

**Enrichissement de la modélisation des processus métiers
par le paradigme des systèmes multi agents**

Denis Berthier, Chantal Morley et Michel Maurice-Demourieux

INT/GET (Groupe des Écoles des Télécommunications)
9 rue Charles Fourier, 91011 Evry Cedex, France

Denis BERTHIER : X68, Professeur à l'INT.

Recherche en logique mathématique puis en intelligence artificielle et en épistémologie des STIC. Auteur de « Le savoir et l'ordinateur » et « Méditations sur le réel et le virtuel ».

tél. : 01 60 76 41 22, mail : denis.berthier@int-evry.fr

Chantal MORLEY : Docteur HEC, Maître de conférences habilitée au Département Systèmes d'Information de l'INT.

Axes de recherche : capitalisation des connaissances en management de projet, modélisation des processus et relation entre genre et technologies de l'information. Co-auteur de « Processus métiers et systèmes d'information » (Dunod 2005).

tél. : 01 46 21 36 98 (rép.-fax), mail : chantal.morley@int-evry.fr,

Michel MAURICE-DEMOURIOUX : Ingénieur d'études au Département Systèmes d'Information de l'INT.

Recherche : raisonnement à partir de cas (CBR) et méthodologies de conception de systèmes d'information.

tél. : 01 60 76 47 35, mail : michel.maurice-demourieux@int-evry.fr,

Enrichissement de la modélisation des processus métiers par le paradigme des systèmes multi agents

Mots-clés : système d'information, modélisation, processus métier, agent

Résumé

Le paradigme agent peut être une source d'inspiration pour enrichir la modélisation des processus métiers. À partir d'un métamodèle synthétisant les représentations classiques, nous proposons une évolution qui permet la réutilisation d'activités, une structuration plus souple, et l'intégration d'activités collaboratives ainsi que d'acteurs dotés d'autonomie.

Keywords : information system, modeling, business process, agent

Abstract

The agent paradigm can be a source of inspiration to improve business process modeling. Starting from a framework including most of the current representations, we propose an evolution that allows the reuse of activities and a more flexible structuring, and that integrates collaborative activities and autonomous actors.

INTRODUCTION

La vision processus, progressivement construite par différentes théories des organisations depuis un demi-siècle, joue un rôle croissant dans le domaine des systèmes d'information. Aujourd'hui, la modélisation des processus métiers est souvent considérée comme un préalable indispensable à la conception de la dynamique d'un système d'information (Butler, 1999 ; Alter, 1999). Cependant, les formalismes de modélisation utilisés sont généralement centrés sur une description détaillée des tâches que les acteurs devront effectuer. Une telle approche, non seulement limite la réutilisation des modèles, mais peut aussi occulter la part d'initiative que l'on entend laisser à certains acteurs, ainsi qu'une éventuelle dimension d'échange et de coopération.

Dans le domaine de l'informatique, les questions de réutilisabilité, d'autonomie et de collaboration entre composants logiciels ont fait l'objet de recherches importantes, qui ont conduit notamment à des développements technologiques regroupés autour du concept d'agent. Le but de cet article est d'utiliser ce que l'on appelle le paradigme agent, c'est-à-dire la logique générale et les principes conceptuels de ces technologies, pour enrichir la modélisation des processus métiers.

Dans une première partie, nous proposons un cadre unifié pour la modélisation classique des processus métiers, sous forme d'un métamodèle dont les concepts respectent les définitions normalisées d'un processus et peuvent être mis en correspondance avec les principaux langages de modélisation actuels, et nous indiquons ses limites de représentation. Dans une deuxième partie, nous décrivons les caractéristiques clés du paradigme des systèmes multi agents. Dans une troisième partie, nous nous inspirons de ce paradigme pour introduire dans notre métamodèle initial les concepts d'*agent*, de *but*, de *service* et d'*interaction*, et nous montrons comment ils peuvent enrichir l'expression des modèles de processus et ouvrir la voie à une modélisation par composants dynamiques.

1. LA MODELISATION DES PROCESSUS METIERS

1.1 Le processus dans le management des systèmes d'information

Commençons par une précision de vocabulaire, pour clarifier la distinction que nous faisons entre système d'information et système informatique. Celle-ci correspond à une répartition des responsabilités au sein de l'entreprise entre gestionnaires et informaticiens.

Le système d'information d'une entreprise est la *partie du réel constituée d'informations organisées, d'événements ayant un effet sur ces informations, et d'acteurs qui agissent sur ces informations ou à partir de ces informations, selon des processus visant une finalité de gestion et utilisant les technologies de l'information.*

Le système informatique est *un ensemble organisé d'objets techniques — matériels, logiciels, applicatifs — dont la mise en œuvre réalise l'infrastructure du système d'information et lui permet de fonctionner.*

La notion de processus joue un rôle majeur dans la maîtrise de l'évolution des systèmes d'information et des systèmes informatiques associés. En effet, à cause de la diversité et de la

plasticité des technologies de l'information, l'efficacité des systèmes installés dépend de la cohérence entre la stratégie, l'organisation et les systèmes d'information (Reix, 2002 ; Georgel, 2005). Dans cette perspective d'*alignement stratégique*, la mission première du système d'information est d'aider l'entreprise à atteindre ses objectifs, sans être dominée par des contraintes techniques. Il en découle deux exigences, l'une pour les informaticiens, l'autre pour les gestionnaires. Pour que le système informatique puisse évoluer de façon réactive, efficiente et efficace, les informaticiens doivent le construire de façon modulaire, comme un assemblage de composants faiblement couplés, permettant de modifier chaque composant de façon quasi indépendante. Par ailleurs, les gestionnaires sont appelés à devenir acteurs de leur système d'information et des services qu'il apporte, et à le structurer de façon à ce qu'il soit flexible. Cela a conduit à définir une notion d'architecture de système d'information avec trois vues (Jean, 2000; Longépé, 2004).

1. La vue métier, qui pilote les deux autres, est celle des activités de l'entreprise, c'est-à-dire ses processus. Une telle représentation est appelée « architecture métier » ou une cartographie des processus.
2. La seconde vue, souvent appelée « architecture fonctionnelle », fait apparaître les fonctions et les informations du système d'information.
3. La troisième vue appelée « architecture applicative ». est celle du système informatique, c'est-à-dire des composants logiciels, applications et des infrastructures techniques.

L'un des enjeux actuels de la maîtrise des systèmes d'information est de construire une architecture métier et une architecture fonctionnelle qui soient suffisamment souples pour permettre de fréquentes évolutions. Une des pistes peut être la construction des processus métiers que nous proposons dans cet article, après avoir fait un état des pratiques de modélisation existantes.

1.2 Les concepts de la modélisation des processus métiers

Un langage de modélisation est un système de signes et de règles permettant d'exprimer une pensée et de communiquer. Chaque signe comporte un symbole et une signification, en soi et par rapport aux autres signes. Lorsque la sémantique attachée aux signes du langage n'établit pas de correspondance exclusive avec une réalité extérieure, on peut utiliser le langage dans différents contextes.

Les éléments composant un langage de modélisation peuvent être eux-mêmes représentés par un modèle (en général un diagramme de classes ou équivalent), appelé *métamodèle*. Il existe différents métamodèles. Certains sont liés à un outil (Scheer, 2000), d'autres visent souvent un aspect particulier, par exemple les processus automatisés (WfWC, 1999), les coûts des activités du processus (Succi, 2000) ou les interactions entre acteurs (Kawalek, 2000 ; Sutcliffe, 2000 ; Te'eni, 2000). Les exigences d'interopérabilité entre les outils de modélisation de processus conduisent depuis quelques années à la construction d'un langage unifié de modélisation de processus (p.ex. Schlenoff, 2000) qui soit un noyau minimum. En revanche, il y a peu de métamodèles visant une synthèse.

