



RAPPORTS DE SYNTHÈSE
Année 2003-2005

Groupe MFI, GDR I3
Mise en forme : P Mathieu, LIFL

Table des matières

1 Rôles, Interactions et Organisations	7
1.1 Objectifs	7
1.2 Le modèle formel	7
1.3 Les interactions	7
1.4 En quoi est-ce un problème dur ?	8
1.5 Perspectives futures	8
2 Modélisation interactionnelle augmentée	9
2.1 Objectifs	9
2.2 Le modèle formel	9
2.3 Les interactions	9
2.4 En quoi est-ce un problème dur ?	10
2.5 Perspectives futures	10
3 Le modèle STROBE et ses évolutions	11
3.1 Objectifs	11
3.2 Le modèle formel	12
3.3 Les interactions	12
3.4 En quoi est-ce un problème dur ?	12
3.5 Perspectives futures	12
4 Validation de tâches avec B. Approches explicite et implicite	13
4.1 Objectifs	13
4.2 Le modèle formel	13
4.3 Les interactions	13
4.4 En quoi est-ce un problème dur ?	13
4.5 Perspectives futures	13
5 Plasticité des systèmes interactifs	15
5.1 Objectifs	15
5.2 Le modèle formel	15
5.3 Les interactions	15
5.4 En quoi est-ce un problème dur ?	16
5.5 Perspectives futures	16
6 Agent émotionnel pour le dialogue	17
6.1 Objectifs	17
6.2 Le modèle formel	17
6.3 Les interactions	18
6.4 En quoi est-ce un problème dur ?	18
6.5 Perspectives futures	18
7 Intentionnalité et agents situés	19
7.1 Objectifs	19
7.2 Le modèle formel	19
7.3 Les interactions	20
7.4 En quoi est-ce un problème dur ?	20
7.5 Perspectives futures	20

8 Interaction entre composants logiciels	21
8.1 Objectifs	21
8.2 Le modèle formel	21
8.3 Les interactions	22
8.4 En quoi est-ce un problème dur ?	22
8.5 Perspectives futures	22
9 Modélisation de Scénarios pédagogiques appliqués au eLearning	23
9.1 Objectifs	23
9.2 Le modèle formel	24
9.3 Les interactions	24
9.4 En quoi est-ce un problème dur ?	24
9.5 Perspectives futures	24
10 La méthode TOOD pour la spécification formelle des interfaces homme-machine	25
10.1 Objectifs	25
10.2 Le modèle formel	25
10.3 Les interactions	26
10.4 En quoi est-ce un problème dur ?	26
10.5 Perspectives futures	26
11 Modélisation de l'interaction HM située	29
11.1 Objectifs	29
11.2 Le modèle formel	29
11.3 Les interactions	30
11.4 En quoi est-ce un problème dur ?	30
11.5 Perspectives futures	30
12 Un modèle d'interaction pour la négociation	33
12.1 Objectifs	33
12.2 Le modèle formel	33
12.3 Les interactions	34
12.4 En quoi est-ce un problème dur ?	34
12.5 Perspectives futures	34
13 Introduction d'interactions dans les processus de décision markoviens	35
13.1 Objectifs	35
13.2 Le modèle formel	35
13.3 Les interactions	36
13.4 En quoi est-ce un problème dur ?	36
13.5 Perspectives futures	36
14 Protocoles de coordination et de conversation dans un système multi-agents	37
14.1 Objectifs	37
14.2 Le modèle formel	37
14.3 Les interactions	38
14.4 En quoi est-ce un problème dur ?	38
14.5 Perspectives futures	38
15 Instrumentation du suivi d'apprenants	39
15.1 Objectifs	39
15.2 Le modèle formel	39
15.3 Les interactions	40
15.4 En quoi est-ce un problème dur ?	40
15.5 Perspectives futures	40
16 Planification située et stratégie d'équipe	41
16.1 Objectifs	41
16.2 Le modèle formel	41
16.3 Les interactions	41
16.4 En quoi est-ce un problème dur ?	42
16.5 Perspectives futures	42

17 Gestion des interactions entre agents : application à la simulation de trafic routier	43
17.1 Objectifs	43
17.2 Le modèle formel	43
17.3 Les interactions	44
17.4 En quoi est-ce un problème dur ?	44
17.5 Perspectives futures	44
18 Intégration des connaissances dans les environnements virtuels pour l'annotation	45
18.1 Objectifs	45
18.2 Le modèle formel	45
18.3 Les interactions	46
18.4 En quoi est-ce un problème dur ?	46
18.5 Perspectives futures	46
19 Moyens de perception du trajet de formation par le tuteur en ligne	47
19.1 Objectifs	47
19.2 Le modèle formel	47
19.3 Les interactions	48
19.4 En quoi est-ce un problème dur ?	48
19.5 Perspectives futures	48
20 De l'usage des ontologies et de la norme Topic Maps pour le e-learning	49
20.1 Objectifs	49
20.2 Le modèle formel	49
20.3 Les interactions	50
20.4 En quoi est-ce un problème dur ?	50
20.5 Perspectives futures	50
21 Modèle de dialogue entre agents	51
21.1 Objectifs	51
21.2 Le modèle formel	51
21.3 Les interactions	51
21.4 En quoi est-ce un problème dur ?	52
21.5 Perspectives futures	52
22 Modélisation des conflits dans l'activité de pilotage	53
22.1 Objectifs	53
22.2 Le modèle formel	53
22.3 Les interactions	54
22.4 En quoi est-ce un problème dur ?	54
22.5 Perspectives futures	54
23 Processus formel de raffinement pour la conception de protocoles et des agents	55
23.1 Objectifs	55
23.2 Le modèle formel	55
23.3 Les interactions	55
23.4 En quoi est-ce un problème dur ?	56
23.5 Perspectives futures	56
24 Une interface conversationnelle pour une aide intelligente	57
24.1 Objectifs	57
24.2 Le modèle formel	57
24.3 Les interactions	58
24.4 En quoi est-ce un problème dur ?	58
24.5 Perspectives futures	58
25 L'intelligence Artificielle Distribuée appliquée aux processus physiques	59
25.1 Objectifs	59
25.2 Le modèle formel	59
25.3 Les interactions	59
25.4 En quoi est-ce un problème dur ?	60
25.5 Perspectives futures	60

Introduction

Le groupe de travail MFI du GDR I3 s'est fixé comme objectif d'étudier les modèles formels d'interactions à travers les trois disciplines que sont les systèmes Multi-Agents, l'Interface homme-machine et les environnements intelligents pour l'apprentissage humain. Les responsables du groupe sont :

- P Mathieu, LIFL, Lille 1, pour la communauté SMA
- C Kolski, LAMH, UVHC, pour la communauté IHM
- P Trigano, HEUDYASIC, UTC Compiègne, pour la communauté EIAH

Le groupe se réunit 2 à 3 fois par an durant une journée pendant laquelle plusieurs exposés et discussions sont proposées.

Pour permettre, à travers une présentation standardisée et commune, de montrer ce qui se fait sur le sujet et de faciliter la comparaison des approches respectives, nous avons décidé de réaliser un rapport d'activité du groupe contenant une fiche par exposé qui aura été effectué au sein du groupe. Notre objectif à termes est de sélectionner les plus pertinents afin d'en faire un numéro spécial d'une revue.

Nous demandons donc lors de chaque réunion, à chaque participant, de fournir une page recto-verso standardisée par des rubriques imposées présentant son exposé/travail de manière synthétique.

C'est ce document que vous avez entre les mains.

Les responsables du groupe

Un protocole d'interaction est ainsi défini à partir des éléments suivants (figure 5.1) : le nom du protocole d'interaction, une description textuelle sommaire du but de ce protocole, la liste des micro-rôles intervenant dans l'interaction et le symbole géométrique qui leur est associé, les ontologies des messages échangés, une liste d'informations nécessaires en entrée et produites en sortie, un graphe d'interaction regroupant les informations comme le déroulement temporel de la conversation, la synchronisation et la nature des messages échangés, les compétences nécessaires aux différents micro-rôles.

1. Rôles, Interactions et Organisations

Yann SECQ

Equipe SMAC, LIFL UMR 8022, USTL

secq@lifl.fr

http://www.lifl.fr/SMAC

Domaine de recherche : Conception de Systèmes Multi-Agents

Résumé

Nous proposons une conception des systèmes multi-agents se reposant sur les notions de rôles, d'organisations et d'interactions. Les protocoles d'interaction sont introduits pour automatiser la gestion des interactions multi-partites dans lesquels différents rôles peuvent intervenir. Tandis que la notion d'organisation permet de structurer la société d'agent en répartissant le contrôle à différents niveaux de l'organisation.

1.1 Objectifs

De nombreux travaux ont été effectués pour spécifier des protocoles d'interaction. Récemment, AgentUML[OPB00] a été défini comme une extension d'UML, pour spécifier les conversations entre agents, notamment en spécialisant les diagrammes de séquence. Cependant, ces spécifications nécessitent l'interprétation de développeurs, qui doivent ensuite les traduire dans leur propre système. Les travaux de Labrou et Finin[RSCP99] explorent l'utilisation des Réseaux de Petri Colorés[Jen92] (RdPC) pour modéliser les conversations entre agents. Dans [MSH02], la même approche est utilisée, mais la notion de Réseau de Pétri Colorés Récursifs est introduite pour faciliter la composition de conversations.

Nos travaux suivent le même principe : représenter l'interaction de manière globale, et utiliser un formalisme reconnu et établi. Cependant, contrairement aux travaux précédents, notre objectif est de produire une *spécification exécutable*. C'est-à-dire, une description du protocole d'interaction qui peut être ensuite directement intégrée dans un système s'exécutant. De plus, les RdPC sont indéniablement adaptés à la modélisation de processus concurrents, et fournissent un intéressant formalisme graphique, mais ils ne s'utilisent malheureusement pas très aisément. C'est pourquoi il nous paraît préférable de définir un langage adapté à la modélisation des protocoles d'interactions, et d'utiliser ensuite un mécanisme de projection permettant de traduire ce langage vers les RdPC.

1.2 Le modèle formel

Ce modèle a pour but de faciliter la spécification, la vérification et le déploiement de protocoles d'interactions au sein de systèmes multi-agents. Sur l'ensemble de ces phases, le concepteur a pour tâche de réaliser la spécification, les autres phases étant *automatisées*. Pour cela, nous avons défini un formalisme représentant la vue globale d'un protocole d'interaction, et un mécanisme de projection qui transforme cette vue globale en un ensemble de vues locales à chaque agent. Une vue locale est réalisée sous la forme d'un automate capable de traiter les messages d'un des rôles intervenant dans l'interaction.

1.3 Les interactions

Notre modèle repose sur la notion de compétence, de micro-rôle et de graphe d'état de conversation. Un protocole d'interaction spécifie formellement le déroulement d'une conversation (considérée comme une loi sociale) entre différentes entités, c'est-à-dire la nature des messages échangés, le flux de ces messages et les compétences que doivent mettre en œuvre les entités à chaque étape de la conversation. Ces entités correspondent à des micro-rôles, et sont caractérisées par leur nom et leurs compétences. On utilise un graphe pour représenter le déroulement de la conversation : les nœuds représentent les micro-rôles qui peuvent être associés à une interface de compétence, et les arcs à un envoi de message entre deux micro-rôles (typés par un motif de message). Un état dans le graphe caractérise donc partiellement un micro-rôle, en indiquant le(s) message(s) qu'il traite/génère, ainsi que la compétence qu'il doit utiliser pour traiter/produire ce(s) message(s), à un instant donné de la conversation.

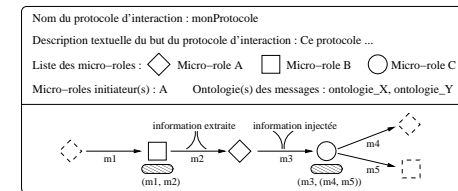


FIG. 1.1 – Cartouche spécifiant les protocoles d'interactions

Le graphe d'interaction est annoté en différents endroits, le détail de ces annotations est donné dans la dernière section. Ce qu'il est important de comprendre est que le concepteur dispose ainsi d'une vue globale de l'interaction : le flux des messages, leur nature (le typage de ces messages correspond aux annotations attachées aux arcs), les compétences nécessaires et les informations utilisées ou produites. D'autre part, toute l'information nécessaire à la gestion du protocole d'interaction est ici centralisée et peut être utilisée pour effectuer la génération de code nécessaire à la gestion de cette interaction pour chacun des micro-rôles. Le concepteur n'a donc qu'à définir le protocole d'interaction en utilisant un outil graphique, le mécanisme de projection se charge de la génération des descriptions pour chacun des micro-rôles, et le modèle générique d'agent peut ensuite exécuter ces descriptions.

1.4 En quoi est-ce un problème dur ?

La plupart des plateformes multi-agents ne proposent pas d'automatisation de la gestion des interactions multi-partites. C'est déjà un problème en soi, mais cela devient encore plus épineux dans le cadre de systèmes ouverts, utilisant différents modèles d'agents et organisationnels. Nous pensons que tant qu'il n'y aura pas d'avancée sur les protocoles d'interaction, la notion de systèmes multi-agents ouverts restera purement théorique.

1.5 Perspectives futures

Une évolution du domaine serait possible si une prise de conscience s'effectuait au niveau de la FIPA pour choisir dans un premier temps un outil de description des protocoles d'interaction suffisant formel pour que des transformations permettant de générer le code puissent être appliquées. Dans un second temps, le même travail devrait être effectué au niveau d'une ontologie des systèmes multi-agents, qui pourrait alors servir de base commune pour le développement de systèmes réellement ouverts.

Bibliographie

- [Jen92] Kurt Jensen. Coloured petri nets - basic concepts, analysis methods and practical use, vol. 1 : Basic concepts. In *EATCS Monographs on Theoretical Computer Science*, pages 1–234. Springer-Verlag : Berlin, Germany, 1992.
- [MSH02] Hamza Mazouzi, Amal El Fallah Seghrouchni, and Serge Haddad. Open protocol design for complex interactions in multi-agent systems. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 517–526. ACM Press, 2002.
- [OPB00] J. Odell, H. Parunak, and B. Bauer. Extending uml for agents, 2000.
- [RSCP99] Tim Finin Yannis Labrou R. Scott Cost, Ye Chen and Yun Peng. Modeling agent conversations with colored petri nets. In *Third Conference on Autonomous Agents (Agents-99), Workshop on Agent Conversation Policies*, Seattle, May 1999. ACM Press.

2. Modélisation interactionnelle augmentée

Marc-Philippe Huget

Leibniz-IMAG/MAGMA 46, avenue Félix Viallet 38031 Grenoble Cedex
 Marc-Philippe.Huget@imag.fr http://www-leibniz.imag.fr/MAGMA

Domaine de recherche : SMA, protocoles, interactions entre agents

Résumé

L'interaction entre agents telle que nous la considérons est réalisée par l'intermédiaire de protocoles d'interaction qui décrivent les séquences de messages valides dans le cadre de l'interaction. Ces protocoles sont représentés à l'aide de techniques de description formelle. Le principal défaut que l'on rencontre sur ces formalismes est le manque d'information : il s'agit avant tout de décrire les échanges de messages sans faire apparaître les conditions sur l'exécution des messages ou les intentions des agents à utiliser tel ou tel message. Nous présentons ici un automate à états fini étendu afin de prendre en compte les notions de conditions, de dimension sociale, d'intentionnalité ou encore de contexte dans l'interaction.

2.1 Objectifs

La représentation des interactions par l'intermédiaire de protocoles d'interaction est maintenant commun mais, la plupart du temps, la représentation se borne à décrire les séquences de messages en omettant les informations concernant le protocole comme les conditions d'exécution, les intentions des agents, etc. C'est par exemple le cas pour le protocole de Contract Net [SD81] où la sémantique du protocole est donnée dans le texte et non sur le protocole. Certains formalismes de description des protocoles considèrent eux aussi des adjonctions mais en nombre limité : par exemple les conditions dans les automates à états finis étendus [LT87] et les engagements des agents dans les jeux de dialogue [McB02]. Ce besoin de créer un formalisme ayant un plus grand pouvoir d'expression est de permettre d'intégrer plus facilement le protocole dans l'agent en connaissance de cause, et de pouvoir générer du code à partir du protocole qui ne se borne pas à un squelette du protocole (comme c'est le cas dans les systèmes distribués [Chu89]) mais ajoute aussi la sémantique associée aux transitions.

2.2 Le modèle formel

Le modèle formel que nous utilisons ici est un automate à états fini que nous augmentons avec un certain nombre d'information. Tout d'abord, il est associé au protocole une intention qui motive l'utilisation de ce protocole. Nous suivons pour cela une logique intentionnelle comme celle proposé par Cohen et Levesque [CL90]. Ensuite, à chaque état de l'automate sont associées des assertions sous forme logique qui vérifient si certaines conditions sont valides. Enfin, il est ajouté des conditions et de la sémantique sur chacune des transitions. Les conditions donnent les conditions à satisfaire pour pouvoir traverser la transition. Ces conditions font référence aux connaissances et aux croyances de l'agent, à ces intentions et engagements, à sa position sociale, ainsi qu'à l'état du monde et les précédents messages envoyés. Il est donc nécessaire de mettre en place un mécanisme d'exception si les conditions étaient insatisfaites. La description des conditions se font en logique déontique, en logique intentionnelle, en logique Belief-Desire-Intention et en logique du premier ordre. Dans la mesure du possible, nous privilégions de ne pas utiliser des formalismes qui pourraient avoir des conséquences sur le type d'architecture de l'agent. Nous préférons utiliser des mots-clefs renvoyant à des concepts. La sémantique de l'envoi et de la réception des messages n'est pas liée à de possibles présupposés sur l'agent destinataire mais décrit simplement les modifications apportées au monde et ce qui peut être considéré comme valide après l'émission du message.

2.3 Les interactions

Les interactions telles que nous les utilisons sont entre agents logiciels et simulent les conversations humaines, i.e. les agents utilisent des actes de langage pour s'exprimer. Toutefois, ces agents ne manipulent pas des langages naturels par souci d'efficacité. En effet, les agents servent la plupart du temps à résoudre des problèmes de coordination et de coopération. L'utilisation du langage naturel et les difficultés d'interprétation seraient préjudiciables à la bonne tenue des coordinations et des coopérations.

L'interaction se différencie ici de la communication dans les systèmes distribués car l'expéditeur et les destinataires modifient leur état après envoi et réception de messages. Par exemple, si un agent envoie le message FIPA `inform`, il considérera que le destinataire croit l'information dès réception du message.

2.4 En quoi est-ce un problème dur ?

La description des protocoles pour les interactions entre agents héritent des travaux dans les systèmes distribués où il s'agit avant tout de décrire quel message peut suivre après un autre. La sémantique associée est donnée dans de volumineux document (appelés *Request For Comments*). En suivant cette voie, les concepteurs de protocoles entre agents ont laissé de côté l'aspect sémantique qui est plus ou moins bien traité. En réduisant l'information sur les protocoles, on réduit la possibilité de générer automatiquement des protocoles pour un agent. La difficulté va être de pouvoir concilier d'un côté une augmentation du pouvoir d'expression et de l'autre de rester le plus indépendant des langages de programmation et de ne pas influencer sur l'architecture de l'agent.

2.5 Perspectives futures

La définition d'une technique de description formelle n'est pas suffisante pour attester de l'intérêt, en particulier dans le domaine des protocoles où il est nécessaire de faire de la validation. Nos deux prochains objectifs seront de fournir un outil pour la conception de protocoles à l'aide de ce formalisme puis de fournir un algorithme pour la validation des protocoles. La validation passera par l'utilisation d'un model checker comme Spin [Hol97]. Il est nécessaire de fournir un algorithme pour la traduction vers le formalisme utilisé dans Spin. Une question reste de plus en suspens concernant les possibilités d'avoir l'équivalence forte et l'équivalence observationnelle pour ce formalisme, i.e. est-il possible de vérifier si deux protocoles représentés à l'aide de formalismes sont équivalents. L'équivalence forte et observationnelle est utilisée dans le cadre de la validation.

Bibliographie

- [Chu89] Peil-Ying Mark Chu. *Towards Automating Protocol Synthesis and Analysis*. PhD thesis, Ohio State University, 1989.
- [CL90] P. R. Cohen and H. J. Levesque. Rational interaction as the basis for communication. In P. R. Cohen, J. Morgan, and Martha E. Pollack, editors, *Intentions in Communication*, pages 221–256. MIT Press, 1990.
- [Hol97] Gerard J. Holzmann. The model checker SPIN. *IEEE Transactions on Software Engineering*, 23(5), May 1997.
- [LT87] Nancy A. Lynch and Mark R. Tuttle. Hierarchical correctness proofs for distributed algorithms. In *Proceedings of the 6th Annual ACM Symposium on Principles of Distributed Computing (PODC'87)*, pages 137–151, Vancouver, British Columbia, Canada, August 1987. Technical Report MIT/LCS/TR-387, Laboratory for Computer Science, Massachusetts Institute of Technology.
- [McB02] Peter McBurney. *Rational Interaction*. PhD thesis, Department of Computer Science, University of Liverpool, 2002.
- [SD81] R.G. Smith and R. Davis. Framework for cooperation in distributed problem solving. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(1) :61–70, January 1981.

3. Le modèle STROBE et ses évolutions

Clément Jonquet

LIRMM – Univ. Montpellier II, 161 Rue Ada 34392 Montpellier Cdx 5

jonquet@lirmm.fr

http://www.lirmm.fr/~jonquet

Domaine de recherche : SMA, communication agent, langages, modèle du partenaire, generation de service

Résumé

Le modèle STROBE a été originellement proposé par Cerri [Cer96, Cer99] comme modèle de communication agent. Il décrit la communication par des flots (STReams) de messages échangés entre agents représentés comme des objets (OBjects) qui interprètent ces messages dans de multiples environnements (Environment). STROBE modélise la notion de point de vue multiple et de représentation du partenaire par le concept d'Environnement Cognitif [Cer96]. Chaque message d'une conversation est évalué dans un environnement cognitif dédié à l'interlocuteur. Ainsi, ces environnements¹ sont modifiés au fur et à mesure des conversations et correspondent aux connaissances situées d'un agent. Récemment [JC03, JC04], le modèle a été modifié pour rajouter le concept d'Interpréteur Cognitif. La structure d'environnement mémorise désormais un interpréteur qui est également dédié à la conversation. Cet interpréteur est dit cognitif car, comme l'environnement, il est capable d'évoluer dynamiquement au fur et à mesure des conversations. Ainsi, l'interprétation des messages d'une conversation est faite dans un environnement donné et avec un interpréteur donné tous les deux dédiés à cette conversation. Considérant, qu'un langage est caractérisé par un couple environnement + interpréteur (cf. section 3.4), le modèle STROBE permet donc aux agents de développer pour chacune de leurs accointances un langage spécifique. Ce qui en fait un modèle "social" fort et très dynamique.

