par ailleurs, ces concepteurs définiront des systèmes d'assistance tout d'abord avec la version de SEPIA n'intégrant pas le modèle d'articulation, puis avec la version intégrant ce modèle. Ces comparaisons devraient nous permettre d'évaluer la pertinence de notre modèle d'articulation et de sa mise en œuvre.

6 Conclusion

Dans cet article, nous avons présenté le modèle d'articulation entre les règles d'un système d'assistance que nous proposons pour compléter le langage aLDEAS. Ce modèle explicite la notion d'articulation entre règles d'un système d'assistance. Il propose pour cela cinq modes d'articulation correspondant aux articulations que nous avons identifiées en nous appuyant notamment sur une étude de systèmes d'assistance existants : *indépendant*, *successif*, *simultané*, *progressif*, *interactif*. Grâce à l'introduction de ce modèle dans notre approche, un système d'assistance n'est plus seulement défini par un ensemble de règles, mais peut également contenir des informations qui explicitent l'articulation entre ces règles. Nous avons implémenté ce modèle dans le système SEPIA, en ajoutant trois fonctionnalités principales à notre système : la spécification des modes d'articulation entre règles, l'application des contraintes sur les règles et la vérification des contraintes sur les règles. Notre utilisation de l'outil ainsi enrichi pour créer un système d'assistance combinant plusieurs modes d'articulations entre règles a démontré la faisabilité notre proposition.

Nous travaillons actuellement pour enrichir enrichir l'éditeur d'assistance de SEPIA afin de proposer au concepteur une visualisation graphique du système d'assistance qu'il crée. Cette interface permettra notamment de visualiser graphiquement l'ensemble des règles d'un système d'assistance et plus particulièrement les modes d'articulations entre elles.

Bibliographie

[1] Cordier, A., Lefevre, M., Jean-Daubias, S. & Guin, N., Concevoir des assistants intelligents pour des applications fortement orientées connaissances : problématiques, enjeux et étude de cas, *IC*, p. 119-131, 2010.

[2] Gapenne, O., Relation d'aide et transformation cognitive, *Intellectica*, p 7-16, 2006.

[3] Ginon, B., Jean-Daubias, S., Champin, P.-A. & Lefevre, M., aLDEAS : un langage de définition de systèmes d'assistance épiphytes, *IC*, p. 137-148, 2014.

[4] Ginon, B., Jean-Daubias, S., Champin, P.-A. & Lefevre, M., Setup of epiphytic assistance systems with SEPIA, *Ec-Tel*, p. 138-151, 2014.

[5] Ginon, B., Jean-Daubias, S., & Champin, P.-A.s Une typologie de l'assistance aux utilisateurs: exemple d'application aux EIAH, *RR-LIRIS-2013-007*, 2013.

[6] Melis, E., Andres, E., Budenbender, J., Frischauf, A., Goduadze, G., Libbrecht, P., Pollet, M. & Ullrich, C., ActiveMath: A Generic and Adaptive Web-Based Learning Environment, *IJAIED*, 12, p. 385–407, 2001.

[7] Paquette, G., Pachet, F., Giroux, S., Girard, J., Epitalk, a generic tool for the development of advisor systems, *IJAIED*, p. 349-370, 1996.

[8] Simonin, J., & Carbonell, N., Interfaces adaptatives Adaptation dynamique à l'utilisateur courant, *Interface numérique*, p. 37-54, 2007.

Representing and Mining Association Rules from Multisource Heterogeneous Simulation Traces

Ben-Manson Toussaint[1,2] Vanda Luengo[1]
[1] Univ. Grenoble Alpes, LIG, F-38000 Grenoble, France
[2] Ecole Supérieure d'Infotronique d'Haïti, Port-au-Prince, Haïti

20, rue Le Corbusier
38400, St-Martin-d'Hères
Ben-Manson.Toussaint@imag.fr

Résumé

Nous présentons dans cet article le framework PeTra conçu pour la représentation et le traitement de traces d'activités hétérogènes multi-sources enregistrées à partir d'environnements d'apprentissage intelligents. Nous avons testé les performances de PeTra sur des traces enregistrées sur un Système Tutoriel Intelligent (STI) dédié à la chirurgie orthopédique percutanée, TELEOS. Les expérimentations réalisées ont démontré que notre framework a permis de générer des séquences à partir desquelles il a été possible, (1) de produire des analyses didactiques sur les liens entre le comportement des apprenants et leurs performances ; (2), d'extraire des règles d'association pertinentes potentiellement réutilisables dans le STI.

Mots Clef

Systèmes Tutoriels Intelligents, Fouille de Données Educationnelles, Fouille de règles séquentielles, chirurgie orthopédique percutanée

Abstract

This paper presents PeTra, a framework proposed for representing and treating multi-source heterogeneous traces from intelligent learning environments. We tested the framework performance on traces from TELEOS, a simulation-based Intelligent Tutoring System (ITS) dedicated to percutaneous orthopedic surgery. This ITS captures learners' interactions from three different and independent sources. The conducted experiment demonstrated that the sequences generated by PeTra fostered efficiently: 1) the learning analytics task of evaluating the influence of visual perceptions on learners' errors; 2) the extraction of interesting association rules potentially reusable for the improvement of TELEOS tutoring services.

Keywords

Intelligent Tutoring Systems, Educational Data Mining, sequential rules mining, percutaneous orthopedic surgery.

1 Introduction

Perceptual-gestural knowledge is multimodal knowledge that involves a combination of actions and/or gestures along with perceptions used as controls for deciding on actions execution or validation [7]. However, in the literature, Intelligent Tutoring Systems (ITS) dedicated to domains involving this type of knowledge often discard the analysis of its perceptual part. This type of knowledge is often tacit and empirical and, thus, hard to capture and model. In fact, capturing perceptual-gestural knowledge in a learning environment requires the use of complementary sensing devices such as eye-trackers for visual perceptions, haptic devices for the touch or computer vision technology to detect postures, facial expressions, etc. The multiplicity of sources generate heterogeneous traces. To provide tutoring services based on these traces, one of the main challenges is to foster their transformation into sequences that reflect consistently the perceptual-gestural aspect of involved knowledge.

The framework PeTra (PErceptual-gestural TRAces treatment) that we present is a proposition to address this challenge. Our case study is TELEOS, a simulation-based ITS dedicated to percutaneous orthopedic surgery. Knowledge involved in this domain is perceptual-gestural [1, 7]. In fact, this surgery requires mastery of coordination of visual analyzes of X-rays, theoretical knowledge on the human anatomy and interpretation of resistance felt on the tools at different points of progression. The evaluations conducted on the validity of our proposition are twofold. We tested first the possibility to analyze learners' performance through their behavior related to perceptual analyses from the proposed representation of generated sequences. Secondly, we evaluated the possibility to extract from the set of generated sequences interesting perceptual-gestural association rules based on domain experts' belief. The rest of the paper is organized as follows. The 2nd section presents related works on analyzing perceptions in ITSs; the 3rd section, the methodology for recording traces in our case study; the 4th section details the traces processing with PeTra; the 5th section, the rules extraction process from the set of transformed sequences; the 6th section, our evaluation and findings and the 7th section, our conclusion and perspectives.

2 Related Works

The literature reports many prominent works on Intelligent Tutoring Systems dedicated to domains where perceptual-gestural knowledge is involved. We can mention ITSs that have been proposed for training helicopters [9] and planes [10] piloting as well as car driving [14, 15]. As one of the most recent related researches, we can also cite CanadarmTutor that was designed to train astronauts of the International Space Station for handling an articulated robotic arm [4]. However, the emphasis is generally carried in these works on actions and gestures and not on the perceptions accompanying these latter. In CanadarmTutor, the manipulation of the robotic arm from one configuration to another is guided by cameras through the operation scenes. Visual perceptions that are likely in play for this guidance would be worth further analysis. Other works have been conducted on the analysis of perceptions in learning contexts. For example, visual perceptions are captured and analyzed to deduce learners' cognitive abilities [12] or their metacognitive skills in exploratory learning [2]. Some researchers would rather use collected perceptual information for measuring the learners' mental workload or cognitive effort [6], or for inferring their behavior in the learning process [3, 8]. In other studies, sensing devices are used for capturing postures, facial expressions and body language as emotional signals [11]. For our part, we believe that perceptions denote knowledge states along with actions they are related to and, therefore, should be analyzed from an epistemic point of view. The aim is to point out the benefits from studying perceptual-gestural knowledge up on its original multimodal characteristics. To realize this, we need first to foster the consistent representation of perceptions-related behaviors and actions/gestures into perceptual-gestural sequences.

3 Recording Perceptual-Gestural Traces: TELEOS Case Study

The simulation interface of TELEOS is composed of sections that represent the main artefacts of a percutaneous operating room. Namely, as illustrated in Fig. 1.a**,** it includes a 3D model section where the patient's model is displayed; the X-rays sections and the settings panel that embeds three settings sub-sections: the fluoroscope settings panel; the cutaneous marks panel and the trocar manipulation panel.

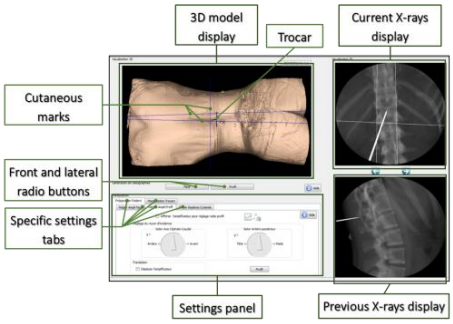


Fig. 1. TELEOS simulation interface

The "fluoroscope" is the unit used to generate X-rays; the "trocar" is the surgical tool used to reach the targeted vertebra; and the cutaneous marks are lines drawn on the patient's skin to spot the insertion point of the surgical tools.

These sections represent the areas of interest (AOI) that are recorded when they are fixed by the user. AOIs related to X-rays embed the points of interest of the targeted vertebra, i.e., specific parts that should be analyzed to support decisions on surgical actions and gestures [5].
For capturing surgical gestures involved in vertebroplasty procedures, an haptic arm has been configured to render the bones and body resistance through the insertion trajectory [7]. The surgical gestures consist of different types of prehension of the trocar, the force applied for its manipulation and the consequent speed of its progression, as well as its incline, orientation and direction of insertion. Finally, punctual actions are captured from the simulation software interface. This is for example the triggering of an X-ray. They are punctual as opposed to the continuity of a visual path or the spatial and temporal dynamism of a gesture. They are therefore recorded only at the moment of their execution as opposed to the continuous recording of information from the complementary sensing devices that are sent on a milliseconds basis: (100 ms for the haptic arm and 200, for the eye-tracker). Furthermore, traces from the three sources are recorded independently. They are heterogeneous in their content types and format, and their time granularities. Traces from the simulator and the eye-tracker are alphanumeric whereas those from the haptic arm are numeric. Their lengths are also different: traces from the simulator contain up to 54 items; those from the haptic arm, 14 items and those from the eye-tracker, 7 items. The challenge is to link those traces into one sequence such that they reflect properly all aspects of involved knowledge.