En examinant les métamodèles associées aux principaux langages de modélisation de processus métier, on distingue trois perspectives : la perspective orientée activité apporte les

concepts d'activité, de lien, d'aiguillage, de ressource, d'entrée et de sortie ; la perspective orientée flux propose les concepts d'acteur et de flux ; la perspective orientée état s'appuie sur les concepts d'état, de transition, de condition, d'événement et de synchronisation. Pour établir des correspondances entre ces formalismes et leurs vocabulaires, nous avons proposé dans une recherche antérieure un métamodèle de synthèse (Morley, 2004 ; Morley, 2005). Sans avoir l'ambition d'épuiser tous les concepts mis en œuvre dans les différents langages de modélisation, il offre un ensemble suffisant pour modéliser la plupart des configurations. Nous en représentons ici la partie centrale (Fig.1) articulée autour des concepts d'Activité, Acteur et Tâche.

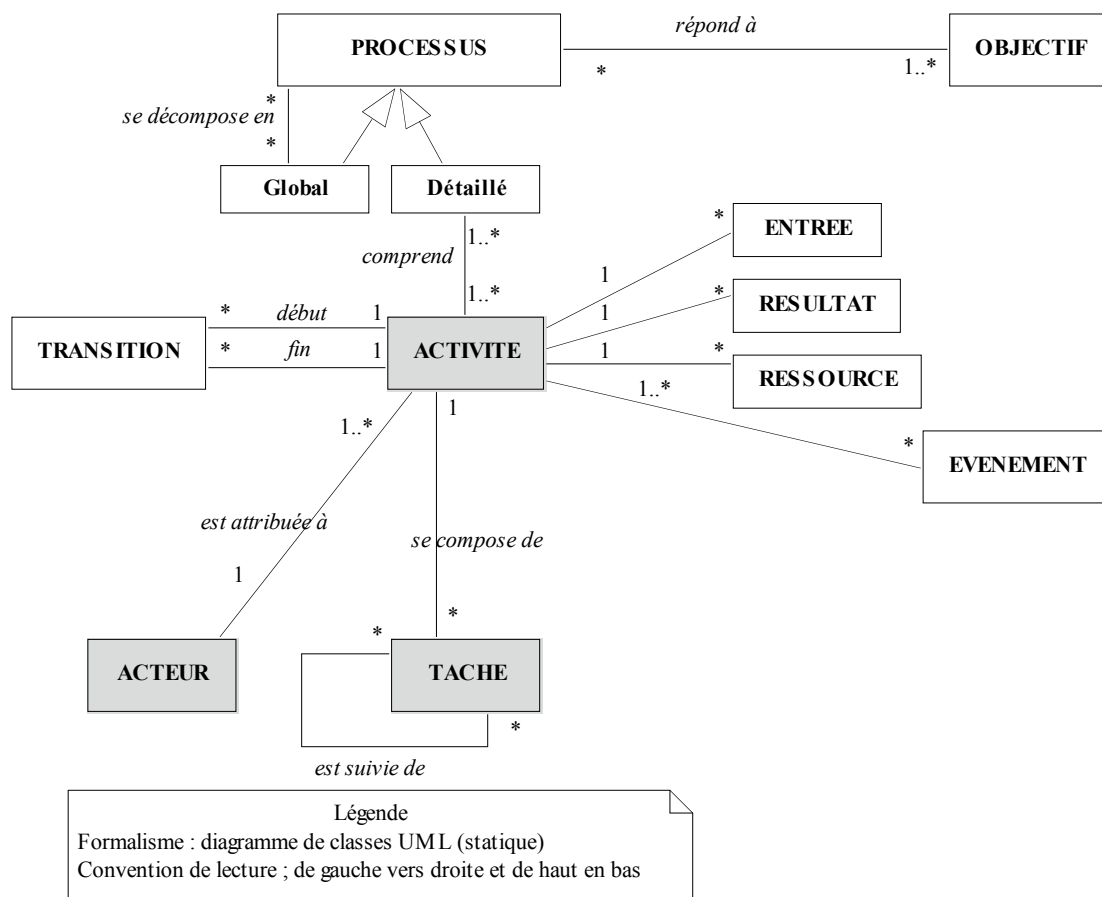


Figure 1 : Métamodèle de synthèse pour la représentation des processus

Ce métamodèle repose sur deux options. D'une part, il accompagne une approche « top down », dans laquelle un processus est toujours finalisé (il correspond à un objectif de gestion) et peut être décrit à différents niveaux de granularité (notamment dans une cartographie), le dernier niveau étant seul décrit de façon détaillée. D'autre part, il place le concept d'activité au coeur de la description, suivant en cela la définition de référence donnée par (ISO9000, 2000)¹ et celle la norme ENV12204 (CEN/CENELEC, 1995), consacrée à la

¹ « Un processus est un ensemble d'activités corrélées ou interactives qui transforme des éléments d'entrée en éléments de sortie ».

modélisation d'entreprise². Dans cette approche, la définition des activités précède l'organisation qui peut être mise en place par leur répartition entre les acteurs.

Un processus métier est ainsi appréhendé comme un ensemble d'activités, pouvant être considérées comme une succession de transformations. Son déroulement peut être indiqué par des transitions, éventuellement caractérisées par des événements. L'agencement des activités correspond à la structure du processus.

1.3 Les différentes structurations des processus métiers

Si l'on s'interroge sur les raisons qui poussent à structurer un processus, on doit distinguer deux grandes catégories de processus : répétitif ou unique. Un processus *répétitif* est amené à être exécuté à de multiples reprises et sa description sous forme de modèle présente un caractère normatif : les activités sont supposées être effectuées conformément à leur description en tâches et à leur ordonnancement. La description vise à s'imposer, que les acteurs soient des êtres humains (individuels ou collectifs) ou que ce soit un logiciel dont elle représente le cahier des charges. Un processus *unique*, dont l'exemple principal est le projet, n'est exécuté qu'une seule fois. Dans ce dernier cas, pourquoi le modéliser ? Ce peut être soit pour des raisons de planification (identification des activités et répartition entre un groupe d'acteurs), soit parce qu'on cherche à représenter une structure générique pouvant servir de support à un outil de collaboration autour d'un processus.

Par ailleurs, on distingue aujourd'hui trois grandes approches de structuration d'un processus (Vidal, 2002 ; Melao, 2000). Certains processus peuvent être définis par une structure qui rend complètement compte de l'ordre des activités, alors que pour d'autres, il est difficile ou peu efficace d'imposer tous les liens entre les activités.

La première approche est parfois qualifiée de « mécaniste » : le rôle du processus est de définir précisément l'ordre et le contenu des activités à effectuer, pour accroître l'efficacité (réduction des moyens) et l'efficacité (meilleure atteinte du but) du travail. La plupart des techniques de modélisation s'inscrivent dans cette approche. Les liens entre activités sont représentés par des transitions, qui marquent des jalons dans la transformation que représente le processus, le résultat d'une activité représentant une entrée pour l'activité suivante.

Dans la deuxième approche, souvent appelée « systémique », on considère que les activités sont des composants réagissant à des événements. Les liens entre activités s'effectuent par les résultats : le résultat d'une activité représente un événement déclencheur pour autre activité. Le déroulement réel d'une instance de processus correspondra à l'un des chemins prévus.

Dans la troisième approche, qualifiée d'« émergente » ou de « construit social », on ne souhaite pas établir de chemin, même multiple, entre les activités. Ce n'est qu'*a posteriori* que l'on peut éventuellement retracer la séquence des activités. Chaque activité est assortie d'événements pouvant la déclencher, l'interrompre ou modifier son cours. Un événement est soit d'origine externe, soit temporel, soit le résultat de la sollicitation d'un autre acteur. Ce type de représentation correspond à un processus dont le déroulement n'est pas déterminé *a*

² «Un processus est un ensemble partiellement ordonné d'activités de l'entreprise exécutées pour réaliser un objectif de l'entreprise».

priori, par exemple un processus unique dans lequel les acteurs possèdent une latitude dans la façon dont ils vont accomplir une activité.