3.1 Objectifs

La notion de service est désormais centrale dans la conception de systèmes distribués. Le Web aujourd'hui se développe grandement dans cette direction : *Web Services, Grid Services*. Cependant, fournir un service ne peut plus seulement se limiter à exécuter une tâche simple pour un client donné telle que c'est le cas dans le modèle classique client/serveur où avec les Web Services aujourd'hui. Aujourd'hui, il faut envisager des services dynamiques capables de s'adapter et de se composer pour en fournir d'autres. L'exécution d'un service ne peut plus seulement se voir comme un "appel distant de procédure", le service doit être fourni par un système, imperméable au client, capable de détecter les besoins des utilisateurs et de s'y adapter. Il ne s'agit plus de simplement fournir des services, mais il s'agit de permettre la **génération dynamique de service**.

Les Systèmes Multi-Agents (SMA) semblent naturellement adaptés à la conception de systèmes orientés service. La communication agent est fondamentale dans la conception de SMA. Elle se caractérise par deux étapes. L'une consiste à définir des langages de communication, et l'autre à modéliser les conversations. Pour la première étape, la plupart des approches, hautement inspirées des travaux de Allen, Cohen et Perrault, considèrent que l'agent possède des états mentaux (e.g. Believe Desire Intention) et que la communication change ces états. Pour la seconde étape, la méthode la plus courante est l'utilisation de protocole d'interaction ou de règle de conversation, qui représentent la structure d'une conversation et spécifient les messages qui doivent être échangés. Ces méthodes font des présuppositions fortes sur les agents comme le langage partagé, le concept de sincérité, et elles limitent souvent les conversations qui peuvent intervenir entre plusieurs agents. Il n'y a pas d'interprétation dynamique des messages, tout est fait de façon à encadrer au maximum les interactions. Cela ne pas toujours suffisant. Face à cela, des approches dites plus "dialogiques" existent. Elles visent à introduire plus de dynamique dans l'interprétation des messages, à ne pas focaliser sur l'état mental, mais plutôt sur le sens caché du message. **L'intention d'un message n'y est pas prédéterminé, mais co-construite au fur et à mesure de la conversation**. Les agents peuvent dialoguer (avoir des conversations) les uns les autres, ou avec des agents humains de façon à construire/générer des services dynamiquement. C'est dans ces approches que le modèle STROBE s'inscrit.

¹Le terme environnement est utilisé ici avec sa signification en langage de programmation, c'est à dire comme structure stockant des variables et des valeurs, et non pas avec sa signification en SMA, c'est à dire comme le monde dans lequel évoluent des agents et avec lequel ils interagissent.

3.2 Le modèle formel

$$\begin{aligned} AGENT_x &= \{ENV_x^x + \{ENV_j^x\}^*\} \forall j \text{ agent} \\ ENV_j^i &= \{INT_j^i + \{BIND_j^i\}^*\} \\ INT_j^i &= \lambda : EXP \rightarrow evaluate_j^i(EXP, ENV_j^i) \\ BIND_j^i &= \{(var, val)^*\} \end{aligned} \quad \text{Avec, } ELT_Y^X = \begin{cases} \text{élément local de X dédié à Y} & \text{si } X \neq Y \\ \text{élément global de X} & \text{si } X = Y \end{cases}$$

Ce qui signifie qu'un agent x , $AGENT_x$, est considéré comme un ensemble d'environnements, non vide, car un agent a obligatoirement un environnement global (privé), ENV_x^x , et éventuellement un ensemble d'environnements correspondants aux modèles de ses partenaires, ENV_j^x . Un environnement, ENV_j^i , est un ensemble de liaisons avec au moins, un interpréteur, INT_j^i , et le reste des liaisons, $BIND_j^i$, du type paire (variable – valeur). Un interpréteur, INT_j^i , est une procédure $evaluate_j^i$, qui interprète une expression EXP , dans l'environnement correspondant, ENV_j^i .

3.3 Les interactions

Les interactions qui nous intéressent sont des communications directes asynchrones entre agents. Les messages échangés lors de ces interactions sont de la forme : $MSG = \{AGENT_s, AGENT_r, PERFORM, CONTENT\}$ et $PERFORM = \{assertion, ack, request, answer, order, executed\}$. Nous considérons que les agents se comportent comme des évaluateurs REPL : Chaque fois qu'un agent lit un message (Read), il sélectionne l'environnement et l'interpréteur dédiés à cette conversation, pour interpréter ce message (Eval), puis il envoie la réponse correspondante (Print) et attend le message suivant (Listen).

3.4 En quoi est-ce un problème dur ?

Construction dynamique de langage. Les langages sont caractérisés par les trois niveaux : *donnée, contrôle, interpréteur*. Le niveau donnée consiste à affecter des valeurs à des variables déjà existantes, ou à définir de nouvelles données. Le niveau contrôle consiste à définir de nouvelles fonctions par abstraction sur celles existantes. Le niveau interpréteur ou méta-niveau consiste à faire évoluer l'interpréteur lui même du langage. Les deux premiers niveaux sont accessibles dynamiquement assez facilement, cela consiste à modifier l'environnement, par contre le troisième est plus compliqué car il s'agit de modifier dynamiquement la machine d'exécution du langage (interprétation, ou compilation/exécution). Le modèle STROBE utilise la dynamique (réflexivité etc.) du langage applicatif et interprété Scheme, pour y arriver.

Génération dynamique de service. L'approche génération dynamique de service peut être vue comme un triplet $\{client, fournisseur de service, processus conversationnel\}$ d'où le rôle fondamental du modèle de communication qui supporte cette approche. Les conversations ne sont pas prédéterminées, et le client construit pas à pas ce qu'il désire en fonction des réactions du fournisseur. Délivrer un produit est simple, mais générer un service est compliqué. Nous ne détaillons pas cela ici, voir [JC04].

3.5 Perspectives futures

Plusieurs expérimentations du modèle STROBE ont déjà été proposées, comme l'apprentissage au méta-niveau par communication, ou la spécification dynamique de problème [JC04]. Ces expérimentations illustrent des larges classes de problèmes. Parallèlement, est apparu le concept de grille (Grid) qui, si il était limité à ses débuts au partage de ressource matérielle (calcul et capacité de stockage), s'étend aujourd'hui de plus en plus vers le partage de tout service au sein de communauté virtuelle. Ainsi, l'objectif est de fournir un *toolkit* (basé sur des expérimentations généralisés), pour la génération dynamique de service basé sur le modèle de communication et de représentation des agents qu'est STROBE.

Bibliographie

- [Cer96] Stefano A. Cerri. Cognitive Environments in the STROBE model. In *European Conference in Artificial Intelligence and Education, EuroAIED'96*, Lisbon, Portugal, 1996.
- [Cer99] Stefano A. Cerri. Shifting the focus from control to communication : the STReam OBjects Environments model of communicating agents. In Padget J.A., editor, *Collaboration between Human and Artificial Societies, Coordination and Agent-Based Distributed Computing*, volume 1624 of *Lecture Note in Artificial Intelligence*, pages 74–101. Springer-Verlag, Berlin Heidelberg New York, 1999.
- [JC03] Clément Jonquet and Stefano A. Cerri. Cognitive agents learning by communicating. In *Colloque Agents Logiciels, Coopération, Apprentissage et Activité Humaine, ALCAA'03*, pages 29–39, Bayonne, September 2003.
- [JC04] Clément Jonquet and Stefano A. Cerri. Agents communicating for dynamic service generation. In *1st International Workshop on GRID Learning Services, GLS'04*, pages 39–53, Maceio, Brazil, September 2004.

4. Validation de tâches avec B. Approches explicite et implicite

Yamine Ait Ameer
LISI-ENSMA, Université de Poitiers
yamine@ensma.fr

Domaine de recherche : Sémantique, preuve, IHM

Résumé

4.1 Objectifs

4.2 Le modèle formel

4.3 Les interactions

4.4 En quoi est-ce un problème dur ?

4.5 Perspectives futures

5. Plasticité des systèmes interactifs

Gaëlle Calvary, Alexandre Demeure
CLIPS-IMAG, Grenoble

Gaëlle.Calvary@imag.fr <http://iihm.imag.fr/calvary>

Domaine de recherche : IHM, Plasticité (adaptation au contexte d'usage)

Résumé

Avec l'informatique ambiante, l'hypothèse implicite de la plate-forme d'exécution unique et du lieu d'interaction fixe ne tient plus : l'utilisateur est mobile et met à profit, de façon opportuniste, son espace interactif pour accomplir sa tâche. Cette (r)évolution a motivé, en Interaction Homme-Machine, l'introduction d'une nouvelle propriété : la plasticité. Par plasticité, on entend la capacité d'une Interface Homme-Machine (IHM) à s'adapter à son contexte d'usage dans le respect de son utilisabilité. Le contexte d'usage est défini comme un triplet \langle Utilisateur, Plate-forme, Environnement \rangle . Notre approche consiste à capitaliser des IHM méta-décrites pour à l'exécution, lorsque le contexte d'usage change, sélectionner, composer, etc. ces IHM afin d'offrir à l'utilisateur le service attendu dans la qualité voulue. Les graphes conceptuels sont ici explorés pour la description et sélection des IHM. Un graphe des descriptions est proposé pour leur capitalisation.

5.1 Objectifs

La Plasticité des IHM[The01] a été introduite comme solution à un problème de variété, variabilité et imprévisibilité du contexte d'usage (\langle Utilisateur, Plate-forme, Environnement \rangle) en informatique ambiante. Elle dénote la capacité d'une IHM à s'adapter à son contexte d'usage dans le respect de son utilisabilité. Son ingénierie pose, en ricochet, des problèmes. Nous nous focalisons sur la méta-description des IHM pour leur capitalisation puis réutilisation à l'exécution lorsque le contexte d'usage change. Il s'agit, selon notre approche, de (1) rendre explicite dans chaque IHM ce qu'elle * est (tâche utilisateur offerte, structure et présentation), * requiert en termes de contexte d'usage (typiquement, en dispositifs d'entrée/sortie) et * satisfait en termes de propriétés d'utilisabilité; (2) capitaliser les IHM ainsi décrites pour (3) à l'exécution, lorsque le contexte d'usage change, adapter l'IHM en sollicitant, si nécessaire, des IHM préfabriquées. Nous explorons les graphes conceptuels et proposons un graphe des descriptions[DC03].

5.2 Le modèle formel

Un graphe conceptuel est un type de réseau sémantique[Sow84]. Un réseau sémantique est un système de représentation graphique des connaissances basé sur des noeuds interconnectés par des arcs. Un graphe conceptuel est un graphe biparti étiqueté par des items lexicaux : des concepts et relations entre concepts (Figure 1a).

Concepts et relations sont définis dans un support. Nous explorons les graphes conceptuels pour décrire une IHM à tout niveau d'abstraction ainsi que ses requis et propriétés.

Grâce à l'opération de projection offerte par les graphes conceptuels, on peut savoir si l'information présente dans un graphe est déductible d'un autre. Ainsi, à l'exécution, lorsque le contexte d'usage change et devient incompatible de l'IHM courante, une recherche d'IHM peut s'opérer parmi les IHM préfabriquées, méta-décrites et capitalisées. La recherche consiste, via une opération de projection, à identifier dans la base des IHM capitalisées (le graphe des descriptions) les IHM dont la description coïncide avec celle de la requête (Figure 1b).

5.3 Les interactions

Un système interactif est un assemblage de composants, donnant lieu à deux types d'interaction : système, d'une part (entre composants); utilisateur, d'autre part (composant-utilisateur). Notre priorité aujourd'hui est de décrire les composants pour ensuite pouvoir raisonner sur leurs interactions. Nous nous concentrons sur les spécificités IHM, à savoir fonction, structure, présentation, requis et propriétés. Il s'agira, bien entendu à terme, d'intégrer la description de contrat

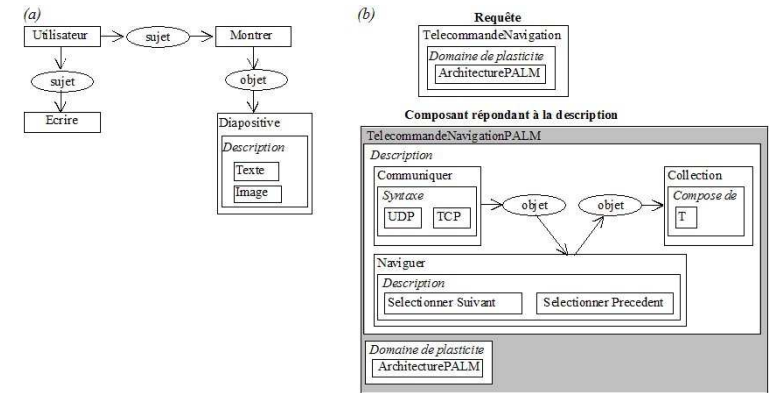


FIG. 5.1 – Exemple de graphe conceptuel exprimant qu'un utilisateur écrit et montre une diapositive contenant du texte et au moins une image. (b) Formulation d'une requête (en haut) décrivant le composant recherché, ici une télécommande de navigation fonctionnant sur PALM. Le composant TélécommandeNavigationPALM (en bas) répond à la description.

syntactique pour permettre la connexion logicielle. Dans nos descriptions, nous appliquons le principe de séparation des préoccupations : chaque niveau d'abstraction (tâche utilisateur, espaces de travail, interacteurs) définit une perspective particulière sur une même IHM et fait l'objet d'une description. Les perspectives sont liées entre elles par le biais de transformations, souvent motivées par une priorité donnée à certains critères d'ergonomie (typiquement la compatibilité par rapport à la tâche). Les transformations font elles-mêmes l'objet de descriptions pour conserver à l'exécution les justifications de la conception. Elles soutiendront aussi la gestion de l'état de l'interaction (sauvegarde et reprise) lors de l'adaptation : les interacteurs pointant vers le modèle des tâches, il devient possible, à l'exécution, de suivre l'état d'avancement de l'utilisateur dans la réalisation de sa tâche et, en conséquence, de restaurer cet état post-adaptation pour éviter à l'utilisateur d'avoir à recommencer.

5.4 En quoi est-ce un problème dur ?

Au-delà du choix des outils conceptuels et implémentationnels, la difficulté tient ici à l'approche elle-même : le défi est de pouvoir tenir à l'exécution des raisonnements jusqu'ici menés à la conception et majoritairement soutenus par la réflexion. La migration de ces raisonnements du concepteur au système nécessite leur explicitation puis formalisation. Si les aspects méthodologiques (étapes de conception) et fonctionnels (tâche utilisateur) semblent aujourd'hui maîtrisés, l'ancrage de l'ergonomie reste un point dur.

5.5 Perspectives futures

Le rapprochement des communautés IHM et IDM (MDE) est une suite logique au travail. Le tandem est aujourd'hui établi : les métamodèles sont en cours de définition ; les outils en cours d'évaluation. Il nous faut à présent nous ouvrir à la communauté système pour mettre à profit les avancées en architectures à services (SOA). Notre vision est de combiner Modèles (MDE) et Services (SOA) pour pleinement couvrir la variété, variabilité et imprévisibilité du contexte d'usage. Il restera alors à soutenir la composition d'IHM. Si les critères d'ergonomie permettent l'évaluation critique d'une composition (défaut d'homogénéité-cohérence par exemple) et, en conséquence, l'orientation vers telle solution plutôt que telle autre, il nous faut définir des transformations et des outils permettant de régler ou d'atténuer les anomalies constatées. Les styles en sont un exemple.

Bibliographie

- [DC03] A. Demeure and G. Calvary. Le modèle d'évolution en plasticité des interfaces : Apport des graphes conceptuels. In *Actes de la 15ème conférence francophone sur l'Interaction Homme-Machine, IHM 2003, Caen, France*, pages 80-87, Nov 2003.
- [Sow84] J.F. Sowa. *Conceptual Structures Information Processing in Mind and Machine*. Addison Wesley, 1984.
- [The01] D. Thevenin. *Adaptation en Interaction Homme-Machine : le cas de la Plasticité*. Thèse pour l'obtention du titre de docteur en informatique, Université Joseph Fourier, Grenoble I, 2001.

6. Agent émotionnel pour le dialogue

Luca Bisognin, Sylvie Pesty

Equipe MAGMA, LEIBNIZ/IMAG,

46, av. Félix Viallet F-38031 Grenoble Cedex

{luca.bisognin,sylvie.pesty}@imag.fr <http://www-leibniz.imag.fr/A3>

Domaine de recherche : SMA, agents conversationnels, modélisation des émotions

Résumé

Depuis 1998, nos travaux sur les agents conversationnels et la modélisation affective de l'apprenant en environnement d'apprentissage nous ont amenés à reconsidérer le traitement du dialogue entre agents naturels et artificiels en tentant d'y introduire des artefacts d'ordre subjectif tels que les émotions. En nous appuyant sur un formalisme BDI (Beliefs, Desires, Intention) propre à la modélisation d'agents rationnels autonomes, nous avons entrepris de remodeler notre architecture d'agent conversationnel pour la rapprocher de ce formalisme. Aussi, dans ce contexte, les interactions entre agents naturels et artificiels ne sont plus considérées comme des processus de type encodage/transport/décodage imposant aux agents un seul et même langage mais comme des échanges permettant aux interactants de co-construire un terrain commun à partir duquel les agents en interaction pourront extraire un potentiel de sens que chacun pourra, à son tour, interpréter. La modélisation des émotions apporte à cette conception des interactions, les instruments nécessaires à l'interprétation différenciée du potentiel de sens.

6.1 Objectifs

Dans le cadre des travaux entrepris depuis 1998 sur les agents conversationnels au sein de l'équipe MAGMA, nous avons implémenté un agent conversationnel dédié à la recommandation de films sur l'Internet [Chi02]. Dès lors, il nous est apparu qu'un traitement exclusivement linguistique, aussi fin que possible, ne suffisait pas pour maintenir un dialogue naturel entre un agent virtuel et un être humain. Par ailleurs, dans le cadre de la modélisation de l'apprenant au sein d'un système d'apprentissage humain et didactique [JPV03], nous avons dégagé plusieurs propriétés relatives à la modélisation d'états psychologiques tels que les affects et les émotions dans les interactions homme-machine. Les perspectives offertes par ces travaux nous ont permis de reconsidérer la notion de dialogue en appuyant le traitement rationnel de l'information qui concourt à son interprétation, à son contrôle et à sa génération sur des considérations d'ordre extra-rationnelles¹ de type émotionnel et subjectif. Cette étude consiste donc à :

1. identifier et mettre en œuvre des mécanismes d'interactions sociales entre agents basés sur les émotions (essentiellement du point de vue cognitif)
2. concevoir une architecture générique d'agent conversationnel autonome.

6.2 Le modèle formel

L'approche formelle employée ici s'inspire des architectures d'agents de type BDI (Beliefs, Desire, Intention). Cette approche permet de prendre en compte des aspects cognitifs au sein d'une architecture formelle d'agent rationnel autonome mettant en relation :

1. les croyances ou *Beliefs* qui correspondent aux connaissances de l'agent sur l'environnement ;
2. les désirs ou *Desires* qui correspondent aux états de l'environnement que l'agent *désirerait* voir se réaliser sur la base de ses croyances ;
3. les intentions qui correspondent aux projets que l'agent tente de réaliser en vue de satisfaire ses désirs.

Cette approche des fonctions cognitives d'un agent rationnel autonome donne ainsi lieu à une représentation formelle composée :

¹Ce qui ne signifie pas qu'elles soient irrationnelles pour autant.

- d'un langage du premier ordre.
- d'opérateurs portant sur les savoirs et les croyances de l'agent (composante informationnelle).
- d'opérateurs portant sur les buts, les plans et les intentions (composante motivationnelle).
- d'opérateurs portant sur le résultat et l'évolution des actions d'un agent (composante dynamique).

Le formalisme BDI fournit ainsi une prise directe avec des notions jusque-là cantonnées au domaine de la philosophie et de la psychologie et offre un cadre de formalisation, d'implémentation et d'évaluation pour la conception d'agents cognitifs autonomes.

6.3 Les interactions

Dans cette étude les interactions sont considérées du point de vue du dialogue homme-machine. Il s'agit donc d'interactions langagières se voulant très proches du langage humain. En outre, reprenant la thèse défendue par Guillaume Chicoine sur la dynamique du dialogue, la notion d'interaction est à considérer du point de vue de la co-construction de sens : l'interaction ne correspond pas seulement au processus d'encodage/transport/décodage du message mais devient le lieu d'un terrain commun sur lequel les interactants co-construisent un potentiel de sens qui sera à son tour interprété par chacun des agents pour en extraire un sens. Cette conception de l'interaction, appartenant au dialogisme de [BP99], permet de considérer des agents autonomes dans un environnement complexe et fortement différencié. Dans ce cadre, les interactions langagières deviennent complexes et la modélisation des états psychologiques tels que les émotions constitue un passage nécessaire à l'identification et le traitement d'indices propres à l'interprétation du potentiel de sens du message. Du point de vue de la modélisation des émotions, les interactions sont conçues également dans une perspective d'interaction que nous qualifions d'*incarnée*. En effet, la modélisation des émotions impose de considérer les aspects non verbaux de la communication et nécessite de prendre en compte le traitement d'indices d'ordre expressif. La communication repose alors non seulement sur des manifestations linguistiques mais également sur des manifestations *corporelles* : traits du visage, gestes, postures corporelles. L'ensemble de ces indices, étudiés notamment par [Dar72], constituent une famille d'artefacts propre à l'induction d'états psychologiques susceptibles de guider l'interprétation de messages et la régulation du dialogue au sein d'un agent conversationnel [BP04].

6.4 En quoi est-ce un problème dur ?

La difficulté inhérente à la modélisation des émotions est multiple : d'une part les émotions reposent sur des considérations qui semblent essentiellement subjectives ; d'autre part, les émotions présentent deux composantes distinctes, l'une expressive, la seconde cognitive dont la relation reste difficile à mettre en œuvre au sein d'une même architecture. La composante expressive nécessite ainsi le développement de notions propres à la communication verbale et non verbale et, de ce fait, de la prise en compte d'une communication incarnée dans laquelle les agents utilisent leur propre *enveloppe corporelle*² pour produire du sens et communiquer. La composante cognitive nécessite à son tour le développement de mécanismes de sélection, de réduction et de décision adaptés aux contraintes courantes du dialogue et reposant sur l'induction d'état émotionnel propres à la situation.

6.5 Perspectives futures

Notre étude présente plusieurs perspectives tant du point de vue de l'interaction homme-machine que des interactions entre agents dans un système multi-agent. Pour le moment, nous étudions essentiellement le premier domaine afin d'envisager dans une perspective à moyen terme l'établissement de nouvelles notions d'interactions entre agents.

Bibliographie

- [BP99] Christian Brassac and Sylvie Pesty. Analyse et simulation de conversations. In *De la théorie des actes de discours aux systèmes multiagents*, chapter Simuler la conversation : un défi pour les systèmes multi-agents, pages 317–345. Limonest, l'interdisciplinaire édition, 1999.
- [BP04] Luca Bisognin and Sylvie Pesty. Agent, langage et emotions : un prototype d'agent émotionnel. In ATALA, editor, *Agental : Agents et Langue*, 2004.
- [Chi02] G. Chicoine. *Dialogue entre agents naturels et agents artificiels. Une application aux communautés virtuelles*. PhD thesis, IMAG - Laboratoire Leibniz, 2002.
- [Dar72] C. Darwin. *The expression of the emotions in man and animals*. John Murray, London, 1872.
- [JPV03] Patricia Jaques, Sylvie Pesty, and Rosa Vicari. An animated pedagogical agent that interacts affectively with the student. In *11th International Conference on Artificial Intelligence in Education*, Sydney, Australia, 2003.

