

# Modélisation des interactions entre individus avec AgentUML

Stéphane Daviet\*,  
Fabrice Guillet\*\*, Henri Briand\*\*  
Adina Magda Florea\*\*\*, Vincent Philippé\*

\* KnoweSis

Atlanpole la Fleuriaye – BP 40703 – 44481 Carquefou Cedex  
{stephane.daviet, vincent.philippe}@kowesis.fr  
<http://www.kowesis.fr>

\*\* LINA – École Polytechnique de l'Université de Nantes  
La Chantrerie – BP 50609 – 44306 Nantes Cedex 3  
{fabrice.guillet, henri.briand}@polytech.univ-nantes.fr  
<http://www.polytech.univ-nantes.fr>

\*\*\* Université "Politehnica" de Bucarest  
Splaiul Independentei 313, Bucarest, Postal Code 060032, ROMANIA  
adina@cs.pub.ro  
<http://www.pub.ro>

**Résumé.** Pour faciliter l'étude de certains phénomènes, des outils de simulation ont été créés dans de nombreux domaines. L'étude du comportement humain a jusque là échappé à cette tendance. Aujourd'hui, les systèmes multi-agents couplés aux avancées des sciences humaines fournissent les bases nécessaires à l'élaboration de ce type d'outil. Cet article s'inscrit ainsi dans cette dynamique avec l'objectif de développer un outil de simulation du comportement d'individus traumatisés crâniens sur une chaîne de production. Cet outil doit permettre la collecte de la connaissance relative au système étudié et fournir une aide à la décision pour les responsables de l'entreprise. Cet article propose une modélisation des interactions entre individus dans le formalisme AgentUML. Une implémentation du modèle au sein d'un outil de simulation fonctionnel et les résultats obtenus seront également présentés. À terme, le but est la production de données de simulation exploitables par des techniques d'ECD.

## 1 Introduction

Certains systèmes réels, au comportement ou à la structure complexe, échappent aux outils d'étude classiques. Ces dernières années, l'informatique a néanmoins permis de développer des outils de simulation de plus en plus performants assurant notamment la traçabilité des phénomènes, la réitération des expériences, la capture de moments clés et la possibilité d'étude de cas inexistantes ou encore jamais survenus dans le système réel (Casti 1997). Les formalismes de modélisation, tels qu'AgentUML (Parunak et al. 2001, Bauer et al. 2001, Odell et al. 2000), permettent de capitaliser et structurer en modèles de connaissance les différents aspects de ces systèmes. De plus, les progrès des sciences humaines dans l'analyse et la compréhension du comportement humain (Gras et al. 2003, Bratman 1987) laissent entrevoir la possibilité de simuler certains aspects de ce

comportement, aussi bien au niveau rationnel qu'émotionnel (Florea et Kalisz 2003). À partir d'une modélisation pertinente du système, on parvient aujourd'hui à simuler son évolution de plus en plus fidèlement, et ainsi à constituer rapidement des bases de données relatives aux phénomènes étudiés.

Le travail présenté dans cet article a été mené au sein du projet "TC&Plus.Virtuel", collaboration entre la société PerformanSe et l'entreprise APARTA (Atelier Protégé d'Aide à la Réinsertion des Traumatisés crâniens Atlantique), dont l'objectif est de développer un outil informatique d'aide à la réinsertion d'individus cérébrolésés dans le milieu professionnel ordinaire. L'originalité du projet est importante à souligner puisqu'il s'agit de concevoir une plate-forme d'expérimentation destinée à produire de l'information exploitable à l'aide de techniques d'ECD. Ce travail s'inscrit donc dans une problématique de formalisation et de gestion de la connaissance nécessaire en amont de ce processus d'extraction. L'objectif est de développer une modélisation pertinente, résolument orientée multi-agents afin de garantir la maintenance et l'extensibilité du modèle et de l'application. L'objectif est également de s'appuyer sur une sémantique précise afin de permettre l'exportation et l'exploitation de la connaissance. Nous montrerons d'ailleurs l'intérêt de l'utilisation d'AgentUML à cette fin.

Cette phase de modélisation a été réalisée en étroite collaboration avec des psychologues afin de dégager les règles comportementales des individus traumatisés, ainsi que de personnes en contact avec le milieu de travail modélisé afin de valider les scénarios d'interactions que nous avons pu élaborer. Le travail présenté est complémentaire avec celui réalisé par Hélène Desmier sur la modélisation interne de l'agent et de ses émotions (Desmier et al. 2005). Nous présenterons dans une première partie les détails du système étudié. Nous détaillerons ensuite l'importance de la prise en compte des interactions pour comprendre et modéliser l'évolution d'un système, ainsi que l'intérêt d'AgentUML comme formalisme de représentation des connaissances pour la modélisation de ces interactions. Enfin, nous présenterons le travail réalisé et les résultats obtenus grâce à l'outil implémenté.