1.4 Les limites du métamodèle

On peut souligner deux principales limites au métamodèle présenté ci-dessus.

La première tient à la façon sous-jacente de concevoir un processus. En effet, malgré la transposition des principes de l'approche objet à la modélisation des systèmes d'information, il reste difficile de concevoir des composants opératoires suffisamment découplés pour pouvoir être utilisés par plusieurs processus. La décomposition d'un processus en activités situées dans un contexte organisationnel réduit fortement leur réutilisabilité. D'ailleurs, les méthodes de modélisation de processus (IDEF, UML, Ossad, Merise, réseaux de Pétri...) n'évoquent pas le recours à des « briques » existantes.

La seconde limite concerne les possibilités de structuration. En effet, le métamodèle est bien adapté à la représentation de processus à structure mécaniste. Il l'est également pour ceux à structure systémique, si leur degré de complexité (c'est-à-dire le nombre de liens entre activités) est limité. En revanche, il ne l'est guère pour des processus à structuration émergente, dans lesquels les acteurs — internes ou externes, uniques ou groupe —disposent d'autonomie pour accomplir des activités.

De façon générale, la représentation d'un processus dans lequel les décisions et les communications entre les acteurs jouent un rôle clé conduit à des modèles lourds à manipuler ou difficiles à représenter. Chaque exécution du processus dépendra en particulier des décisions de chaque acteur (par exemple solliciter un autre acteur ou répondre à une sollicitation), de la disponibilité de certaines ressources (qui peuvent diminuer le recours à d'autres acteurs) et de la participation effective de chaque acteur.

Ces deux limitations, qui peuvent être mises en relation avec les questions de communication entre agents logiciels dotées d'autonomie et de réutilisation de composants logiciels, nous conduisent à nous inspirer des concepts du paradigme agent pour enrichir le métamodèle des processus métiers.

2. LE PARADIGME AGENT

Dans le monde du logiciel, l'objectif majeur de réutilisation a été partiellement réalisé par l'approche objet. Cependant, l'objectif initial de réutilisation a été reformulé en termes plus ambitieux d'interopérabilité, auxquels entend aujourd'hui répondre le paradigme des systèmes multi agents.

2.1 De la problématique de la réutilisation à celle de l'interopérabilité

Le problème de la réutilisabilité consiste à concevoir des « briques » logicielles pouvant être réutilisées dans diverses applications. Les langages typés puis orientés objet, dans lesquels des éléments génériques peuvent être progressivement spécialisés, ont été développés essentiellement en réponse à ce souci. Le succès incontestable de l'approche objet rencontre toutefois une limite : des classes d'objets conçues indépendamment les unes des autres ou écrites dans des langages informatiques différents ne peuvent pas être directement intégrées

dans un même logiciel. D'où les deux idées étroitement liées : celle d'une approche par « composants » réutilisables en un sens plus général que ne le sont les objets et celle d'interopérabilité.

L'interopérabilité consiste à faire « travailler » ensemble, sans les réécrire, des logiciels conçus séparément, dans des langages différents, tournant sur des machines de constructeurs différents, avec des systèmes d'exploitation différents, et généralement distantes. On peut la considérer comme une version très exigeante de réutilisation après coup de logiciels existants.

Il serait illusoire de viser à une architecture unique des systèmes informatiques qui leur permettrait de communiquer simplement entre eux, ou à un langage universel qui les autoriserait à travailler directement sur des objets et données communs. Une telle uniformité serait synonyme d'une part du blocage de tout progrès et d'autre part de l'impossibilité d'optimiser des langages ou systèmes à des fins particulières. Elle imposerait en outre la réécriture, économiquement impensable, de tous les logiciels existants. Il en est de même pour les systèmes de communication et d'échange d'information. L'hétérogénéité des systèmes informatiques et de communication s'impose donc comme une réalité incontournable. Bien entendu, l'interopérabilité ne saurait se présenter comme une alternative à la normalisation, toujours indispensable pour éviter une multiplication désordonnée et inutile de systèmes incompatibles ; elle en est au contraire le complément, les systèmes qu'on veut voir interopérer étant en principe déjà normalisés. L'interopérabilité des systèmes informatiques est donc fondamentalement ce qui pallie leur impossible et indésirable uniformité, mais qui concerne des systèmes déjà normalisés. Elle est ce qui permet de réaliser un compromis entre d'un côté hétérogénéité et évolutivité et de l'autre uniformité et normalisation (Berthier, 2002).

Classiquement, l'interopérabilité est attendue tant du côté des données que du côté des programmes.

Du côté des bases de données, elle est recherchée, de manière classique, selon deux voies complémentaires : l'une syntaxique générale, avec la normalisation des langages de requête (comme SQL), l'autre sémantique, au cas par cas, avec l'intégration des schémas conceptuels des bases concernées ; mais, avec les avancées du Web, le problème de l'interopérabilité des données déborde désormais le cadre des bases de données et s'étend, tout particulièrement, aux données structurées en XML.

Du côté des programmes orientés objets, la question devient celle de systèmes (distribués) hétérogènes et les réponses se nomment CORBA, COM+ ou Java RMI ; leur principe commun repose sur : 1°) un langage normalisé (IDL) permettant de spécifier les comportements élémentaires de chaque classe d'objets (ses « méthodes »), de manière abstraite, indépendante de toute implémentation ; 2°) la mise en place d'un « middleware », c'est-à-dire d'un ensemble de services système sur des ordinateurs qu'on regroupe en une « plateforme » ; celle-ci a pour tâche de mettre en œuvre automatiquement ces descriptions abstraites pour localiser et invoquer les objets informatiques ainsi décrits, quel que soit l'endroit précis où ils résident sur les différentes machines qu'elle rassemble.

Plus récemment sont apparus les services Web, résultat de la transposition des techniques d'interopérabilité à *la* Corba vers les technologies fondamentales du Web. Leur but est de permettre à une application de trouver automatiquement sur Internet le service dont elle a besoin et d'échanger des données avec lui (voir par exemple Chauvet, 2002). Ces services Web répondent, à un premier niveau, à un besoin de plus en plus fort de communication et d'interopérabilité entre d'une part les données et programmes du système informatique classique (essentiellement centré sur des bases de données) et d'autre part les nouvelles sources d'information et d'accès à celles-ci que permet le Web (qu'il soit intranet, extranet ou internet) ; mais, de manière plus ambitieuse, ils visent aussi à résoudre le problème de l'EAI (Enterprise Application Integration) en s'appuyant sur les technologies du Web.

2.2 Les agents

Alors que les services Web apparaissent comme étant essentiellement d'ordre technologique et ont vocation à fournir des services relativement simples et unitaires, les agents peuvent être appréhendés comme un véritable paradigme de conception et de programmation (Maes, 1991 ; Shoham, 1993 ; Bradshaw, 1997 ; fipa.org). En particulier, ils visent à prendre en compte des capacités complexes ou demandant des interactions complexes entre agents. On distingue deux aspects des agents, l'un « communicationnel », hérité du génie logiciel, l'autre « mentaliste » ou « cognitif », hérité de l'intelligence artificielle (IA).

Relativement à l'aspect « communicationnel », la technologie des systèmes multi agents (SMA), telle qu'elle est normalisée par le FIPA (Foundation for Intelligent Physical Agents, qui joue un rôle homologue à celui que joue l'OMG pour les objets), s'inscrit dans la problématique de l'interopérabilité et vise à pousser un cran plus loin les facultés définies ci-dessus. Techniquement, les agents sont des types d'objets particuliers, définis par leur capacité à « communiquer » entre eux par des messages au format standardisé ; chaque message est une instance d'une primitive formelle de communication de haut niveau conceptuel (faisant en particulier abstraction de toute technique et de tout protocole au niveau des réseaux de communication) ; il y a un nombre fini de telles primitives et l'ensemble de ces primitives constitue un langage universel (par exemple le FIPA-ACL, si l'on adopte la référence FIPA). Cela distingue les agents des objets, pour lesquels les « méthodes » (même si elles sont parfois dénommées abusivement des « messages ») ne sont en fait que des appels de procédures spécifiques à chaque type d'objet. Les agents permettent la réutilisation des logiciels existants, par des techniques d'« enrobage » (wrapping) : chaque source d'information (base de données, pages XML, etc.), chaque objet, chaque programme, etc., est enrobé dans une interface qui le fait apparaître de l'extérieur comme s'il était un agent. Par ailleurs, le paradigme des SMA permet de concevoir des architectures de systèmes *a priori* beaucoup plus variées, plus souples et plus robustes que le classique client-serveur.