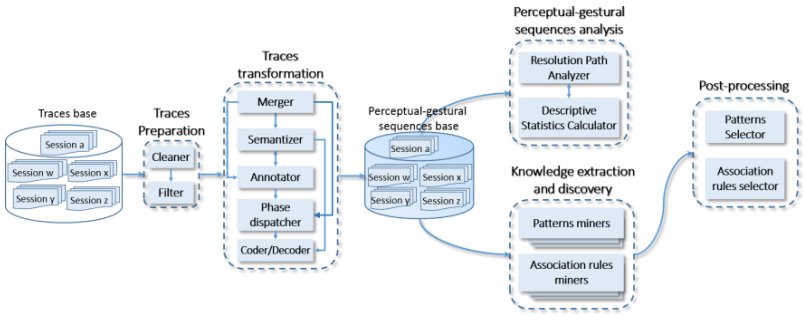


Fig. 2. Overview of the framework PeTra

4 The Treatment Process with PeTra

The PeTra framework is composed of several single-function software that we call "operators". As illustrated in Fig. 2, the operators are structured as a process and dispatched into 5 phases of treatments.

The preparation operators are applied to clean traces from known formatting errors and to discard if needed parameters that are not relevant for the treatment. The transformation phase embeds the operators that generate perceptual-gestural sequences from the multisource traces. We describe further the main operators used in this study.

4.1 Transforming the Multisource Traces

The Merger. The merger is used to link action to the perceptions that support their execution. The result expected is a set of sequences that render the perceptual-gestural aspect of each interaction. This operation is realized in two steps. First, traces from the different sources are joined in one set and then ordered sequentially. Depending on the didactic interest, perceptions linked to one action can precede or follow this latter. In other words, this link is either descending (perceptions to action) or ascending (perceptions from action). In the second step, the operator checks the chosen configuration and merges the parameters of the related perceptions and action into one sequence. In our case study, the link between actions and perceptions is descending.

The Semantization Operator. This operator is used for transposing changes in the coordinates of the simulation tools into semantic states. Those are the consequence of executed actions and gestures. This semantic information can translate a discrete manipulation (e.g. *"the fluoroscope has a caudal incline"*) or a continuous one (e.g. *"the trocar is quickly inclined on the sagittal axis"*). The semantic denominations used for our case study are based on the standard anatomical terms of location. To transpose discrete manipulations of the surgical tools, the operator considers the current and previous position coordinates of each tool. For continuous manipulations, it considers the continuum of sequences from the first that reports a position change to the last that reports a homogeneous evolution of this change (e.g. a change on the same axis and the same direction through several sequences).

The Annotator. This operator is used to enrich sequences with expert assessments. In TELEOS, elements of knowledge that are put into play by learners are diagnosed by a Bayesian Network based on their suitability to a set of expert-defined controls [9]. Those controls are elements of knowledge formulated by expert surgeons and integrated in the knowledge model of the simulator. We refer to the assessment items returned by the Bayesian Network based on these controls as "situational variables". Table 1 shows an example of control and situational variable as well as the actions they are associated to and steps of the simulation in which execution of these actions can be observed.

Table 1. An example of control and situation variable.

Action	Phase	Control	Situation Variable
Check the position of the trocar on a lateral radio	Insertion	At the cutaneous entrance spot, the trocar must be tipped towards the pedicle.	Orientation of the trocar at the cutaneous entrance spot.

The Phase Dispatcher. The phase dispatcher classifies sequences within phases when the resolution of exercises provided in the learning environment is realized through different phases. To determine the phase to which an action belongs, the operator takes as input either the predefined lists of actions for each phase or the list of characteristics that define each phase. At this stage of the treatment process, we have a representation of the perceptual-gestural sequences from which we can proceed to more advanced treatment like learning analytics and data mining tasks.

4.2 Processing Generated Perceptual-Gestural Sequences

The Resolution Path Analyzer. The resolution path analyzer helps in the analysis of exercises whose resolution is a progression of phases' validation. The operator records each phase validation as well as backward steps due to validation errors. It also records both regular or corrective actions and gestures executed at each point of the resolution path along with associated perceptions.

The Miner and Rules Selector. The miner embeds several algorithms proposed in the literature for mining frequent sequential patterns and association rules [13]. For this study, we are interested in extracting sequential rules from the set of generated perceptual-gestural sequences. We proposed a novel algorithm, PhARules, for the extraction process. Fig. 3 presents the pseudo-code of the algorithm that we presented in detail in [13].

INPUT : a phase descriptors database, a sequence database D, minSeqSup, minSeqConf
OUTPUT : the set of phase-related sequential rules
PROCEDURE:

1. Consider the sequence database D as a transaction database D'
2. Group sequences by phase based on phase descriptors and generate transaction databases $D'_1, \ldots, D'_n$ (n = number of phases)
3. Find all association rules for each database D'_i. Select minsup = minSeqSup and minconf = minSeqConf
4. Compute seqSup and seqConf of each association rule r D'_i in D_i. Eliminate each rule r in D_i such that:
   a. seqSup(r) < minSeqSup
   b. seqConf(r) < minSeqConf
5. Return the set of phase-related sequential rules

Fig. 3. The PhARules algorithm

The rules selector is an operator used to filter the mining results so as to select patterns and rules based on user-specified characteristics. For example, in our case study, we are interested in rules reporting perceptions, states of the tools, expert evaluation and at least one punctual actions or one gesture. An example of extracted rule is given in Fig. 4.

{The fluoroscope is positioned for front X-rays with a cranial incline}; {the learner positions the cutaneous marks slider on the patient's body}; {the learner takes a front X-ray}; {the learner visualizes the position of the spinous of the targeted vertebra on the X-ray}
⇒ {the transverse cutaneous mark is correct}; {the left cutaneous mark is correct}

Fig. 4. An example of selected rule

5 Evaluations and Findings

The traces used for this experiment were recorded from 9 simulation sessions of vertebroplasty performed by 5 interns and 1 expert surgeon of the University Hospital of Grenoble. The proposed simulation exercises consisted of treating a fracture of the 11th and/or the 12th thoracic vertebra. Each session is recorded after a preliminary overview of the simulator by the intern and lasts around one hour. The interns have never used the simulator before but have already assisted to at least one vertebroplasty operation in real life. We integrated an expert in the group so as to define a reference scope for the performance of a simulated vertebroplasty. Table 2 presents the characteristics of collected and treated data.

Table 2. Collected and treated data characteristics

Profiles	Session	Treated vertebra	#Raw Traces	#Annotated p.-g. seq.	#Fixations	#Incorrect SV	#Validation errors	#Correction sequences
Intern	S01	11th T	2702	113	2033	750	9	11
Intern	S02	11th T	1636	37	885	178	4	4
Intern	S03	12th T	118	33	690	208	3	5
Intern	S04	11th T	5107	128	2482	644	10	39
	S05	12th T	1677	41	858	174	6	10
Expert	S06	11th T	3432	59	1452	249	4	31
	S07	12th T	1828	47	1040	239	5	9
Intern	S08	11th T	5068	117	2514	644	20	36
	S09	12th T	1496	41	869	193	4	22

Our aim is to demonstrate that the proposed representation of perceptual-gestural sequences generated with the framework PeTra fosters key learning analytics and knowledge extraction results involving perceptions along with gestures and actions to which they are associated. To achieve this, we first evaluate the influence of the visual perceptions on learners' errors over operation simulations; secondly, we evaluate the interestingness of perceptual-gestural association rules extracted from the set of generated sequences and the significance of visual perceptions in this estimate. Further, to test our assumption on the pertinence of representing state of the simulation in the sequences, we also evaluate the significance of reported states of the tools in the estimate of rules interestingness. The evaluation of selected perceptual-gestural rules interestingness has been completed by 5 surgeons specialized in percutaneous surgery at the University Hospital of Grenoble.

5.1 Analyzing Influence of Visual Perceptions on Learners' Performance

We are interested in the analysis of the number of validation errors, the number of corrective actions and gestures applied on these errors and, the number and type of visual perceptions associated to these corrective actions and gestures for each session. The graph of Fig. 5.a summarizes the distribution of visual perceptions, incorrect situational variables and validation errors for each session. The session with the highest rate of perceptions (24.6) reports 19% fewer incorrect situational variables than the others, in average. The same link can be observed between visual analysis and validation errors for all the sessions, except for S08. This can be explained by the fact that the subject performed few corrective actions and visual analysis to support these actions. In fact, in the graph b of Fig. 5, we can notice that this session has one

of the lowest averages of corrective sequences (1.7) along with the lowest average of visual perceptions (15.5) associated to these sequences. As a comparison, session S02 reported the lowest average of corrective actions (1.0. See Fig. 5.b) but numerous visual analyzes (20.5) for supporting validation decisions and consequently limiting the number of errors (4. See Fig. 5.a). Moreover, it can be seen in Fig. 4.c that few visual analysis in S08 are of verification type (7.7 against 13.8 of decision perceptions). S09 was performed by the same intern. Conversely, less validation errors and fewer incorrect situational variables were observed even with approximately the same rate of visual perceptions. This is the consequence of the reversal of the amount of visual perceptions and the execution of more corrective actions (See Fig. 5.b).

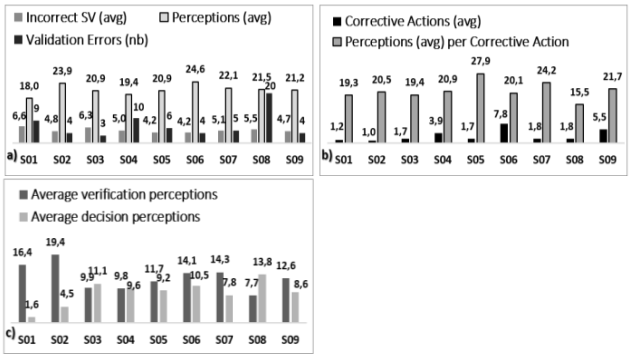
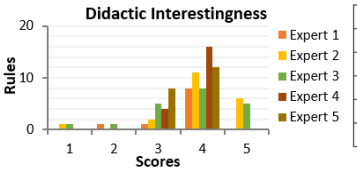


Fig. 5. a) Histogram of average incorrect situational variables, average visual perceptions and number of validation errors per session. b) Histogram of corrective actions and average associated visual perceptions per session. c) Histogram of average verification perceptions and average decision perceptions.

5.2 Evaluating the Extracted Perceptual-Gestural Association Rules

A total of 188 803 rules have been mined with a minimum support of 0.3 and a minimum confidence of 0.7. The rules selector identified 3 895 out of those based on the constraint that each rule should contain actions and/or gestures, visual perceptions, states of the simulation tools and situational variables in either its *if*-clause or its *then*-clause. We randomly pulled out 20 rules to be evaluated by 5 experts in vertebroplasty. 4 of these experts are teaching surgeons. They were asked to estimate from their belief the educational interestingness of each rule in a Likert scale ranging from 1 to 5, 1 being *very low* and 5, *very high*. Besides this overall evaluation, we asked them to provide as well their rating for 1) the reusability of the rules in a teaching context; 2) the novelty of the rules; 3) the pertinence of the reported visual perceptions and 4) the pertinence of the reported states of the tools. To measure the level of agreement among experts surgeon we computed the Jaccard distance of their answers.