## 2 Présentation du système étudié

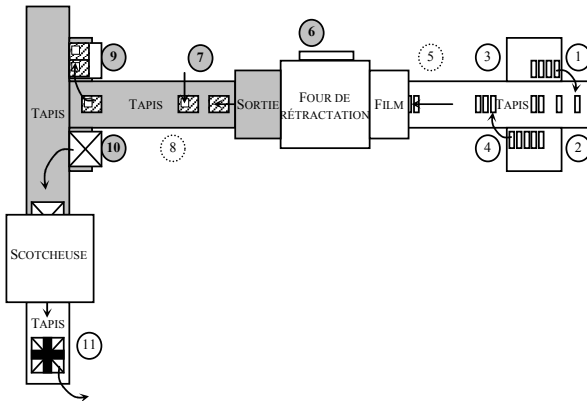


FIG. 1 - Chaîne de conditionnement.

Le système étudié est une chaîne de re-conditionnement de lots. Sur cette chaîne représentée sur la figure 1, les employés 1 et 4 déconditionnent des lots qui leur arrivent sur palette. Puis, ces lots sont déposés par les employés 2 et 3 sur un tapis roulant, acheminés jusqu'à une filmeuse et un four de rétraction du film. En sortie, une étiquette est déposée sur le lot conditionné par l'employé 7. Le lot est ensuite emballé par les employés 9 et 10 et le carton une fois plein passe dans une scotcheuse avant d'être palettisé par l'employé 11. Les employés figurés en pointillés (5 et 8) peuvent assister leurs collègues sur la chaîne en cas de besoin, leur présence n'est pas systématique. Nous avons réduit notre champ de représentation aux postes 6, 7, 9 et 10 (partie grisée sur la figure 1) afin, d'une part, de partir d'un sous-ensemble pour le complexifier ensuite comme le préconise l'approche systémique (Mélèse 1972), et d'autre part, de faciliter la validation et la correction de notre modèle.

Le système étudié est décomposé en différentes entités :

- Le poste de travail constitue une première entité, elle-même composée de sous-entités : four, tapis de sortie de four, tapis d'acheminement des cartons. Ces entités sont modélisables par des agents purement réactifs (Ferber 1996). Le comportement de la machine étant relativement simple, l'architecture d'agent réactif correspond bien au mode de fonctionnement à modéliser : la machine reçoit un stimulus (tel que l'actionnement du bouton de mise en marche) et réagit (ici, se met donc en marche). Ensuite, les lots et les cartons peuvent être considérés comme des entités inertes (ou objets) du système pouvant être manipulés par les agents du système,
- Chaque individu travaillant sur la chaîne constitue également une entité. Cette entité est modélisée par un agent cognitif émotionnel, ce type d'agent étant particulièrement adapté à la modélisation des comportements humains. Parmi ces agents, seul l'agent 6, encadrant ayant la charge de s'assurer du bon fonctionnement de la chaîne, n'est pas traumatisé crânien. Notre agent est constitué d'une mémoire (correspondant aux croyances d'un agent BDI (*Belief, Desire, Intention*) (Bratman 1987, Rao et Georgeff 1995), de caractéristiques personnelles (émotions, sentiments, tempérament, état psychologique, état physique) et d'un système délibératif basé sur une architecture BDI avec une prise en compte des caractéristiques personnelles. Dans cet article, nous utiliserons de manière indifférenciée les termes 'avatar' et 'agent émotionnel',
- Nous considérons enfin une dernière entité appelée environnement qui ne possède pas de réelle existence, mais va permettre de gérer l'ensemble des phénomènes ne dépendant pas des entités déjà citées. On peut donner l'exemple de la diffusion de la lumière ou des lois physiques qui vont réduire la zone d'audition d'un agent ou sa visibilité. L'environnement prend en fait le rôle d'administrateur du système. C'est grâce à lui que l'on va définir l'ensemble des lois, externes aux agents, qui régissent le système.

### 3 Modéliser les interactions : pourquoi et comment ?

#### 3.1 L'importance de la modélisation des interactions

Une interaction apparaît dès que deux agents ou plus sont mutuellement dépendants comme c'est le cas sur une chaîne de conditionnement. L'interaction est alors selon (Pasquier et Chaïb-Draa 2002) "*un type d'action collective où une entité effectue une action ou prend une décision, laquelle décision est influencée par une autre entité*". Ce type de situation est

courant dans le système auquel nous nous intéressons. Un individu dépend toujours des individus en amont de la chaîne de production pour l'exécution de sa tâche. Si, par exemple, un paquet n'est pas correctement étiqueté, l'emballageur devra le prendre en compte pour choisir la tâche qu'il va effectuer. L'ensemble de ces dépendances se traduit par des interactions à l'origine d'une dynamique de groupe et de l'émergence de comportements. On ne peut se limiter à l'agent en tant qu'entité autonome. *"There's no such thing as a single agent system."* (Wooldridge 2001). Dans tout système pouvant être caractérisé comme complexe, il est donc nécessaire, pour comprendre et modéliser son fonctionnement, de caractériser l'ensemble des interactions clés du système.