Relativement à l'aspect « mentaliste » ou « cognitif » ou « intentionnel », les agents satisfont au minimum à la définition de (Newell, 1982) : un agent est régi par des « buts », il dispose de « connaissances », qu'il utilise rationnellement pour atteindre ses buts. Dans ce domaine, la plus grande prudence, pas toujours respectée par l'IA, s'impose en matière de vocabulaire : les « connaissances » dont il s'agit ici sont des connaissances symboliques formelles, hybrides mi-humaines mi-informatiques, compréhensibles par un humain ne

sachant rien des subtilités informatiques d'une machine capable de les mettre en œuvre, et exploitables par un ordinateur ne sachant rien des subtilités psychiques d'un humain les comprenant (Berthier, 2002). En pratique, ces connaissances se présentent souvent sous forme de règles logiques, exprimant, dans le vocabulaire du domaine et de l'application, l'expertise nécessaire pour traiter les problèmes que l'agent est censé résoudre, et le terme rationnel signifie que le système est capable de faire de l'inférence formelle pour les exploiter. D'autres composants « mentalistes » peuvent aussi intervenir : des « intentions », des « croyances », des « engagements », etc., qui permettent d'exprimer, par exemple, des règles générales régissant les interactions entre agents.

Ces deux aspects des agents, quoique conceptuellement et techniquement bien distincts, sont en fait fortement liés par le niveau conceptuel auquel ils font sens, comme cela se voit dans les messages, d'où sont exclus tous les aspects techniques de l'informatique ou des réseaux : 1°) chaque message exprime une « intention de communication », au sens de la théorie des actes de parole d'Austin et Searle (Austin, 1956 ; Searle, 1969), telle qu'elle a été ultérieurement formalisée (Searle & Vanderveken, 1985 ; Vanderveken, 1990) ; 2°) chaque message se réfère à une ontologie (applicative) particulière et à un langage de contenu précis (dans lequel l'ontologie et le contenu sont exprimés) ; et 3°) le contenu de chaque message, qui en est la partie la plus spécifique, se réfère aux « connaissances » de l'agent, exprimées dans les termes mêmes de l'application.

Nous insistons sur ces deux aspects et sur leur lien étroit, car nous allons les retrouver ensemble dans la modélisation de processus métiers. Conjointement, ils représentent une considérable montée en abstraction du point de vue technique de l'informatique, mais une non moins considérable avancée vers le concret du point de vue de l'application.

De même que l'approche objet a permis d'enrichir la représentation conceptuelle des systèmes d'information, notamment la partie statique avec l'introduction de la relation de généralisation/spécialisation, nous proposons d'utiliser l'approche agent pour apporter une réponse aux limites de la représentation dynamique, plus précisément à la modélisation des processus métiers.

2.3 Protocoles d'interaction et « conventions sociales » entre agents

Un autre aspect important distingue les agents des services Web, tout en leur conférant une autre dimension conceptuelle : leur organisation « sociale ». En effet, si, à la base, les interactions entre agents d'un système reposent sur l'échange de messages, il est généralement nécessaire de définir pour ceux-ci des niveaux d'organisation supérieurs.

Au premier niveau, une « conversation type » définit le graphe des chemins possibles (i.e. les suites de types de messages) dans la conversation.

A un deuxième niveau, un « protocole d'interaction » est un ensemble de règles qui régissent la communication d'informations entre agents et le suivi de l'avancement d'un plan d'action commun. Il décrit, entre autres :

- 1°) quel type d'information chaque agent peut échanger avec quels agents, et dans quelles conditions, par exemple à son initiative ou à celle d'un tiers ;
- 2°) quel type de tâches il peut déléguer, à qui, et dans quelles conditions ;
- 3°) quel type de compte rendu d'avancement il doit faire, à qui, à quel moment.

Un tel protocole d'interaction est suffisant dans un système basé sur une organisation strictement hiérarchique, où la planification et la répartition des tâches se font « par le haut ». Mais les interactions sociales complexes entre humains (coopération, négociation, travail d'équipe laissant à chacun une certaine autonomie) sont plus difficiles à modéliser et à introduire dans un système d'agents artificiels. De telles interactions sont néanmoins nécessaires pour assurer au système une certaine souplesse et éviter les blocages inhérents à toute organisation hiérarchique rigide.

Une « convention sociale » de coopération est à la fois plus souple et plus précise qu'un simple protocole d'interaction (plus précise car un protocole d'interaction présuppose une certaine organisation sociale implicite assez rigide, qu'il n'a pas à détailler). Elle définit la manière dont les engagements des divers agents envers des plans et méthodes communs (et leurs conséquences individuelles) peuvent être pris, réévalués ou remis en cause. Elle présuppose évidemment que les concepts fondamentaux permettant de l'exprimer aient été sélectionnés, parmi le vaste champ de possibilités évoqué ci-dessus. Une telle convention peut être réduite à son minimum, c'est-à-dire à un protocole d'interaction, dans un système hiérarchique rigide. Une convention plus ouverte décrira, par exemple :

1°) quels rôles (normalement liés à ses compétences, mais aussi le cas échéant à des pré-affectations préférentielles dans le système) chaque agent peut tenir dans divers scénarios, et les relations entre les divers rôles ; c'est-à-dire une certaine pré-organisation sociale souple de la communauté ;

2°) les méthodes pour parvenir à un accord entre agents sur leurs buts, plans et méthodes, les types de conversations nécessaires à cette fin ;

3°) dans quelles conditions un agent peut s'engager à traiter quels types de problèmes ;

4°) dans quelles conditions il peut réévaluer et/ou abandonner ses engagements, quelles actions corrélatives à cet abandon il doit avoir pour garantir la cohérence du système (actions d'information d'autres agents, par exemple) ;

5°) plus généralement, quelles initiatives il peut prendre, et dans quelles conditions.

Une « convention sociale » peut être basée sur un modèle général d'interaction sociale : négociation, théorie des jeux, ou planification, sur la libre collaboration des agents ou sur leur rémunération selon certains modèles économiques.

Comme indiqué précédemment, c'est des divers aspects de ce paradigme agent que nous allons maintenant nous inspirer pour étendre notre métamodèle initial.

3. ENRICHISSEMENT DE LA MODELISATION DES PROCESSUS METIERS

3.1 Les travaux antérieurs

Différents chercheurs ont cherché à rapprocher la modélisation des SMA et la modélisation des systèmes d'information, dans différentes optiques.

Certains ont proposé des modèles pour représenter les SMA, dont (Kishore, 2004) fait une large revue. Dans cette perspective, les seuls acteurs sont des agents intelligents, et les acteurs humains sont en général considérés comme des utilisateurs du système. Certains auteurs ont

utilisé des langages de modélisation issus du génie logiciel, notamment UML ou en dérivant, pour représenter de tels systèmes (Paik, 2004 ; Bauer, 2004).

D'autres ont cherché à modéliser les communications entre organisations, en utilisant notamment le concept de contrat (Weigand, 2002 ; Hoffner, 2000). D'autres encore ont proposé une vision d'un processus métier comme une « conversation » (Aurimaki, 1988) ou une succession de cycles de communication entre un client et un fournisseur (Mentzas, 2001), utilisant la théorie des actes de paroles. (Kishore, 2004), propose un cadre unificateur entre la modélisation des SMA et la modélisation des systèmes de coordination, permettant de modéliser les interactions entre processus métiers.