²Pas nécessairement anthropomorphe.

7. Intentionnalité et agents situés

Pierre-Alexandre FAVIER
 ENIB Brest, CS 73 862 - 29 238 Brest Cedex 3
 favier@enib.fr

Domaine de recherche : SMA, modèle comportemental d'agents situés, prototypage interactif

Résumé

Le modèle d'agent intentionnel introduit la notion d'intentionnalité comme lien sémantique entre les différentes parties du modèle comportemental d'un agent situé. Cette approche a pour but de constituer un cadre de spécification de comportement et, grâce à cet apport sémantique, d'offrir un modèle d'exécution favorisant le prototypage interactif. Le modèle est un modèle mixte (impératif / déclaratif) de sélection d'action offrant des facilités de spécification comportementale, de mise en place de mécanismes d'apprentissage et d'héritage de comportement. Le but à terme est d'établir une méthodologie de spécification de comportement basée sur l'utilisation de ce modèle et de l'outil supportant son exécution développé en parallèle.

7.1 Objectifs

Les travaux présentés sont développés dans le cadre de la thèse de l'auteur au Laboratoire d'Ingénierie Informatique, au sein du projet ARéVi (Atelier de Réalité Virtuelle). Ce projet a pour objectif le développement d'une plateforme dédiée à l'exécution de simulations impliquant des agents autonomes situés.

L'élaboration du modèle d'agent intentionnel a pour but de favoriser la spécification du comportement de tels agents dans un objectif de prototypage interactif : les phases de conception et de modification du modèle comportemental du cycle de développement sont alors confondues, l'utilisateur pouvant altérer le modèle alors que la simulation est en cours.

Au delà de la seule spécification de comportement, ces travaux constituent un examen prospectif de l'apport d'un lien sémantique fort, *via* la notion d'intentionnalité, entre la partie décisionnelle de l'agent et la définition de ses actions et perceptions. Notre objectif est d'exploiter les bénéfices possibles de l'explicitation au sein du modèle du "sens" des actions (et perceptions) dans l'abord des problématiques de démarche de conception interactive, de génération automatique d'explication, d'héritage de comportement...

Le dessein à terme est de proposer les fondations d'un langage orienté agent tel que [EFSA03] prenant en compte nativement les contraintes de la simulation de comportements d'agents situés.

7.2 Le modèle formel

Le but principal étant le prototypage interactif de l'agent par l'utilisateur dans un environnement dynamique, ces travaux n'ont pas pour but la formalisation du comportement final de l'agent. Les actions de l'utilisateur étant par nature imprévisibles, nous ne saurions valider le comportement issu du prototypage dynamique par une preuve *a priori*.

Par contre, le modèle d'agent intentionnel [Fav03] formalise une *classe* de comportements : notre but est d'offrir un *framework*, c'est à dire un cadre de développement. Tous les comportements ne sont pas possibles à spécifier comme instance du modèle intentionnel, mais ceux qui le sont peuvent assurément être exécutés sur la plateforme développée : SMAIN (Système Multi-Agents INTentionnels).

Le modèle intentionnel est un modèle de sélection d'action. Il repose sur la déclaration sous forme qualitative est symbolique des effets des méthodes sur des propriétés clés pour établir le lien entre des règles comportementales et l'exécution des actions. La description des effets des actions constitue une indirection entre la définition des méthodes et le processus de sélection. La motivation de l'introduction de cette indirection est d'augmenter la sémantique du modèle en manipulant au maximum les concepts métiers du domaine abordé.

Ce modèle repose essentiellement sur les concepts suivants :

Les méthodes permettent de définir les actions et perceptions de l'agent vis-à-vis de son environnement. SMAIN reposant sur la bibliothèque ARéVi écrite en C++, il offre des fonctionnalités de compilation à la volée afin de satisfaire aux objectifs de prototypage interactif.

Les propriétés sont des abstractions des attributs. Elles sont consultables mais non modifiables par la partie décisionnelle du modèle. Elles facilitent la séparation entre la spécification de comportement et l'aspect implémentation.

Les connaissances sont constituées de code prolog. A la discrétion du programmeur, elles permettent à celui-ci d'affiner le processus de sélection d'action proposé par défaut ou de stocker un historique pour de l'apprentissage par exemple.

Les règles comportementales ont des conditions d'activation arbitrairement complexes, ces dernières sont écrites en prolog. Lorsqu'elles sont actives, elles génèrent des *intentions* : des *tendances* (symboliques) d'évolution souhaitées pour tout ou partie des *propriétés* de l'agent. Elles permettent au spécificateur d'interagir avec le modèle au travers de ses concepts métiers.

Les descriptions d'action établissent le lien entre les *méthodes* et leurs effets sur les *propriétés*, guidant la sélection d'action. Elles constituent donc aussi le lien entre les concepts métiers et la partie implémentation.

7.3 Les interactions

Le modèle intentionnel spécifie l'organisation des différentes parties constitutives du comportement de l'agent, les liens qu'elles entretiennent et le modèle d'exécution permettant de simuler ce comportement. Nous pouvons donc parler d'interaction entre les méthodes et les attributs, donc au niveau interne de l'agent. Il s'agit ici de modification de l'état interne de l'agent par ses méthodes (actions ou méthode de perceptions), donc d'une modification des valeurs de ses attributs. D'autre part, l'agent est un agent situé dans un environnement dynamique. Il est donc susceptible de subir des interactions avec cet environnement et avec les autres agents (collisions, communication...). Dans le cas du modèle intentionnel, l'agent prend en compte ces interactions par le biais de ses méthodes de perception (perceptions proprioceptives et extéroceptives). Les interactions se ramènent donc toujours à une modification de l'état interne de l'agent, qu'elles soient ou non à son initiative. Le dernier niveau d'interaction est celui qui lie l'utilisateur au modèle au travers du prototypage interactif : en manipulant les concepts du modèle, l'utilisateur interagit avec l'agent en modifiant dynamiquement son comportement.

7.4 En quoi est-ce un problème dur ?

La mixité du paradigme d'implémentation (impératif / déclaratif) complique considérablement le développement de la plateforme (ordonnancement d'activités de natures différentes, constantes de temps d'ordre très différents...). D'autre part, la principale difficulté de ces travaux tient à la définition des symboles manipulés (notion de *tendance*, de *propriété*...), la spécification de leur lien et leur exploitation par le modèle d'exécution. Un ensemble important de symbole augmente la souplesse du modèle et accroît le cardinal de la classe de comportements qu'il est possible de spécifier avec le modèle intentionnel. Sa restriction au contraire limite la complexité des comportements abordés mais améliore la cohérence sémantique de l'ensemble. Ceci renforce le cadre de développement et guide plus efficacement le spécificateur dans sa tâche.

7.5 Perspectives futures

Le but à terme de ces travaux est de stabiliser l'ensemble des concepts à mettre en œuvre par le développement de nombreux exemples grâce à SMAIN. Cette expérience permettra d'affiner le modèle intentionnel par un cycle de conception en spirale (le modèle induisant les fonctionnalités de l'outil nécessaires). Une fois le modèle stabilisé autour d'un ensemble réduit de concepts clés, nous tenterons de formaliser une méthodologie de spécification de comportement telle que [MPS03] reposant sur ce *framework* (modèle intentionnel + outil associé).

Bibliographie

- [EFSA03] Suma A. El Fallah-Seghrouchni A. Claim : Un langage de programmation pour des agents autonomes, intelligents et mobiles. In Jean-Pierre Briot et Khaled Ghédira, editor, *Déploiement des systèmes multi-agents, JFSMA '03*, pages 91–105. Lavoisier, 2003.
- [Fav03] Pierre-Alexandre Favier. Cross-paradigm programming for behavioural specification in multi-agent systems, intentionality : From decision to action. In *First Indian International Conference on Artificial Intelligence Proceedings, IICAI-2003*, number ISBN 0-9727412-0-8, pages 302–314, Hyderabad, India, 18–20December 2003. Bhanu Prasad.
- [MPS03] P. Moraitis, E. Petraki, and N. Spanoudakis. Engineering jade agents with the gaia methodology. In et al R. Kowalszyk, editor, *Agent Technologies, Infrastructures, Tools, and Applications for e-services*, number 2592 in LNAI, pages 77–91. Springer-Verlag, 2003.

- le point de vue « Run time environment ». Il est dédié à l’implantation du système. Il permet de prendre en compte les plates-formes d’exécution cibles et de transformer le PIM pour obtenir un ou plusieurs sous-graphes d’interaction représentant des PSM (Platform Specific Model). C’est à ce niveau qu’est gérée l’interopérabilité technique.

8. Interaction entre composants logiciels

Philippe Aniorté

LIUPPA, IUT - Dept Informatique, Place Paul Bert, F-64100 Bayonne

philippe.aniorte@iutbayonne.univ-pau.fr

Domaine de recherche : EIAH, Systèmes d’information

Résumé

La complexité croissante des Systèmes d’Information (SI) et leur incessante évolution rend leur développement plus laborieux, plus coûteux et moins fiable. Dès lors, il émerge des entreprises des besoins très forts en matière de réutilisation. D’autre part, les mutations économiques et technologiques de ces dernières années ont bouleversé profondément leur organisation, conduisant à la multiplication et à la dissémination d’informations. Aussi, de plus en plus, les SI s’avèrent distribués et hétérogènes.

Les domaines de recherche liés à cette problématique sont nombreux. Pour notre part, nous nous intéressons à l’ingénierie de systèmes distribués hétérogènes basée sur la réutilisation. Nous situons notre approche dans la lignée des recherches récentes pour développer de nouveaux systèmes [SG01]. Nous avons centré nos travaux sur le paradigme « composant ». L’objectif consiste à réutiliser des composants logiciels et à les intégrer au sein du système à développer, de sorte à les faire interopérer (concept d’interopérabilité). Pour ce faire, nous nous appuyons sur le concept d’interactions (entre composants). Les composants que nous nous proposons de traiter peuvent être qualifiés de haut niveau dans le sens où ils sont autonomes en terme d’exécution. De plus, ils n’ont pas été conçus, a priori, pour être réutilisés.

8.1 Objectifs

Les objectifs des travaux visent à proposer des solutions permettant d’accompagner la démarche d’ingénierie du système, depuis sa conception, jusqu’à son implantation. Ces solutions sont articulées autour :

- d’un (méta)modèle de composants s’inscrivant dans l’architecture de métamodélisation à 4 niveaux proposée par l’OMG (Object Management Group). Ce modèle répond à un double objectif de réutilisation et d’intégration de composants. Nous détaillons ce point dans la section suivante.
- d’un processus de développement inspiré de la démarche MDA (Model Driven Architecture) préconisée par l’OMG [R00]. Ce processus est basé sur l’instanciation et la transformation de modèles et permet une synergie entre l’interopérabilité conceptuelle et l’interopérabilité technique (à l’implantation).
- d’un ensemble d’outils permettant d’utiliser le métamodèle et d’instrumenter le processus.

8.2 Le modèle formel

Le (méta)modèle, dénommé « Ugatze »[Ani][Ani04][SF04], est un modèle MOF (Meta Object Facilities) de niveau M2 dans l’architecture de métamodélisation. Une représentation graphique et des règles de bonne formation (« well-formedness rules ») exprimées en OCL (Object Constraint Language) lui sont associées. Il repose sur différents principes comme le haut niveau de découplage entre composants, l’abstraction et la variabilité, et la distinction données / contrôle. Son organisation est basée sur la séparation des préoccupations. Il comprend trois points de vue correspondant à trois paquetages :

- le point de vue « Component ». Il est dédié à la (re)spécification des composants en vue de leur réutilisation. Chaque composant est « emballé » (« wrapping ») dans une interface définie par un ensemble de points d’interaction. Différents critères (entrée / sortie, données / contrôle, synchrone / asynchrone / flot continu) permettent de catégoriser ces points.
- le point de vue « Interaction ». Il est dédié à l’intégration des composants par le biais d’interactions basées sur les points d’interaction des composants. Il permet de concevoir l’architecture logicielle du système représentée par un graphe d’interaction, matérialisation du PIM (Platform Independent Model dans la terminologie MDA), instanciation du métamodèle Ugatze. L’interopérabilité conceptuelle est gérée à ce niveau. Les interactions sont détaillées dans la section suivante

8.3 Les interactions

Une interaction est un dispositif permettant à plusieurs composants d’interagir (interopérer). Elle peut impliquer un nombre de composants supérieur à 2. Plus précisément c’est une relation entre 1 ou plusieurs points d’interaction d’entrée et 1 ou plusieurs points d’interaction de sortie. Nous distinguons :

- les interactions prédéfinies, proposées dans le paquetage « Interaction ». Elles ont un caractère générique et sont à la disposition de l’architecte du système. Elles se répartissent en différentes catégories, fonction du type des points d’interaction sur lesquels elles s’appuient.
- les interactions « à façon ». Elles sont conçues par l’architecte pour des besoins spécifiques. Elles constituent une réponse au problème classique du nombre restreint d’éléments prédéfinis ne permettant pas de résoudre l’ensemble des cas pouvant se présenter. C’est particulièrement vrai dans notre contexte de recherche où le niveau de découplage entre composants est de facto très élevé.

8.4 En quoi est-ce un problème dur ?

La réutilisation dans les SI est généralement déclinée en 2 activités, d’une part l’ingénierie de composants réutilisables (« design for reuse »), et d’autre part l’ingénierie par réutilisation de composants (« design by reuse »), la deuxième activité s’appuyant sur la première. Ce type de réutilisation est qualifié « d’a priori ».

En ce qui nous concerne, nous nous intéressons à la réutilisation « a posteriori », i.e. à la réutilisation de composants n’ayant pas été conçus pour être réutilisés, hérités de systèmes existants (« legacy systems »), d’où un niveau de découplage entre composants très élevé. Il en résulte des problèmes d’interopérabilité important, amplifié par la distribution des composants (hétérogénéité).

La volonté d’intervenir sur plusieurs éléments du cycle de vie (de la conception à l’implantation) constitue une difficulté supplémentaire. Le concept d’interaction, tel que nous l’appréhendons, permet de gérer l’interopérabilité au niveau conceptuel. Le modèle, le processus et les outils associés permettent une synergie entre l’interopérabilité conceptuelle et l’interopérabilité technique, rendant possible in fine l’implantation du système.

8.5 Perspectives futures

Les perspectives envisagées sont de 2 ordres. Premièrement, il s’agit dans la lignée des travaux actuels validés par un projet européen (ASIMIL) d’enrichir le modèle en termes de points d’interaction et d’interaction, et de développer des outils permettant d’instrumenter la démarche (référentiel MOF, validation du PIM et automatisation de certaines étapes).

Deuxièmement, nous comptons nous pencher sur l’adaptation de nos propositions pour gérer des « Grid Services » dans le cadre d’un projet européen (ELeGI) en phase de démarrage.

Bibliographie

- [Ani] Philippe Aniorté. A distributed adaptable software architecture derived from a component model. In *Computer Standards & Interfaces (CSI)*.
- [Ani04] Philippe Aniorté. La dimension processus dans l’ingénierie par réutilisation de composant : De l’infrastructure de réutilisation à l’architecture logicielle distribuée. *Numéro spécial Revue L’Objet*, 2004. Hermes, M. Oussalah et D. Rieu (eds).
- [R00] Soley R. Model driven architecture - white paper. *OMG Staff Strategy Group*, 2000.
- [SF04] Aniorté P. Seyler F. Model-based middleware integration. In *6th International Conference on Enterprise Information Systems (ICEIS 2004)*, avril 2004.
- [SG01] HCSS (High Confidence Software and Systems) Coordinating Group. Hcss research needs : a white paper. *Interagency Working Group in Information Technology Research and Development (IWG/IT R&D), White House National Science and Technology Council, USA*, 2001.

9. Modélisation de Scénarios pédagogiques appliqués au eLearning

Ecaterina Giacomini

HeuDiasyC/UTC

UMR CNRS 6599, BP 20529, 60205 Compiègne

egiacomi@hds.utc.fr

Domaine de recherche : IHM, EIAH, e-learning, instructional theories design

Résumé

L'abondance et la diversité du matériel pédagogique numérisé nous déterminent à reconnaître aujourd'hui l'intérêt d'adopter des normes et des standards. Ces normes et standards se présentent comme un " langage commun " servant à désigner, catégoriser et décrire les ressources éducatives numérisées. Ce langage commun qui a l'avantage d'être " interprétable " autant par les machines que par les humains, constitue le coeur de l'interopérabilité des systèmes et des logiciels qui traitent le matériel pédagogique numérisé. Plusieurs normes pour la représentation des contenus pédagogiques, comme SCORM et IMS LD, sont actuellement à notre disposition et peuvent être appliquées en fonction de nos objectifs [CRE03]. Ainsi, pour la représentation de scénarios pédagogiques nous avons choisi le langage normalisé IMS Learning Design [IMS03] qui permet de prendre en compte la description des contenus pédagogiques ainsi que l'interaction entre les différents acteurs dans une activité pédagogique donnée

9.1 Objectifs

Les travaux présentés se situent dans le cadre du projet de recherche CEPIAH (Conception et Evaluation des Produits Interactifs pour l'Apprentissage Humain) au sein du laboratoire HeuDiasyC de l'UTC de Compiègne. Nous développons une méthodologie pour la conception et l'évaluation des environnements hypermédia pédagogiques : Quelles sont les étapes de création et de développement d'un produit hypermédia pédagogique ? Comment pouvons nous concevoir un site Web éducatif de point de vue ergonomique ? Quelles sont les éléments pédagogiques pertinents pour différentes méthodes d'apprentissage ? Pour répondre à ces questions nous avons besoin d'outils permettant de concevoir, caractériser et évaluer le produit hypermédia d'apprentissage. Ainsi, nous développons un guide interactif accessible sur le Web, composé de trois modules : Aide à la Conception, Aide à l'Evaluation et Modèles Pédagogiques de sites web. Globalement, l'objectif de cet environnement interactif est d'aider les auteurs de sites web éducatif dans la conception et l'évaluation de leurs prototypes. Les deux premiers modules étant développés, nous nous intéressons à présent au troisième module, Modèles Pédagogique [TG04]. Ainsi, l'objectif de ce module est de proposer une série de modèles prédéfinis de sites web éducatifs, tout en respectant les normes et les standards pour la description des contenus pédagogiques. Dans ce cadre de conception nous prenons en compte deux aspects majeurs, à savoir : l'IHM des sites web (couleurs, formes de menus et de boutons, etc.) et la modélisation par IMS Learning Design de scénarios pédagogiques qui sont basés sur différentes approches théoriques et méthodes d'enseignement. Afin d'éviter le problème combinatoire pour la conception artisanale d'un grand nombre de squelettes de sites web pédagogiques, nous concevons actuellement, un outil de génération automatique de ces sites qui prend en entrée un questionnaire interactif pédagogique et un autre questionnaire interactif sur les préférences des éléments IHM pour le web.

Tous ces aspects rendent notre approche différente par rapport à d'autres environnements de conception de modules de formation à distance, comme par exemple OASIF [Gal01] et Scenari [Cro02]. OASIF c'est un outil collaboratif d'aide à la scénarisation de modules de formation ouverte et à distance alors que Scenari c'est un environnement qui permet la conception industrialisée des cours de formation.

9.2 Le modèle formel

Le langage de modélisation de scénarios pédagogiques que nous utilisons est composé de trois niveaux de représentation : le niveau A est intégré dans le niveau B et ce dernier est intégré dans le niveau C.

Le niveau A est celui de base et permet la spécification des activités pédagogiques ordonnées dans le temps et effectuées par les apprenants et par les professeurs dans le contexte d'un milieu qui consiste en objets d'apprentissage et des services. L'élément de base est la méthode qui est une séquence d'éléments permettant de définir la dynamique du processus d'apprentissage. Une méthode part des conditions initiales (pré requis) qui doivent être satisfaites et qui a comme but d'atteindre certains objets d'apprentissage. Elle consiste en un ou plusieurs scénarios qui sont exécutés en parallèle. En générale il y a deux rôles qui peuvent être interprétés : étudiant et professeur.

Le niveau B intègre le niveau A en ajoutant des propriétés et des conditions permettant une représentation du processus d'apprentissage plus avancée et plus flexible. Les propriétés sont utilisées pour stoker les informations sur une personne ou sur un groupe des personnes.

Le niveau C intègre le niveau B en ajoutant les notifications. C'est un niveau de représentation " évènementiel ".

9.3 Les interactions

Alors qu'au niveau A nous définissons les éléments pédagogiques qui composent un cours de formation, au niveau B, les conditions permettent de décider sur l'évolution d'un scénario pédagogique à un moment donné. Par l'évaluation d'une expression on peut décider, en fonction de son résultat, quel parcours suivra le scénario. Par exemple, si pour un test donné le résultat de l'apprenant est au-dessus d'un certain niveau, l'apprenant peut dépasser certaines activités. Au niveau C est permise la transmission des messages d'un rôle (enseignant, apprenant) ou l'ajout des nouvelles activités associées à un rôle, comme effet de l'apparition des événements pendant le processus d'apprentissage. Par conséquent, nous remarquons que l'interactivité entre enseignant/apprenant ou encore avec l'environnement, se situe seulement aux niveaux B et C du IMS LD

9.4 En quoi est-ce un problème dur ?

La conception de l'outil de génération automatique de sites web éducatifs est une tâche complexe car ceci nécessite la création d'une base de connaissance intégrant les connaissances pédagogiques à partir des théories et modèles d'enseignement et d'apprentissage. Un autre problème complexe est la prise en compte de l'interactivité, au niveau B, dans la modélisation de scénarios pédagogiques. A ce niveau nous devons bien gérer les conditions et les propriétés afin de proposer des modèles de scénarios interactifs et pertinents avec les atteintes des utilisateurs.

9.5 Perspectives futures

A court terme, nous envisageons à réaliser une expérimentation de notre outil de génération automatique de modèles pédagogiques auprès des enseignants de notre université (UTC). En ce qui concerne l'implémentation de notre outil, dans une perspective à moyen terme, nous envisageons l'intégration du niveau de représentation " évènementiel ".

Bibliographie

- [CRE03] CREPUQ. La description normalisée des ressources : vers un patrimoine éducatif. Technical report, Novasys inc., Montreal, Canada, 2003.
- [Cro02] Stéphane Crozat. *Ingénierie multimédia dans des contextes éducatifs*. PhD thesis, Laboratoire Heudiasyc - UTC, 2002.
- [Gal01] Arnold Galisson. un outil collaboratif d'aide à la scénarisation de modules de formation ouverte et à distance. In *TICE'2001*, Lyon, France, 2001.
- [IMS03] *IMS Learning Design Best Practice and Implementation Guide*. 2003.
- [TG04] Philippe Trigano and Ecaterina Giacomini. Toward a web based environment for evaluation and design of pedagogical hypermedia. *Educational Technology and Society Journal*, 2004.