Cette étape de modélisation des interactions est donc fondamentale pour la réalisation d'une simulation. En effet, la plupart du temps, on ne trouve qu'une description structurelle de l'agent et de son fonctionnement interne. Cela suffit dans la plupart des cas où il s'agit de faire collaborer des agents pour l'exécution de tâches. Les agents sont donc spécifiquement conçus pour accomplir une tâche donnée et les normes de langage de communication prennent déjà en compte la plupart des interactions que l'on peut rencontrer à ce niveau. Dans notre cas, il s'agit de simuler le comportement d'un système réel. Les agents ne doivent alors plus être conçus en fonction d'une tâche à exécuter mais doivent avoir un comportement qui correspond à celui des entités que nous devons modéliser. Il est alors nécessaire de caractériser ce comportement par recueil de la connaissance d'experts du système à modéliser. Nous nous basons d'une part sur le travail déjà réalisé par la société PerformanSe sur les profils comportementaux de l'individu (Gras et al. 2003, Peter et al. 2001), ainsi que sur l'aide de psychologues et d'encadrants travaillant avec des personnes cérébrolésées pour construire notre modélisation.

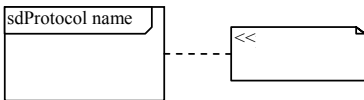
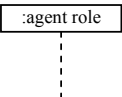
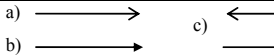
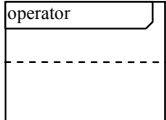
### 3.2 L'intérêt d'AgentUML

Les différences entre le concept d'objet et celui d'agent (cf. tableau 1) ont rendu nécessaires des outils de modélisation et de programmation orientés agent (Shoham 1993). C'est pour répondre à ce besoin que le langage de modélisation AgentUML (AUMML) (Parunak et al. 2001, Bauer et al. 2001, Odell et al. 2000) de la FIPA (Foundation for Intelligent Physical Agent) et les nombreuses plates-formes de développement multi-agents telles que JADE (Java Agent DEvelopment framework) (Rimassa 2003, Bellifemine et al. 2000) ont été créés.

	<i><b>Orienté objet</b></i>	<i><b>Orienté agent</b></i>
<i><b>Unité de base</b></i>	<i>objet</i>	<i>agent</i>
<i><b>Paramètres définissant l'état de l'unité de base</b></i>	<i>pas de contraintes</i>	<i>croyances, décisions, obligations, habiletés, ...</i>
<i><b>Processus de calcul</b></i>	<i>envoi de messages et méthodes pour la réponse</i>	<i>envoi de messages et méthodes pour la réponse</i>
<i><b>Types de messages</b></i>	<i>pas de contraintes</i>	<i>informer, demander, offrir, promettre, accepter, rejeter, ...</i>
<i><b>Contraintes sur les méthodes</b></i>	<i>pas de contraintes</i>	<i>consistance, vérité, ...</i>

TAB. 1 - Tableau comparatif objet-agent (Shoham 1997).

AgentUML est une initiative apparue en 2000 visant à proposer un ensemble de notations mieux adaptées à la programmation orientée agent pour qu'elles soient intégrées à la future mouture du langage UML. Les apports les plus significatifs d'AgentUML figurent au niveau des diagrammes d'interactions, famille de diagrammes regroupant le diagramme de séquence, le diagramme général des interactions (*Interaction Overview Diagram*), le diagramme de collaboration et les diagrammes temporels. Nous présentons dans le tableau 2 (Odell et al. 2000) une partie des notations proposées pour le diagramme de séquence. Elles permettent de prendre en compte la complexité des processus de communication entre agents qui impliquent de nombreuses structures conditionnelles assez lourdes à représenter avec la notation UML classique. Nous n'avons présenté dans ce tableau que les notations qui nous ont été utiles. La partie suivante fournit des exemples concrets d'utilisation de ces notations pour la modélisation de notre système.

<b>Objet</b>	<b>Notation</b>	<b>Explication</b>
<i>protocol</i>		<i>Le protocole contient la séquence de messages, le rôle de l'agent et les contraintes sur les interactions.</i>
<i>lifeline</i>		<i>La ligne de vie définit la présence des agents dans l'interaction par leur identité ou par leur rôle dans l'interaction.</i>
<i>message</i>		<i>a) message asynchrone  b) message synchrone  c) message asynchrone avec copie envoyée à l'émetteur</i>
<i>constraint</i>	a) [condition]      b) <<blocking>>[condition]	<i>a) contrainte non bloquante  b) contrainte bloquante</i>
<i>splitting / merging path</i>		<i>L'opérateur de partage peut être alternative, option, break, parallel, weak, strict, negative, critical, ignore, consider, assertion ou loop.</i>

TAB. 2 - Notations AUML pour le diagramme de séquence.