(Wagner, 2003) propose un métamodèle basé sur la distinction entre les agents, éléments actifs, des objets, éléments passifs, et orienté vers la représentation des communications.

Nous avons choisi une approche d'enrichissement du métamodèle classique, passant par un affinement du concept d'activité et une spécialisation du concept d'activité.

3.2 Évolution vers un métamodèle Activité-Acteur-Agent

Nous inspirant des principaux concepts du paradigme agent, nous proposons de faire évoluer la structure du métamodèle de deux façons (Fig. 2) :

1. Le concept d'acteur est spécialisé selon le degré d'autonomie, pour distinguer l'exécutant et l'agent.
2. Le concept d'activité est spécialisé selon la nature de la description qu'on va lui attacher, pour autoriser la conception d'activités réutilisables dans différents contextes.

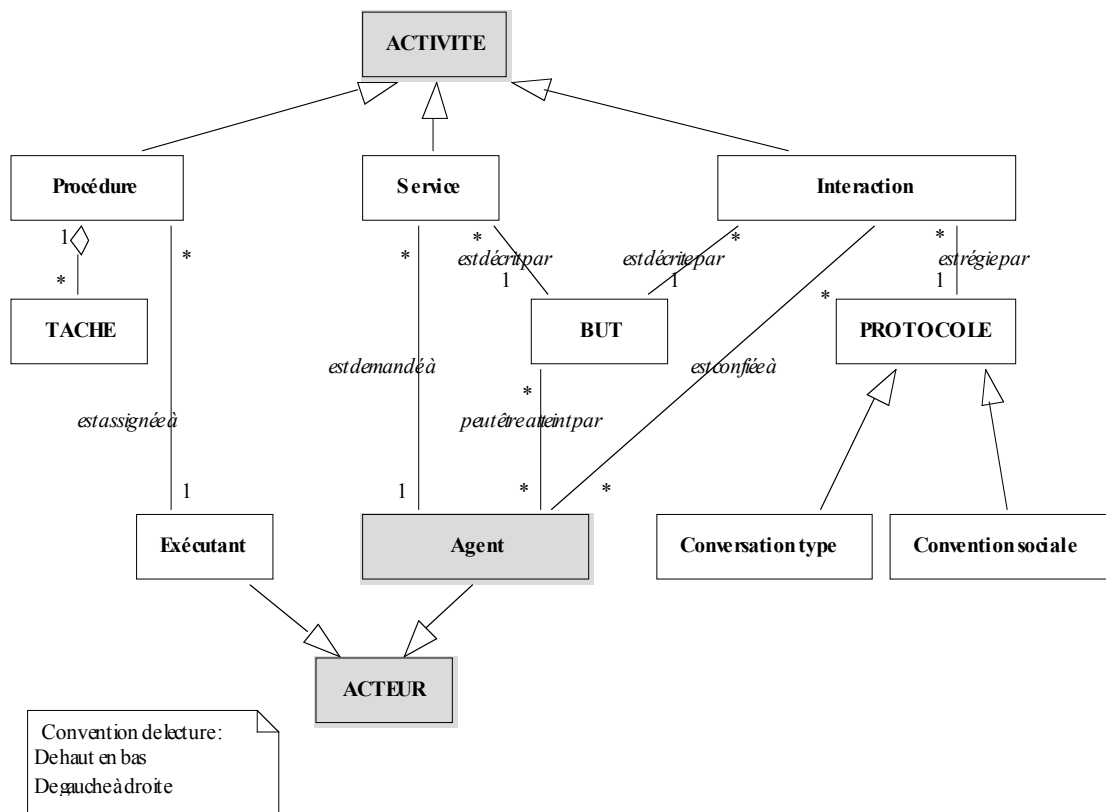


Figure 2 : Métamodèle Activité-Acteur-Agent

Nous allons détailler les nouveaux concepts introduits.

L'**acteur** (tout comme dans le métamodèle précédent) est un élément (personne physique, groupe d'individus, entité organisationnelle ou logiciel) qui est susceptible de jouer un rôle dans la définition d'un processus, ce qui se concrétise lors de l'attribution de la responsabilité de certaines activités. L'acteur peut être interne ou externe à l'entreprise, et un processus peut ainsi être exécuté par plusieurs partenaires qui coopèrent. Il est soit un agent, soit un exécutant.

Le concept d'**agent** permet de caractériser certains acteurs. Un agent est défini comme un acteur ayant capacité à effectuer des activités de façon « autonome », c'est-à-dire sans qu'il soit nécessaire de lui indiquer le mode opératoire, mais seulement le but à atteindre (l'autonomie est ici celle des moyens, pas nécessairement celle des fins). Il peut être interne ou externe à l'entreprise considérée. Ce peut être aussi bien un être humain (individuel ou collectif) qu'un agent artificiel (« intelligent »).

Un **exécutant** est défini comme un acteur qui ne dispose pas d'autonomie et dont on attend qu'il accomplisse des tâches conformément à leur description. Il est forcément interne à l'entreprise de référence. Ce peut être aussi bien un être humain qu'un programme.

Le concept de **but** permet de décrire une finalité, plus limitée que celle qui est attachée au concept d'objectif. On peut utiliser le but aussi bien pour caractériser une activité (« une activité vise à atteindre un but ») qu'un agent (« un agent a capacité à atteindre des buts »).

Une **activité** est spécifiée, de façon exclusive, soit par un but, soit par les tâches qui la composent. Par ailleurs, une activité ne peut être affectée à un agent que si son but correspond à l'un des buts que peut assumer l'agent.

On distingue trois types d'activités.

L'activité de type **Procédure** est une activité définie par les tâches qui la composent et constituent son mode opératoire. C'est, en général, une activité conçue de façon *ad hoc*, pour un processus précis, donc faiblement réutilisable. Elle doit être accomplie par un acteur de type exécutant.

L'activité de type **Service** est définie par un but, elle sera affectée à un agent, doté d'autonomie pour l'accomplir.

L'activité de type **Interaction** est également une activité définie par un but, mais elle ne peut être accomplie que par une coopération entre plusieurs agents. Celle-ci, bien que non spécifiée de façon procédurale par les tâches à effectuer, doit néanmoins s'inscrire dans un cadre organisé. Cela conduit à introduire le concept de Protocole.

Un **protocole** est un ensemble de dispositions qui régissent les échanges entre les agents en vue d'accomplir une activité de type Interaction. On en distingue deux catégories, selon la nature des directives contenues dans le protocole.

Un protocole de type **Conversation-type** est décrit par un graphe des enchaînements possibles de messages entre acteurs types. Sans indiquer les tâches incombant à chacun, il guide la « conversation » des acteurs avec les variations autorisées.

Un protocole de type **Convention-sociale** se compose d'un ensemble de principes d'organisation du travail entre les agents en charge de l'activité (rôles, responsabilités, règles du jeu...). Il est moins directif qu'une conversation-type et possède de ce fait un haut degré de réutilisation.

Le nouveau métamodèle synthétique est donné à la figure 3.