	Ex1	Ex2	Ex3	Ex4	Ex5
Ex1	0				
Ex2	.19	0			
Ex3	.20	.15	0		
Ex4	.13	.10	.15	0	
Ex5	.15	.20	.22	.14	0

Fig. 6. Histogram of the distribution of the rules scores for didactic interestingness by each expert and the corresponding Jaccard distance matrix among experts' ratings two-by-two.

The overall average of the scores for the didactic interestingness of the rules is 3.8 out of 5. The weakest average rating is 3.0 for one rule; the average ratings for the 19 other rules range from 3.2 to 4.4, including 12 with an average score greater than or equal to the overall score of 3.8. This reveals that the selected rules are likely of somewhat high or very high didactic interestingness in average from the experts' point of view. The Jaccard distance computed upon the experts' ratings of this variable denotes a good level of agreement among them. In fact, the largest distance observed is of 22% between experts 3 and 5. Fig. 6 shows the histogram for the distribution of the rules on the Likert scale by each expert and the corresponding Jaccard distance matrix among experts' ratings two-by-two for the estimated didactic interestingness of the rules. The same tendency is observed for the ratings of the reusability of the rules, the pertinence of reported visual perceptions and reported states of the tools. The scores report either high or very high rating for most of the rules for these variables along with little disagreement among the experts. Conversely, the scores for novelty range from moderate to very low for most of the rules. The agreement among the experts for this variable is also moderate (up to 52%).

6 Conclusion and Future Works

We presented in this paper the framework PeTra implemented for addressing the challenge of processing multi-source heterogeneous traces from ITSs dedicated to domains involving perceptual-gestural knowledge. The aim is to consistently connect perceptions to actions and gestures they support so as to foster the consideration of all aspects of involved knowledge. We demonstrated the efficiency of our proposition for processing traces recorded on TELEOS, a simulation-based Intelligent Tutoring System (ITS) dedicated to percutaneous orthopedic surgery. We showed that the proposed representation and treatment of traces recorded from three sources in the learning environment further the analysis of the influence of visual perceptions upon interns' performance in simulation sessions of vertebroplasty. We also demonstrated the possibility of mining didactically interesting perceptual-gestural association rules from the set of sequences generated by the framework. The

reusability of the rules in a teaching context as well as the pertinence of reported visual perceptions and states of the simulation tools were also rated as high by experts of the domain with significant agreement among those. Conversely, the novelty of the rules was rated as moderate at best but with relatively low agreement between the experts. However, the study is of rather small scale. We plan to go further by confronting the quality of traces treatment performed by our framework to the evaluation of a larger panel of experts. We also consider extending our analyses to the measure of the effective gain from taking into account perceptual aspect of multimodal knowledge compared to treatment that discard either facet of this type of knowledge. As next step on improving the framework, we want to increase the efficiency of the rules selection operator based on main characteristics shared by the top-rated rules. Furthermore, the next evaluation planned will be conducted on the genericity of the framework. An experiment is presently underway for testing its operators on traces from a flight simulator.

Acknowledgements

This work has been partially supported by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01). The authors would like to thank Elena Elias for her participation to the data collection and Nadine Mandran for her assistance for the data analysis process. They are also thankful to Drs. M. Boudissa, L. Bouyou-Garnier, G. Kershbaumer and S. Ruatti for their participation in this study.

References

[1] Ceaux, E., Vadcard, L., Dubois, M., & Luengo, V.: Designing a learning environment in percutaneous surgery: models of knowledge, gesture and learning situations. Paper presented at the EARLI Symposium "Simulation-Based Learning: Analyzing and Fostering Complex Skills in the Context of Medical Education", Amsterdam (2009)

[2] Conati, C., Merten, C.: Eye-tracking for user modeling in exploratory learning environments: An empirical evaluation. Knowl.-Based Syst, Vol. 20(6) (2007) 557-574

[3] D'Mello, S. Olney, A., Williams, C., Hays, P.: Gaze tutor: A gaze-reactive intelligent tutoring system. *Int. J. Hum.-Comput. Stud., Vol.* 70(5) (2012) 377-398

[4] Fournier-Viger, P., Nkambou, R., Mayers, A., Mephu Nguifo, E., Faghihi, U. A Hybrid Expertise Model to Support Tutoring Services in Robotic Arm Manipulations. In Proc. of the 10th Mexican Intl Conf. on Artificial Intelligence. LNAI 7094, Springer, (2011) 478–489

[5] Jambon, F., Luengo, V. : Analyse oculométrique « on-line » avec zones d'intérêt dynamiques : application aux environnements d'apprentissage sur simulateur. In Actes de la Conférence Ergo'IHM sur les Nouvelles Interactions, Créativité et Usages, Biarritz France (2012)

[6] Lach, P.: Intelligent Tutoring Systems: Measuring Student's Effort During Assessment. In: Zaïane, O.R. and Zilles, S. (eds.): Advances in Artificial Intelligence. Lecture Notes in Computer Science, Vol. 7884. Springer, Berlin Heidelberg (2013) 346-351

[7] Luengo, V., Larcher, A., Tonetti, J.: Design and implementation of a visual and haptic simulator in a platform for a TEL system in percutaneous orthopedic surgery. In Medecine Meets Virtual Reality 18. (eds.): Westwood J.D., Vestwood, S.W. (2011) 324–328

[8] Mathews, M., Mitrovic, A., Lin, B., Holland, J., Churcher, N.: Do Your Eyes Give It Away? Using Eye Tracking Data to Understand Students' Attitudes towards Open Student Model Representations. In: Cerri, S.A., Clancey, W.J., Papadourakis, G., Panourgia, K. (eds.): Intelligent Tutoring Systems. Lecture Notes in Computer Science, Vol. 7315. Springer, Berlin Heidelberg (2012) 422-427

[9] Mulgund, S.S., Asdigha, M., Zacharias, G. L., Ma, C., Krishnakumar, K., Dohme, J.A., Al, R.: Intelligent Tutoring System for Simulator-Based Helicopter Flight Training. Flight Simulation Technologies Conference. American Institute of Aeronautics and Astronautics Baltimore MD U.S.A. (1995)

[10] Remolina, E., Ramachandran, S., Fu, D., Stottler, R., Howse, W.R.: Intelligent Simulation-Based Tutor for Flight Training. In: Interservice/Industry Training, Simulation, and Education Conference. (2004) 1-13

[11] Ríos, H.V., Solís, A.L., Aguirre, E., Guerrero, L., Peña, J., Santamaría, A.: Facial Expression Recognition and Modeling for Virtual Intelligent Tutoring Systems. In: Cairó, O., Sucar, L.E., Cantu, F.J. (eds.): Advances in Artificial Intelligence. Lecture Notes in Computer Science, Vol. 1793. Springer, Berlin Heidelberg (2000) 115-126

[12] Steichen, B., Carenini, G., Conati, C.: User-adaptive information visualization: using eye gaze data to infer visualization tasks and user cognitive abilities. In: Proceedings of the 2013 Int. Conf. on Intelligent User Interfaces. ACM, New York NY USA (2013) 317-328.

[13] Toussaint, B-M., Luengo, V.: Mining Surgery Phase-Related Sequential Rules from Vertebroplasty Simulations Traces. To be published in Proc. of the 15th International Conference on Artificial Intelligence in Medicine (AIME 2015). Pavia, Italy (2015)

[14] Weevers, I., Kuipers, J., Brugman, A.O., Zwiers, J., van Dijk, E.M.A.G., Nijholt, A.: The Virtual Driving Instructor, Creating Awareness in a Multi-Agent System. In: Xiang, Y., Chaib-Draa, B. (eds.): Proceedings of the 16th Canadian society for computational studies of intelligence conference on Advances in artificial intelligence. Springer-Verlag, Berlin Heidelberg (2003) 596-602.

[15] de Winter, J.C.F., de Groot, S., Dankelman, J., Wieringa, P.A., van Paassen, M.M., Mulder, M.: Advancing simulation-based driver training: lessons learned and future perspectives. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services.* ACM, New York NY USA (2008) 459-464.

Vers une approche collaborative segmentation-classification pour l'analyse d'images de télédétection

A. Troya-Galvis[1] P. Gançarski[2]

ICube, Université de Strasbourg
300 Bd Brant - CS 10413 67412 cedex - France

[1] troyagalvis@unistra.fr
[2] gancarski@unistra.fr

Résumé

L'interprétation automatique d'images de télédétection à très haute résolution spatiale est une tâche complexe. Les approches d'analyse d'images orientée objets sont couramment utilisées afin de résoudre ce problème ; mais leurs résultats dépendent fortement de la segmentation d'images. Or, il n'existe pas de méthode de segmentation universelle permettant d'isoler correctement les différents objets à extraire. Nous proposons ici un cadre formel de collaboration entre un segmenteur et un classeur pour améliorer conjointement la segmentation et la classification pour une classe thématique donnée.

Mots Clef

Segmentation, Classification, Approches collaboratives, Analyse d'images.

Abstract

Automatic interpretation of very high spatial resolution remote sensing images is a complex task. Object based image analysis approaches are commonly employed in order to solve this problem. The results of this kind of methods depend on an early image segmentation step. However, there is no universal algorithm for segmenting every kind of object of interest. We propose a collaboration framework between segmentation and classification agents in order to improve both aspects at the same time.

Keywords

Segmentation, Classification, Collaborative methods, Image Analysis.

1 Introduction

L'interprétation automatique d'images de télédétection est une tâche complexe dont les applications sont variées, telles que l'analyse de l'analyse de l'évolution urbaine [12], la prévention de désastres naturels [16], ou le suivi d'écosystèmes [4]. Les méthodes orientées pixel sont devenues obsolètes avec l'apparition des images à très haute résolution spatiale (**THRS**). En effet, les pixels de celles-ci représentent une surface terrestre allant de 50 cm$\times$ 50 cm à 2 m$\times$ 2 m, à ce niveau de détails un seul pixel ne suffit plus pour décrire un objet géographique. Le paradigme d'analyse d'images basé objets (*Object Based Image Analysis*, **OBIA** [1]) s'est donc imposé comme technique de prédilection pour traiter ce type d'images. Ces approches reposent sur l'hypothèse que les objets à extraire sont représentés dans l'image par des ensembles de pixels ayant des caractéristiques similaires, une première étape de segmentation est donc effectuée sur l'image pour regrouper au mieux les pixels qui représentent chaque objet d'intérêt. Ensuite, divers attributs sont calculés sur ces segments afin de mieux les décrire tels que des descripteurs géométriques ou texturaux permettant ainsi de ne plus se limiter aux seules propriétés radiométriques des objets. Finalement, des méthodes d'apprentissage supervisé ou non peuvent être appliquées afin de classer les segments ou pour proposer de nouvelles classes potentiellement intéressantes pour l'expert.