## 4 Modélisation des interactions

### 4.1 Diagrammes de cas d'utilisation

Dans un premier temps, nous allons utiliser une modélisation UML. Nous proposons d'utiliser les diagrammes de cas d'utilisation pour fournir une vue globale des différentes interactions qui peuvent exister entre les agents du système. Après recueil de l'expertise auprès des personnes en contact avec le milieu que nous devons modéliser, nous avons dégagé un certain nombre d'interactions entre individus observables au sein de l'atelier. Avec le souci de ne conserver qu'un ensemble restreint d'exemples illustratifs, nous avons choisi

un certain nombre d'interactions caractéristiques du milieu modélisé et à l'origine de la dynamique dans les rapports sociaux. Les interactions sélectionnées sont simples et à but illustratif et privilégient l'aspect réactif de l'agent à l'aspect dialogique. Le diagramme de cas d'utilisation présenté sur la figure 2 présente ces différentes interactions d'un point de vue général. Nous avons distingué les interactions qui consistent à *communiquer* par la parole avec autrui, à *apparaître* à autrui, à *agir* sur l'environnement de manière physique et à *percevoir* son environnement de manière volontaire (regarder une entité de l'environnement et observer son état apparent) et involontaire (capter le bruit émis par la machine). Ces catégories d'interaction sont mises en évidence par la notation *extends* sur le diagramme. Au niveau de chacun des postes, nous avons ensuite mené la même démarche mais cette fois-ci uniquement pour les actions de l'avatar sur la machine, les autres interactions n'étant pas spécifiques à un poste donné. La figure 3 présente le diagramme établi pour le poste 7 et correspond donc à une vue plus détaillée des interactions pour ce poste.

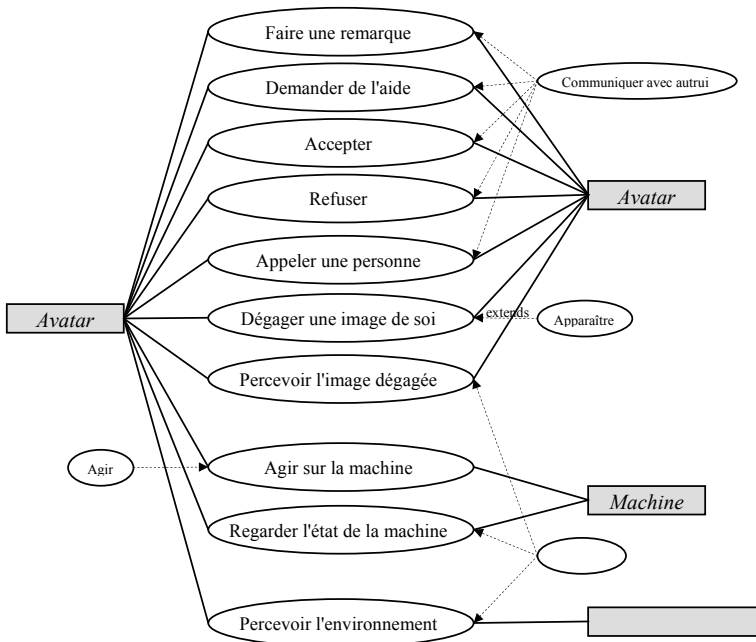


FIG. 2 - Diagramme de cas d'utilisation général.

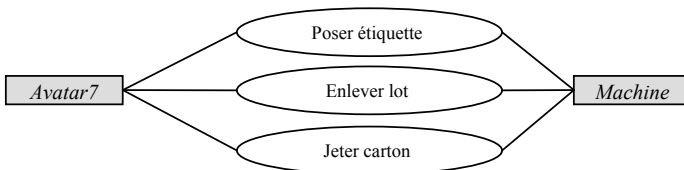


FIG. 3 - Diagramme de cas d'utilisation relatif au poste 7.

## 4.2 Modélisation des interactions par des diagrammes d'activités

Une description précise du fonctionnement de la chaîne de production est nécessaire pour identifier l'ensemble des situations auxquelles peut être confronté chaque avatar. Pour cela, nous avons d'abord établi un diagramme d'activités correspondant au fonctionnement normal de la machine (figure 4). Par normal, on entend que chaque individu travaille correctement sur la machine. Ce diagramme fournit une vue globale non détaillée du processus de travail sur la chaîne. À chaque agent, correspond un couloir d'activité dans lequel chaque tâche (ou activité) est représentée par un rectangle arrondi. Dans celui de la machine, une barre de synchronisation horizontale permet de modéliser l'itération des paquets sortant du four, cela pouvant conduire à un état de bourrage où intervient l'avatar 6. Les tâches sont liées séquentiellement par des flèches pleines ou par des flux (flèches pointillées) d'objets (rectangle) dont l'état peut être modifié par les agents. Ce premier diagramme permet de visualiser l'ensemble des dépendances pouvant exister entre les différentes entités de la chaîne de production.