10. La méthode TOOD pour la spécification formelle des interfaces homme-machine

Mourad Abed

Laboratoire LAMIH CNRS 8530

UVHC, 59313 Valenciennes Cedex 9

mourad.abed@univ-valenciennes.fr

Domaine de recherche : Génération d'IHM, Spécif formelle, Objets et Réseaux de Petri.

Résumé

Nos travaux visent la génération automatique des interfaces homme-machine. Pour ce faire, nous avons développé une méthode, nommée TOOD (Task Object Oriented Design) basée sur une approche formelle qui donne des résultats quantitatifs pouvant être utilisés par les concepteurs et offre la possibilité d'effectuer des vérifications mathématiques sur les modèles. Son formalisme de modélisation est fondé sur l'utilisation conjointe de l'approche objet et des réseaux de Petri de haut niveau. Les concepts empruntés à l'approche objet permettent de décrire les aspects statiques de tâches et les réseaux de Petri la dynamique et le comportement. Un outil support de la méthode permet l'édition, la simulation du modèle de tâches et la génération partielle du code de l'interface.

10.1 Objectifs

Une des solutions apportées par la recherche est la génération de prototypes d'IHM fondée sur le paradigme de la conception d'interface utilisateur basée sur le modèle (Model-Based user interface Design, noté MBD). Ce paradigme se réfère, dans l'absolu, à une description explicite, largement déclarative, capturant la sémantique de l'application et toute la connaissance nécessaire à la spécification tant de l'apparence que du comportement du système interactif. Il s'inscrit dans la lignée des UIMS (User Interface Management Systems). Il propose une réelle alternative à la construction des interfaces. Il prône une appréhension descendante et requiert des concepteurs, non plus une programmation informatique, mais la rédaction de spécifications formelles. Ces connaissances, décrites dans un langage de haut niveau spécialisé, sont traduites ou interprétées pour une génération totale ou partielle du code de l'application.

Dans cette démarche nous avons contribué par une méthode appelée TOOD (Task Object Oriented Design) [MDC03][DM02] qui se base sur la transformation d'une série de modèles formels pour couvrir le cycle de développement d'un système interactif, de l'analyse de la tâche à l'implémentation de l'interface homme-machine.

10.2 Le modèle formel

Le modèle de la tâche spécifie "le quoi" et montre "le quand" les traitements principaux sont réalisés. Deux types de modèles sont issus de cette étape de spécification : (i) un *Modèle Statique de la Tâche (MST)*, basé sur l'approche orientée objet, permet d'exprimer la structure de la tâche, (ii) un *Modèle Dynamique de la Tâche (MDT)* utilise les Réseaux de Petri Objet (RPO) pour décrire comment la tâche gère et traite les données et les ressources qu'elle manipule en fonction de son état interne. Dans TOOD, les données sont appelées *Objets du Domaine (OD)*. Ces OD sont représentés dans le *Modèle des Objets du Domaine (MOD)* et formalisés de manière similaire aux tâches (volets : statique et dynamique). L'étape de conception affine chaque tâche terminale jusqu'à aboutir à des traitements élémentaires (appliqués aux OD) clairement détaillés; il s'agit du *modèle opérationnel*. Ce modèle décrit ainsi les interactions entre les *objets interactifs* (ressources système) et l'utilisateur (ressource humaine) respectivement décrits dans le *Modèle Local de l'Interface (MLI)* et le *Modèle de l'Utilisateur (MU)*. Le MLI décrit le comportement de ces objets, c'est-à-dire comment ils réagissent aux stimuli extérieurs (événements utilisateur et/ou application) et gèrent le contrôle des OD en fonction de leur état interne.

Le MU, de façon similaire au modèle de la tâche, possède deux volets : (i) le volet statique qui définit le rôle de l'utilisateur, son niveau d'expertise, etc. ; (ii) le volet dynamique qui modélise le comportement de l'utilisateur en procédures d'actions. Ces procédures contiennent tous les éléments nécessaires pour décrire précisément comment atteindre un but. De même que le modèle de la tâche, la formalisation du MLI et du MU se basent sur l'utilisation des Réseaux de Petri Objet (RPO).

La dernière étape de la méthode consiste à implémenter les éléments du modèle de l'interface issus de l'étape de conception vers une architecture multi-agents de type PAC (Présentation, Contrôle, Abstraction). Dans TOOD, la démarche est globalement descendante. Cependant, à l'approche de l'implémentation, elle devient ascendante par le fait que la définition de la structure globale des IHM est construite à partir de l'ensemble des objets-composants des modèles locaux de l'IHM, comme le montre la figure 1. En effet, cette structure globale n'est être une simple juxtaposition des objets-composants IHM unitaires mais un processus d'agrégation de classes d'objets interface ayant des caractéristiques semblables (attributs, opérations, ObCS, etc.).

TOOD est supportée par un Environnement de Développement d'Interface nommé TOOD-IDE (Task Oriented Object Design - Interface Development Environment). TOOD-IDE vise à poursuivre, dans la lignée des environnements basés sur les modèles, le développement structuré et intégré des systèmes interactifs.

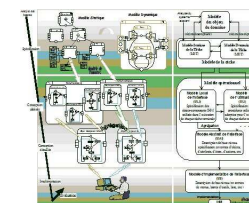


FIG. 10.1 – Cycle de développement de la méthode TOOD

10.3 Les interactions

Notre modélisation fait apparaître les interactions qu'un utilisateur aura avec l'interface ainsi que les interactions inter-objets de l'interface ; c'est à dire le composant de dialogue qui est de loin le plus difficile à décrire. En effet, pour le spécifier il faut à la fois décrire les différents états acceptables de l'application interactive, les actions que peut réaliser l'application, les événements auxquels elle réagit et la réaction à un événement. Ensuite, il faut générer l'application interactive à partir de ces spécifications plus ou moins formelles.

10.4 En quoi est-ce un problème dur ?

La percée des nouvelles technologies de l'information et de la communication met l'accent sur la prépondérance des systèmes interactifs à tous les niveaux : bornes interactives pour le grand public, organisation multi-sites des sociétés, migration des salles de contrôles des systèmes complexes (Contrôle aérien, contrôle des centrales nucléaires...), développement de sites Internet interactifs et personnalisables. Toute cette effervescence et cette démocratisation des systèmes interactifs font naître un besoin encore plus grand de génération automatique des interfaces Homme-Machine en intégrant les besoins et les habitudes des utilisateurs. Ces nouvelles interactions posent de nombreux problèmes théoriques et méthodologiques en termes de conception et d'évaluation. C'est dans ce contexte que se situe l'étude des modèles d'interaction et architectures adaptées à ces nouvelles interactions.

10.5 Perspectives futures

Les principales perspectives de recherche que nous envisageons dans le domaine de l'implémentation de systèmes interactifs se situent dans le prolongement des travaux qui portent sur la mise en œuvre d'un système interactif à partir de spécifications formelles et sur la réalisation d'un environnement de support à l'édition des modèles puis leur interprétation dans le but de permettre un prototypage du système final (tout type de système : plasticité) à partir de spécifications.

Bibliographie

[DM02] Tabary D. and Abed M. A software environment task object oriented design (etood). *Journal of Systems and Software*, 60 :129–140, 2002.

- [MDC03] Abed M., Tabary D., and Kolski C. *The Handbook of Task Analysis for Human-Computer Interaction*, chapter Using Formal Specification Techniques for the Modelling of Tasks and Generation of HCI Specifications., pages 503-529. Lawrence Erlbaum Associates, Mahwah, New Jersey, 2003.

11. Modélisation de l'interaction HM située

Laurent Le Bodic, Pierre De Loor
 CERV 25 rue Claude Chappe BP38 29480 Plouzané
 lebodid@enib.fr <http://www.enib.fr/~lebodid>

Domaine de recherche : IHM, multimodalité, SMA, ingénierie cognitive, simulation.

Résumé

Nos travaux consistent en l'élaboration d'une simulation du comportement d'un utilisateur d'artefacts multimodaux, permettant d'évaluer l'utilisabilité de ces derniers. Cette approche, qualifiée de naturaliste par les ergonomes, intègre l'environnement de l'utilisateur ainsi que les interactions associées. La simulation s'appuie sur un scénario de simulation, une description fonctionnelle des artefacts multi-modaux, un système déductif simulant les prises de décision de l'utilisateur en intégrant des aspects cognitifs et perceptifs ainsi que l'environnement physique qui a une grande influence sur l'usage des interfaces multimodales. En collaboration avec des ergonomes de l'IRIT et de France Télécom, ces travaux ont abouti à l'élaboration d'un démonstrateur logiciel nommé SIHMM (Simulation d'Interaction Homme Machine Multimodale).

11.1 Objectifs

Au sein de la communauté des interfaces homme-machine, il est reconnu que l'évaluation d'interfaces multi-modales reste un domaine peu exploré. La cause principale est certainement liée à la nature même de la multimodalité dont l'évaluation ne peut être séparée de la prise en compte de l'environnement, capable de perturber les sens de l'utilisateur (son, lumière, etc...). Actuellement, les évaluations de la multimodalité se font uniquement de façon empirique [CKZ⁺01], [Ovi99]. Ces évaluations ont mis en lumière des tendances, mais il faut essayer de comprendre les actions de l'utilisateur afin de pouvoir juger de la qualité de l'interface. Il faut alors établir d'autres méthodes, qui prennent en compte les perturbations liées à l'environnement et toutes les possibilités offertes par la multimodalité. Il faut également un modèle de l'utilisateur qui permettrait de simuler son comportement.

La simulation de l'interaction HM située a ainsi pour objectif d'évaluer l'adéquation du triplet <environnement, artefact, utilisateur> en terme d'usage des modalités. L'ergonome, ou le designer, jouera sur les différents paramètres des modélisations afin de tester de nouvelles mises en situation. Cette approche par simulation ne cherche pas à remettre en cause les méthodes d'évaluations classiques mais cherche plutôt à proposer une aide au prototypage des IHMs mobiles et ainsi permettre de tester des situations scénaristiquement intéressantes avant la réalisation d'un prototype fonctionnel.

11.2 Le modèle formel

Notre modélisation suppose une formalisation des différentes parties du triplet support de l'interaction [LDCK04]. Il y a donc plusieurs modèles, certains définis formellement d'autres empiriquement mais tous sont en interaction. La formalisation de l'interaction entre ces modèles est un de nos sujets d'étude. Parmi les modèles formels se trouvent

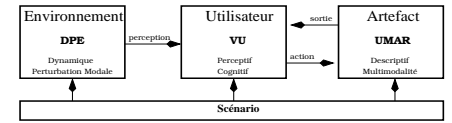
- Un modèle utilisateur (VU) basé sur une intégration d'aspects perceptifs et cognitifs, aboutissant à une prise de décision en terme d'actions à effectuer. Les mécanismes perceptifs sont basés sur les cartes cognitives floues [Kos86] tandis que la modélisation cognitive fait appel aux notions de l'ingénierie cognitive [Cac98]. Cette dernière est écrite en PROLOG.
- Un modèle de description des artefacts multimodaux simulable (UMAR), portant notamment sur les modalités d'action, de contrôle et de sortie d'une interaction. UMAR permet de décrire un artefact multimodal dans un langage graphique interprétable par toutes les personnes intervenant dans le cycle de conception. Ce modèle peut par ailleurs être utilisé découpé du simulateur à des fins de descriptions et d'analyses de l'utilisabilité (pattern, commande à effet tunnel unimodale, etc ..).

L'environnement (DPE) est basé sur un système multi-agents, dont le rôle est de perturber le VU. Basé sur les concepts classiques de la simulation d'agents situés il n'est pas explicitement formalisé même s'il permet une interaction avec les autres modèles.

11.3 Les interactions

Notre modélisation fait apparaître les interactions multimodales d'un artefact ainsi que les interactions inter-modèles. En ce qui concerne les interactions multimodales, UMAR propose une modélisation à base d'un graphe d'états hiérarchique, capables d'exprimer de façon concise les propriétés CARE de la multimodalité [CN95].

On retrouve les interactions inter-modèles entre les trois blocs principaux représentés ci-contre. Ainsi qu'au sein du modèle de VU, composé lui même de deux sous modèles et issu des travaux initiés par [CMN83].



En résumé, les interactions inter-modèle suivent cette boucle : Partant de la perception de son environnement (interaction perceptive), la carte cognitive définit les modalités d'interaction possibles qui sont fournies au moteur décisionnel (interaction cognitive). Lequel, en fonction des buts, des préférences d'usage de l'utilisateur et du scénario de la simulation, choisit une action à effectuer sur l'artefact (interaction motrice). Artefact qui, en retour, associé à l'environnement, influencera les perceptions de l'utilisateur.

11.4 En quoi est-ce un problème dur ?

Une approche multi-modèle : la volonté d'étudier le comportement humain dans un contexte écologique, nécessite la formalisation de plusieurs modèles de simulation -de natures très différentes- et de leurs interactions, ce qui révèle de nombreuses difficultés conceptuelles et applicatives.

Une approche globale de l'interaction HM : Les IHMs mobiles sont par définition destinées à être utilisées dans des milieux ouverts ou la complexité et le nombre des paramètres (affordances de [Gib58], facteurs de [RR97]), est tellement important qu'il est fastidieux de les observer et de les corrélés. La simulation doit permettre de centrer les expérimentations réelles sur des scénarii problématiques. Cependant, une autre difficulté provient du manque de recul des sciences informatiques et humaines sur l'interaction multimodale pour élaborer les modèles sous-jacents.

11.5 Perspectives futures

Nous travaillons en collaboration avec des ergonomes, qui ont pour charge la réalisation d'expérimentations réelles dans un contexte naturel. Ce travail viendra dans un avenir proche alimenter notre modélisation. Il permet de disposer de retours d'expériences sur l'impact de l'environnement sociologique vis à vis du comportement humain. De façon plus générale, la corrélation entre les expériences in situ et in virtuo permettra de démontrer l'usage possible de la réalité virtuelle et de la simulation pour l'évaluation prototypique.

Bibliographie

- [Cac98] P.C. Cacciabue. Modelling and simulation of human behavior in system control. Springer, 1998.
- [CKZ⁺01] G. Calvet, J. Kahn, M. Zouinar, P. Salembier, J.C. Briois, L. Nigay, G. Rey, and L. Pasqueletti. Etude empirique de la multimodalité avec un ordinateur de poche. In Vanderdonck J. Blandford A. and Derycke A. Eds, editors, *IHM-HCI 2001*, pages 5–8. CEPADUES Publ, 2001.
- [CMN83] S.K. Card, T.P. Moran, and A. Newell. *The psychology of human-computer interaction*. Lawrence Erlbaum, London, 1983.
- [CN95] J. Coutaz and L. Nigay. Four easy pieces for assessing the usability of multimodal interaction : The care properties. In Arnesen S.A & Gilmore D. Eds., editor, *INTERACT'95*, pages 115–120. Chapman & Hall Publ., 1995.
- [Gib58] J. Gibson. Visually controlled locomotion and visual orientation in animals. *British Journal of psychology*, 49 :182–194, 1958.
- [Kos86] B. Kosko. Fuzzy cognitive maps. *International Journal of Man Machine Studies*, 24 :65–75, 1986.

- [LDCK04] L. Le Bodic, P. De Loor, G. Calvet, and J. Kahn. Sihmm : Simulateur de l interaction homme machine multimodale. In L. Nigay, editor, *XVI conférence sur l'interaction Homme Machine, Annexes des actes*, pages 31–34, Namur, September 2004.
- [Ovi99] S. Oviatt. Ten myths of multimodal interaction. *Commun. ACM*, 42(11) :84–81, 1999.
- [RR97] F. Redmill and J. Rajan. Human factors in safety-critical systems. Butterworth Heinemann, 1997.

12. Un modèle d'interaction pour la négociation

Marie-Hélène Verrons

SMAC - LIFL - USTL, Cité Scientifique, 59655 Villeneuve d'Ascq cedex
verrons@lifl.fr http://www.lifl.fr/SMAC/projects/genca

Domaine de recherche : SMA, négociation entre agents

Résumé

Lorsque de nombreux agents interagissent, des conflits peuvent survenir. Pour les résoudre, différentes méthodes peuvent être utilisées, comme la coordination, les systèmes de vote et la négociation. Celle qui nous intéresse dans cette présentation est la négociation à base de contrats portant sur des ressources. L'objectif de notre travail est de proposer un modèle général de négociation qui reprend tous les points communs aux différents types de négociation et qui utilise un protocole général de négociation, paramétrable afin d'instancier un protocole spécifique à un type de négociation donné. Le modèle formel que nous proposons, appelé *GeNCA*, est basé sur une architecture à trois niveaux : communication, négociation et stratégie pour lui permettre une réelle généralité. Les interactions considérées dans ce travail sont celles induites par un mécanisme de négociation et sont décrites au sein du protocole général de négociation que nous avons défini. La diversité des négociations rend la réalisation d'un système général difficile, d'autant plus qu'il y a presque autant de définitions de la négociation qu'il y a de chercheurs à ce sujet.

12.1 Objectifs

L'objectif de notre travail est de concevoir un modèle général de négociation (appelé *GeNCA* : *Generic Negotiation of Contracts API*) et une implémentation de ce modèle. Ceci permet à un utilisateur souhaitant développer une application de négociation de ne pas devoir tout réaliser mais de pouvoir utiliser un modèle qui lui facilitera le travail. Notre proposition de modèle général de négociation a plusieurs objectifs, dont la généralité, la portabilité, l'uniformisation et l'automatisation des envois de messages. Nous avons en effet l'intention de fournir un modèle de négociation permettant de réaliser plusieurs formes de négociation différentes, sans demander de lourde charge de travail à un utilisateur. Différentes plateformes ont été réalisées, comme Magnet [CYJ] de l'université du Minnesota et la plateforme des laboratoires HP [BPJ02] qui sont dédiées aux marchés électroniques, ou SilkRoad [Str00] développé par IBM pour l'e-business. A notre connaissance, aucune plateforme générale à différents types d'applications de négociation n'existe à part *GeNCA*.

12.2 Le modèle formel

Nous avons défini *GeNCA* (*Generic Negotiation of Contracts API*) [MV03, MV05], un modèle général de négociation en trois couches : communication, négociation et stratégie. Ces trois couches sont indépendantes les unes des autres, à la manière de briques logicielles. En effet, la façon dont les agents communiquent n'a pas d'influence sur leur façon de négocier. De même, les stratégies de négociation sont intrinsèquement liées à une application donnée et il est bien évident que la stratégie à utiliser lors d'enchères anglaises (où les prix augmentent) et lors d'enchères hollandaises (où les prix diminuent) n'est pas la même. La couche de communication fournit le moyen de communication utilisé par les agents, tels que la communication fournie au sein d'une plateforme multi-agents, la communication par envoi d'e-mails, etc. La couche de négociation forme le cœur de notre modèle, c'est elle qui contient le protocole de négociation utilisé ainsi que la gestion des différentes négociations. La couche de stratégie définit le squelette d'une stratégie de négociation qu'il convient d'instancier pour une application donnée.

12.3 Les interactions

Les interactions que nous considérons sont les messages de négociation échangés entre les agents. Ces différents messages sont définis par un protocole de négociation général et paramétrable. Les paramètres du protocole permettent de définir un délai d'attente des réponses ainsi qu'une réponse par défaut commune à tous les participants, le nombre minimal d'accords nécessaires au succès de la négociation, le nombre de tours de parole des participants au cours de la négociation, le nombre de modifications pouvant être envoyées par tour de parole et de préciser si la rétractation est autorisée auquel cas un nombre de renégociations maximal est défini. Ce protocole permet d'effectuer de la négociation de 1 vers n agents, de faire des contre-propositions, de se rétracter et de renégocier automatiquement un contrat. Grâce aux paramètres de délai d'attente des réponses et du nombre de tours de parole des participants, les deadlocks peuvent être évités. La gestion des négociations entrant en conflit sur des ressources se fait soit séquentiellement, soit simultanément. Les négociations n'entrant pas en conflit sont toutes négociées simultanément.

12.4 En quoi est-ce un problème dur ?

La diversité des négociations rend la réalisation d'un système général difficile, d'autant plus qu'il y a presque autant de définitions de la négociation qu'il y a de chercheurs à ce sujet. L'uniformisation est une notion très importante dans ce domaine de recherche, car à notre connaissance, il y a très peu de travaux sur ce sujet. Seules une classification, dite de Londres et une taxonomie dite de Montréal tentent de répondre au problème de la caractérisation des négociations. Un système général de négociation doit prendre en compte le plus de types de négociation possibles, mais ces types possèdent de nombreuses caractéristiques, ce qui rend difficile leur intégration dans un système général. Par exemple, le nombre de tours de parole des participants n'est pas le même selon que l'on utilise des enchères à offres scellées (1 seul tour) ou des enchères anglaises (plusieurs tours). De même, le nombre de participants d'accords (satisfaits) avec la proposition de contrat nécessaire à la conclusion du contrat peut varier. Pour le cas des enchères, un seul participant d'accord est nécessaire, ce sera le vainqueur de l'enchère, en revanche, lorsqu'on négocie une réunion, il faut que 75% des participants contactés acceptent d'y participer, par exemple. D'autres problèmes comme la gestion des deadlocks, la gestion des négociations entrant en conflit et la renégociation sont autant d'éléments qui compliquent encore plus la réalisation d'un système général de négociation.

12.5 Perspectives futures

Nous envisageons d'extraire notre protocole de négociation du modèle afin de pouvoir en changer facilement et de proposer une bibliothèque de protocoles utilisables dynamiquement au sein de *GeNCA*. Pour cela, il est nécessaire de définir les protocoles dans un formalisme qui soit clair pour l'utilisateur et facilement intégrable du point de vue génie logiciel. Une étude des différents formalismes de représentation des protocoles sera donc nécessaire. Nous aimerions fournir à l'utilisateur un outil lui permettant de définir un protocole de négociation et de l'utiliser au sein de *GeNCA*. D'autres perspectives comme l'intégration d'autres formes de négociation (combinées, par argumentation), la réalisation de nouvelles applications de négociation et l'amélioration du niveau stratégique sont envisagées.

Bibliographie

- [BPJ02] Claudio Bartolini, Chris Preist, and Nick R. Jennings. Architecting for reuse : A software framework for automated negotiation. In *Proc. 3rd Int Workshop on Agent-Oriented Software Engineering*, pages 87–98, Bologna, Italy, 2002.
- [CYJ] John Collins, Ben Youngdahl, and Scott Jamison.
- [MV03] Philippe Mathieu and Marie-Hélène Verrons. ANTS : an API for creating negotiation applications. In *Proceedings of the 10th ISPE International Conference on Concurrent Engineering : Research and Applications (CE2003), track on Agents and Multi-agent systems*, pages 169–176, Madeira Island, Portugal, July, 26-30 2003.
- [MV05] Philippe Mathieu and Marie-Hélène Verrons. GeNCA : Un modèle général de négociation de contrats. *Revue d'Intelligence Artificielle*, 2005. à paraître.
- [Str00] Michael Ströbel. A Framework for Electronic Negotiations Based on Adjusted Winner Mediation. In *Proceedings of the DEXA 2000 Workshop on e-Negotiations*, pages 1020–1028, London, UK, 2000.