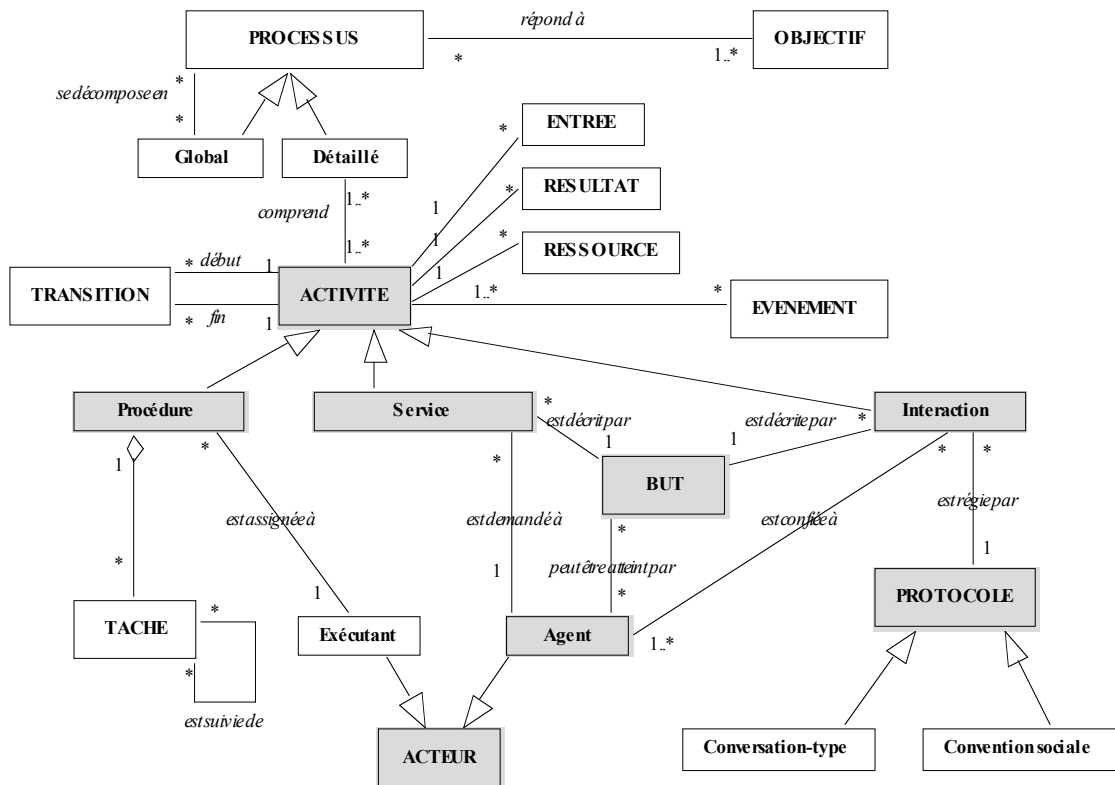


Figure 3 : Nouveau métamodèle de synthèse

Dans ces propositions d'évolution du métamodèle, nous avons cherché à conserver un haut degré de flexibilité dans la conception et l'évolution d'un processus, à travers deux options.

D'une part, nous avons maintenu le découplage entre la description conceptuelle du processus et ses choix organisationnels : c'est ce qui conduit à maintenir le concept central d'activité et à considérer qu'un processus est « un ensemble finalisé d'activités interopérantes, effectuées par des acteurs et/ou des agents ». Les acteurs, agents ou non, ne communiquent que dans le contexte des activités dont ils ont la responsabilité.

D'autre part, on a établi une séparation entre objectif et but. Contrairement à d'autres auteurs (Rolland, 1998), nous n'avons pas considéré qu'un but représente une partie de la décomposition d'un objectif, car les deux concepts ne se situent pas au même niveau sémantique : l'objectif relève du niveau système de gestion et doit pouvoir être mis en correspondance avec des orientations stratégiques de l'entreprise. Pour que le processus atteigne l'objectif, on peut concevoir différents chemins, le choix étant concrétisé par les activités du processus détaillé. En revanche le but se situe au niveau d'une fonctionnalité, et

correspond soit à un résultat pouvant être produit par une activité, soit à un service pouvant être rendu par un agent.

3.3 Apports du métamodèle AAA

Un métamodèle est non seulement un langage de description des modèles (aspect théorique), mais également un guide pour le modélisateur dans la reconfiguration d'un processus métier (aspect pratique). L'apport de notre proposition concerne particulièrement ce dernier aspect.

L'introduction de ces nouveaux concepts et de leurs associations apporte plusieurs ouvertures lorsque l'on construit un processus.

1. Modéliser un processus à l'aide de composants dynamiques

Si l'on s'appuie sur la distinction entre agent et acteur et sur l'autonomie des agents, on peut faire apparaître dans la modélisation des processus certaines activités, uniquement définies en terme de but, et placée sous la responsabilité d'un agent (humain ou logiciel) dont on masque le comportement. Si plusieurs agents sont en charge de l'activité (Interaction), le protocole, notamment celui de type convention sociale représente un cadre standard.

Ceci apporte une solution à la question de la réutilisation. En effet, une activité définie par un but sera potentiellement réutilisable par plusieurs processus, et peut donc être considérée comme un composant de la dynamique.

2. Modéliser les processus inter-organisationnels

On peut également considérer qu'un agent est un système d'information externe susceptible d'effectuer des activités sous forme de service et représenter ainsi des processus faisant intervenir des agents externes au système d'information de l'entreprise. Une activité confiée à un agent externe ne fera l'objet d'aucune description sous forme de tâches, mais n'apparaîtra qu'en terme de but et de résultat fourni à une activité subséquente.

3. Modéliser un processus à structuration émergente

L'autonomie des acteurs permet de représenter des processus, dans lesquels certaines parties sont structurées avec des tâches précises à accomplir, d'autres parties font appel à des agents (individus, organisation ou logiciel) capables de rendre des services, d'autres encore sont spécifiées comme devant être le résultat d'un travail collaboratif dont on indique plus ou moins précisément le cadre à l'aide d'un protocole.

3.4 Illustration

Un exemple, donné à la figure 4, présente un processus « Organisation de livraisons » modélisé à l'aide de composants dynamiques. Il met en jeu les trois types d'activités évoquées : activités procédurales, activités de service et activités d'interaction (intra organisationnelles et inter organisationnelles).

L'entreprise initiatrice est une société de vente par correspondance. Son objectif est, ici, de préparer la livraison des commandes à réaliser pour le lendemain. Cette préparation est fonction des contraintes de délais et de disponibilités exprimées par les clients au passage de leurs commandes.

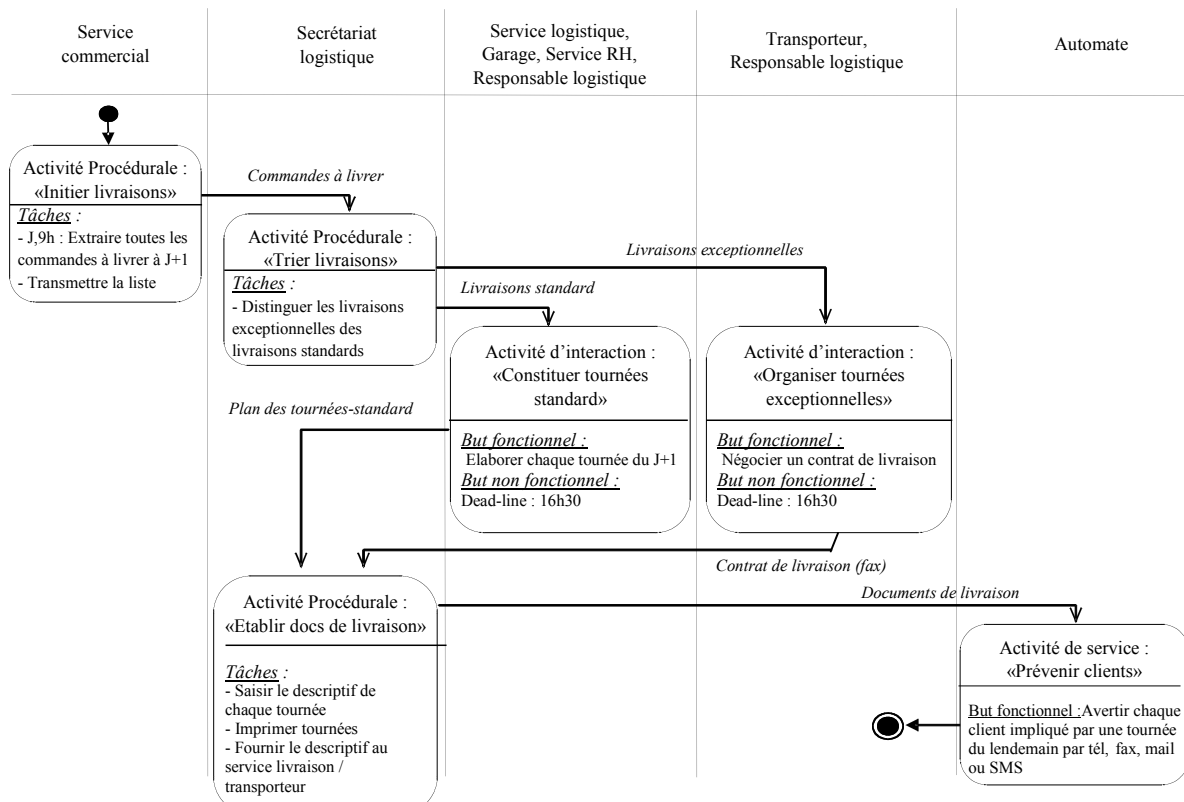


Figure 4 : Exemple de processus basé sur le métamodèle Activité-Acteur-Agent

Le processus met en jeu six activités, dont trois sont décrites par les tâches et trois par les buts.