Cependant, la segmentation d'images est un problème mal posé, la notion de qualité étant subjective et dépendante du contexte. De fait, il n'existe pas de méthode universelle permettant de segmenter de manière optimale la totalité des objets géographiques d'intérêt. Or la segmentation de l'image est une étape critique puisque ses résultats influent directement sur la pertinence des attributs calculés et donc sur la performance des algorithmes de classification : il est primordial de fournir à l'algorithme de classification la segmentation la plus à même d'optimiser celle-ci. Ainsi, il semble à la fois naturel et intéressant de faire interagir ces deux paradigmes entre eux afin d'améliorer conjointement et progressivement leurs résultats mutuels.

Dans cet article, nous proposons un cadre formel de collaboration segmentation-classification pour l'analyse d'images de télédétection.

La suite de cet article se déroule comme suit : dans la section 2 l'état de l'art en interprétation automatique d'images de télédétection est présenté ; dans la section 3 un

cadre pour la collaboration multi-paradigme segmentation-classification est proposé ; dans la section 4 des premiers résultats expérimentaux sont présentés ; finalement dans la section 5 nous introduisons nos perspectives de recherche à court et moyen termes.

2 État de l'art

De nombreux travaux dans la littérature ont traité la combinaison segmentation-classification d'images de télédétection. Par exemple, Lizarazo et Elsner ont proposé dans [9] un cadre de segmentation floue à partir de classeurs flous pour chaque classe thématique. Dérivaux et al. ont mis en place dans [5] une méthode de segmentation supervisée qui adapte les paramètres de l'algorithme watershed à partir d'exemples données par un expert et d'algorithmes évolutionnaires. Kurtz et al. emploient dans [8] une méthode hiérarchique et multi-résolution pour la segmentation et l'extraction d'objets urbains complexes. Mahmoudi et al. ont proposé dans [10] un système multi-agent pour la classification des images de télédétection où chaque agent est spécialisé dans l'extraction d'une classe thématique donnée et un agent particulier permet la communication entre les différents agents afin de faciliter la résolution de conflits. Récemment, Hoffman et al. ont formalisé dans [7] un cadre pour l'analyse d'image basée agents, l'objectif étant de combiner les concepts du paradigme multi-agent aux approches OBIA pour faciliter la complète automatisation du processus, spécialement la sélection de règles de classification ainsi que des paramètres de segmentation.

3 Mieux classifier pour mieux segmenter

Les approches classiques de combinaison segmentation-classification appliquent la segmentation puis la classification de manière séquentielle, en assumant que la segmentation utilisée pour la classification est la meilleure. Farmer et al. ont défini dans [6] un cadre de segmentation guidée par la classification. La particularité de cette approche est qu'elle effectue la segmentation de manière conjointe avec la classification. L'approche conjointe consiste à effectuer des aller-retours entre la segmentation et la classification : la qualité de la classification est prise comme mesure de qualité pour la segmentation. L'objectif de Farmer et al. est néanmoins différent à celui de l'analyse d'images de télédétection. En effet, il s'agit d'extraire un seul objet de l'image dont on connaît approximativement la position. En revanche, en télédétection il s'agit de l'extraction d'un ou plusieurs types d'objets dont on ne connaît pas *a priori* ni le nombre ni la position.

Comme il nous semble possible d'effectuer des segmentations indépendantes pour chaque classe thématique recherchée [11], ce qui permet de centrer les effort de l'algorithme de segmentation sur les propriétés particulières des objets de chaque classe, en nous inspirant du modèle de Farmer et al., nous proposons un cadre formel permettant l'interaction entre un segmenteur et un classeur pour l'extraction d'une classe thématique donnée.

3.1 Définitions préliminaires

Soit C la classe thématique à extraire, nous définissons les concepts suivants :

Segmenteur de classe : agent noté S_C, doté d'un critère de qualité permettant de déterminer si un segment est sur-, sous-, ou bien segmenté par rapport aux objets de la classe C. Cet agent est muni d'opérateurs permettant de modifier localement les segments (e.g. fusion, scission, agrandissement, rétrécissement, etc.).

Extracteur de classe : agent noté E_C, doté d'un modèle de classement M supposé optimal appris au préalable et donnant pour un segment R_i de l'image sa probabilité d'appartenance à la classe C : $P(R_i \in C|M)$.

Classe-Segmenteur : agent noté CS_C, composé d'un couple (S_C, E_C) et gérant la collaboration entre un segmenteur et un extracteur de la classe C visant à optimiser l'identification des objets de la classe thématique C

Décision sur la classification. On définit deux seuils $T_\in$ (proche de 1) et $T_\notin$ (proche de 0) donnant pour chaque segment R_i de l'image son appartenance ou non à la classe C par :

$$\begin{cases} P_C(R_i) = 1 \text{ si } P(R_i \in C|M) \geq T_\in \rightarrow R_i \in C \\ P_C(R_i) = 0 \text{ si } P(R_i \in C|M) \leq T_\notin \rightarrow R_i \notin C \\ P_C(R_i) = P(R_i \in C|M) \text{ sinon } \rightarrow R_i \text{ n'est pas décidé} \end{cases} \quad (1)$$

3.2 Classe-segmentation

Nous appelons classe-segmentation le processus d'optimisation géré par un Classe-Segmenteur CS_C, l'objectif de la collaboration étant de réduire au maximum le nombre de segments non décidés.

Le processus consiste en une suite d'échanges de messages/actions entre le segmenteur S_C et l'extracteur E_C en deux grandes étapes :

1. La segmentation courante (donnée initialement par l'expert) est classée par l'extracteur. La classification obtenue est évaluée suivant un critère donné.
2. Si la qualité désirée n'est pas atteinte, un segment non décidé (Eq. 1) est sélectionné et un opérateur de modification lui est appliqué. Retour en 1

Afin de spécifier ce processus, il est nécessaire de définir au préalable :

— le critère de qualité Q_{cs} à optimiser par le classe-segmenteur ;
— la stratégie pour choisir le segment à modifier ;
— la stratégie de modification du segment candidat ;
— la stratégie permettant d'éviter la convergence prématurée.

Critère de qualité. Le critère Q_{cs} estime la qualité locale du Classe-Segmenteur à chaque étape de la collaboration. Il est défini comme un score borné entre 0 (aucun segment n'est décidé) et 1 (tous les segments sont décidés) par :

$$Q_{cs} = \frac{1}{N_R}\left(\sum_{i|P_C(R_i)>T_{\in}}^{N_R} P_C(R_i) + \sum_{i|P_C(R_i)<T_{\notin}}^{N_R} 1 - P_C(R_i)\right) \quad (2)$$

Avec N_R le nombre de segments dans la segmentation. Ainsi, lorsque Q_{cs} est optimale, le nombre de segments non décidés est minimal. En effet, Q_{cs} calcule la probabilité moyenne qu'un segment soit positive ou négativement classé ; les segments non décidés sont pénalisés puisque la moyenne est effectuée sur la totalité des segments, et leur probabilité de classement n'est pas prise en compte dans le calcul.
Ce critère nous permet d'évaluer la qualité du résultat de manière non supervisée pourvu que l'extracteur soit capable de d'extraire correctement les objets de la classe C de ceux qui n'y appartiennent pas.

Sélection du segment candidat. Définir une bonne stratégie pour choisir le segment devant être modifié est primordiale. En effet, il faut s'assurer que les modifications ont effectivement une forte probabilité d'améliorer la qualité du classe-segmenteur. L'idée ici est donc de choisir à chaque fois le segment le plus "ambigu" (c'est à dire le moins décidé) en espérant qu'après modification, celui-ci aura une meilleure probabilité d'appartenance (ou non appartenance) à la classe C. Soit T_{med}, la valeur médiane entre les deux seuils $T_{\in}$ et $T_{\notin}$: $T_{med} = \frac{T_{\in} - T_{\notin}}{2}$
Pour une segmentation $\mathfrak{S}$ le segment dont la probabilité de classification est la plus proche de T_{med} est sélectionné comme suit :

$$candidat(\mathfrak{S}) = \arg\min_{R_i} |P_C(R_i) - T_{med}| \quad (3)$$

Modification locale. Les modifications à appliquer au segment candidat R_c sont choisies en fonction de la qualité de segmentation de celui-ci, l'algorithme 1 illustre ce traitement. En effet, si R_c est sur-segmenté (trop petit), il sera fusionné avec son plus proche voisin en termes de radiométrie ; si R_c est sous-segmenté (trop large), il sera rétréci par une opération d'érosion morphologique ; et si R_c est bien isolé, alors plusieurs modifications sont effectués de manière indépendante et celle dont la probabilité de classement est la plus haute est conservée. Finalement, la modification n'est appliquée définitivement que si elle améliore la probabilité de classement initiale.

Arrêt et convergence prématurée. Pour évaluer la qualité de la solution courante, le critère de qualité *globale* Q_{cs} est utilisé. Or une modification *locale* du segment candidat peut amener la méthode à dégrader le résultat courant. Ne pas admettre de telles dégradations peut amener à une convergence prématurée vers un minimum local. Afin de tenter d'éviter celle-ci, des étapes dégradant le résultat sont admises avec retour à la meilleure solution en cas de non amélioration. Après un nombre donné de retours en arrière sans amélioration ou en cas de stabilisation de la qualité du résultat, la classe-segmentation s'arrête.

Algorithme 1 : Modification_locale

Data : Segment R_c, Segmenteur S_C, Extracteur E_C
Result : Segment R_m
if $sursegmentation(R_c)$ **then**
  $R_n \leftarrow fusion(R_c, plusProcheVoisin(R_c))$
else if $soussegmentation(R_c)$ **then**
  $R_n \leftarrow decroissance(R_c)$
else
  $R_n \leftarrow \arg\max_{R_{tmp}} P_{C_i}(R_{tmp})$;
  $R_{tmp} \in \{croissance(R_c), decroissance(R_c), fusion(R_c, plusProcheVoisin(R_c))\}$
if $P_{C_i}(R_n) > P_{C_i}(R_c)$ **then**
  $R_m \leftarrow R_n$
else
  $R_m \leftarrow R_c$

3.3 Processus collaboratif classification-segmentation

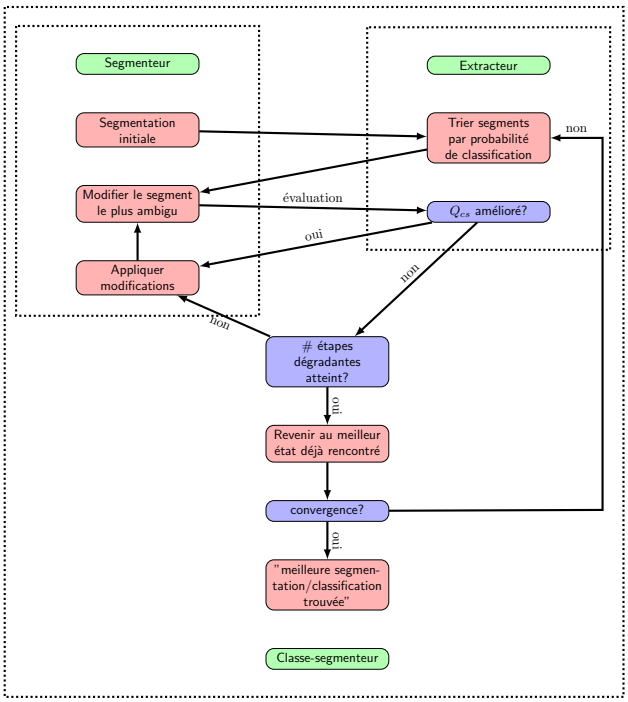


FIGURE 1 – Schéma de la classe-segmentation.