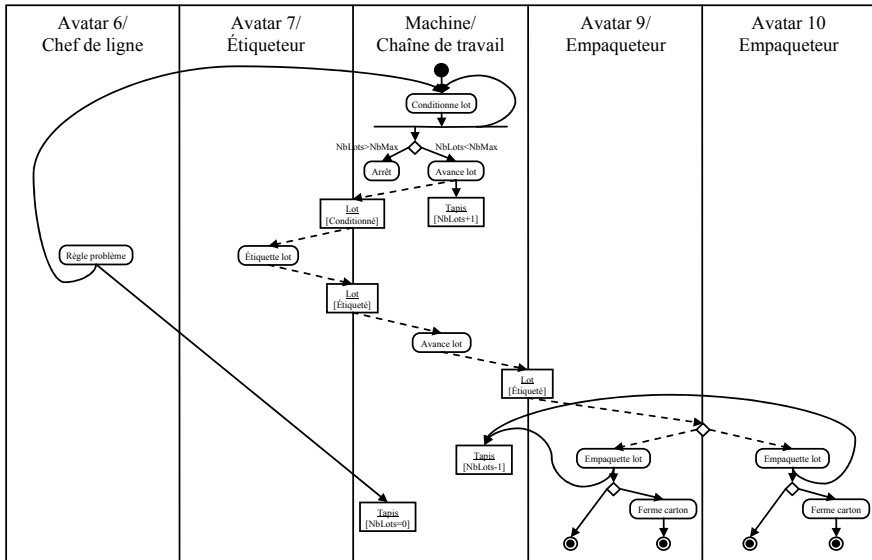


FIG. 4 - Diagramme d'activités du fonctionnement normal de la chaîne.

Le diagramme précédent est complété par des diagrammes plus précis correspondant à chaque poste et permettant de faire figurer des comportements plus particuliers. La figure 5 ne comporte que la machine, l'avatar 7 et le chef de ligne qui intervient dans la résolution des problèmes. On voit que l'agent 7 peut avoir à faire à des lots plus ou moins bien conditionnés. Devant un lot bien conditionné, il peut choisir d'accomplir sa tâche courante (étiqueter le lot) ou bien de ne rien faire. Devant un lot mal conditionné, il peut l'étiqueter quand même, enlever le lot, jeter le lot et dans tous les cas choisir ou non de prévenir le chef

de ligne. Chacun de ces choix, ou plutôt comportements, l'action accomplie n'étant pas entièrement délibérée puisqu'elle dépend en partie de composantes de l'état interne (émotionnelles, psychologiques, etc.), est conditionné par des règles comportementales établies par des psychologues.

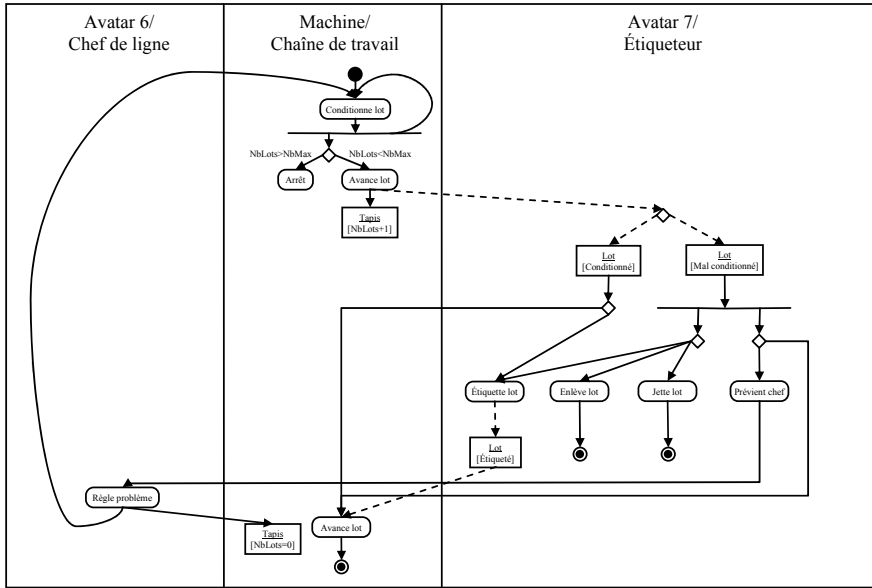


FIG. 5 - Diagramme d'activités du poste 7.