1. L'activité « Initier livraisons » est définie par les tâches, c'est-à-dire que l'on a indiqué la séquence des tâches à accomplir par l'exécutant (le service commercial) à partir de 9 heures : Extraire toutes les commandes pour lesquelles il a été convenu avec le client une livraison pour le lendemain ; Transmettre au secrétariat logistique la liste des commandes obtenues.
2. L'activité « Trier livraisons », exécutée par le secrétariat logistique (considéré, ici, comme un exécutant), est également définie par les tâches : Distinguer les livraisons à effectuer le lendemain selon qu'elles soient standard ou exceptionnelles ; Transmettre la liste des livraisons standard et la liste des livraisons exceptionnelles au responsable du service logistique. Une commande à livrer deviendra une livraison standard si elle est livrable avec les ressources de la société de vente par correspondance. Dans le cas contraire, elle sera dite exceptionnelle et il s'agira de trouver un transporteur disposant des moyens permettant de la livrer. Deux activités vont alors se dérouler en parallèle : « Constituer tournées standard » et « Organiser tournées exceptionnelles ».
3. L'activité « Constituer tournées standard » est définie par un but. Elle est ainsi décrite « Élaborer chaque tournée standard à réaliser le lendemain ». Elle doit être achevée à 16h30 au plus tard. Cette activité ne fera pas l'objet d'une description détaillée sous la forme d'une succession de tâches, mais sera régie par un protocole

- (de type convention sociale) qui va définir les rôles et les responsabilités de chacun des agents qu'elle va impliquer. Ce protocole indique qu'elle sera placée sous la responsabilité du responsable du service logistique. Les autres agents invoqués seront le service RH qui devra fournir les informations concernant la disponibilité des livreurs, le garage qui devra fournir les informations concernant la disponibilité des camions et le service logistique qui devra élaborer les trajets correspondant à chacune des tournées et leur affecter les ressources nécessaires (camion, chauffeur). Le responsable du service logistique, le garage, le service RH et le service logistique sont considérés comme des agents. Les quatre acteurs (tous internes à l'entreprise considérée) ont toute latitude pour organiser leur collaboration et la façon de procéder de chacun d'eux pour produire le résultat demandé dans les délais impartis.
4. L'activité « Organiser tournées exceptionnelles » est définie par un but. Elle est ainsi décrite (pour chaque tournée exceptionnelle à effectuer) : « Trouver un transporteur apte à effectuer la livraison souhaitée et élaborer avec lui un contrat de service ». Elle doit être achevée à 16h30 au plus tard. Cette activité est définie par un protocole (de type convention sociale) qui fixe les rôles et obligations de chacun des agents impliqués. Le résultat produit par cette activité (un contrat) est le fruit d'une collaboration entre le responsable du service logistique (agent responsable de l'activité) et une société externe qui joue le rôle d'un prestataire de service (agent fournisseur). Le déroulement de cette activité (organisation, tâches) n'est pas visible dans la description du processus. Les agents sont autonomes dans la façon de s'organiser pour atteindre le but fixé dans les délais impartis.
 5. L'activité « Établir documents de livraison » est définie par les tâches, c'est-à-dire que l'on a indiqué la séquence des tâches à accomplir impérativement par l'exécutant (le service logistique) : Saisir le descriptif de chaque tournée à réaliser le lendemain ; imprimer ces descriptifs et les envoyer au service livraison et dans le cas des tournées exceptionnelles au transporteurs sélectionnés.
 6. L'activité « Prévenir client » est une activité de service, elle est définie par un but et réalisée par un agent unique. Dans le contexte décrit par cet exemple, elle pourrait correspondre à l'appel d'une activité réutilisable, en l'occurrence une application développée par le service informatique, qui se présentera sous la forme d'un service web interne (pattern « visiteur »). Dans la description de ce processus, on perçoit cette activité comme une boîte noire et on n'affinera jamais sa description. C'est pourquoi l'acteur impliqué est considéré comme un agent.

Ajoutons quelques remarques sur la démarche de modélisation de ce processus.

Ce processus peut être modélisé en plusieurs cycles. Dans un premier temps, le concepteur peut dégager des activités définies à « gros grain » ; cela va lui permettre de se focaliser sur la structuration du processus en étapes vues comme des « boîtes noires », puis par la suite revenir sur chacune d'elles pour les décrire de façon plus détaillée. Les activités procédurales vont être alors ensuite décrites sous la forme d'un enchaînement de tâches assignées à un exécutant (et donc devenir des « boîtes blanches »). Les activités de service et

d'interaction resteront des « boîtes noires », mais le concepteur les approfondira en cherchant soit un service qui va permettre de la réaliser (cela inclut l'agent responsable de cette réalisation pour les activités de service), soit à agréger un ensemble de compétences complémentaires afin de parvenir au but visé. Dans ce dernier cas, il sera nécessaire de décrire le protocole de collaboration entre les agents ayant ces compétences, c'est-à-dire de préciser les rôles et responsabilités de chacun d'eux : le collectif d'agents aura alors toute l'autonomie pour parvenir au résultat attendu dans le cadre des délais fixés.

3.5 Limites et perspectives

La recherche présentée ici relève de la catégorie « artefact », selon la distinction développée par (Hevner, 2004) : son but est d'améliorer les capacités des modélisateurs. Elle se situe dans l'articulation entre organisation et technologie. Le métamodèle proposé a été validé sur un cas d'école. Il fera par la suite l'objet d'une validation approfondie. Celle-ci devra apporter une double preuve :

1. Celle de l'utilité conceptuelle, c'est-à-dire l'apport effectif des possibilités nouvelles de modélisation en termes de précision, de richesse et de souplesse des modèles produits. Cela renvoie au rôle du métamodèle comme guide pour le modélisateur de processus métier.

2. Celle de faciliter le passage entre niveau conceptuel et niveau logique : en effet, initialement inspiré des SMA, ce métamodèle devrait rendre plus aisée la conception de systèmes informatiques mettant en oeuvre simultanément des SMA et des technologies classiques.

Pour cela, nous nous proposons d'appliquer le modèle sur un cas réel dans le domaine du e-commerce (système de gestion de place de marché). Les processus seront représentés aux deux niveaux, conceptuel et logique, ainsi que la passerelle permettant le passage. Ensuite, une validation à plus grande échelle sera mise en place, probablement avec la collaboration d'un éditeur d'outils.

De plus, dans le prolongement de la démarche actuelle, nous nous inspirerons des méthodes de structuration des « sociétés d'agents » pour proposer des éléments complémentaires de modélisation pour représenter les processus collaboratifs.

CONCLUSION

Dans le but d'améliorer la maîtrise du système d'information, nous nous sommes focalisés sur la modélisation des processus métier. Partant d'un métamodèle qui synthétise les représentations classiques des systèmes d'information, nous en avons montré quelques limites et nous nous sommes inspirés du paradigme des systèmes multi agents pour tenter de les dépasser. Nous avons ainsi produit un nouveau métamodèle qui permet la réutilisation d'activités, une structuration plus souple et l'intégration d'activités collaboratives ainsi que d'acteurs dotés d'autonomie. Ce métamodèle enrichit le précédent par des concepts nouveaux (comme Agent, Exécutant, Service, Interaction, But, ...). Il a été illustré par un exemple de modélisation d'un processus inter-organisationnel.