La figure 1 décrit le processus global que nous proposons. A partir d'une segmentation initiale donnée :

1. Appliquer le classeur à tous les segments (modifiés) et calculer les probabilités d'appartenance (po-

sitives, négatives et non décidées).

2. Les segments classés positivement ou négativement sont fixés et ne seront plus modifiés ;
3. Sélectionner et modifier un segment
4. Répéter tant Q_{cs} n'est pas satisfaisante et tant qu'il y a amélioration.

A la fin du processus nous obtenons un ensemble de segments classés positivement (resp. négativement) appartenant (resp. n'appartenant pas) à la classe C et un ensemble de segments non décidés en espérant que ce dernier ensemble soit le plus petit possible. L'algorithme associé (Algo. 2) décrit le déroulement complet de la collaboration. L'algorithme comporte 5 paramètres : une segmentation initiale, de préférence adaptée aux objets de la classe C ; un segmenteur et un extracteur spécialisés dans l'extraction de la classe C ; et deux entiers D et B qui permettent de modifier le comportement de la stratégie pour éviter la convergence prématurée. D correspond au nombre d'étapes dégradantes qui sont admises par le processus, et B correspond au nombre maximum de retours en arrière autorisés sans rencontrer une amélioration des résultats.

Algorithme 2 : Classe-segmentation

Data : Segmentation $\mathfrak{S}_{init}$, Segmenteur $S_i(\mathfrak{S}_{init})$, Extracteur E_i, Entier D, Entier B
Result : Segmentation $\mathfrak{S}_{final}$, Nombre[] probas, Nombre qualité

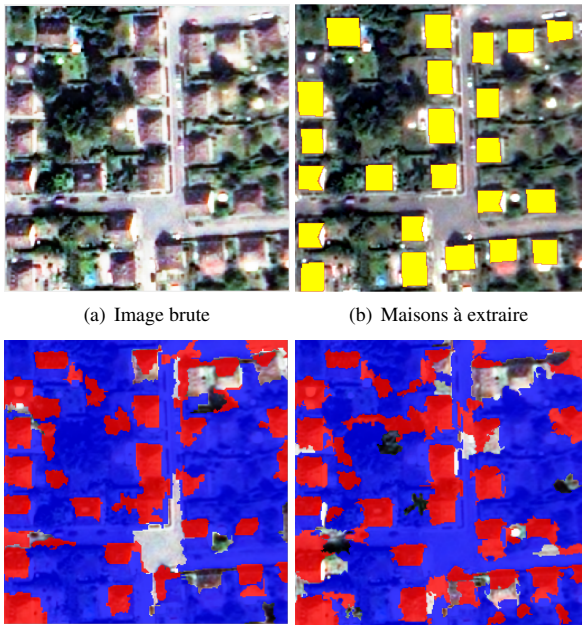
```
begin
    D_count ← 0
    B_count ← 0
    S_current ← S_init
    S_best ← S_init
    while B_count < B do
        R_c ← candidat()
        R_m ← Modification_locale(R_c, S_current, E_i)
        S_tmp ← apply(R_m, S_current)
        if Q_cs(S_tmp) > Q_cs(S_current) then
            S_current ← S_tmp
            S_best ← S_current
            D_count ← 0
            B_count ← 0
        else if D_count < D then
            D_count ← D_count + 1
            S_current ← apply(R_m)
        else
            B_count ← B_count + 1
            D_count ← 0
            S_current ← S_best
    S_final ← S_best
```

4 Expérimentation

Afin de valider notre proposition nous avons effectué diverses expérimentations. Dans cette section nous présentons un cas d'étude sur un extrait de 256×256 pixels d'une image Pléiades de la ville de Strasbourg (Fig. 2(a)). L'extrait correspond à une zone résidentielle. Nous nous intéressons à l'extraction des maisons individuelles (Fig. 2(b)).
Le segmenteur utilise la métrique UOA [15] avec l'entropie comme indice d'homogénéité. Ceci nous permet d'une part d'initialiser la segmentation avec des segments adaptés à la taille des maisons et d'autre part d'avoir une estimation de la qualité locale des segments (sur-, sous-, ou bonne segmentation).
L'extracteur est basé sur l'algorithme de régression nu-SVR [2] entrainé au préalable avec des exemples différents de ceux de la figure 2(b).
Nous remarquons que les maisons ont été globalement bien détectées bien qu'il y ait encore quelques faux positifs et un certain nombre de segments non décidés. Les résultats sont tout de même très encourageants.
Le tableau 1 donne la qualité du résultat final suite au processus de classe-segmentation par rapport à la vérité-terrain (métriques F-measure [14], Rand index [13] et coefficient Kappa [3]) et ce pour différentes valeurs du paramètre D (Algo 2) qui donne le nombre de modifications dégradantes acceptables. La première ligne correspond à la classification obtenue pour la segmentation initiale ; c'est équivalent à l'approche classique de segmenter puis classifier. On remarque que de façon générale les résultats obtenus sont prometteurs, la collaboration permet d'améliorer les résultats par rapport à ceux de la segmentation initiale, et les meilleurs scores sont obtenus avec 12 étapes dégradantes (figure 2(c)) pour l'indice de Rand, et avec 15 étapes (figure 2(d)) pour la F-measure et le Kappa. Au delà de cette valeur, les résultats chutent de manière considérable car les segments commencent à être trop modifiés de manière non pertinente à la classification. La figure 3 illustre les résultats obtenus en répétant la classe-segmentation pour l'extraction des routes et de la végétation. Les résultats sont également positifs, pour la végétation nous obtenons *F-measure*=0.66454, *Rand*=0.66394, et *kappa*=0.36662 ; et pour les routes nous obtenons *F-measure*=0.66895, *Rand*=0.90602, et *kappa*=0.61489.

5 Conclusion

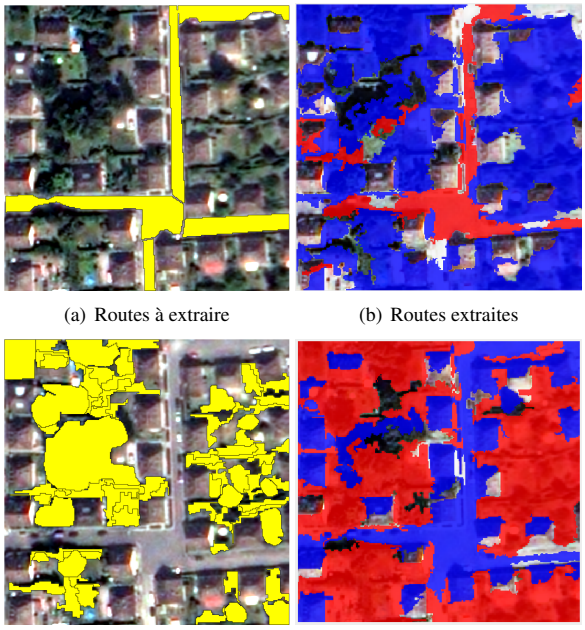
La segmentation d'images est la première étape dans la chaîne de traitements des approches OBIA et l'efficacité de ces dernières en dépend fortement. Notre intuition est que la segmentation idéale est celle qui permet de mieux classifier l'image. Dans cet article nous avons proposé un cadre de collaboration permettant de mener les processus de segmentation et de classification de manière conjointe afin d'améliorer les deux résultats simultanément.
Nos expériences montrent que la collaboration entre ces deux paradigmes est bénéfique pour l'extraction d'une classe donnée. Le nombre d'étapes dégradantes admises dans le processus a une influence considérable sur les résultats et peut les améliorer considérablement, mais le temps de calcul est alors fortement augmenté. Il est nécessaire de trouver un bon compromis. Il peut s'avérer également inté-



(a) Image brute (b) Maisons à extraire

(c) Classe-segmentation, $D = 12$ (d) Classe-segmentation, $D = 15$

FIGURE 2 – Visualisation des résultats : dans l'image (c) les segments positivement classés apparaissent en rouge, les segments négativement classés en bleu et les segments non décidés sans couleur. ©CNES2012, Distribution Astrium Services / Spot Image S.A., France, All rights reserved.



(a) Routes à extraire (b) Routes extraites

(c) Végétation à extraire (d) Végetation extraite

FIGURE 3 – Exemples de classe-segmentation pour l'extraction des routes et de la végétation. Sur la colonne de gauche, la vérité terrain des deux classes. Sur la colonne de droite, les résultats de la classe-segmentation avec $D = 15$.

TABLE 1 – Résultats de la classe-segmentation en fonction du nombre d'étapes dégradantes permises.

D	F-measure	Rand	Kappa
intiale	0.57888	0.79069	0.45213
1	0.57855	0.79096	0.45198
2	0.54600	0.77096	0.40608
3	0.55570	0.76723	0.41937
4	0.53730	0.78383	0.41045
5	0.56287	0.78271	0.43695
6	0.55487	0.77424	0.41919
7	0.55662	0.78403	0.42656
8	0.58047	0.78197	0.45349
9	0.55095	0.77451	0.41822
10	0.58128	0.77435	0.44673
11	0.58170	0.79908	0.46171
12	0.59330	**0.81746**	0.48169
13	0.53463	0.77798	0.40129
14	0.57036	0.77643	0.43690
15	**0.62263**	0.80837	**0.50899**
16	0.56386	0.80402	0.44532
17	0.58064	0.79017	0.45698
18	0.57950	0.80156	0.46053
19	0.58240	0.810203	0.46962
20	0.52157	0.7635919	0.38625

ressant d'adapter cette valeur à la volée et de manière plus fine en fonction du segment qui est en cours de modification.

Le cadre que nous proposons est générique, ce qui permet d'employer des classeurs quelconques. On pourrait imaginer par exemple un classeur dont le modèle est incrémentale et peut être amélioré à chaque lancement par l'évaluation manuelle des résultats précédents, notamment par la pénalisation des faux positifs et la prise de décision concernant les segments non décidés.

D'autres paramètres, tels que le critère de qualité, ou la stratégie pour sélectionner le segment candidat, pourraient avoir une influence sur les résultats de la méthode. Une étude approfondie sur ces aspects s'avère intéressante à faire dans nos prochains travaux. De plus, nous envisageons étendre ce cadre pour permettre l'interaction entre N classe-segmenteurs afin de pouvoir gérer un ensemble de N classes thématiques et donc l'interprétation totale de l'image.

Remerciements

Ces travaux de recherche ont été possibles grâce au financement de l'Agence Nationale de la Recherche dans le cadre du projet COCLICO (ANR-12-MONU-0001).

Références

[1] T. Blaschke. Object based image analysis for remote sensing. *ISPRS J Photogramm*, 65 :2–16, 2010.