### 4.3 Modélisation des interactions par des diagrammes de séquence

Le diagramme de séquence permet de modéliser sous un autre angle le déroulement d'une tâche et d'y faire figurer plus de détails sur les conditions de déclenchement de chaque action (figure 6). Sur ce diagramme, les notations proposées par AUML nous ont été très utiles pour fournir une meilleure lisibilité. Ce diagramme correspond à l'exécution d'une tâche de travail sur le poste 7 dans le cas où l'agent n'a qu'un paquet dans sa zone de travail (situation ordinaire). Nous avons donc utilisé ici plusieurs structures conditionnelles imbriquées. Au niveau le plus haut, on distingue les cas 'bien scellé' et 'mal scellé'. Ensuite, suivant ces deux cas, on retrouve les tâches du diagramme 6, l'avantage est qu'ici on peut spécifier des conditions à l'exécution de chacune de ces tâches (si la personne est négligente et détendue, elle laissera le paquet sans rien faire). On voit également que la notation *ref* permet d'imbriquer un autre diagramme de séquence, ici le diagramme des appels (appel du chef de ligne), à celui-ci. Ce dernier diagramme est très utile car il permet de faire figurer explicitement le lien entre l'état interne de l'agent (émotionnel, psychologique, etc.), la mise à jour (*maj*) de cet état et ses interactions avec le système. De plus, il se traduit directement au niveau de l'implémentation par des structures conditionnelles correspondant au choix par



l'agent d'un comportement à adopter en fonction de son état interne. On voit donc que cette démarche descendante (diagramme de cas d'utilisation, diagramme d'activités puis diagramme de séquence) a fourni les briques élémentaires à la conception de l'outil dont nous allons parler dans la prochaine partie.

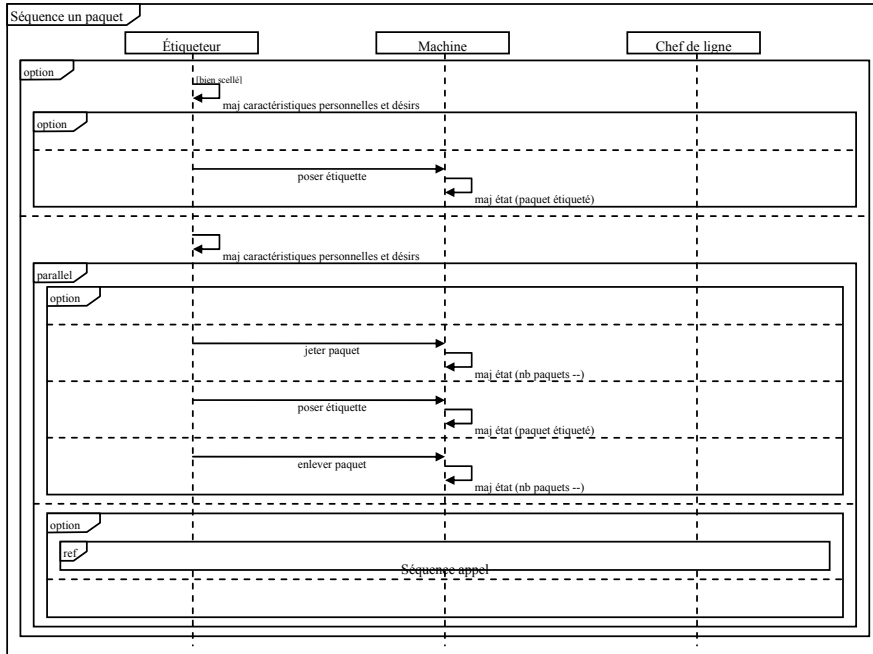


FIG. 6 - Diagramme de séquence du poste 7.

## 5 Implémentation et résultats obtenus

Un prototype démonstratif avait été développé en 2003 ce qui nous a permis d'avoir un certain recul sur nos besoins pour la réalisation de la plate-forme. Nous avons alors opté pour le langage de développement Java, mais n'avions pas utilisé de plate-forme pour nous assister dans le développement et l'exécution de notre outil. Nous avons alors été confrontés au problème de la concurrence entre agents (gestion des *threads*), c'est pourquoi nous nous sommes dirigés vers le choix d'une plate-forme de développement multi-agents mieux adaptée à nos besoins et éprouvée dans de nombreux outils. C'est aussi afin de garantir l'évolution et la diffusion de l'outil développé. Nous avons fait notre choix en nous basant sur des comparatifs entre plates-formes (Garneau et Delisle 2002). Nous nous sommes également basés sur nos essais personnels différentes plates-formes. La plate-forme JADE est un outil développé par CSELT (*Centro Studi E Laboratori Telecomunicazioni*, Laboratoire de Telecom Italia (Tilab)) en Java fournissant des bibliothèques de classes pour

le développement d'agents et permettant d'animer ces agents au sein d'une plate-forme SMA. Cet outil est largement utilisé dans la communauté multi-agents et prend en charge certaines caractéristiques essentielles aux multi-agents telles que la communication et la concurrence sans pour autant contraindre exagérément la structure de l'agent. JADE a ainsi grandement facilité la réalisation du système en fournissant des moyens simples et efficaces pour implémenter la plupart des composants de notre modèle.