13. Introduction d'interactions dans les processus de décision markoviens

Vincent THOMAS (équipe MAIA)

LORIA, Campus Scientifique - BP 239, 54506 Vandoeuvre Cedex

Vincent.thomas@loria.fr <http://webloria.loria.fr/~vthomas/>

Domaine de recherche : SMA, Modèles markoviens, Résolution collective de problèmes

Résumé

Nos travaux cherchent à construire de manière décentralisée les comportements d'agents réactifs afin de résoudre un problème global. Nous proposons un cadre formel inspiré des processus de décision markoviens. Celui-ci permet de représenter des problèmes de prise de décisions dans lesquels les agents peuvent agir indépendamment les uns des autres ou interagir entre eux. Nous cherchons, en séparant les décisions collectives des décisions individuelles, à trouver un compromis entre un apprentissage centralisé coûteux et des apprentissages individuels. En outre, la présence de récompenses différées, partiellement perçues par les agents, nécessite d'introduire des mécanismes de négociation et de transfert de récompense pour pouvoir tirer parti d'apprentissages individuels égoïstes.

13.1 Objectifs

Nos travaux se focalisent sur les systèmes multi-agents réactifs coopératifs. Dans ce cadre, nous cherchons à construire, de manière automatique et décentralisée, les règles stimulus-réponse des agents afin de résoudre de manière distribuée un problème posé à la collectivité.

Les MDP (Markov Decision Process) en introduisant la rationalité au niveau individuel permet à un agent seul d'apprendre son comportement de manière optimale par planification ou apprentissage [RG98]. Des extensions multi-agents de ces modèles comme les DEC-MDPs ont été proposées pour représenter plusieurs agents cherchant à influencer simultanément le même processus [BZI00]. Construire les comportements des agents dans ce cadre reste un problème ouvert puisqu'il n'existe pas de garantie de convergence pour des apprentissages individuels effectués en parallèle [SPG03] et qu'une approche centralisée doit faire face à une explosion combinatoire du nombre d'états en fonction du nombre d'agents.

Nous pensons néanmoins que dans beaucoup de problèmes, il est possible de trouver un compromis entre un apprentissage centralisé coûteux mais optimal et des apprentissages individuels égoïstes simples mais insuffisants. Pour cela, nous proposons un nouveau cadre formel plus restrictif basé sur la notion d'interaction afin de pouvoir raisonner à différents niveaux d'abstraction en séparant les décisions individuelles des décisions collectives. Notre objectif consiste alors à trouver un processus permettant à des agents égoïstes de résoudre un problème collectif en s'aidant de phases d'interactions ponctuelles.

13.2 Le modèle formel

L'approche formelle s'inspire des modèles markoviens et sépare le modèle d'action du modèle d'interaction. Dans ce cadre formel, chaque agent à la possibilité d'agir dans son espace local et d'interagir avec les autres agents du système.

Un modèle d'action est un MDP $\langle S_i, A_i, T_i, R_i \rangle$. Il représente un agent, son espace local, ses actions locales et ses lois d'évolution locales. Chaque agent peut évaluer la tâche qu'il est en train de réaliser par la fonction locale de récompense R_i . Les actions des agents (et donc ces MDPs) sont totalement indépendants les uns des autres. En outre, chaque agent ne perçoit que partiellement les états locaux des autres agents. A ce niveau d'abstraction, des apprentissages individuels sont envisagés.

Le modèle d'interaction caractérise les moyens dont dispose un agent pour influencer les espaces locaux des autres agents. Ces moyens sont des interactions impliquant plusieurs agents. Chaque interaction utilisée a pour conséquence l'évolution des espaces locaux des agents impliqués. Cette évolution dépend d'un choix sur les différents résultats d'interaction possibles. Pour faire ce choix, des négociations réactives entre les agents impliqués sont envisagées.

L'objectif du système consiste à déterminer les actions des agents, l'utilisation individuelle des interactions et leur résolution collective afin de maximiser la somme des récompenses perçues par tous les agents $R = \sum_i R_i$.

13.3 Les interactions

Le modèle d'interaction introduit des couplages entre les entités du système. Les interactions que nous utilisons sont définies comme des 'influences mutuelles réciproques ponctuelles'. Chaque agent peut décider d'utiliser ces interactions. Pour cela, il doit dans un premier temps décider de manière individuelle quelle interaction utiliser et avec quels agents interagir. Dans un second temps, les agents impliqués décident de manière semi-centralisée par négociation quelle action jointe coordonnée autorisée par l'interaction permet d'améliorer au mieux les récompenses globales reçues à long terme.

L'originalité de notre travail réside dans le fait que les couplages entre agents sont localisés au niveau des interactions. Cette conception de l'interaction permet d'explicitier les moments où il peut être utile de prendre une décision de manière semi-centralisée. Elle permet en outre de définir une nouvelle entité : les ensembles d'agents impliqués dans une interaction. Il est ainsi possible d'envisager de nouveaux mécanismes opérant sur cette entité comme les transferts de récompenses entre agents.

13.4 En quoi est-ce un problème dur ?

La résolution de problèmes posés dans ce cadre reste difficile. Comme dans les MDPs, les agents doivent prendre en compte des récompenses différées pour agir de manière optimale. Mais en outre, les agents ne perçoivent qu'une partie de la récompense globale à maximiser et la présence d'interactions introduit des influences à long terme entre leurs comportements. Nous cherchons désormais à savoir s'il est possible de trouver une approche utilisant la programmation dynamique pour réduire la complexité de la recherche des politiques tout en conservant une approche décentralisée. Pour ce faire, nous envisageons plusieurs mécanismes : des mécanismes de négociation basés sur des apprentissages individuels préalables pour en tirer parti et des mécanismes de transfert de récompense lors des interactions pour mettre à jour localement les comportements individuels à partir de récompenses plus globales que celles partiellement observées par un agent.

13.5 Perspectives futures

Notre objectif à long terme est de pouvoir proposer une approche permettant d'utiliser des techniques d'apprentissages par renforcement dans des cadres multi-agents réalistes pour lesquels la récompense globale et l'état du monde sont perçus partiellement par les agents, pour lesquels les communications entre agents sont limitées et qui nécessitent de mettre à jour localement les comportements de chaque agent.

Bibliographie

- [BZI00] D.S. Bernstein, S. Zilberstein, and N. Immerman. The complexity of decentralized control of markov decision processes. In *16th Conference on Uncertainty in Artificial Intelligence (UAI'00)*, Stanford, California, pages 32–37, 2000.
- [RG98] Sutton R. and Barto G. *Reinforcement Learning : an introduction*. Bradford Book, MIT Press, Cambridge, MA, 1998.
- [SPG03] Y. Shoham, R. Powers, and T. Grenager. Multi-agent reinforcement learning : a critical survey. Technical report, 2003.

14. Protocoles de coordination et de conversation dans un système multi-agents

Abdelkader GOUAICH

LIRMM, 161 rue ADA, 34090, Montpellier.

gouaich@lirmm.fr

http://www.lirmm.fr/~gouaich

Domaine de recherche : SMA, interaction et coordination

Résumé

Un système multi-agents (SMA) est défini comme une agrégation cohérente d'entités autonomes et interactionnelles appelées agents. On peut le voir également comme une société virtuelle composée de citoyens autonomes et interactionnels que sont les agents. L'autonomie des agents est vue comme leur capacité à contrôler leurs lois internes de comportement et le cas échéant pouvoir les modifier sans intervention externe. Evidemment, pour qu'une agrégation d'agents autonomes se comporte de façon cohérente il faut établir certaines règles communes qui vont lier des comportements agents indépendants. Il ne suffit pas d'explicitier ces règles ou normes communes ; il faut également être capable de les vérifier (dynamiquement) durant le cycle de vie du système. Dans cette présentation, nous allons considérer les normes relatives à la coordination et aux conversations entre agents et montrer quelle est la relation entre un protocole de coordination et le protocole conversation sous-jacent. Nous présentons également une approche incrémentale et décentralisée pour contrôler la validité d'une conversation entre les agents par rapport à un protocole de coordination.

14.1 Objectifs

Notre objectif est de contrôler dynamiquement les normes sur la coordination dans un SMA [GOU03]. Pour parvenir à cet objectif nous avons établi les étapes suivantes :

- définir un modèle et une plate-forme de déploiement d'agents autonomes qui permet de contrôler les actions des agents. Cet environnement de déploiement, nommé MIC*, a été déjà défini dans des travaux ultérieurs [GGM03] ;
- définir un cadre de coordination basé sur les travaux de Malone et Crowson [MC94] ;
- définir un lien formel entre le protocole de coordination exprimé sous forme de graphe de dépendance et la conversation sous-jacente ;
- présenter un moyen pour valider une conversation entre agents par rapport au graphe de coordination en utilisant des réseaux de Petri à file ;
- intégrer les travaux sur la coordination et MIC*.

14.2 Le modèle formel

L'environnement de déploiement MIC* est une structure algébrique. L'utilisation de l'algèbre linéaire nous a permis de pouvoir définir la composition de plusieurs SMA. Ceci a été particulièrement intéressant pour la conception et l'implémentation de systèmes informatiques ubiquistes. L'environnement MIC* implémente le principe d'influence/réaction [JJ95] : les actions des agents ne modifient pas directement l'état de leur environnement ; mais l'environnement possède ses propres lois qui valident ou pas les actions des agents. Concernant la coordination, les protocoles de coordination sont exprimés comme des graphes orientés. Deux types de noeuds sont définis : les activités et les ressources. Une activité peut produire ou consommer une ou plusieurs ressources. De même, une ressource peut être produite ou consommée par une ou plusieurs activités. Quand une activité produit plusieurs ressources, ceci est considéré comme une production simultanée sans ordre particulier ; Quand une ressource est consommée par plusieurs activités, ceci est considéré comme un choix exclusif, une et une seule des activités peut s'exécuter. Le graphe de coordination est par la suite découpé en plusieurs partitions regroupant les activités d'un même rôle. Toutes les ressources qui vont apparaître entre les différentes partitions de rôles sont appelées la jointure des rôles. Etant donnée la propriété d'autonomie des agents, un observateur externe ne peut pas accéder aux informations se trouvant à l'intérieur des rôles ; cependant, il peut analyser les ressources qui

transitent dans la jointure des rôles. Nous allons utiliser l'observation des ressources dans la jointure des rôles comme moyen pour valider la cohérence de la conversation entre les agents par rapport au protocole de coordination prédéfini.

14.3 Les interactions

L'interaction est interprétée dans le contexte de ce travail comme la transition d'une ressource d'un agent à un autre. Par exemple, dans les SMA, les messages qui transigent entre les agents sont considérés comme des ressources. Une conversation est définie comme une séquence de ressources (ou de messages) échangées. Le graphe de coordination impose de facto une relation d'ordre partiel entre les différentes ressources. Nous allons vérifier que cet ordre partiel est respecté par la conversation entre les agents. Si une conversation ne respecte pas cet ordre, on peut conclure que le protocole de coordination a été rompu. D'un point de vue théorique, nous avons utilisé un système de réécriture pour définir l'ensemble des séquences générées par un graphe de coordination. Ainsi, à partir d'un graphe de coordination donné, on peut théoriquement définir le protocole de conversation sous-jacent. D'un point de vue pratique, nous avons utilisé des réseaux de Petri à file pour valider de façon incrémentale les conversations entre les agents par rapport à un graphe de coordination.

14.4 En quoi est-ce un problème dur ?

- Au niveau conceptuel, il est important de partir des propriétés fondamentales d'un SMA et de déduire les conséquences sur le système informatique qui sera construit. Il ne faut pas remettre en cause ces fondements en cours de route. Ainsi, tous les modèles formels et les structures informatiques développés s'inscrivent dans cette cohérence paradigmatique. Par exemple, l'axiome d'autonomie nous interdit de consulter l'état interne des agents ; en conséquence, la validation des protocoles de coordination n'est effectuée que par l'observation des conversations.
- Les protocoles de coordination exprimés sous forme de graphes de coordination sont simples, mais restent assez expressifs : on peut définir un protocole de coordination entre plusieurs rôles ; représenter des branchements dans le processus de coordination ; exprimer des productions simultanées asynchrones de ressources. Cependant, comme le graphe de coordination n'exprime qu'un ordre partiel entre les différentes ressources, décider de la validité d'une conversation de façon incrémentale n'est pas facile. C'est pourquoi nous avons défini les réseaux de Petri à file pour : capturer facilement le parallélisme induit par l'ordre partiel entre les ressources ; modéliser les différentes alternatives dues aux conflits sur les ressources (choix exclusifs) et aux ambiguïtés sur les ressources ; stocker l'état de la conversation qui est représentée par le marquage du réseau de Petri.
- La dernière difficulté est liée à la façon de contrôler les conversations entre les agents. En effet, si on suppose qu'il existe une seule entité de contrôle centralisée, le contrôle est facile et il suffit d'appliquer directement les résultats présentés. Cependant, si on se met dans le contexte distribué (ou même ubiquiste), l'état global de la conversation ne peut pas être maintenu de façon cohérente. C'est pourquoi nous avons adopté un contrôle décentralisé qui effectue un contrôle local sur chaque rôle participant à la conversation.

14.5 Perspectives futures

Les travaux présentés ont été implémentés. Ainsi, le concepteur d'un SMA utilise un outil graphique pour exprimer le graphe de coordination entre les différents rôles et la génération de code du contrôleur de conversation est automatique. Cependant, il reste les points suivants à développer :

- compléter l'intégration du cadre de coordination et l'environnement de déploiement MIC* ;
- offrir un cadre d'ingénierie de SMA utilisant la métaphore sociale pour construire des sociétés artificielles d'agents. La coordination des tâches sociales sera exprimée comme des graphes de coordination et l'environnement de déploiement garantira leur consistance.

Bibliographie

- [GGM03] Abdelkader GOUAICH, Yves GUIRAUD, and Fabien MICHEL. Mic* : An agent formal environment. 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2003), 7 2003. Orlando, USA.
- [GOU03] Abdelkader GOUAICH. Coordination and conversation protocols in open multi-agent systems. *Engineering Societies in the Agents World : 4th International Workshops, ESAW 2003, LNCS (3071/2004)*, pages 182–199, 2003.
- [JJ95] Ferber J. and Muller J.P. Influences and reaction : A model of situated multiagent systems. *International Conference on Multi-Agent Systems*, 1995.
- [MC94] Thomas W. Malone and Kevin Crowston. The interdisciplinary study of coordination. *ACM Computing Surveys (CSUR)*, 26(1) :87–119, 1994.

15. Instrumentation du suivi d'apprenants

Erik Gebers

UTC - Heudiasyc - CR de Royallieu BP20529 60205 Compiègne cedex

Erik.Gebers@hds.utc.fr <http://www.hds.utc.fr/~egebers>

Domaine de recherche : EIAH, ingénierie documentaire, standards FOAD

Résumé

Dans le cadre de la FOAD articulée sur des documents numériques, nous nous intéressons à la problématique du suivi des apprenants instrumenté par un système informatique. Nous proposons une approche fondée sur l'ingénierie documentaire et les standards du domaine pour réaliser cette instrumentation. Les résultats souhaités seront des principes de modélisation des échanges documentaires et des contributions aux normes du domaine.

15.1 Objectifs

Dans le cadre de la Formation Ouverte et A Distance (FOAD), mais plus particulièrement les formations articulées sur des documents numériques, les efforts de standardisation permettent de rendre progressivement les plates-formes utilisées interoperables. Nous nous intéressons aux standards pour les supports de formation numériques, qui cherchent à permettre l'échange de contenus entre les différentes plates-formes du domaine, que nous distinguons en deux classes fonctionnelles : les plates-formes de déroulement de formations (LMS pour Learning Management System) et celles de création de contenus (LCMS pour Learning Content Management System). Si c'est à la première qu'incombe la gestion de la tâche de suivi, les plates-formes de la deuxième catégorie doivent néanmoins produire des contenus permettant l'instrumentation de cette tâche.

Nous abordons dans nos recherches les problématiques de suivi des apprenants et son instrumentation selon une approche issue de l'ingénierie documentaire, centrée sur les contenus de formation. Notre problématique se présente alors comme la définition de structures documentaires explicitant les utilisations des contenus, i.e. leur manipulation par des apprenants [BC04]. Il s'agit d'étudier comment la structuration documentaire peut permettre l'interaction ainsi que le contrôle du déroulement de la formation et son suivi.

15.2 Le modèle formel

Nous n'avons pas à ce jour développé de modèle formel pour l'interaction de l'apprenant avec le contenu, ni pour l'interaction du tuteur avec celui-ci. L'objectif de nos travaux étant la définition de structures documentaires, nous nous sommes approchés du groupe MFI dans la perspective de pouvoir étudier la forme prise par les productions des apprenants dans les différents modèles étudiés par ce groupe. Cette phase d'étude devrait permettre de déterminer les fonctionnalités requises pour la réalisation du diagnostic, étude que nous découpons selon quatre axes :

Etat : Il s'agit de l'état de la relation entre l'apprenant et le contenu, i.e. l'ensemble de ses productions et l'état de complétude de la formation. Cet axe permet une analyse des travaux de l'apprenant selon la dimension espace.

Trace : Il s'agit cette fois de l'évolution de la relation apprenant-contenu, i.e. les différentes versions des productions de l'apprenant ainsi que les actions entreprises par l'apprenant pour atteindre l'état courant. Cet axe permet une analyse des travaux de l'apprenant selon la dimension temps.

Pré-programmation : Il s'agit d'intégrer le suivi dans le déroulement du contenu par pré-programmation, c'est la traduction d'une intention d'accompagnement tutoral dans un comportement particulier du contenu (proposition d'informations supplémentaires suite à des réponses erronées...).

A posteriori : C'est l'activité de suivi tutoral par excellence, le suivi étant réalisé suite à la production de ressources interprétables dans le système (une réponse particulière ou un rapport d'activité par exemple). Il dépend donc de l'état et de la trace.

15.3 Les interactions

L'ingénierie documentaire place le contenu de formation au centre du dispositif informatique : c'est sur celui-ci que s'inscriront les interactions de l'apprenant. Les interactions se présentent comme des possibilités de manipulation du contenu de formation, l'apprenant réécrit son contenu de formation, au sens large du terme (i.e. ajout de productions personnelles au sein du document sous forme de textes, dessins, enregistrements vocaux...). Or ces productions, qui constituent des ressources documentaires, doivent être structurées de manière à assurer leur utilisation à l'intérieur mais aussi à l'extérieur du contenu auquel elles se rapportent (pour la réalisation d'un suivi sur le LMS par exemple). Par cette structuration même, elles offrent des nouvelles possibilités de manipulation, grâce à la calculabilité des ressources digitales, permettant des nouvelles formes d'interaction (par exemple, un apprenant peut choisir de consulter l'ensemble de ses notes sur un cours, de manière transversale au contenu).

15.4 En quoi est-ce un problème dur ?

L'instrumentation du suivi s'accompagne d'une complexité inhérente à la tâche de suivi, les ressources pertinentes pour le diagnostic n'étant pas toutes clairement identifiées, la jeunesse des pratiques de la FOAD aidant, et étant donné la difficulté rencontrée par les formateurs à formaliser leur "modèle" de suivi [LB00]. Nous proposons dans [GA04] un ensemble de fonctionnalités issu d'une étude des principaux standards pour les contenus de formation (SCORM¹, AICC², IMS SSP et IMS QTT³) et des besoins des contenus de formation SCENARI. Il s'agit, à partir de cet ensemble, d'identifier les fonctions nécessaires à la constitution d'un environnement de formation permettant un suivi efficace et d'en dégager les structures documentaires à mettre en place dans les contenus et les productions des apprenants afin de permettre sa mise en oeuvre.

15.5 Perspectives futures

En partant des fonctions d'instrumentation du suivi répertoriées dans notre étude ascendante [GA04] ainsi que des travaux théoriques sur l'instrumentation du suivi ([Des01] par exemple), nous proposons de réaliser un modèle générique du processus de suivi des apprenants. L'objectif est double : ce modèle permettra d'avoir un référentiel pour la comparaison de la qualité du suivi assuré par les différents standards, dégageant ainsi des axes d'amélioration de ces standards, et nous permettra de définir les structures documentaires nécessaires à la gestion des interactions et à l'instrumentation du suivi. Cette dernière tâche devrait répondre à notre problématique d'explicitation de structures manipulables pour l'interaction et le contrôle des documents numériques.

Nous espérons également proposer à la suite de nos travaux des contributions aux normes du domaine, en poursuivant notre activité au sein de l'AFNOR et de l'IEEE et en l'étendant à la communauté européenne via le CEN (Comité Européen de Normalisation) et le réseau PROLEARN.

Bibliographie

- [BC04] B. Bachimont and S. Crozat. Réinterroger les structures documentaires : de la numérisation à l'informatisation. *Revue I3*, 4(1), 2004.
- [Des01] C. Desprès. *Modélisation et conception d'un environnement de suivi pédagogique synchrone d'activités d'apprentissage à distance*. PhD thesis, Université du Maine, 2001.
- [GA04] E. Gebers and M. Arnaud. Standards et suivi des apprenants : Possibilités offertes pour le suivi des apprenants par les standards du e-learning. *Article soumis à Distance et savoirs*, 2(4), 2004.
- [LB00] K. Lund and M.J. Baker. Interprétations par des enseignants des interactions d'élèves médiatisées par ordinateur. In *3ème Colloque international Recherche et formation des enseignants : Didactique des disciplines et formation des enseignants : approche anthropologique*, Marseille, février 2000.

¹<http://www.adlnet.org/>

²<http://www.aicc.org/>

³<http://www.imsglobal.org/specifications.cfm>

16. Planification située et stratégie d'équipe

Damien Devigne

SMAC - LIFL - USTL, Cité Scientifique, 59655 Villeneuve d'Ascq cedex
devigne@lifl.fr <http://www.lifl.fr/SMAC/projects/cocoa/>

Domaine de recherche : SMA, simulation, agents situés, stratégies d'équipes

Résumé

Nous présentons une solution permettant de gérer des équipes d'agents situés dirigés par un chef. Celui-ci est chargé d'établir le plan d'équipe et de distribuer les tâches à effectuer aux agents en fonction des spécificités de chacun. Il est important de donner de l'autonomie de décision aux agents afin de leur permettre de s'adapter à la configuration de leur environnement, qui n'est pas toujours connue lors de la planification par le chef. Nous présentons une manière de créer des plans d'équipes abstraits qui n'interfèrent pas avec le rôle des agents. Le moteur de planification utilisé par le chef pour établir ce plan est le même que celui d'un agent simple. Cette approche permet à une équipe d'agents hétérogènes de réaliser des tâches qui nécessitent l'utilisation des capacités de plusieurs agents.