[2] C.-C. Chang and C.-J. Lin. Libsvm : A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3) :27 :1–27 :27, May 2011.

[3] J. Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1) :37, 1960.

[4] P. Coppin, I. Jonckheere, K. Nackaerts, B. Muys, and E. Lambin. Review articledigital change detection methods in ecosystem monitoring : a review. *Int J Remote Sens*, 25 :1565–1596, 2004.

[5] S. Derivaux, G. Forestier, C. Wemmert, and S. Lefèvre. Supervised image segmentation using watershed transform, fuzzy classification and evolutionary computation. *Pattern Recogn Lett*, 31 :2364–2374, 2010.

[6] M. Farmer and A. Jain. A wrapper-based approach to image segmentation and classification. *Image Processing, IEEE Transactions on*, 14 :2060–2072, Dec. 2005.

[7] P. Hofmanna, P. Lettmayerb, T. Blaschkea, M. Belgiua, S. Wegenkittlb, R. Grafb, T. J. Lampoltshammera, and V. Andrejchenkoa. Abia - a conceptional framework for agent based image analysis. *South-Eastern European Journal of Earth Observation and Geomatics*, 3(25) :125–129, 2014.

[8] C. Kurtz, N. Passat, P. Gançarski, and A. Puissant. Extraction of complex patterns from multiresolution remote sensing images : A hierarchical top-down methodology. *Pattern Recognition*, 45 :685–706, 2012.

[9] I. Lizarazo and P. Elsner. Segmentation of remotely sensed imagery : moving from sharp objects to fuzzy regions. *Image Segmentation*, 2011.

[10] F. T. Mahmoudi, F. Samadzadegan, and P. Reinartz. Object oriented image analysis based on multi-agent recognition system. *Computers & Geosciences*, 54 :219–230, 2013.

[11] M. Musci, R. Feitosa, and G. Costa. An object-based image analysis approach based on independent segmentations. In *Urban Remote Sensing Event (JURSE), 2013 Joint*, pages 275–278, April 2013.

[12] H. M. Pham, Y. Yamaguchi, and T. Q. Bui. A case study on the relation between city planning and urban growth using remote sensing and spatial metrics. *Landscape Urban Plan*, 100 :223–230, 2011.

[13] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336) :846–850, 1971.

[14] C. J. V. Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979.

[15] A. Troya-Galvis, P. Gancarski, N. Passat, and L. Berti-Equille. Unsupervised quantification of under- and over-segmentation for object-based remote sensing image analysis. *IEEE J STARS*, PP(99) :1–10, 2015.

[16] C. V. Westen. Remote sensing and gis for natural hazards assessment and disaster risk management. In *Treatise on Geomorphology*, pages 259–298. Academic Press, 2013.

Business Rule Sets as Programs: Turing-completeness and Structural Operational Semantics

Olivier Wang[1,2] Changhai Ke[1] Leo Liberti[2] Christian de Sainte Marie[1]

[1] IBM France, 9 Rue de Verdun, 94250 Gentilly, France
[2] CNRS LIX, Ecole Polytechnique, 91128 Palaiseau, France

olivier.wang@polytechnique.edu

Abstract

Production Rules or Business Rules are of the form if condition then action. *Sets of such rules can be executed on input data according to execution algorithms, or analyzed semantically using operational semantics, but the universality of systems or sets of such rules has never been characterized formally, to the best of our knowledge.*
Our goal is to show that for a simple execution algorithm, sets of Business Rule form a universal programming language. In other words, we show that Business Rules are Turing-complete. The proof consists in showing that a Business Rule program can be transformed in a WHILE program using additional variables. The implications of such a WHILE form include a type of structural operational semantics which may facilitate semantic analysis of Business Rule Sets.

Keywords

Business Rules, Production Rules, Turing-completeness, Structural Operational Semantics, WHILE program.

1 Introduction

In recent years, academic research on Rule Systems has mostly focused on knowledge representation and learning, through Classification Rules for example. In contrast, we look at Rule Systems as programs. In particular, Business Rules (BR) (also called Production Rules in RuleML [1]) have a behavior much closer to standard imperative programs (flowcharts, WHILE programs) than any other type of rules. They are notably used in decision automation through Business Rules Management Systems (BRMS). Where Induction Rules can be used to prove the internal consistency of an ontology and deduce theorems from axioms, Business Rules behave more like WHILE programs, when executed via an appropriate execution algorithm.
The rules we consider in this paper are BRs. BRs are written using meta-variables stored in a different symbol table than the input variables, which we will keep calling variables. The meta-variable table matches variables appearing in a BR to variable names, and the variable table matches the latter to stored values. Interpretation is in two stages: first, the BR program is turned into a set of *rule instances*, where the meta-variables are replaced by corresponding variables; then, rule instances are run by an execution algorithm, which includes a conflict resolution method to decide the order of rule instance execution. We remark that BR is a typed language. Every meta-variable and input variable must have a type, and arbitrary new types can be defined within the BR program itself. The matching from meta-variables to variables is typed: each meta-variable has a type and can only match to a variable of the same type. We only consider execution algorithms which include a main loop.
BR programming can be seen as programming for non-programmers, in the following sense. The two most difficult concepts for a layperson to understand are loops and function calls. BR disposes of the former by automatically executing programs over a loop construct, and of the latter by resorting to meta-substitutions (meta-variables are replaced by variable names), so that a rule *if* condition *then* action is automatically matched to many conditions and actions sharing the same structure.
There are many variations of BR execution algorithms. Our goal is not to characterize them or compare them. We use a very simple algorithm consisting of the following steps:

1. Match variables to type-appropriate meta-variables in rules to create all possible rule instances
2. Select the rule instances for which the condition is true, using the current values of the variables
3. Execute the action of the first rule instance in the current selection or stop if there is no such rule instance
4. Restart from step 1.

The order of rule instances used in step 3 is in our algorithm defined by the lexicographic order derived from a predefined order on the rules in the rule set to execute and a predefined order on input variables. Such an algorithm has a simplistic conflict resolution strategy: whenever more than one rule instance could be executed, the one selected is obtained from a fixed total order on rule instances. An

example of the execution of such an algorithm is described in Fig. 1.

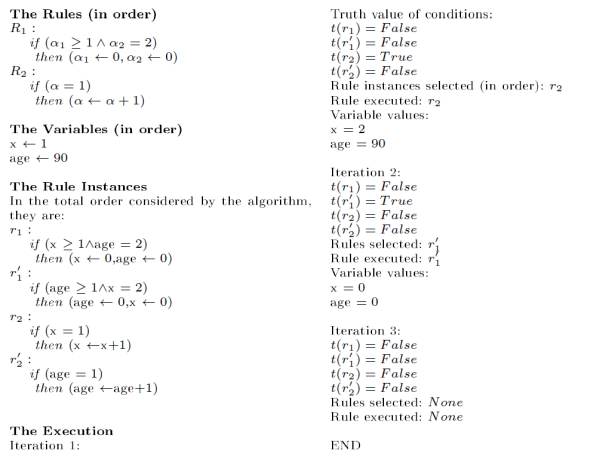


Figure 1: Example illustrating the execution algorithm

2 Turing-completeness

A Universal Turing Machine (UTM) is a Turing Machine (TM) which can simulate any other TM on arbitrary input [10, 11]. Let L be a programming language for the UTM U, described for example by its grammar. By means of a special program $\mathscr{I}$ called *interpreter*, programs written in L can be executed on U [6].

Definition 1. *If a programming language L can be used to program a UTM via an interpreter, then L is Turing-complete.*

We can replace "UTM" in Defn. 1 by any universal computer described in any Turing-complete language L', since interpreters can be composed. More precisely: (i) let U' be a program in L' describing a UTM, and $\mathscr{I}'$ is the interpreter from L' to the UTM; (ii) let U be a program in L describing $\mathscr{I}'(U')$, and $\mathscr{I}$ be an interpreter from L to L'. Then $\mathscr{I}(U) = \mathscr{I}'(U')$ is a UTM. Moreover, since a UTM is defined as a TM which is able to simulate any other TM, we can prove L Turing-complete by showing that for any TM, L can be used to describe that TM via its interpreter, as was done in [3]. According to the Church-Turing thesis [2, 12, 4], any effectively calculable function is Turing-computable. In other words, no device or program can compute a function that a UTM cannot.

2.1 Business Rule programs

Regardless of the execution algorithm, a BR program consists in a set of type declarations and a set of rules. A type declaration consists of either the creation of a type ($create\_new\_type$(type)) or the assignment of a type to a variable ($type$(var) ← type), where var can be either a variable or a meta-variable. A rule is defined as follows. Given α the typed meta-variables and x the typed variables, a rule is written:

if $T(\alpha, x)$ **then**
  $\alpha \leftarrow A(\alpha, x)$
  $x \leftarrow B(\alpha, x)$
end if

where T is the condition and the couple (A, B) describes the action. At least one of the variables, say x_1 without loss of generality, is selected to be the *output* of the BR program.

The first part of a BR program interpreter is to compile the rule instances derived from each rule. At compile time, α is replaced by every type-feasible reordering of the x input variable vector. For $x \in \mathbb{R}^n$, the explicit set of rule instances compiled from this rule is the type-feasible part of the following code fragments, using $(\sigma_j \mid j \in \{1, ..., n!\})$ the permutations of $\{1, ..., n\}$:

if $T((x_{\sigma_1(1)}, ..., x_{\sigma_1(n)}), x)$ **then**
  $(x_{\sigma_1(1)}, ..., x_{\sigma_1(n)}) \leftarrow A((x_{\sigma_1(1)}, ..., x_{\sigma_1(n)}), x)$
  $(x_1, ..., x_n) \leftarrow B((x_{\sigma_1(1)}, ..., x_{\sigma_1(n)}), x)$
end if
...
if $T((x_{\sigma_n(1)}, ..., x_{\sigma_n(n)}), x)$ **then**
  $(x_{\sigma_n(1)}, ..., x_{\sigma_n(n)}) \leftarrow A((x_{\sigma_n(1)}, ..., x_{\sigma_n(n)}), x)$
  $(x_1, ..., x_n) \leftarrow B((x_{\sigma_n(1)}, ..., x_{\sigma_n(n)}), x)$
end if

The size of this set varies. The typing of the variables and meta-variables matters, and depending on T, A and B some of these operations might also be computationally equivalent. The number of rule instances compiled from a given rule can be 0 for an invalid rule ($T(\alpha, x) = false$), 1 for a static rule ($T(\alpha, x)$ and $B(\alpha, x)$ do not vary with α, $A(\alpha, x) = \alpha$), and up to $n!$ for some rules if every variable has the same typing.

An interpreter $\mathscr{I}$ for a BR program takes the Business Rules and values of the variables as input, and executes valid rule instances one at a time according to a conflict resolution strategy (this ranges from simple to complex algorithms) until it either executes a $Stop$ instruction or runs out of valid rule instances. It then returns the value of x_1. The most basic interpreter $\mathscr{I}_0$ uses the order given earlier to choose which rule instance to execute, executing assignment actions whenever conditions are True. In order to show Turing-completeness, we only consider the basic interpreter, meaning that we expect any more complicated ones to be able to simulate the most basic.