La simulation mettait en scène les éléments présentés dans la première partie de l'article (cf. figure 7). Elle a apporté des résultats jugés intéressants par les experts psychologues relativement au comportement réel du groupe d'individus sur la chaîne de production. D'une part, le déroulement des actions des agents sur la chaîne de production, des points de vue de la cadence et du séquençage des actions, est correct. Chaque agent effectue en effet les tâches qui lui sont dévolues tel que l'étiquetage ou l'emballage de paquet de sorte que la chaîne de production a un comportement conforme à la réalité. Le processus industriel de reconditionnement a donc été correctement modélisé. Pour ce qui est des échanges entre agents au niveau notamment des remarques en cas de dysfonctionnement du processus de conditionnement, elles sont également échangées comme elles peuvent l'être en milieu réel. Le processus social a donc lui aussi été correctement modélisé.

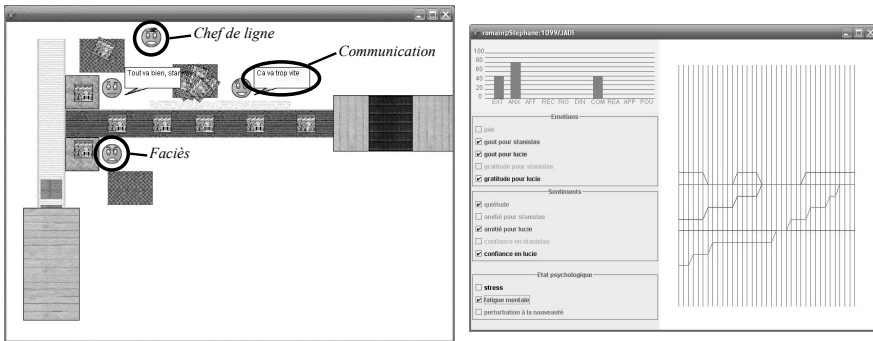


FIG. 7 - Captures d'écran de la simulation.

D'autre part, du point de vue psychologique et après concertation avec les experts, les observations recueillies avec des paramètres d'initialisation différents au niveau des individus ont montré que les conditions de déclenchement des réactions de l'agent permettent d'obtenir un comportement psychologiquement cohérent. L'exemple du poste 7, poste d'étiqueteur, dont nous avons précédemment exposé la modélisation des interactions, est un cas intéressant. En effet, il est apparu, selon les simulations, que ce poste était particulièrement stressant quel que soit le profil de la personne qui l'occupe. Le personnel encadrant a confirmé cette tendance et des investissements ont d'ailleurs été effectués afin d'y remédier. Enfin, des comportements typiques des traumatisés crâniens sont également apparus. Citons encore l'exemple du poste 7 où un agent de type paranoïaque confronté à un trop grand nombre d'aléas techniques se met à jeter les lots sortant du four au sol sans informer son supérieur du problème, alors qu'un agent de type consciencieux va les mettre de côté et informer le chef de ligne du dysfonctionnement.

## 6 Conclusion

Optimiser les conditions de travail afin d'augmenter le confort des employés et donc obtenir de meilleurs résultats tant en terme d'ambiance qu'en terme de production, détecter les facteurs aggravants pouvant conduire à une situation de crise et détecter des phénomènes qui échappent en temps normal à l'observateur sont autant de raisons qui justifient le développement d'un outil de simulation des comportements d'individus dans leur milieu de travail. L'étape de modélisation des interactions du système multi-agents est fondamentale pour la réalisation de ce type d'outil. Elle permet d'obtenir un modèle comportemental précis et maintenable de l'agent. Elle assure l'évolutivité de ce modèle et une implémentation cohérente de ces comportements au sein de l'agent. Les modèles obtenus permettent d'obtenir rapidement une plate-forme fonctionnelle en assurant des fondements solides à son développement. Ils fournissent surtout un support lisible et compréhensible pour le transfert de la connaissance des experts en psychologie à intégrer dans la simulation.

Dans cet article, nous avons décrit la façon dont nous avons caractérisé les entités d'un système et les interactions qu'elles pouvaient entretenir, afin de construire un outil de simulation. Nous avons d'abord insisté sur l'importance de la prise en compte des interactions qui constituent la base de l'évolution et de la richesse du système. Nous avons également montré l'intérêt de disposer d'outils de modélisation adaptés au domaine des multi-agents tels qu'AgentUML, ainsi que d'une plate-forme de développement et de déploiement des agents telle que JADE. La prise en compte, en collaboration étroite avec des psychologues, de caractéristiques humaines telles que les émotions, les sentiments et la personnalité dans les processus d'interactions entre agents était également au centre de ce travail.