16.1 Objectifs

L'objectif de notre travail est de concevoir un environnement de simulation de comportements d'équipes d'agents hétérogènes. Nous recherchons la rationalité des comportements, ce qui n'est pas forcément synonyme d'optimalité. Un comportement est jugé rationnel lorsque la décision prise par l'agent aurait pu être prise raisonnablement par un être humain confronté à la même situation. Notre approche collaborative vise essentiellement à donner de l'autonomie de décision à chacun des agents qui composent l'équipe, en fonction de ses spécificités. Pour cela, nous limitons le rôle du chef d'équipe à la planification de haut niveau (établissement de plans abstraits), à la distribution des tâches et au contrôle du bon déroulement du plan, à l'instar de certaines organisations humaines. D'autres approches utilisent des agents réactifs [CGGG03] dont les comportements sont difficiles à contrôler finement ou les réseaux de Pétri [CGLM04] représentant les plans statiques des agents. Ces plans sont calculés "hors-ligne" et ne peuvent pas s'adapter facilement à un changement de la situation qui n'a pas été prévu dans le plan initial.

16.2 Le modèle formel

Nous utilisons un modèle d'interaction original "peut-subir/peut-effectuer" [MPR03]. Dans ce modèle, les agents possèdent une propriété *peut-subir* qui liste les interactions dont ils peuvent être la cible (par exemple, l'agent *porte* peut subir l'interaction *ouvrir*). Certains agents, que nous appelons agents animés, ont en plus une propriété *peut-effectuer* qui liste les interactions qu'ils connaissent et sont capables d'effectuer. Une interaction *i* peut être déclenchée par l'agent *a* sur l'agent *b* si *a* peut effectuer *i* et *b* peut subir *i*. Les agents et les interactions sont indépendants ce qui permet de les réutiliser séparément dans diverses expériences.

Une équipe est un regroupement d'agents dont un agent particulier qui joue le rôle de chef. Le chef possède la liste des interactions que peuvent effectuer les membres de son équipe afin de pouvoir leur attribuer des tâches en fonction du but que l'équipe doit réaliser.

16.3 Les interactions

Les interactions permettent d'exprimer les règles qui régissent l'environnement dans lequel se trouve l'agent. La connaissance exprimée par une interaction est une connaissance globale qui ne dépend ni de l'agent ni de l'environnement. L'interaction *ouvrir* peut par exemple s'appliquer à tous les objets "ouvrables", que ceux-ci soient des portes ou des boîtes. Une interaction est constituée de trois parties :

- la partie *prémisses* qui désigne les conditions à remplir pour pouvoir déclencher l'interaction,
- la partie *garde* qui représente une condition spécifique portant sur la position géographique de l'agent par rapport à la cible de l'interaction,
- la partie *actions* qui exprime les conséquences de l'exécution de l'interaction.

Pour établir un plan, les agents utilisent ces interactions grâce à un chaînage arrière. Lorsqu'une interaction est donnée comme but à un agent, les conditions de l'interaction deviennent à leur tour de nouveaux buts que l'on peut résoudre grâce à d'autres interactions. Le plan de l'agent est un arbre constitué des interactions à effectuer pour parvenir au but initial.

Les *gardes* des interactions permettent d'exprimer les contraintes liées au caractère situé des agents. Il est possible d'obtenir des plans abstraits (du point de vue spatial) en ne tenant pas compte des gardes. Le plan peut alors être "instancié" en fonction de la situation géographique dans laquelle l'agent est placé.

Dans une équipe, le chef possède une connaissance sur les capacités des agents membres de l'équipe. La partie *prémisses* et la partie *garde* de ces interactions sont cachées au chef afin de répartir l'effort de planification entre les agents : lorsque le chef utilise l'une de ces interactions, son chaînage s'arrête car il n'y a pas de conditions à satisfaire. Ces interactions constituent alors des feuilles de l'arbre qui représente le plan de l'équipe, et elles sont données comme but aux membres de l'équipe par le chef. Chacun des agents, pour résoudre son but, se réfère à sa connaissance. Il considère alors l'interaction avec les prémisses et les gardes et établit un plan. Les agents participent ainsi activement au travail de planification.

16.4 En quoi est-ce un problème dur ?

L'intégration de la position géographique des agents dans la planification pose des problèmes spécifiques : la position relative des agents peut radicalement changer le plan. En effet, pour se déplacer vers la cible d'une interaction, il est parfois nécessaire d'établir un plan permettant de contourner ou de déjouer certains obstacles. De plus, la connaissance des agents est imparfaite à cause de la dynamique de l'environnement et de la présence d'autres agents interagissant avec les mêmes objets dans la simulation. Cette dynamique requiert la vérification des connaissances à chaque instant et l'adaptation rapide du plan en cas de modification dans l'environnement.

L'établissement de plans d'équipe qui tirent pleinement partie de l'hétérogénéité des agents pose plusieurs problèmes, notamment la représentation de la connaissance de l'équipe, la répartition de cette connaissance, et la communication entre les agents.

16.5 Perspectives futures

Nous aimerions pouvoir gérer les échecs au niveau de l'équipe. Lorsqu'un agent ne peut pas atteindre le but donné par son chef, il faut que le chef puisse analyser la raison de l'échec pour adapter le plan et parvenir quand même au but. Pour y parvenir, cela peut nécessiter la réorganisation de l'équipe ou le recrutement d'autres agents. Nous aimerions aussi pouvoir gérer la synchronisation des agents pour permettre des stratégies d'équipe de style "encerclément" de manière cognitive.

Bibliographie

- [CGGG03] D. Capera, J-P. Georgé, M-P. Gleizes, and P. Glize. The amas theory for complex problem solving based on self-organizing cooperative agents. In *Proceedings of the Twelfth IEEE International Workshops on Enabling Technologies : Infrastructure for Collaborative Enterprises (WETICE'03)*, pages 1080–1383, 2003.
- [CGLM04] M. Castilho, A. Guedes, T. Lima, and J. Marynowski. A petri net based representation for planning problems. In *International Planning Competition - IPC'04*, pages 27–29, 2004.
- [MPR03] P. Mathieu, S. Picault, and J-C. Routier. Simulation de comportements pour agents rationnels situés. In *Actes des Secondes Journées Francophones sur les Modèles Formels de l'Interaction (MFI'03)*, pages 277–282, mai 2003.

17. Gestion des interactions entre agents : application à la simulation de trafic routier

Arnaud Doniec

LAMIH UMR 8530, UVHC/ Equipe MSIS, INRETS

adoniec@univ-valenciennes.fr

Domaine de recherche : SMA, simulation

Ce travail s'inscrit dans le cadre d'une thèse co-financée par l'INRETS et la région Nord/Pas-de-calais. Elle fait l'objet d'une collaboration INRETS (S. Espié) et LAMIH (R. Mandiau & S. Piechowiak).

Résumé

La simulation comportementale de trafic a pour objectif de produire du trafic routier en modélisant le plus fidèlement possible les comportements des conducteurs à partir des études menées en psychologie de la conduite. Dans cette approche, le trafic est le résultat émergeant de la somme des interactions entre les différents acteurs de la simulation : conducteur, piéton, infrastructure, etc... Cette approche comportementale est souvent appliquée dans un cadre multi-agent [SBH98], [BCKR04], [Esp95].

Dans le modèle ARCHISIM [Esp95], développé à l'INRETS, chaque acteur de la simulation est un agent logiciel autonome disposant d'un ensemble de buts (un itinéraire, une vitesse désir,...) qu'il essaye de satisfaire en se déplaçant sur un réseau routier et en interagissant avec d'autres agents potentiellement en conflit avec lui.

Dans ce contexte, simuler le trafic en carrefour peut se ramener à un problème de coordination multi-agent compétitive en environnement dynamique : chaque conducteur simulé essaye de traverser l'intersection en évitant les accidents et les interblocages.

17.1 Objectifs

La simulation de trafic en milieu urbain pose le problème du franchissement d'intersections pour lequel les agents doivent se coordonner de manière compétitive (la tâche de conduite est généralement non-coopérative) dans le but de résoudre leurs conflits. Dans ce domaine, nous pouvons citer comme travaux significatifs : l'approche automate [RS93], l'approche contextuelle [TLD98] et la coordination à base de jeux [CEMK03].

L'utilisation du mécanisme de coordination à base de jeux proposé par A. Champion [CEMK03] apporte une première réponse à notre problématique dans le cas de carrefours simples et isolés. Notre objectif est donc d'étendre ces travaux de manière à gérer les interactions entre agents dans les situations de trafic complexes comme par exemple les situation inter-bloquantes.

17.2 Le modèle formel

Le mécanisme de coordination proposé par Champion et actuellement utilisé dans ARCHISIM ne s'avère pas suffisant pour la simulation de trafic dans les carrefours complexes puisqu'il ne permet pas la gestion de certains aspects comme l'occupation des zones de stockage, les remontées de queue et les interblocages.

A la problématique de coordination nous ajoutons donc un deuxième axe qui est la prise en compte de l'anticipation. L'anticipation fait partie de la tâche de conduite et d'autre part constitue d'un point de vue multi-agent un axe prometteur permettant de créer des comportements réalistes et complexes [Lai01].

Notre approche consiste à fournir à l'agent, une couche anticipative lui permettant d'évaluer l'impact de ses actions dans un horizon temporel plus grand, tout en tenant compte du contexte et de la situation de trafic environnante.

Un de nos objectifs étant d'éviter les interblocages infinis, nous utilisons une déclinaison de l'anticipation : l'*anticipation préventive*. Elle consiste à prévenir un certain nombre de situations non-désirées en essayant de projeter un ensemble d'actions dans le futur [Dav03]. Pour ce faire, notre modèle se compose d'une représentation de l'environnement perçu par l'agent et d'une fonction d'évaluation des effets des actions de l'agent.

17.3 Les interactions

Nous distinguons ainsi deux niveaux d'effets : 1) un niveau local portant uniquement sur les interactions entre l'agent et les autres conducteurs avec lesquels il est directement en conflit ; 2) un niveau global portant sur l'ensemble des interactions perçues par l'agent et incluses dans sa représentation de la situation.

Le niveau local pouvant être facilement calculé en fonction des informations relatives à la vitesse de l'agent et à la position des autres véhicules, nous proposons d'évaluer les effets d'une action au niveau global en utilisant : une représentation égo-centrée sous la forme d'un réseau de contraintes et les techniques de propagation de contraintes.

Trois types de contraintes, exprimant différents degrés de blocage, sont utilisées : le blocage physique, le blocage physique anticipé et le blocage par priorité. Le domaine associé à chaque agent présent dans la représentation exprime un horizon temporel représenté sous la forme d'un intervalle. La projection d'une action correspond donc à un ajout de contraintes suivi d'une propagation. Les situations non-désirées prennent alors la forme d'inconsistances au sein du réseau de contraintes.

17.4 En quoi est-ce un problème dur ?

L'obtention d'un trafic réaliste en situation de carrefour ne se résume pas à la simple coordination de flux de trafic conflictuels (intersection et fusion). Elle nécessite également la prise en compte de nombreux paramètres liés au contexte de la situation : géométrie du carrefour, espaces et zones de stockage au centre du carrefour, effets de bords entre intersections proches, etc.

L'anticipation et la reconnaissance du contexte sont donc nécessaires pour assurer un trafic émergeant visuellement et statistiquement réaliste. La modélisation de l'anticipation dans les systèmes intelligents fait parti des problèmes difficiles en intelligence artificielle ([Ros85],[Dav03]). Il faut en effet généralement composé avec un environnement dynamique pour lequel l'agent ne dispose que d'une vision partielle et par conséquent incomplète et imparfaite.

17.5 Perspectives futures

Jusqu'à présent seule l'accélération longitudinale a été gérée, ce qui du point de vue de l'agent revient à ne considérer que deux stratégies : *Go* et *Stop*. Nous visons donc à étendre notre modèle en associant à l'action *Go* une composante relative l'accélération latérale : *Go_droite*, *Go_gauche*. Cette extension devrait en particulier permettre d'anticiper sur l'occupation des espaces au carrefour et ainsi gérer plus finement les zones de stockage.

La seconde perspective de notre travail concerne la validation de notre approche. Cette validation porterait sur deux niveaux : une validation au niveau macroscopique à partir de données trafic mesurées en situation réelle et une validation au niveau microscopique à partir d'un visuel 3D de la scène.

Bibliographie

- [BCKR04] M. Balmer, N. Cetin, K., and B.K. Raney. Towards truly agent-based traffic and mobility simulations. In *Proceedings of AAMAS'04*, New-York, USA, 2004.
- [CEMK03] A. Champion, S. Espié, R. Mandiau, and C. Kolski. A game-based, multi-agent coordination mechanism - application to road traffic and driving simulations. In *Summer Computer Simulation Conference*, pages 644-649, Montréal, Québec, Canada, juillet 2003.
- [Dav03] P. Davidson. A framework for preventive state anticipation. In Springer, editor, *Anticipatory Behavior in Adaptive Learning Systems, LNAI, vol 2684*. 2003.
- [Esp95] S. Espie. ARCHISIM : Multi-actor parallel architecture for traffic simulation. In *Second World Congress on Intelligent Transport Systems*, Yokohama, Japon, 1995.
- [Lai01] J.E Laird. It knows what you're going to do : Adding anticipation to a quakebot. In *Proceeding of Autonomous Agent 2001*, Montreal, Canada, 2001.
- [Ros85] R. Rosen. *Anticipatory Systems*. Pergamon Press, 1985.
- [RS93] D.A. Reece and S.A. Shafer. A computational model of driving for autonomous vehicles. *Transportation Research*, 1993.
- [SBH98] R. Sukthankar, S. Baluja, and J. Hancock. Multiple adaptive agents for tactical driving. *International Journal of AI*, 1998.
- [TLD98] H. Trannois, A. Lebrun, and J.L. Deleage. A multi-agent framework for car traffic simulation. In *The Third International Conference and Exhibition on the Practical Application of Intelligent Agents and Multiagent Technology PAAM 98*, 1998.

18. Intégration des connaissances dans les environnements virtuels pour l'annotation

Stéphane Aubry

Heudiasyc/UTC UMR CNRS 6599, BP 20529, 60205 Compiègne

stephane.aubry@utc.fr

http://www.hds.utc.fr/~aubrys01

Domaine de recherche : IHM, réalité virtuelle, environnements collaboratifs

Résumé

Dans les projets de design collaboratif, et en particulier lorsque ceux-ci se réalisent à distance, une bonne communication entre les différents membres est un aspect central du projet. Un certain nombre de solutions existent pour faciliter la collaboration (environnements partagés, environnements virtuels collaboratifs ...), cependant, il existe encore un certain nombre de problèmes que nous détaillerons plus bas. Nous présentons ici certains aspects de l'utilisation d'annotations en environnement virtuel pour faciliter cette communication. Les annotations en environnement virtuel ont des spécificités, que nous présentons dans le modèle formel, et dont les deux aspects principaux sont les inscriptions de l'annotation, ainsi que la manière dont celle-ci est contextualisée dans le document de référence. Nous avons implémenté nos idées dans une application de test, et une mise en place de cette dernière nous a conduit à étudier l'intégration d'un modèle de connaissances qui permettra de faciliter la manipulation des produits annotés.

18.1 Objectifs

L'objectif général, est de faciliter la communication entre les différents membres d'un projet de design collaboratif de produit. Lors du déroulement d'un projet de ce type on constate certaines pertes d'informations ainsi que des difficultés dans l'utilisation des informations transmises sur le produit. On note en particulier :

- Des informations perdues lorsque le projet passe d'une équipe à une autre,
- Des problèmes de compréhension entre les différents membres du projet,
- Un manque de capitalisation des informations sur le design du produit, et des difficultés à accéder à l'information pertinente dans un ancien projet.

L'une des sources principales de ces problèmes vient du fait que le sujet du discours des designers (le produit) ainsi que ce discours (les informations qu'ils échangent) sont dans deux espaces différents. Nous proposons donc d'intégrer ces deux entités dans le même espace, afin de faciliter leur interprétation, ce qui nous conduit à la notion d'annotation en environnement virtuel. [ATL03] D'un point de vue plus théorique, notre objectif est d'étudier la possibilité d'associer dans un même environnement des données de natures différentes, et représentées par des modalités différentes, mais dont la compréhension nécessite une lecture simultanée.

18.2 Le modèle formel

Nous basons notre modèle d'annotations sur les modèles 2D déjà existant [ADM03] en mettant en avant les spécificités des annotations 3D par rapport aux annotations 2D. Nous définissons une annotation comme étant un document ou une marque ayant les propriétés suivantes :

- Son discours traite d'un autre document (le document cible), dans le cas qui nous intéresse, le document cible est un modèle virtuel du produit dans l'espace 3D.
- Il est indissociable de la cible (c'est-à-dire que le contenu ne peut être interprété qu'en présence de la cible)
- Mais il est perceptiblement distinguable de l'objet lui-même [BCGP00]

Notre modèle d'annotation est basé sur une décomposition de l'annotation en quatre composantes : les inscriptions de l'annotation, les méta-données, les éléments de contextualisation, et enfin un lien vers un modèle de connaissances. On note que les inscriptions de l'annotation 3D sont une extension des inscriptions des annotations classiques, dans le sens où l'on retrouve les éléments d'une annotation classique (texte, image, son, vidéo ...) mais y sont ajoutés des entités spécifiques à

l'espace 3D (formes 3D et données haptiques). D'autre part, les annotations 3D se distinguent de leurs équivalents 2D par leurs éléments de contextualisation. En effet dans un modèle d'annotation 2D, l'annotation est contextualisée par l'ancre seule, alors que les annotations 3D sont contextualisées à la fois par l'ancre et par le point de visualisation, défini par la position et l'orientation de la caméra au moment de la création de l'annotation. D'autre part, nous liions nos annotations à un modèle de connaissances permettant de disposer d'outils facilitant la manipulation d'annotations. Ces connaissances sont regroupées sous deux ontologies : une ontologie d'annotations qui renseigne sur la place de l'annotation dans le schéma argumentatif de la collaboration [ATLG05].

18.3 Les interactions

On peut considérer deux types d'interactions dans notre système : d'une part les interactions entre les utilisateurs et le système, et d'autre part les interactions d'un utilisateur à un autre. D'un point de vue des interactions entre les utilisateurs et le système, l'une des spécificités des interactions dans notre système est la nécessité de manipuler de manière conjointe des données qui appartiennent à deux espaces distincts : le produit appartenant à l'espace 3D et les annotations ainsi que les connaissances associées appartenant à un espace abstrait. Les interactions entre les utilisateurs ne sont pas directes, du fait qu'elles sont médiatisées par le produit auquel elles se réfèrent. Une classification de ces interactions a été développée, en les classant selon le rôle qu'elles occupent dans le processus d'argumentation.

18.4 En quoi est-ce un problème dur ?

Les annotations en environnement virtuel représentent une manière de collaborer innovante, et par conséquent, une partie du travail consista à la mise en place des modalités d'utilisation de ces annotations, ainsi qu'à déterminer l'ensemble des propriétés de ce support. L'une des questions auxquelles nous sommes confrontés concerne la modélisation des connaissances mises en place dans le système. En effet, des choix concernant la conception de l'ontologie, tels que par exemple son niveau de granularité, ou son niveau de formalisme ont une grande influence sur les possibilités d'exploitation de ces connaissances. D'autre part, la représentation et la manipulation de ces connaissances posent problème : comment représenter dans l'espace 3D des données aussi abstraites que le déroulement d'une collaboration, ou encore la sémantique d'une relation entre deux objets, et quels modèles de manipulation proposer ?

18.5 Perspectives futures

Nous travaillons actuellement à l'intégration notre modèle de connaissances à notre environnement d'annotation, et nous souhaitons évaluer les bénéfices de l'ajout de connaissances. Nous considérons en particulier trois situations différentes servant de briques de base à la collaboration : la lecture de produit annoté, l'écriture d'annotation, ainsi que la recherche d'annotation dans un ensemble de produits annotés.

Bibliographie

- [ADM03] F. Azouaou, C. Desmoulins, and D. Mille. Formalismes pour une mémoire de formation à base d'annotations : articuler sémantique implicite et explicite. In *EIAH 2003*, Strasbourg, France, 2003.
- [ATL03] S. Aubry, I. Thouvenin, and D. Lenne. Virtual training studio for collaborative mechanical design. In *Virtual Concept 2003*, Biarritz, France, 2003.
- [ATLG05] S. Aubry, I. Thouvenin, D. Lenne, and A. Guénand. Vr annotations for collaborative design. In *HCII 2005*, Las Vegas, US, 2005.
- [BCGP00] M. Baldonado, S. Cousins, J. Gwizdka, and A. Paepcke. Notable : At the intersection of annotations and handheld technologies. In *Proceedings of HUC Conference*, 2000.

19. Moyens de perception du trajet de formation par le tuteur en ligne

Philippe Teutsch

LIUM - Université du Maine - 72085 Le Mans cedex 9

Philippe.Teutsch@lium.univ-lemans.fr www-lium.univ-lemans.fr/~teutsch

Domaine de recherche : EIAH, FOAD, instrumentation

Résumé

Nos travaux s'intéressent à la conception d'instruments interactifs dédiés aux tuteurs en ligne. Ces acteurs essentiels des dispositifs de Formation Ouverte et À Distance (FOAD) ont besoin de connaître la situation de l'apprenant pour l'accompagner dans son activité d'apprentissage. Cette situation d'apprentissage peut être définie en termes de profil d'apprenant, de trajet déjà effectué et de situation d'interaction courante entre l'apprenant et le dispositif. L'objectif est de faciliter à la fois la perception du tuteur sur la situation de l'apprenant concerné par le suivi et la démarche d'apprentissage et d'autonomisation de l'apprenant lui-même.

19.1 Objectifs

Le tutorat en ligne correspond à un nouveau métier d'enseignant valorisant principalement les rôles de personne-ressource, d'animation, de parité et de régulation [Gue05]. La régulation s'appuie sur l'articulation des trajets particuliers des individus avec le curriculum planifié auquel ils sont confrontés. Les notions concernées sont le parcours planifié, le trajet effectif, le bilan personnel. L'outil visé doit aider à recontextualiser l'apprentissage afin d'offrir à l'utilisateur un regard immédiat, et discret, sur le parcours de l'apprenant [Lin01].

Nous nous intéressons aux questions d'accès et de traitement des traces d'activité d'apprentissage, traces gérées par les plateformes supports aux dispositifs de formation à distance [Des01].

L'objectif de la recherche est de définir les modèles permettant de gérer la visualisation des différentes informations composant le trajet de formation (le parcours planifié, les bilans personnels, les situations de blocage, les questions et productions de l'apprenant), ainsi que l'articulation entre ces différents points de vue. Il s'agit d'étudier comment la structuration des informations issues des traces d'activité, leur mise en forme et leur mise en relation peut faciliter et enrichir l'interaction entre le tuteur et les données.

Nous nous inscrivons dans une démarche d'instrumentation des activités de formation en environnement médiatisé [Tch02, HL03].