La suite de ce travail visera à valider le métamodèle de manière plus approfondie.

REFERENCES

- Alter, S. (1999), *Information Systems : A Management perspective*, 3e éd., Addison-Wesley.
- Aurimaki E., Lehtinen E. et Lyytinen K. (1988), A speech-act-based office modeling approach, *ACM transaction on Office Information Systems*, 6, 2.
- Austin J. (1956), *How to do things with words*, Clarendon Press, Oxford.
- Bauer B. & Odell J. (2004), UML 2.0 and agents : how to build agent-based systems with the new UML standard, *Engineering Applications of Artificial Intelligence*.
- Berthier D. (2002), *Le savoir et l'ordinateur*, L'Harmattan, Paris.
- Berthier D. (2004), Discordances ontologiques et questions d'interopérabilité, Colloque « *Source et ressources pour les sciences sociales* », EHESS, Paris.
- Bertziss A.T. (1999), Domain Analysis for Business Software Systems, *Information Systems*, 24, 7, 555-568.
- Bradshaw Jeffrey ed. (1997), *Software Agents*, AAAI Press / MIT Press, Cambridge, Mass.
- Bustard D., Kawalek P. & Norris M. eds. (2000), *System modeling for business process improvement*, Artech House.
- Bustard D., Kawalek P. et Norris M. eds (2000), *System modeling for business process improvement*, Artech House.
- Butler K.A., Esposito C. & Hebron R. (1999), Connecting the design of software to the design of work, *Communications of the ACM*, 42, 1, 39-46.
- Castro J., Kolp M. & Mylopoulos J. (2002), Towards requirements-driven information systems engineering : the Tropos project, *Information Systems*, 27, pp. 365-389.
- CEN/CENELEC (1995), ENV12204, Advanced Manufacturing Technology – Systems Architecture – Constructs for enterprise modelling.
- Chauvet J.M. (2002), Services Web avec SOAP, WSDL, UDDI, ebXML, Eyrolles, Paris.
- Cohen P. & Levesque H. (1997), Communicative Actions for Artificial Agents, in (Brashaw, 1997).
- Cohen P., Morgan J. & Pollack M. (1990), *Intentions in Communication*, MIT Press, Cambridge, Mass.
- FIPA : <http://www.fipa.org>
- Georgel F.(2005), *IT Gouvernance*, Dunod.
- Green P. & Rosemann M. (2000), Integrated process modelling : an ontological evaluation, *Information Systems*, 25, 2, 73-87.
- Hevner A.R., March S., Park J., Ram S. (2004), Design Science in Information Systems Research, *MIS Quarterly*, .28, 1, pp.75-105.
- Hoffner Y., Lwdwig H., Gulcu C. & Grefen P. (2000), An architecture for cross-organisational business processes, *Advanced Issues of E-Commerce and Web-Based Information Systems*. WECWIS.
- Horrocks I., Patel-Schneider P. & van Harmelen F. (2003), From SHIQ and RDF to OWL : the making of a Web Ontology Language, *Web Semantics : Science, Services and Agents on the World Wide Web*, Vol. 1, n° 1, pp. 7-26, Elsevier.
- Inverno d',M. & Luck M. (2001), *Understanding Agent Systems*, Springer,.
- ISO9000 (2000), *Qualité et systèmes de management ISO 9000*, Afnor.

- Jean G. (2000), *L'urbanisation du business et des SI*, Hermès.
- Jennings N. (1994), *Cooperation in Industrial Multi agents Systems*, World Scientific, Singapore.
- Kawalek P. et Greenwood R.M. (2000), The organization, the process, and the model, in (Bustard, 2000), pp. 61-80.
- Kishore R., Zhang H. & Ramesh R. (2004), Enterprise integration using the agent paradigm : foundations of multi agents-based integrative business information systems, *Decison Support Systems*.
- Levesque Hector & Pirri Fiora (1999), *Logical Foundations for Cognitive Agents*, Springer.
- Longépé C. (2004), *Le projet d'urbanisation du système d'information*, 2e éd., Dunod.
- Maes P., ed. (1991) *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*. MIT Press..
- Melão N. & Pidd M. (2000), A conceptual framework for understanding business process and business process modeling, *Information Systems Journal*, 10, 105-129.
- Mentzas G., Halaris C. & Kavadias S. (2001), Modelling business processes with workflow systems : an evaluation of alternative approaches, *International Journal of Information Management*, 21, 123-135.
- Morley C. (2004), Un cadre unificateur pour la représentation des processus, Pre-ICIS.
- Morley C., Hugues J., Leblanc B. & Hugues O., (2005), *Processus métiers et systèmes d'information*, Dunod.
- Mougin Y. (2004), *La cartographie des processus*, Edition organisation.
- Newell A. (1982), The Knowledge Level, *Artificial Intelligence*, Vol 59, pp 87-127.
- O'Donnell E. & David J. (2002), How information systems influence user decisions : a research framework and literature review, *International Journal of Accounting Information Systems*, 1, pp. 178-203.
- Österlé H., Brenner W. & Hilbers H. (1993), *Total Information Systems Management*, Willey.
- Paik I., Takami S. & Watanabe F. (2004), Intelligent agent to support design in supply chain based on semantic web services, *Proceedings of the 4th International Conference on Hybrid Intelligent Systems*.
- Reix R. (2002), *Systèmes d'information et management des organisations*, Vuibert
- Rolland C., Nurcan S. & Grosz G. (1998), A unified framework for modeling cooperative design processes and cooperative business processes, *IEEE proc. 31st annual Hawaii International Conference on System Sciences*.
- Rowe F. éd. (2002), *Faire de la recherche en systèmes d'information*, Vuibert – Fnege.
- Scheer A.-W. (2000), *ARIS Business Process Modeling*, 3^e éd., Springer-Verlaag.
- Schlenoff C. (2000) , Gruninger, M., Tissot F., Valois J., Lubell, J. , Lee, J.D., The Process Specification Language (PSL) Overview and Version 1.0 Specification, NISTIR 6459.
- Searle J. & Vanderveken D. (1985), *Foundations of Illocutionary Logic*, Cambridge University Press, New York.
- Searle J. (1969), *Speech Acts : an Essay in the Philosophy of Language*, Cambridge University Press, Cambridge, U.K.
- Shoham Y. (1993), Agent-oriented programming, *Artificial Intelligence*, Vol. 60, pp. 51-92.

- Succi G., Predonzani P. et Vernazza T. (2000), « Business process modeling with objects, costs and human resources », in (Bustard, 2000), pp. 47-60.
- Sutcliffe A.G. (2000), Business modeling interprocess relationships, in (Bustard, 2000), 117-134
- Te'eni D. (2000), Modeling organizational communication : top-down analysis and bottom-up diagnosis, in (Bustard, 2000), pp. 249-261.
- Vanderveken M. (1990), *Meaning and Speech Acts*, Vol. 1 : *Principles of language use* ; Vol. 2 : *Formal semantics of success and satisfaction*, Cambridge University Press, Cambridge, Masss.
- Vidal P. et Nurcan S. (2002), Coordination des actions organisationnelles et modélisation des processus, in (Rowe, 2002)
- Wagner G. (2003), The Agent-Object-Relationship Metamodel : towards a unified view of state and behavior, *Information Systems*, 28, pp. 475-504.
- Weigand H. et van den Heuvel W.J. (2002), Cross-organizational workflow integration using contracts, *Decision Support Systems*, Volume 33, Issue 3, pp. 247-265.
- WFMC (1999), The Workflow Management Coalition Terminology and Glossary. Document Reference WFMC-TC-1011, 3.0.