2.2 WHILE programs

A WHILE program has the canonical (recursive) form:

while $\mathsf{T}_0(x)$ **do**
  ifblock$_1(\mathsf{T}_1, \mathsf{A}_1, x)$
  ...
  ifblock$_K(\mathsf{T}_K, \mathsf{A}_K, x)$
end while

where, for each $k \leq K$, **ifblock**$_k(\mathsf{T}_k, \mathsf{A}_k, x)$ is defined either as:

if $\mathsf{T}_k^1(x)$ **then**
  $x \leftarrow \mathsf{A}_k^1(x)$
  ifblock$(\mathsf{T}_k, \mathsf{A}_k, x)$
end if

or as an empty command. The interpretation of the symbols $\mathsf{T}_k^i(x)$ and $\mathsf{A}_k^i(x)$ is: T are tensors of Boolean conditions on the variables x, which evaluate to True or False, and A is a tensor of functions of x yielding values to be assigned to the variables.

In other words, a WHILE program is a single conditional loop containing a sequence of embedded test conditions followed by a conditional assignment action (note the action could be empty by setting the corresponding A function to the identity).

It is well known that WHILE programs are Turing-complete [5].

2.3 WHILE form of rule sets

A canonical WHILE program equivalent to a BR program is easy to establish using the execution algorithm we consider. Although this reduction plays no role in proving Turing-completeness of BR programs, it is still interesting to remark that BR programs *can* be written as WHILE programs, and having a standardized WHILE form for BR programs has some interesting applications. The main idea is to introduce an additional integer variable x_0 to serve as a control variable, and to write each rule instance explicitly in the WHILE program itself. This allows for an easier adaptation to different interpreters, such as those used in commercial BRMS. It must be noted that as explained in 2.1, each rule corresponds to a sequence of rule instances. The order of the instances in that sequence is determined by our algorithm: it is the order described in 2.1. The formal definition is somewhat more complicated, of course.

For a set of Business Rules $R_1, ..., R_m$ with conditions $T_1, ..., T_m$ and actions $(A_1, B_1), ..., (A_m, B_m)$, a set of typed variables $x_1, ..., x_n$, and a set of typed meta-variables $\alpha_1, ..., \alpha_n$, the WHILE program equivalent to the BR program with the above-mentioned execution mode is written as below. It uses a single additional integer variable x_0, and we note $(\sigma_j \mid j \in \{1, ..., n!\})$ the permutations of $\{1, ..., n\}$. The σ_j are ordered in lexicographic order on $\sigma_j(1, ..., n)$.

1: $x_0 \leftarrow -1$
2: **while** $x_0 \neq 0$ **do**
3: **if** $x_0 = -1$ **then**
4: $x_0 \leftarrow \min\{(i-1) \times n! + j$
5: $\mid T_i(x_{\sigma_j(1)}, ..., x_{\sigma_j(n)}) = true\}$
6: **else if** $x_0 = (i-1) \times n! + j$ **then**
7: $(x_{\sigma_j(1)}, ..., x_{\sigma_j(n)}) \leftarrow$
8: $A_i((x_{\sigma_j(1)}, ..., x_{\sigma_j(n)}), (x_1, ..., x_n))$
9: $(x_1, ..., x_n) \leftarrow$
10: $B_i((x_{\sigma_j(1)}, ..., x_{\sigma_j(n)}), (x_1, ..., x_n))$
11: $\triangleright$ x_0 is unique for each (i,j) couple
12: $x_0 \leftarrow -1$
13: **else**
14: $x_0 \leftarrow 0$
15: **end if**
16: **end while**

The use of the min() function and of the conditioned set is not strictly speaking allowed in WHILE programs, and the expression $T_i(x_{\sigma_j(1)}, ..., x_{\sigma_j(n)}) = true$ should be closer to checking that the substitution $\alpha_i \leftarrow x_{\sigma_j(i)}$ is type appropriate AND that the test is true. Even so, line 4 and line 5 could easily be replaced by series of try...catch... and if...then... instructions. For example, line 4 would be replaced by:

if $type(\alpha_1) = type(x_{\sigma_1(1)}) \wedge \cdots \wedge$
 $type(\alpha_n) = type(x_{\sigma_1(n)}) \wedge$
 $T_1((x_{\sigma_1(1)}, ..., x_{\sigma_1(n)}), (x_1, ..., x_n)) = true$ **then**
  $x_0 \leftarrow 1$
else if $type(\alpha_1) = type(x_{\sigma_2(1)}) \wedge \cdots \wedge$
 $type(\alpha_n) = type(x_{\sigma_2(n)}) \wedge$
 $T_1((x_{\sigma_2(1)}, ..., x_{\sigma_2(n)}), (x_1, ..., x_n)) = true$ **then**
  ...
else if $type(\alpha_1) = type(x_{\sigma_{n!}(1)}) \wedge \cdots \wedge$
 $type(\alpha_n) = type(x_{\sigma_{n!}(n)}) \wedge$
 $T_1((x_{\sigma_{n!}(1)}, ..., x_{\sigma_{n!}(n)}), (x_1, ..., x_n)) = true$ **then**
  $x_0 \leftarrow n!$
else if ... **then**
  ...
else if $type(\alpha_1) = type(x_{\sigma_{n!}(1)}) \wedge \cdots \wedge$
 $type(\alpha_n) = type(x_{\sigma_{n!}(n)}) \wedge$
 $T_n((x_{\sigma_{n!}(1)}, ..., x_{\sigma_{n!}(n)}), (x_1, ..., x_n)) = true$ **then**
  $x_0 \leftarrow n \times n!$
end if

```
The typed Variables (in order)
int x ← 1
int age ← 90

The WHILE program

 1: x0 ← −1
 2: while x0 ≠ 0 do
 3:     if x0 = −1 then
 4:         if x ≥ 1∧age = 2 then
 5:             x0 ← 1                    ▷ (i,j) = (1,1)
 6:         else if age ≥ 1∧x= 2 then
 7:             x0 ← 2                    ▷ (i,j) = (1,2)
 8:         else if x = 1 then
 9:             x0 ← 3                    ▷ (i,j) = (2,1)
10:         else if age = 1 then
11:             x0 ← 4                    ▷ (i,j) = (2,2)
12:         end if
13:     else if x0 = 1 then
14:         x ← 0
15:         age ← 0
16:         x0 ← −1
17:     else if x0 = 2 then
18:         age ← 0
19:         x ← 0
20:         x0 ← −1
21:     else if x0 = 3 then
22:         x ←x+1
23:         x0 ← −1
24:     else if x0 = 4 then
25:         age ←age +1
26:         x0 ← −1
27:     else
28:         x0 ← 0
29:     end if
30: end while
```

Figure 2: WHILE form of the rule set in Fig. 1

The Fig. 2 shows the WHILE form of the example in Fig. 1.

2.4 Equivalence

We now prove that the programming language defined using business rules and the simple looping algorithm chosen is Turing-complete. This is based on creating a simulation of WHILE programs using rules. In other words, after having transformed a BR program in a WHILE program, we now make sure that the converse is possible for any WHILE program.

Theorem 2 (Equivalence with WHILE programs)**.** *The set of programs computable using WHILE programs is the same as the set of programs computable using Business Rules.*

We prove this by showing that a generic WHILE program can be interpreted into a BR program. The only requirement of the interpretation is to be computable. We first prove this for WHILE programs without embedded **if** statements, then we discover a sequence of syntactical steps on the symbols of a generic WHILE program which transforms it into a WHILE program without embedded **if** statements.

Lemma 3. *Any WHILE program without embedded* **if** *statement can be simulated using a BR program.*

Proof. Given the following WHILE program without embedded **if** statements:

1: **while** $\mathsf{T}_0(x)$ **do**
2: **if** $\mathsf{T}_1(x)$ **then**
3: $x \leftarrow \mathsf{A}_1(x)$
4: **end if**
5: ...
6: **if** $\mathsf{T}_K(x)$ **then**
7: $x \leftarrow \mathsf{A}_K(x)$
8: **end if**
9: **end while**

We can write an equivalent rule set with $K+1$ rules. They are static rules, because they do not make use of meta-variables and thus produce only one rule instance each:

if $\neg\mathsf{T}_0(x_1, ..., x_n)$ **then**
$Stop$
end if
if $\mathsf{T}_1(x_1, ..., x_n)$ **then**
$x \leftarrow \mathsf{A}_1(x)$
end if
...
if $\mathsf{T}_K(x_1, ..., x_n)$ **then**
$x \leftarrow \mathsf{A}_K(x)$
end if

Using previous notations, rule R_0 has $T_0(\alpha, x) = \neg\mathsf{T}_0(x)$ with $Stop$ as action, while for $k \in \{1, ..., K\}$ rule R_k has $T_k(\alpha, x) = \mathsf{T}_k(x)$, $A_k(\alpha, x) = \alpha$ and $B_k(\alpha, x) = \mathsf{A}_k(x)$. This is obviously equivalent to the WHILE program considered.

In the (admittedly rare) case where static rules are not allowed, this is still possible using variable types. In our case, using the same execution algorithm, we use n meta-variables $\alpha_1, ..., \alpha_n$ contained in a vector α to write the following non-static BR program:

for all $i \in \{1, \ldots, n\}$ **do**
$create\_new\_type(t_i)$
$type(x_i) \leftarrow t_i$
$type(\alpha_i) \leftarrow t_i$
end for
if $type(\alpha_1) = type(x_1) \wedge \cdots \wedge type(\alpha_n) = type(x_n) \wedge \neg\mathsf{T}_0(\alpha_1, ..., \alpha_n)$ **then**
$Stop$
end if
if $type(\alpha_1) = type(x_1) \wedge \cdots \wedge type(\alpha_n) = type(x_n) \wedge \mathsf{T}_1(\alpha_1, ..., \alpha_n)$ **then**
$\alpha \leftarrow \mathsf{A}_1(\alpha)$
end if...
if $type(\alpha_1) = type(x_1) \wedge \cdots \wedge type(\alpha_K) = type(x_K) \wedge \mathsf{T}_K(\alpha_1, ..., \alpha_n)$ **then**
$\alpha \leftarrow \mathsf{A}_K(\alpha)$
end if

Using previous notations, the typing conditions are part of the definition of rule instances, and rule R_0 has $T_0(\alpha, x) = \neg\mathsf{T}_0(\alpha)$ with $Stop$ as action, while for $k \in \{1, ..., K\}$ rule R_k has $T_k(\alpha, x) = \mathsf{T}_k(\alpha)$, $A_k(\alpha, x) = \mathsf{A}_k(\alpha)$ and $B_k(\alpha, x) = x$. This is also equivalent to the WHILE program considered, since the type declarations ensure that the only viable rule instances will be the ones where $\alpha = x$.
■

Lemma 4. *Any WHILE program can be transformed into an equivalent WHILE program without embedded* **if** *statements.*