19.2 Le modèle formel

Nous ne disposons pas encore de modèle formel de description de l'interaction du tuteur avec les données dont il dispose, mais simplement d'un modèle de situation d'apprentissage qui se décompose en quatre composants principaux : identité, profil, trajet et situation courante [TBG04]. L'identité indique la localisation, la situation socio-culturelle, les conditions et les habitudes de travail de l'apprenant. Le profil décrit l'état des compétences de l'apprenant dans chacun des trois espaces composant l'environnement d'apprentissage : discipline, dispositif technico-pédagogique et autonomie d'apprentissage. Le trajet décrit l'état général de la situation de l'apprenant. Il correspond à une synthèse de la progression dans les ressources sous forme d'historique, de bilans ou d'étapes remarquables. La situation courante décrit les aspects les plus dynamiques des interactions entre l'apprenant et le dispositif : participation aux activités proposées, résultats, productions liées aux tâches proposées, échanges avec les autres acteurs du dispositif.

Ce modèle répond aux besoins de perception permettant de mettre en oeuvre une régulation assurée par un formateur qui va porter assistance à l'apprenant tout au long de son parcours d'apprentissage : mise en relief de son trajet personnel, analyse de l'activité, évaluation de ses productions, conseils en remédiation et en stratégies d'apprentissage.

19.3 Les interactions

La conception de dispositifs de FOAD présente de réelles problématiques de visualisation [SC05] et de manipulation de données volumineuses (parcours et activités de nombreux apprenants), disparates et hétérogènes (informations sur le contenu de la tâche, mais aussi sur les modalités, le contexte, l'(a)synchronicité, ...) et néanmoins nécessaires à la conduite de la tâche de régulation (vues d'ensemble, vues détaillées, zoom, filtres, ...).

Les premières mises à l'essai d'un dispositif de formation en langue étrangère nous ont permis de définir trois besoins essentiels au tuteur pour assurer son rôle de régulation. Il s'agit des besoins de perception suivants :

- Contexte d'émergence. Besoin de disposer du contexte d'émergence (phase d'activité) des sollicitations (question ou production de l'apprenant, avertissement du système) afin de permettre le travail réactif du tuteur.
- Articulation. Besoin de mise en relation de contextes différents afin de faciliter l'analyse et la synthèse de la situation. Articulation des différentes observables (traces de navigation, productions, échanges par messagerie), différentes de part leur nature, leur volume, leur granularité.
- Bilans. Besoin d'effectuer des bilans et des comparaisons entre le trajet réalisé et le parcours prévu (différence entre " curriculum vécu " et " curriculum planifié ").

19.4 En quoi est-ce un problème dur ?

La problématique des interfaces dédiées au suivi d'apprentissage ouvre un terrain de réflexion sur les interfaces liées à l'apprentissage et à sa représentation. D'une part en ce qui concerne les interfaces logicielles : leur pertinence tient à leur qualité d'incitation à l'action et aux modalités d'appropriation et d'intervention sur les données qu'elles offrent. Du point de vue du tuteur, l'interface a pour rôle de faciliter la perception et l'appropriation de la situation. Du point de vue de l'apprenant, l'interface qui apparaît renvoie aux hypothèses faites sur les usages auxquels elle est destinée et l'utilisateur apprenant doit pouvoir confronter son mode d'apprentissage à cette mise en forme. Nous abordons ici une seconde dimension de l'interface : pour s'en saisir et pour la mettre en oeuvre, l'utilisateur apprenant doit en élaborer une représentation intellectuelle qui, à terme, peut modifier son mode d'appropriation des connaissances. Cette compréhension du fonctionnement et de la pertinence de l'outil renvoie à la notion d'interface cognitive. Le dialogue avec l'interface concrétise la dimension métacognitive des apprentissages requise par l'exploitation d'un dispositif : réflexion sur le curriculum (planification, calendrier, gestion du temps, identification des objectifs), réflexions sur les stratégies déployées (savoir-faire, recours aux outils, dialogue avec le tuteur, gestion des feedbacks), ...

19.5 Perspectives futures

Notre proposition d'interfaces de tutorat vise à faciliter le travail du tuteur en ligne. Elle aide le tuteur à modéliser son propre travail en l'amenant à développer une attitude de régulation plus que de contrôle. Il est en parallèle intéressant d'offrir à l'apprenant les mêmes outils de perception de la situation d'apprentissage (la sienne) afin de lui faciliter, à lui aussi, la tâche de prise de conscience de ses actions et l'autocontrôle de son apprentissage.

Bibliographie

- [Des01] C. Després. *Modélisation et conception d'un environnement de suivi pédagogique synchrone d'activités d'apprentissage à distance*. PhD thesis, Université du Maine, Le Mans, 2001.
- [Gue05] O. Gueye. *Instrumentation des activités du tuteur en ligne, le cas de Croisières, dispositif de Formation Ouverte et À Distance en langues*. PhD thesis, Université du Maine, Le Mans, 2005.
- [HL03] R. Hotte and P. Leroux. Technologies et formation à distance. volume 10. 2003.
- [Lin01] M. Linard. Concevoir des environnements pour apprendre : l'activité humaine, cadre organisateur de l'interactivité technique. volume 8, pages 211–238. 2001.
- [SC05] B. Shneiderman and Plaisant C. *Designing the User Interface : Strategies for Effective Human-Computer Interaction*. Addison-Wesley, Fourth Edition, Chapter 14, 2005.
- [TBG04] Ph. Teutsch, JF. Bourdet, and O. Gueye. Perception de la situation d'apprentissage par le tuteur en ligne. In *Actes de TICE'2004*, pages 59–66, Compiègne (France), 2004.
- [Tch02] P. Tchounikine. Pour une ingénierie des environnements informatiques pour l'apprentissage humain. volume 2, pages 59–95. 2002.

20. De l'usage des ontologies et de la norme Topic Maps pour le e-learning

Ahcene BENAYACHE

HeuDiasyC/UTC UMR CNRS 6599, Compiègne

abenayac@hds.utc.fr

Domaine de recherche : e-learning, ontologies, Topic Maps, web sémantique

Résumé

La croissance et la diversité des ressources informationnelles utilisées dans le cadre d'une formation de type e-learning génèrent des problèmes d'accès, de classification et de gestion. Avec le projet MEMORAe1 (MEMOIRE ORGANISATIONNELLE Appliquée à l'e-learning), nous abordons ces problèmes et proposons d'y répondre par le concept de mémoire organisationnelle à base d'ontologies. Ainsi, pour chaque enseignement envisagé, une ontologie conceptuelle est élaborée. Cette dernière permet d'une part de définir les notions à appréhender et d'autre part, de mieux indexer les ressources décrivant ces notions en tenant compte du contexte de l'enseignement considéré.

20.1 Objectifs

Une formation est constituée d'acteurs (apprenants, enseignants, personnels administratifs, etc.), de différentes ressources (définitions, exercices, études de cas, etc.) rédigées sous différentes formes (livres, rapports, sites Web, etc.) et sur différents supports (papier, vidéo, audio, etc.), ainsi que de connaissances et de compétences qu'elle doit apporter. En ce sens, une formation est une organisation; c'est pourquoi dans le cadre du projet MEMORAe, nous proposons de réaliser une mémoire organisationnelle de formation basée sur des ontologies et de l'exploiter comme contenu mis à disposition des apprenants d'une formation e-learning. Cette mémoire se distingue d'une mémoire organisationnelle classique [DR00] par sa finalité qui est de mettre à disposition des utilisateurs un contenu et de le présenter pédagogiquement. Ce contenu est le résultat de la capitalisation des connaissances, ressources et informations relatives à la formation. Il est constitué des ressources nécessaires à la formation ainsi que d'index permettant d'accéder à ces ressources. Ces index représentent les granules ou notions à appréhender traités dans les ressources. Notre mémoire permet finalement de capitaliser toute connaissance, document, ou information traitant de ces notions. Parmi les documents représentés, certains (documents électroniques) sont directement stockés dans la mémoire, alors que les autres ne figurent que sous la forme de références. Cette capitalisation touche aussi toute information relative à l'environnement de la formation qui peut être utile à un utilisateur, telle que les informations concernant : les acteurs (membres), les dossiers d'inscription, etc. Afin que les utilisateurs de la mémoire puissent accéder aux ressources capitalisées, et puissent communiquer et collaborer entre eux, il est nécessaire :

- Qu'ils emploient un vocabulaire commun ; Que l'accès aux ressources soit le plus significatif possible.

Dans ce contexte, nous avons choisi une approche "Web cognitivement sémantique" pour modéliser notre mémoire. Cette approche privilégie la problématique de l'indexation à celle de l'inférence. Elle permet néanmoins des inférences "basiques" à partir de représentations dont la "sémantique opérationnelle" est plus faible que celle de représentations basées sur des langages formels supportant des traitements puissants [Cau02]. Elle permet aux utilisateurs d'accéder aux diverses informations concernant la formation choisie de manière significative et selon des points de vues multiples.

20.2 Le modèle formel

Dans le contexte du projet MEMORAe, l'utilisation d'ontologies nous permet de :

- lever les ambiguïtés dans les échanges entre les personnes, entre les personnes et les systèmes, et entre les systèmes ;
- définir les notions à appréhender lors de la formation (définition granulaire du contenu de la mémoire), ces notions servent d'index pour accéder aux ressources traitant de ces dernières ;

- structurer la mémoire de la formation.

Dans notre mémoire nous distinguons deux types d'ontologies [Ben04] :

1. Une ontologie du domaine : elle définit des concepts nécessaires à toute formation, tels que : livres, supports de cours, enseignant, apprenant, etc.
2. Des ontologies d'applications : elles définissent les concepts propres à une application donnée (notions à appréhender) dans un contexte précis. Il s'agit donc de définir une ontologie d'application par formation envisagée. Actuellement, nous en développons deux :
 - une ontologie pour le module NF01, enseignement d'initiation à l'algorithmique et à la programmation Pascal suivi par les étudiants de première année à l'Université de Technologie de Compiègne (UTC). Elle définit des concepts tels que : tableau, boucle, etc. ;
 - une ontologie pour le module B31.1, un enseignement de mathématiques appliquées suivi par les étudiants de l'IUP MIAGE de l'UPJV d'Amiens. Elle définit des concepts tels que : ensemble, cardinal, etc.

Pour exploiter le résultat de ces ontologies, nous avons choisi le formalisme des Topic Maps [IEC99]. Les concepts du formalisme (topic, association, occurrence, rôle, etc.) restent cependant transparents pour l'utilisateur.

20.3 Les interactions

Nous avons développé un environnement d'aide à l'apprentissage à distance qui met l'accent sur la visualisation et sur la navigation pour faciliter l'accès aux sources de connaissances. Il intègre les outils nécessaires aux activités d'autoformation par exploration.

L'intégration de ces outils au sein d'un même environnement est rendue possible par l'utilisation des deux ontologies évoquées auparavant, qui interviennent dans plusieurs phases de l'apprentissage par exploration.

20.4 En quoi est-ce un problème dur ?

Dans le cadre du projet MEMORAe, nous avons fait le choix d'évaluer l'apport d'une mémoire organisationnelle dans le contexte d'une formation e-learning. Cette approche consiste à utiliser des méta-données reposant sur des ontologies : une ontologie du domaine de la formation et une ontologie spécifique pour chaque application donnée. Ces méta-données permettent de repérer plus efficacement les documents en facilitant la recherche par descripteur ou marqueur. Afin d'assurer une bonne exploitation de la mémoire, MEMORAe intègre le formalisme des Topic Maps qui permet de décrire au mieux les ressources informationnelles de la mémoire. Nous situons ainsi notre approche au carrefour de trois domaines de recherche :

- l'Ingénierie Educative, pour la détermination des méta-données, des notions à appréhender et des liens entre ces notions,
- l'Ingénierie des Connaissances, pour le choix de l'organisation et la gestion du contenu d'une formation au moyen d'une mémoire organisationnelle basée sur des ontologies.
- le Web Sémantique, pour la structuration des informations de la mémoire en utilisant la norme ISO Topic Maps.

20.5 Perspectives futures

Après une première expérimentation, l'approche que nous avons choisie s'est révélée être un choix judicieux. Les étudiants ont apprécié l'indexation par le biais d'une ontologie. D'ailleurs, ils ont demandé à pouvoir poursuivre l'utilisation de la mémoire afin de préparer leur examen.

Une deuxième évaluation est en cours. En fonction des résultats de ces évaluations, nous essayons d'améliorer l'environnement développé et l'enrichir avec des annotations sémantiques. Nous développons actuellement un moyen d'import/export de données au format TXM (XML Topic Maps) afin de permettre un bon interfaçage avec d'autres environnements de même type.

Bibliographie

- [Ben04] Barry C.-Chaput B. Abel M.-H. Benayache, A. Construction of application ontology for the purpose of e-learning. In *E-learn 2004*, Washington, CD, USA, 2004.
- [Cau02] Cahier J.-P. Zacklad M. Charliet J. Caussanel, J. Les topic maps sont-ils un bon candidat pour l'ingénierie du web sémantique? In *IC'02*, Rouen, 2002.
- [DR00] Giboin A. Golebiowska J. Matta N. Ribière M Dieng R., Corby O. *Méthodes et outils pour la gestion des connaissances*. Dumod, 2000.
- [IEC99] *Topic Map*. International Standard ISO/IEC 13250, 1999.

21. Modèle de dialogue entre agents

Maxime MORGE

SMAC/LIFL - Bat M3 - F-59655 Villeneuve d'Ascq

menge@lifl.fr

http://www.lifl.fr/~menge

Domaine de recherche : Système multi-agents, Dialogue, Argumentation.

Résumé

Nous proposons un modèle formel de dialogue entre agents, intitulé **DIAL** (*DIALOGue Is an Argumentative Labour*) [?, ?]. Celui-ci permet de formaliser un processus de prise de décision collective et débattue. Ce modèle circonscrit un dispositif formel au travers duquel les agents jouent et arbitrent pour atteindre un accord. À cette intention, nous proposons un modèle de raisonnement argumentatif qui permet de gérer les conflits entre des arguments ayant des forces différentes selon l'agent qui les évalue. Nous proposons également un modèle d'agents argumentatifs qui justifient les hypothèses sur lesquelles ils s'engagent et prennent en compte les engagements de leurs interlocuteurs. Ainsi, dans le dispositif que nous avons circonscrit, la décision finale est du ressort d'un agent tiers. Celui-ci résout les conflits entre deux joueurs en fonction de leur compétence respective et des arguments avancés.

21.1 Objectifs

Dans le domaine de la gestion territoriale, l'adhésion d'un large panel d'acteurs est nécessaire pour garantir le succès des décisions. Le processus doit donner aux acteurs la conviction que la décision est équitable. Ce constat est à l'origine d'un changement de perspective dans l'élaboration des procédures démocratiques. Alors que la démocratie représentative consiste en un processus d'agrégation des préférences individuelles, la **démocratie dialogique** est un processus participatif de composition des perspectives et des intérêts [?]. Notre objectif consiste à formaliser ce type de processus à l'aide d'un système multi-agents.

21.2 Le modèle formel

DIAL (*DIALOGue Is an Argumentative Labour*) [?, ?] permet de modéliser formellement un processus de prise de décision collective et débattue à l'aide d'un système multi-agents.

Afin que les agents puissent gérer les conflits entre des arguments éventuellement contradictoires provenant de différentes sources, ils seront munis d'un modèle formel de raisonnement argumentatif [?]. Pour pouvoir exprimer des connaissances représentant l'état du monde, ce modèle de raisonnement s'appuie sur un langage logique sous-jacent. Dans le dessein de munir chaque agent de sa propre échelle de valeur, chacun d'eux doit évaluer indépendamment la force des arguments. À cette intention, nous définissons une logique argumentative qui constitue le modèle de raisonnement des agents. Cette **logique argumentative à base de valeurs** permet de gérer les conflits entre des arguments contradictoires qui sont plus ou moins forts selon l'agent envisagé.

Ayant défini le modèle de raisonnement des agents, nous spécifions dans le modèle DIAL comment les arguments peuvent être avancés et comment ils sont adoptés. À cette intention, nous proposons **un modèle d'agent argumentatif** qui leur permettent d'échanger des arguments et de raisonner conjointement.

Afin de garantir l'obtention d'un résultat au terme du processus et d'évaluer sa qualité, nous spécifions dans le modèle DIAL comment gérer le flux d'information en régulant l'enchaînement des arguments afin de mener à bien le processus. À cette intention, nous proposons un dispositif formel intitulé **système dialectique** dans lequel le processus se déroule.

21.3 Les interactions

Pour interagir, des agents logiciels doivent partager un langage commun, appelé langage de communication d'agents (ACL). En s'inspirant de la théorie des actes de langage [?], l'approche mentaliste spécifie la pragmatique des messages

en terme d'état mentaux. Pour les trois raisons suivantes, cette approche ne permet pas de modéliser une prise de décision concertée :

1. Les messages constituent des briques élémentaires qui doivent être ordonnées dans une conversation, conformément à un protocole d'interaction. D'une part, ces protocoles sont trop rigides. D'autre part, les propriétés d'un protocole ne permettent ni de garantir l'obtention d'un résultat ni de préjuger de sa qualité.
2. La sémantique des actes de langage est privée et subjective. Un agent qui assiste à une conversation ne confère pas le même sens à un acte de langage que son destinataire. *A fortiori*, il ne réalise pas les mêmes inférences.
3. Le modèle de raisonnement sous-jacent ne permet pas à un agent de gérer les incohérences entre ses propres croyances et les informations reçues.

Pour pallier cette dernière lacune, nous nous sommes orientés vers un modèle de raisonnement argumentatif. L'argumentation constitue un modèle de raisonnement adapté au processus cognitif d'un agent autonome et social pour gérer les interactions entre les arguments internes qui explicitent ses croyances et les arguments externes qui créditent des croyances contradictoires provenant d'autres agents.

Néanmoins, l'argumentation ne constitue pas un modèle satisfaisant pour étudier la dynamique des flux informationnels dans un processus d'argumentation. C'est pourquoi nous nous sommes intéressés à la dialectique. Par dialectique, on désigne la conception de dispositifs au travers desquels l'argumentation est mise en situation [?]. Un système dialectique est un dispositif au travers duquel un ensemble de participants jouent des coups dialogiques. D'une part, un dialogue est régulé par une convention, *i.e.* un ensemble de règles dialectiques. Ces règles spécifient, étant donné un contexte dialogique, les locutions qui sont autorisées (ou non). D'autre part, les déclarations sont stockées au fur et à mesure dans des structures de données intitulées tableaux d'engagements.

Ainsi DIAL consiste est un modèle de dialogue entre agents, *i.e.* un modèle d'interaction directe entre agents (par envoi de messages) qui s'inspire de la dialectique. Contrairement à l'approche mentaliste, cette approche permet de modéliser une prise de décision concertée :

1. Les règles dialectiques constituent une alternative au protocole d'interaction. D'une part, ce sont des structures collaboratives de communication flexible qui permettent de coordonner l'activité dialogique des agents. D'autre part, le système dialectique constitue un dispositif formel qui permet de garantir l'obtention d'un résultat et d'évaluer sa qualité.
2. Les engagements pris sont publics et objectifs. Un agent qui assiste au dialogue confère le même sens aux actes de langage et réalise les mêmes inférences qu'il en soit le destinataire ou non.
3. Le raisonnement des agents s'appuie sur un logique argumentative. Ce modèle de raisonnement permet aux agents de gérer les incohérences entre ses propres croyances et les informations transmises par d'autres agents.

21.4 En quoi est-ce un problème dur ?

Ce travail de recherche combine et réconcilie un ensemble de techniques éparses (argumentation, dialogue, SMA, ...) provenant de différentes disciplines (philosophie, sociologie, linguistique, logique, intelligence artificielle) dans un cadre formel cohérent.

21.5 Perspectives futures

Les dialogues envisagés portent sur des formules qui représentent un état du monde. Il pourrait être intéressant de prendre en considération des décisions qui devront être (ou non) mises en œuvre. L'objectif consiste alors à élargir le cadre pour pouvoir y jouer des négociations.

22. Modélisation des conflits dans l'activité de pilotage

Frédéric Dehais, Charles Lesire, Catherine Tessier

Onéra SUPAERO, av E. Belin 31055 Toulouse Cedex 4

dehais@supaero.fr; {tessier, lesire}@onera.fr <http://www.supaero.fr>

Domaine de recherche : IHM, modélisation de comportements humains, systèmes hybrides, aéronautique

Résumé

Ces travaux se proposent d'apporter une contribution théorique et pratique à la résolution de la problématique de l'erreur humaine en général et à l'aéronautique en particulier. Ils se concentrent notamment sur l'étude du conflit, concept marginal, peu étudié dans le domaine ou mal défini dans la littérature. Pourtant la revue d'incidents aéronautiques montre que l'apparition de conflits dans la gestion du vol est un précurseur remarquable d'accidents. Ceux-ci peuvent se produire par exemple entre l'équipage et le contrôle aérien, entre l'homme et les systèmes embarqués, dans la représentation de la position de l'aéronef ou dans la compréhension des procédures de vol. La détection formelle de ces conflits dans les interactions pilote-système mais aussi la mise au point de contre-mesures cognitives pour assister les équipages représentent les principaux enjeux de ces recherches.

22.1 Objectifs

Nous cherchons à définir une méthode et des outils pour identifier des précurseurs de la dégradation de l'activité de pilotage et à mettre en œuvre des moyens pour empêcher la détérioration des interactions pilote-système. L'approche classique est celle de la modélisation des erreurs des opérateurs, définis comme des écarts à la norme [Lep97]. Pourtant au regard des limites épistémologiques et opérationnelles de cette approche, nous sommes en mesure de nous demander si la modélisation de ce concept constitue une approche nécessaire et suffisante pour rendre compte de l'état de dysfonctionnement d'un système homme-machine.

Au contraire, l'étude du conflit en psychologie sociale [Sim03] montre que sa présence est révélatrice d'une dynamique de tension et d'opposition au sein d'une société; en intelligence artificielle distribuée, son apparition permet de diagnostiquer un état de dysfonctionnement dans les interactions entre les agents artificiels. Par ailleurs, le conflit engendre des comportements dangereux, tels que l'inhibition, l'agressivité, l'exclusion ou l'incompréhension, qui peuvent remettre en cause la sécurité du vol lorsqu'ils se produisent au sein de l'équipage.

La modélisation des conflits dans le cadre des interactions pilote/système représente une alternative intéressante et novatrice pour détecter et prédire l'apparition d'événements aéronautiques.

22.2 Le modèle formel

Nous proposons que la notion de conflit recouvre l'impossibilité pour un agent ou un groupe d'agents de satisfaire un but important. Plus formellement, la modélisation du conflit repose sur l'étude de l'incohérence au sein de la connaissance portée par des agents, l'idée consistant à montrer que pendant une période temps donnée il existe des connaissances qui ne peuvent être satisfaites simultanément [Cas00]. Pourtant, l'incohérence d'un ensemble de connaissances ne représente qu'un conflit larvé ou potentiel. C'est seulement si une incohérence entraîne des conséquences cruciales pour les agents que la situation va devenir conflictuelle [EFKN94]. Le formalisme utilisé pour représenter les connaissances et détecter les conflits est celui des réseaux de Petri hybrides particulières [LT05].