Nous disposons à présent d'un modèle fiable pouvant être enrichi, ainsi que d'une première version de notre outil de simulation nous permettant de valider des résultats obtenus et la viabilité de notre démarche. La suite de ce travail consistera à développer et mettre en œuvre des outils d'extraction de connaissances adaptés à la spécificité de nos données de simulation afin de caractériser et prévoir les comportements collectifs émergents.

## Références

- Bauer B., Müller J. P. et Odell J. J. (2001), Agent UML : A Formalism for Specifying Multiagent Interaction, *Proceedings of the Agent-Oriented Software Engineering Workshop*, pp 91-103, Paolo Ciancarini and Michael Wooldridge Eds.
- Bellifemine F., Poggi A. et Rimassa G. (2000), Developing multi-agent systems with JADE, *Proceedings of the Seventh International Workshop on Agent Theories, Architectures, and Languages ATAL'00*, pp 89-103, Springer-Verlag.
- Bratman M. E. (1987), Intentions, plans and practical reason. Harvard University Press.
- Casti J. L. (1997), Would-be worlds : How simulation is changing the frontiers of science. John Wiley & Sons Eds.
- Desmier H., Guillet F., Briand H., Florea A. M. et Philippé V. (2005), Modélisation d'un agent émotionnel en UML et RDF, *Actes de la conférence Extraction et Gestion des Connaissances EGC'05*, (à paraître), CEPADUES.
- Ferber J. (1996), Reactive distributed artificial intelligence, *Foundations of Distributed Artificial Intelligence*, pp 287-317, G. M. P. O'Hare et N. R. Jennings Eds, John Wiley.

Florea A. M., Kalisz E. (2003), Behavior Anticipation Based on Beliefs, Desires and Emotions, *Proceedings of the 6th international conference on Computing Anticipatory Systems (CASYS 03)*. Springer-Verlag.

Garneau T. et Delisle S. (2002), Programmation orientée-agent : évaluation comparative d'outils et environnements, *Actes des 10ème Journées Françaises d'Intelligence Artificielle Distribuée et de Systèmes Multi-Agents JFIADSMA'02*, pp 111-123, Hermès Science.

Gras R., Peter P., Baquédano S. et Philippé J. (2003), Structuration de comportements de réponse à un questionnaire par des méthodes multidimensionnelles, *Numéro spécial Extraction et Gestion des Connaissances, Extraction des Connaissances et Apprentissage*, 17(1-3):105-118, Hermès Science.

Mélèse J. (1972), L'analyse modulaire des systèmes de gestion, A.M.S., Hommes et techniques Éd.

Odell J.J., Parunak H. Van Dyke et Bauer B. (2000), Extending UML for Agents. *Proceedings of the Agent-Oriented Information Systems, Workshop at American Association for Artificial Intelligence AAAI'00*, pp 3-17.

Parunak H. Van Dyke, Odell J. J. et Bauer B. (2001), Representing Agent Interaction Protocols in UML, *Proceedings of the Agent-Oriented Software Engineering Workshop*, pp 121-140, Paolo Ciancarini and Michael Wooldridge Eds.

Pasquier P. et Chaïb-Draa B. (2002), Cohérence et conversation entre agents : vers un modèle basé sur la consonance cognitive, *Actes des 10ème Journées Françaises d'Intelligence Artificielle Distribuée et de Systèmes Multi-Agents JFIADSMA'02*, pp 189-203, Hermès Science.

Peter P., Gras R., Philippé J. et Baquédano S. (2001), L'analyse implicative pour l'étude d'un questionnaire de personnalité, *Numéro spécial Extraction et Gestion des Connaissances, Extraction des Connaissances et Apprentissage*, 1(1-2):181-187, Hermès Science.

Rao A. S. et Georgeff M. P. (1995), BDI Agents : from theory to practice. *Proceedings of the First International Conference on Multiagent Systems ICMAS'95*.

Rimassa G. (2003), Runtime Support for Distributed Multi-Agent Systems, *Rapport de Thèse*, Université de Parme.

Shoham Y. (1993), Agent-oriented programming, *Artificial Intelligence*, 60(1), pp 51-92. Elsevier Science Publishers Ltd.

Shoham Y. (1997), An Overview of Agent-Oriented Programming, *Software agents*, pp 271-290, MIT Press.

Wooldridge M. (2001), An Introduction to MultiAgent Systems, John Wiley & Sons Eds.

## Summary

Simulation tools to allow the study of complex phenomena have been created in various fields. However, simulation of all aspects of human behaviour is still a challenge for both computer and human sciences. In this context, the paper presents our approach of simulating brain damaged people's behaviour on a production chain by mixing techniques of artificial intelligence, multi-agent systems and psychological sciences. The focus of the paper is on the dynamics of human interactions in such an environment. We propose a model and its associated description using AgentUML, describe the first version of the simulation tool that we have developed, and report on the obtained results. The final goal of our work is to produce simulation data valuable to a KDD process.