Proof. We prove the lemma by reasoning on the **ifblocks**. We consider the assertion: Any **ifblock** can be transformed into a finite sequence of **ifblocks** without embedded **if** statements.
We reason inductively on the depth of the deepest embedded **if** statement. If it is 0 or 1, the property is trivial (one is a pass instruction and the other already of the correct form).
Let $n \in \mathbb{N}$ such that we can transform any **ifblock** of depth n into a sequence of **if** statements. Let us consider an **ifblock** of depth $n+1$. It can be written as:

if $\mathsf{T}_1(x)$ **then**
$x \leftarrow \mathsf{A}_1(x)$
if $\mathsf{T}_2(x)$ **then**
$x \leftarrow \mathsf{A}_2(x)$
ifblock$(\mathsf{T}_3, \mathsf{A}_3, x)$
end if
end if

where **ifblock**(T_3, A_3, x) is an **ifblock** of depth $n-1$. It is equivalent to the following:

if $\mathsf{T}_1(x) \wedge \mathsf{T}_2(x)$ **then**
$x \leftarrow \mathsf{A}_2(\mathsf{A}_1(x))$
ifblock$(\mathsf{T}_3, \mathsf{A}_3, x)$
end if
if $\mathsf{T}_1(x) \wedge \neg\mathsf{T}_2(x)$ **then**
$x \leftarrow \mathsf{A}_1(x)$
end if

Which is a sequence of two **ifblocks** of depth n. As we can tranform each of them into a sequence of **ifblocks** without

embedded **if** statements, we have the property for **ifblocks** of depth $n+1$, and the induction is valid.
The property applied to each **ifblocks** of a WHILE program transforms it into the form we wanted. ■

2.5 Turing-machine

While the above proof is sufficient to justify the Turing-completeness of BRs, we can also use a much more direct proof by exhibiting a rule set that simulates a universal Turing machine (UTM). Such a rule set is exhibited in Fig. 3.

We suppose the variables include the following:
- many (static) state objects of type "state": $q_1, \ldots, q_Q$
- many (static) symbol objects of type "symbol": $s_1, \ldots, s_S$
- a (static) finite set of terminal states of type "terminal": T_{er}
- a (static) blank symbol of type "symbol": s_b
- a (static) set of Turing rules of type "rules", of the form $(\text{state}_{\text{initial}}, \text{symbol}_{\text{initial}}, \text{right}|\text{left}|\text{stay}, \text{state}_{\text{next}}, \text{symbol}_{\text{written}})$:
  $R = \{(q_r^i, s_r^i, \text{act}_r, q_r^f, s_r^f) \mid \text{act}_r \in \{\text{"left"}, \text{"right"}, \text{"stay"}\}\}_r$
- the current state of type "state": q
- the length of the visible tape data, of type "length": l
- the current visible tape data of type "tape": $T = \{(i, s_i) \mid i \in \mathbb{N}, 0 \leq i \leq l-1\}$ where l is the length of the visible tape data
- the current place on the tape of type "position": p

We use the following meta-variables in the BR program that simulates a UTM:
- α_{qf} of type "state"
- α_{sf} of type "symbol"

The rule set to simulate a UTM is then written in a compact form:

R_1:
if
  $(q, T(p), \mathsf{act}, \alpha_{qf}, \alpha_{sf}) \in R$
then
  $q \leftarrow \alpha_{qf}$
  $T \leftarrow (T \setminus \{(p, T(p))\}) \cup \{(p, \alpha_{sf})\}$
  $\alpha_p \leftarrow$ ("position", $p \pm 1$)(Depending on the value of act)
  $\alpha_l \leftarrow$ ("length", $l \pm 1$)(Depending on the respective values of act, p and l)

R_2:
if
  $(q \in T_{er})$
then
  Stop;

Figure 3: Universal Turing Machine

This universal Turing machine has a straightforward input, and terminates correctly for any valid Turing machine. We have made simplifications for the sake of clarity: R_1 should clearly be at least three different rules each replacing act with one of {"left", "right", "stay"}, and its complete formally correct form would in fact have two more rules, to be able to increase the length of the tape as needed (using the variable s_b as necessary).

3 Applications

3.1 WHILE form with other execution algorithms

While we have used a very simple execution algorithm where conflict resolution is based on a fixed ordering of rules and variables, most BR execution algorithms are more complex. Almost all of those can simulate the basic algorithm (except in extreme cases, such as a conflict resolution strategy leading to a bounded number of execution loops). As such, they are also Turing-complete. For all of those, a WHILE form can be defined (although a canonical one might not always seem obvious).

The most common conflict resolution strategies combine at least the following three elements [9]:

- *Refraction* which prevents a rule instance from firing (being selected by the conflict resolution algorithm) again unless its condition clause has been reset.
- *Priority* which is a kind of partial order on rules, leading of course to a partial order on rule instances.
- *Recency* which orders rule instances in decreasing order of continued validity duration (when rule instances are created at run time, it is often expressed as increasing order of rule instance creation time).

These elements do not forbid the conversion of a rule set to a WHILE program. While the specifics depend on each execution algorithm, the broad strokes are that:

- *Refraction* results in the use of an additional boolean variable per rule instance in the WHILE program.
- *Priority* partially decides the order of the if-then-else of the WHILE program.
- *Recency* is the most complicated. An easy workaround would add an incremental integer variable per rule instance in the WHILE program and use a max() function in the tests of the if-then-else.

3.2 Structural Operational Semantics

Not all operational semantics for BR follow the structural approach introduced by Plotkin [7] in that they use small-step semantics. The standard semantics defined in W3C's Production Rule Dialect of the Rule Interchange Format (RIF-PRD) [9] do, but are tailored to the syntax and algorithm of the standard. Other attempts at creating operational semantics for BRs have focused on being compatible with either declarative semantics or ontologies (or both) [8, 13]. Such semantics keep the structure of the rule set divided into rules, which helps with making sense of complex rule sets and creating better user experiences.
At the price of losing the knowledge representation linked to the use of rules, we can use the canonical WHILE form to obtain a structural operational semantic interpretation that is not unique to a syntax or execution algorithm, which can allow for comparison of different BR programs. Furthermore, using this semantics makes proving properties of some rule programs easier, as the working memory will not include lists of rules anymore. In a way, this semantics is a tradeoff between data complexity (RIF-PRD has rule instances in working memory) and number of small steps taken (any semantics derived from a WHILE form will go through many steps corresponding to inactive rule instances). By keeping the conflict resolution separate from the evaluation of condition clauses, the execution of the rule set is algorithmically correct yet adds complexity to the semantics that is unnecessary in most cases. The comparison of a flowchart representation of each semantic analysis for the example of Fig. 1 is made in Fig. 4 to make

the difference easier to visualize. We keep the steps in both cases bigger than they could be, so that we do not get bogged down in the evaluation of condition clauses for example.

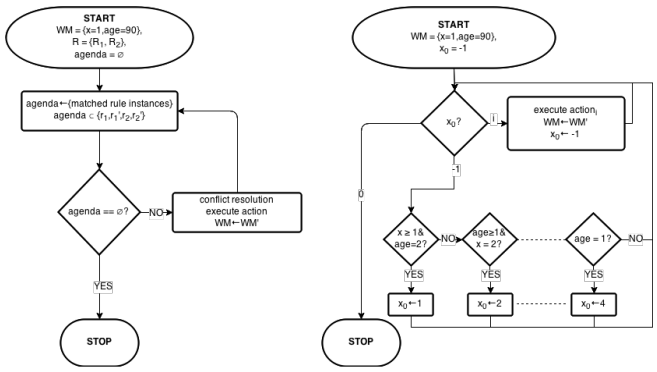


Figure 4: Representation of two semantic interpretations of the example in Fig. 1

4 Conclusion

Business Rules seem simple enough, repeatedly treating data according to a simple algorithm. The complexity of BR programs actually comes from the interpreters. In particular, almost any interpreter that uses a looping algorithm can make BR programs Turing-complete, as is the case with the simplistic algorithm we have presented in this paper. The proof of such is simple, yet it is a result that has been overlooked so far (to the best of our knowledge). The Turing-completeness of BR programs can have important theoretical implications: it links the usual Rules research on Inference Rules and ontologies with more traditional research on programming languages and computability.

We have further provided a tool to study this link in detail through the introduction of WHILE forms of BR programs. WHILE programs are simple programming languages that are easily linked to others, and can already provide some insight on their own. Using structural operational semantic analysis techniques on the transformed BR programs highlights a marked difference with the existing operational semantics of Business Rules. These might inspire new techniques for studying the properties of BR programs. The use of WHILE forms also provides a way to compare BR programs that use different interpreters without needing to examine the details of either interpreter, as long as a canonical WHILE form is agreed upon.

Both directions, bringing programming to Rules via WHILE forms and bringing Rules to programming through theoretical work, seem interesting enough to merit further work.

Acknowledgments

The first author (OW) is supported by an IBM France/ANRT CIFRE Ph.D. thesis award.

References

[1] Boley, H., Paschke, A., Shafiq, O., 2010: RuleML 1.0: the overarching specification of web rules. In Proceedings of the 2010 international conference on Semantic web rules (RuleML'10), Dean M., Hall J., Rotolo A., and Tabet S. (Eds.). Springer-Verlag, Berlin, Heidelberg, 162-178.

[2] Church, A., 1936: An Unsolvable Problem of Elementary Number Theory. American Journal of Mathematics 58, 345-363.

[3] Curtis, M.W., 1965: A Turing Machine Simulator. Journal of the Association for Computing Machinery, Vol. 12, No. 1 (January, 1965), pp. 1-13.

[4] Gandy, R., 1980: Church's Thesis and the Principles for Mechanisms. In The Kleene Symposium, Barwise H.J., Keisler H.J., and Kunen K. (eds). North-Holland Publishing Company. pp. 123-148.

[5] Harel, D., 1980: On folk theorems. Communications of the ACM 23, 7 (July 1980), 379-389.

[6] Minsky, M., 1972: Computation: Finite and infinite machines. Prentice-Hall, London.

[7] Plotkin, G. 1981: A structural approach to operational semantics. DAIMI FN-19, Computer Science Department, Aarhus University.

[8] Rezk, M., Kifer, M., 2012: Formalizing Production Systems with Rule-Based Ontologies. In Foundations of Information and Knowledge Systems (pp. 332-351). Springer Berlin Heidelberg.

[9] de Sainte Marie, C., Hallmark, G., Paschke, A., 2013: RIF Production Rule Dialect (Second Edition). W3C Recommendation, www.w3.org/TR/2013/REC-rif-prd-20130205/

[10] Shannon, C., 1956: A universal Turing machine with two internal states. In Automata studies, volume 34 of annals of mathematics studies, Shannon C., McCarthy J. (eds). Princeton University Press, Princeton, pp 157-165

[11] Turing, A., 1937: On computable numbers, with an application to the Entscheidungsproblem. Proceedings of the London Mathematical Society 42(1), 230-265.

[12] Turing, A., 1939: Systems of Logic Based on Ordinals (Ph.D. thesis). Princeton University. p. 8.

[13] Zaniolo, C., 1994: A Unified Semantics for Active and Deductive Databases. In Proceedings of the 1st International Workshop on Rules in Database Systems, Edinburgh, Scotland, pp 271-287. Springer London.