22.3 Les interactions

Les interactions que nous considérons dans le cadre de notre application sont relatives au dialogue pilote/système embarqué (ex : IHM, pilote automatique...) L'identification de conflits dans ces interactions repose (1) sur l'analyse des d'incohérences entre les actions de l'équipage sur l'interface d'une part, et de l'état ou la réponse des systèmes embarqués d'autre part (ex : commande, alarme..) (2) sur la prédiction des conséquences de ces incohérences sur la sécurité du vol (ex : collision).

Par ailleurs, la réalisation des expérimentations en simulateur de vol pour valider nos modèles des conflits dans les interactions pilote/système a permis de montrer que les pilotes, lorsqu'ils sont confrontés à un conflit, ont tendance à se focaliser sur sa résolution au détriment de la surveillance du vol (ex : insensibilité aux alarmes) [DTC03].

L'identification d'un tel comportement nous a amenés à modifier les dialogues interactifs entre l'IHM et le pilote lorsque celui-ci se produit : l'information sur laquelle se focalise le pilote disparaît puis est remplacée momentanément par un message pertinent en terme de sécurité.

22.4 En quoi est-ce un problème dur ?

Cette recherche s'intéresse à l'identification et à la prédiction automatique de situations psychologiques qui peuvent conduire à la dégradation de l'activité de pilotage. L'approche que nous suivons s'inscrit dans un esprit de transdisciplinarité et de pluridisciplinarité, puisque les fondements du conflit sont formels, biologiques, psychologiques et sociologiques et ensuite, parce que son identification et sa résolution nous amènent à utiliser des théories, des méthodes et des outils issus de l'intelligence artificielle, de la psychologie expérimentale, de l'ergonomie et de la neuropsychologie. La première difficulté est donc d'arriver à intégrer, dans un même système, toutes ces dimensions.

De plus le système (avion/pilote) étudié est *hybride*, son comportement est complexe et on ne dispose pas de modèle du pilote. Il est également nécessaire de gérer l'incertitude liée au modèle et au système, et le problème des "fausses détections".

22.5 Perspectives futures

Les perspectives s'orientent vers la réalisation d'expérimentations dans le simulateur de vol de SUPAERO pour évaluer avec des pilotes la capacité de nos modèles à détecter les conflits dans la gestion du vol.

Par ailleurs, un des objectifs est de pouvoir disposer d'un vecteur d'état "physiologique du pilote" de manière à pouvoir estimer le niveau de stress du pilote et de le corrélérer avec la détection de conflits. L'idée est de compléter la modélisation avec des grandeurs physiologiques et conduire des expérimentations avec des appareils de mesures physiologiques (eye tracking, RED, electro-cardiogramme..)

Bibliographie

- [Cas00] C. Castelfranchi. Conflict ontology. In H.-J. Müller and R. Dieng, editors, *Computational conflicts - Conflict modelling for distributed intelligent systems*, pages 21–40. Springer Verlag, 2000.
- [DTC03] F. Dehais, C. Tessier, and L. Chaudron. Ghost : experimenting conflicts countermeasures in the pilot's activity. In *Ijcai conference*, Acapulco, Mexico, Août 2003.
- [EFKN94] S. M. Easterbrook, A. C. W. Finkelstein, J Kramer, and B. A. Nuseibeh. Coordinating distributed viewpoints : the anatomy of a consistency check. *Journal of Concurrent Engineering : Research and Applications*, 2(3), 1994.
- [Lep97] J. Leplat. *Regards sur l'activité en situation de travail : contribution à la psychologie ergonomique*. Paris, PUF, 1997.
- [LT05] C. Lesire and C. Tessier. Particle Petri nets for aircraft procedure monitoring under uncertainty. In *26th International Conference on Application and Theory of Petri Nets and Other Models of Concurrency (ATPN'05)*, 2005.
- [Sim03] G. Simmel. *Le conflit*. Circé/poche, 2003.

pour le réseau, par conséquent, les agents n'embarquent que le protocole et les connaissances nécessaires pour pouvoir interagir avec les autres agents et instancier correctement les messages.

23.4 En quoi est-ce un problème dur ?

Décrire formellement un protocole et le valider ne pose plus de problèmes mais intégrer ce protocole dans l'agent reste encore peu considéré dans la littérature. En particulier, la relation avec les actions et les connaissances de l'agent, et comment s'assurer que le protocole conserve les bonnes propriétés édictées. Le raffinement formel et la validation de propriétés tout au long permet de s'assurer d'une meilleure prise en compte du protocole dans l'agent.

23.5 Perspectives futures

Les règles de raffinement reste à décrire précisément et en particulier à assurer de leur exhaustivité et qu'elles permettent de définir un agent utilisant correctement le protocole. La suite des travaux portera à considérer un raffinement pour un système multi-agent complet.

23. Processus formel de raffinement pour la conception de protocoles et des agents

Marc-Philippe Huet

B.P. 806 74016 Annecy Cedex

Marc-Philippe.Huet@univ-savoie.fr <http://marcphilippe.huet.free.fr>

Domaine de recherche : Systèmes Multi-Agents, Protocoles d'interaction, Raffinement

Résumé

La conception de protocoles, et en particulier leur implémentation dans les agents, demeure une activité difficile. Il s'agit de bien comprendre comment le protocole doit s'articuler dans l'agent. Nous choisissons ici une approche par raffinement pour concevoir le protocole puis pour aller du modèle formel vers le code et l'agent. L'agent est dans le cas de notre recherche spécialisé dans une problématique spécifique (négociation d'une ressource, participation à une vente aux enchères, etc.)

23.1 Objectifs

Généralement la description formelle d'un protocole ne pose pas de problèmes aux concepteurs de systèmes multi-agents puisque la difficulté réside dans l'intégration de ce protocole dans l'agent. S'agit-il d'une partie distincte de l'agent ou au contraire est-ce que l'échange de messages est intégré dans les actions de l'agent ? Afin d'aider le concepteur dans l'intégration du protocole dans l'agent, nous proposons une approche par raffinement. Le concepteur décrit petit à petit le protocole puis progressivement passe à l'agent qui est alors une couche englobante au protocole. En effet, dans nos travaux, l'agent est considéré comme un agent jetable qui répond à un besoin unique d'interaction et qui disparaît après. Par l'intermédiaire des règles de raffinement, le concepteur va définir le protocole et l'agent tout en validant à chaque pas de raffinement que le processus respecte bien certaines propriétés définies par le concepteur, par exemple : l'anonymat de l'agent ou la fiabilité des envois de messages.

23.2 Le modèle formel

Les travaux de recherche dans le domaine du raffinement propose plusieurs langages formels comme les langages Z et B ou le pi-calcul. Nous proposons ici d'utiliser une approche par automates à états finis. Le choix des automates à états finis est conditionné par deux motifs : (1) les automates à états finis est un langage fréquemment employé pour représenter les protocoles dans les systèmes multi-agents, et (2) il est facile de valider des propriétés sur des automates à états finis, soit directement s'il s'agit de propriétés structurelles sur les automates, soit par le biais de model checkers. De plus, dans le cadre du raffinement, les automates pourront facilement être réécrits pour correspondre aux changements que l'on y apporte, par exemple retirer un message, remplacer un message par un autre, instancier un message, etc. Aux automates à états finis doivent être adjoints des règles de raffinement qui indique ce qui est possible de faire sur les automates et comment s'effectue la transformation. Nous choisissons une formule de la logique du premier ordre pour les conditions à satisfaire avant d'exécuter la règle et un mini-langage pour la partie actions qui correspondent aux actions à effectuer sur l'automate, comme ajouter/retirer/modifier un état, ajouter/modifier/retirer une transition, instancier, modifier des messages, etc.

23.3 Les interactions

Les interactions qui nous intéressent ici sont des interactions pour des agents effectuant une unique tâche d'interaction, par exemple négociation d'une ressource, participation à une vente aux enchères, etc. Ces interactions interviennent dans le domaine des agents mobiles qui doivent satisfaire les objectifs que nous leur donnons sans toutefois être trop volumineux

24. Une interface conversationnelle pour une aide intelligente

Emerson PARAIISO, Jean-Paul A. BARTHES

Université de Technologie de Compiègne - Heudiasyc - UMR CNRS 6599

eparaiso@utc.fr

<http://www.hds.utc.fr/~eparaiso>

Domaine de recherche : IHM, SMA, agents assistants, ontologies, dialogue, interface conversationnelle

Résumé

Notre recherche concerne la conception d'une interface conversationnelle pour les agents assistants personnels, travail qui s'inscrit dans un processus de développement d'un modèle générique d'agent assistant personnel. Nous estimons qu'une interface conversationnelle spécialement conçue pour l'agent assistant, simplifiera son pilotage, entraînant une amélioration de la qualité de l'aide apportée par rapport aux interfaces classiques purement graphiques (connues comme WIMP - window, icon, menu, pointer). Ceci diminuera la charge cognitive de l'utilisateur, puisque le travail d'interprétation et la reconnaissance des intentions de ce dernier devient la responsabilité de la machine. Nous postulons aussi que le comportement intelligent de l'agent peut contribuer à l'amélioration de la performance liée à une telle interface, grâce à une politique d'affichage d'informations.

24.1 Objectifs

Le but de notre travail est la conception d'une interface conversationnelle pour les agents assistants personnels [PB04], travail qui s'inscrit dans un processus de développement d'un modèle générique d'agent assistant personnel. Après avoir examiné les divers types d'interfaces utilisées par des agents, nous avons choisi les interfaces conversationnelles [Kol99] comme interfaces idéales pour nos besoins. Nous nous sommes donc focalisés sur la définition et la conception d'une interface conversationnelle pour un agent assistant, en tenant compte de plusieurs contraintes et hypothèses liées à l'agent lui-même, à son pilotage et à la nature du rapport utilisateur-agent, mais aussi aux caractéristiques des applications potentielles [PB05]. Dans cette recherche, nous proposons une interface conversationnelle permettant à l'utilisateur d'utiliser le langage naturel parlé pour piloter son assistant. De notre travail est né le concept d'interface conversationnelle pour une aide intelligente - ICAI. Une interface conversationnelle pour une aide intelligente est le résultat de l'union d'un mécanisme conversationnel en langage naturel parlé et de la gestion intelligente de ce mécanisme, permettant le déroulement d'un dialogue coopératif, et capable de gérer le déclenchement de plusieurs tâches à la demande de l'utilisateur, tout en n'exigeant qu'un minimum d'effort de la part de ce dernier.

Les principales contributions de notre recherche sont :

- la conception de l'interface conversationnelle, basée sur :
 - la définition d'un système de dialogue fondé sur les actes de langage directifs (ordre, question et réponse);
 - l'adoption d'une stratégie coopérative pour le système de dialogue;
 - l'ancrage sémantique à travers les ontologies;
 - la séparation physique des connaissances de domaine et des tâches;
 - la mise en place d'une politique d'affichage d'informations, un premier pas vers une politique de présentation;
- une liste de principes de base à respecter pour élaborer une ontologie pour la gestion de dialogues avec un agent assistant.

24.2 Le modèle formel

Notre approche s'appuie sur les ontologies, fil conducteur de nos travaux, qui nous permettent d'interpréter les messages, provenant de l'utilisateur ou d'ailleurs. Les ontologies sont la seule forme de représentation des connaissances dont nous avons besoin. Notre travail montre pourquoi et comment il est possible d'utiliser les ontologies au cours des analyses

syntactique et sémantique dans le cadre d'un système de dialogue spécialement conçu pour être intégré à l'interface conversationnelle de l'agent assistant.

Après l'analyse syntaxique, nous trions les énoncés suivant une politique "d'encadrement" de l'espace de dialogue. L'objectif est de caractériser les buts de l'utilisateur en identifiant les actes de langage présents dans l'énoncé. Dans notre travail, nous nous intéressons plus particulièrement aux actes illocutoires, car ils explicitent les intentions d'utilisateur. Nous avons restreint l'analyse des énoncés à trois catégories d'actes de langage : question, réponse et ordre.

Ce résultat est utilisé au cours de l'analyse sémantique, qui résout la construction d'une représentation formelle de l'énoncé.

24.3 Les interactions

Dans cette recherche les interactions sont considérées du point de vue du dialogue homme-machine ainsi que du point de vue du système multi-agents. Du point de vue du dialogue homme-machine, les interactions sont en langage naturel parlé. Un gestionnaire de dialogue est responsable de la gestion de la conversation entre l'agent et son maître. Du point de vue du système multi-agents les interactions sont vues dans une perspective de gestion et délégation de l'exécution des tâches par des agents de service. L'agent assistant délègue l'exécution des tâches à des agents de service qui lui rendent les résultats.

24.4 En quoi est-ce un problème dur ?

La conception et la réalisation d'une interface conversationnelle pour l'agent assistant ne sont pas évidentes. Les principaux défis au développement d'une telle interface, sont les suivants :

- la conception d'un mécanisme d'analyse syntaxique robuste, capable de traiter des énoncés parlés, pas forcément bien reconnus;
- la conception d'un analyseur sémantique, capable d'interpréter les énoncés, pas toujours liés au domaine de l'application;
- la conception d'un gestionnaire de dialogue capable de gérer des conversations sur des domaines spécifiques, capable de déclencher plusieurs tâches simultanément et capable de bien gérer l'arrivée de messages provenant de plusieurs sources différentes (d'autres agents);
- la mise en place d'une structure de mémoires, capable de fournir des informations à la gestion du contexte de la conversation;
- la conception d'une politique d'affichage d'informations, gérée par l'agent assistant, pour organiser la façon d'interrompre l'utilisateur : pour lui poser des questions et pour lui présenter des informations diverses, comme des résultats d'exécution des tâches;
- la capacité d'ancrer le raisonnement de l'agent sur les ontologies dont l'agent dispose.

24.5 Perspectives futures

Concernant les voies de recherches futures, nous identifions plusieurs chemins possibles, dont la conception et la mise en place d'une politique de présentation, l'étude de l'impact de la mobilité de l'utilisateur et l'approfondissement de l'étude du rôle des ontologies dans l'interprétation et dans la personnalisation de l'interaction entre l'utilisateur et son agent assistant.

Bibliographie

- [Kol99] A. Kolzer. Universal dialogue specification for conversational systems. In *Sixteenth International Joint Conference on Artificial Intelligence - IJCAI 1999*, Stockholm, Suède, 1999.
- [PB04] Emerson Cabrera Paraiso and Jean-Paul A. Barthes. Une interface conversationnelle pour les agents assistants appliqués à des activités professionnelles. In *16th Conférence Francophone sur l'Interaction Homme-Machine*, Namur, Belgique, 2004.
- [PB05] Emerson Cabrera Paraiso and Jean-Paul A. Barthes. Speechpa : An ontology-based speech interface for personal assistants. In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, Compiègne, France, 2005.

25. L'intelligence Artificielle Distribuée appliquée aux processus physiques

DAPOIGNY Richard

LISTIC/ESIA Université de Savoie BP 806 74016 ANNECY Cedex
richard.dapoigny@univ-savoie.fr http://www.listic.univ-savoie.fr

Domaine de recherche : modélisation, Information Flow, systèmes distribués, contrôle intelligent

Résumé

La conception et de la mise en oeuvre de systèmes de mesure et de contrôle intelligents en milieu industriel nécessite la spécification de structures formelles. Ces structures conceptuelles définissent un cadre ontologique, où l'aspect distribué suggère l'introduction de la notion de service. La nature dynamique des services les rend capables de s'adapter et de se composer. Leur processus de composition via une approche type flot d'informations, peut être automatisé sur la base d'une logique distribuée. Cet aspect est essentiel pour garantir la consistance de la composition de services au travers de la notion de dépendance fonctionnelle. Parmi les extensions possibles, il est envisagé de généraliser la représentation des connaissances à d'autres domaines, et d'étendre la théorie IF par les institutions.

25.1 Objectifs

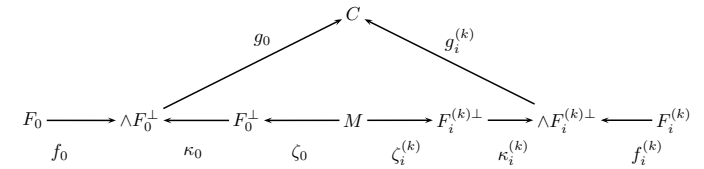
Dans le cadre de la conception et de la mise en oeuvre de systèmes de mesure et de contrôle intelligents en milieu industriel reposant sur un formalisme orientée but, les approches proposées utilisent principalement des hiérarchies de buts formels [vL01]. Le principal inconvénient de ces approches réside dans la faible expressivité de leur représentation [BPML05]. Par ailleurs, cette insuffisance a un impact certain sur l'aptitude du modèle à composer de telles hiérarchies en milieu distribué. L'approche proposée consiste donc en la spécification d'une structure formelle de buts reposant sur la notion de service. Cette structure permet des manipulations algébriques et s'intègre dans un modèle ontologique de description des connaissances, où chaque service est associé à une ontologie fonctionnelle de buts. L'automatisation du processus de composition des services est rendue possible par l'introduction d'un modèle centré sur l'approche Information Flow (IF), théorie formalisant les échanges d'information entre systèmes complexes [BS97].

25.2 Le modèle formel

Le modèle proposé s'inspire d'une subdivision de la connaissance ontologique reposant sur trois ontologies, i.e. structurelle, fonctionnelle et comportementale. La description de l'ontologie structurelle de l'environnement du système fait ressortir les liens méronymiques et topologiques entre les entités physiques, où deux entités sont connectées si elles sont susceptibles d'échanger de l'énergie ou de l'information [DBF03]. L'ontologie structurelle introduit la notion de contexte physique $c = (r, \mu(r), \varphi_1, \varphi_2, \dots, \varphi_{\mu(r)})$. A partir des contextes physiques, les hiérarchies de buts sont construites via un mécanisme utilisant l'analyse formelle de concepts (FCA)[DBBF05].

25.3 Les interactions

La recherche de dépendances entre buts se ramène à la construction progressive d'une logique distribuée. Les buts sont de nature dynamiques et capables de s'adapter et de se composer. La connaissance structurelle et fonctionnelle locale au système est constituée d'une connaissance initiale à laquelle vient s'ajouter la connaissance distante au fur et à mesure des échanges. Celle-ci est rattachée logiquement à la hiérarchie locale. Pour cela, la génération de services dynamique utilise la construction d'un canal d'information (IF channel) à partir de deux entités fondamentales, les classifications et les infomorphismes. IF est (entre autres) utilisé en partages d'ontologies pour identifier les types communs entre différentes



ontologies. Cette approche classique est étendue dans ce travail en identifiant les liaisons entre buts appartenant à des services distants utilisant la même structure formelle. En mapping d'ontologies, \vdash représente la subsumption de concepts, alors que dans la présente approche elle représente la dépendance fonctionnelle.

On modélise la recherche de dépendances en utilisant des classifications pour chaque service (F_0 , origine et candidat(s) potentiel(s) F_i). Pour construire le canal IF, on extrait de chaque hiérarchie potentielle le contexte physique qui correspond au contexte(s) recherché(s). On se limite au cas du contrôle distribué. L'équivalence est formalisée par une classification M reliant les types (α, β, \dots) avec toutes les instances possibles. Puis, on introduit les transposées des classifications F_0 et $F_i^{(k)}$ (service i sur système k) pour raisonner sur les contextes. L'alignement partiel de contextes est réalisé via la classification M où des relations telles que $\zeta_0^{sup}(a) = c_r$ et $\zeta_i^{(k)sup}(a) = c_q^{(k)}$ signifient que c_r et $c_q^{(k)}$ sont des types identiques dans les classifications F_0^\perp et $F_i^{(k)\perp}$. L'équation de l'infomorphisme permet de calculer les tokens de F_0^\perp et $F_i^{(k)\perp}$. On introduit les partitions conjonctives $\wedge F_0^\perp$ et $\wedge F_i^{(k)\perp}$ qui classifient les types de buts en fonction des conjonctions de contextes physiques. L'étape suivante consiste à construire la classification C du canal IF qui traduit la notion de flot d'information en fonction des identifications de contextes. C représente la colimite du processus. On en déduit la logique centrale $LogC$, puis la logique distribuée $DlogC(L)$ sur la somme $F_0 + F_i^{(k)}$ qui est l'image inverse de $LogC$. L'interopérabilité se traduit par des contraintes faisant intervenir des buts distants dans le membre de gauche des séquents.

25.4 En quoi est-ce un problème dur ?

Ce travail propose une modélisation de l'aspect intensionnel de la composition de services distribués en environnement industriel. Une première difficulté était de trouver un compromis entre efficacité et modularité pour une représentation de la connaissance pouvant être intégrée dans des agents à mémoire limitée. Les ontologies décrivant des environnements limités mais susceptibles de se composer répondent à cette contrainte. Une seconde difficulté était de trouver une théorie permettant la composition d'ontologies hétérogènes qui soit simultanément complète et valide. La solution adoptée représente les dépendances fonctionnelles à partir de la colimite du diagramme catégoriel décrivant les transferts d'information. Le formalisme catégoriel rend le modèle mathématiquement fondé et offre un support ré-utilisable et généralisable à d'autres domaines. La principale limitation vient de l'hypothèse d'alignement préalable des contextes physiques. Dans le cas d'ontologies différentes, une phase préliminaire de mapping d'ontologies structurales pourrait s'avérer fructueuse.

25.5 Perspectives futures

Un intéressant sujet d'investigation serait d'étendre la représentation des connaissances à d'autres domaines (business engineering, web, ...). Il est également souhaitable de disposer d'un outil graphique d'aide à la conception (en cours). Enfin, une évaluation et des tests du processus de composition via un SMA est actuellement en cours (les buts dont il est question ici correspondent aux hard goals, tandis que les buts au niveau agent correspondent aux soft goals).

Bibliographie

- [BPML05] Lars Braubach, Alexander Pokahr, Daniel Moldt, and Winfried Lamersdorf. Goal representation for bdi agent systems. In R.H. Bordini et al., editor, *PROMAS 2004*, number 3346 in LNAI, pages 44–65. Springer, 2005.
- [BS97] Jon Barwise and Jerry Seligman. *Information Flow*, volume 44 of *Cambridge tracts in Theoretical Computer Science*. Cambridge University Press, 1997.
- [DBBF05] Richard Dapoigny, Patrick Barlatier, Eric Benoit, and Laurent Foulloy. Formal goal generation for intelligent control systems. In *18th Int. Conf. on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems*, number 3533 in LNAI, pages 712–721. Springer, 2005.
- [DBF03] Richard Dapoigny, Eric Benoit, and Laurent Foulloy. Functional ontology for intelligent instruments. In *Foundations of Intelligent Systems*, number 2871 in LNAI, pages 88–92. Springer, 2003.
- [vL01] A. van Lamsweerde. Goal-oriented requirement engineering : a guided tour. In IEEE Computer Society Press, editor, *5th International Symposium on Requirement Engineering*, 2001.