

## DU PROCESSUS COLLABORATIF AU MEDIATEUR DES SYSTEMES D'INFORMATION DES PARTENAIRES

Frédéric BENABEN, Jihed TOUZI,  
Hervé PINGAUD

DRGI, Ecole des Mines d'Albi-Carmaux  
Route de Teillet  
81013 ALBI Cedex 9

[frederick.benaben@enstimac.fr](mailto:frederick.benaben@enstimac.fr),  
[jihed.touzi@enstimac.fr](mailto:jihed.touzi@enstimac.fr), [herve.pingaud@enstimac.fr](mailto:herve.pingaud@enstimac.fr)

Jean-Pierre LORRE

EBM WebSourcing  
Parc Technologique du Canal,  
10 avenue de l'Europe  
31520 Ramonville St Agne

[jean-pierre.lorre@ebmwebsourcing.com](mailto:jean-pierre.lorre@ebmwebsourcing.com)

**RESUME :** Dans un contexte de collaboration industrielle ou inter-organisationnelle, la qualité de l'intégration des différents partenaires dépend grandement de la capacité de leurs systèmes d'information (SI) à interagir efficacement. Nous proposons dans cet article d'aborder cette problématique selon l'angle de l'interopérabilité de ces systèmes d'information. Cette interopérabilité des SI des partenaires s'appuie conceptuellement sur deux caractéristiques : (i) la faculté de ces SI à se conformer à une orientation services (SOA pour Service-Oriented Architecture) et (ii) le positionnement d'un système d'information médiateur au sein du réseau de partenaires destiné à assurer l'intégration du « système de systèmes » ainsi créé. Cet article traite précisément de la conception de ce médiateur selon une démarche MDA (Model Driven Architecture). Notre approche consiste à considérer que la connaissance contenue dans un modèle de processus collaboratif (en tant que dimension fonctionnelle de la collaboration) pourrait permettre d'alimenter une démarche de modélisation de ce SI médiateur. Après avoir exposé les fondements théoriques de ces propositions, nous allons présenter un certain nombre de règles de traduction ainsi qu'un aperçu de l'outil logiciel les supportant, permettant d'assurer cette modélisation du médiateur.

**MOTS-CLES :** Système d'information, Interopérabilité, Processus collaboratif, Services, Système de systèmes.

### 1. INTRODUCTION

D'un paysage figé et fractionné au sein duquel il fallait s'implanter et pérenniser sa position, le tissu industriel et économique évolue vers un environnement mouvant, fluide et chargé, dans le flot duquel il faut naviguer. Cette fluidification de la structure du tissu industriel s'avère à la fois porteuse de nouvelles opportunités pour les entreprises (inhérentes à cette malléabilité accrue) mais également de nouvelles contraintes d'agilité, plus critiques et plus exigeantes : La capacité à collaborer n'est plus seulement un atout marginal dans cet univers industriel fluide, c'est une nécessité de survie tant l'environnement tend à la rendre incontournable

Directement impactée par cette tendance, la notion de collaboration d'entreprise subit également une évolution notable que l'on peut illustrer par un adage simpliste et parodique : « plus loin, plus vite, plus fort » :

- Rayonnement (*plus loin*) : l'accessibilité des clients, partenaires ou fournisseurs s'est largement étendue,
- Réactivité (*plus vite*) : aux relations durables et inscrites dans le fonctionnement même des partenaires s'ajoutent à présent des collaborations plus ponctuelles et opportunistes,

- Intensité (*plus fort*) : les collaborations se veulent plus profondes dans le partage et l'échange qu'elles impliquent (données, informations, applications, services, processus, etc.).

Cette tendance à la fluidification efficace et réactive du milieu industriel, cette exigence d'agilité correspond à une finalité unique : l'intégration des composants du tissu économique et industriel. Notre problématique relève donc de la notion d'intégration des entreprises. Nous proposons d'aborder ce sujet selon sa dimension système d'information en nous intéressant en particulier à la question de l'interopérabilité des SI comme support de cette intégration des entreprises.

Au sein de cette vaste ambition, cet article propose de se concentrer sur la démarche de conception d'un SI médiateur garant de l'interopérabilité des SI des partenaires de la collaboration. Pour ce faire, nous allons tout d'abord présenter notre vision de l'interopérabilité des systèmes d'information en tant que capacité des SI des partenaires à constituer un système de systèmes (dont l'homogénéité sera justement assurée par ce SI médiateur). Dans un second temps, nous étudierons conceptuellement et pratiquement la conception d'un tel médiateur. Enfin, nous discuterons des perspectives et mise en application de ces travaux.

## 2. L'INTEROPERABILITE DES SI

L'interopérabilité, définie par le réseau d'excellence européen INTEROP NoE<sup>1</sup> comme *la capacité de systèmes à pouvoir travailler ensemble sans effort particulier pour les utilisateurs de ces systèmes* (Konstantas *et al.*, 2005), peut être positionnée comme un moyen de tendre vers le concept d'intégration (Vernadat, 2006). L'objectif de nos travaux est en effet d'accompagner (conceptuellement et techniquement) la transition qui amène les acteurs du domaine industriel à passer d'un *isolement compensé par l'ouverture technologique de leur système d'information à une intégration optimale de ces mêmes systèmes d'information en un système de systèmes agile et pertinent* (cf. figure 1). Cette notion de « système de systèmes » caractérisée par (Maier, 1998) selon cinq critères :

- Indépendance opérationnelle des éléments,
- Indépendance managériale des éléments,
- Développement évolutif,
- Comportement émergent,
- Distribution géographique.

Dans le contexte dont il est question dans cet article, les SI des partenaires peuvent opérer indépendamment (et le font effectivement) ce qui couvre les deux premiers critères. Le système constitué des SI des partenaires doit par ailleurs s'adapter aux évolutions de la situation, ce qui correspond au troisième point. Le fait que l'ensemble des SI fonctionne sur la base de processus collaboratif relève du quatrième point. Enfin, ces SI se contentent d'échange de données (pas de flux d'énergie ou de matière), ce qui correspond au dernier critère de Maier.

La question qui nous intéresse dans cette section 2 concerne moyen d'obtenir concrètement cette interopérabilité de SI et ce système de système (*SoS pour System of Systems*).

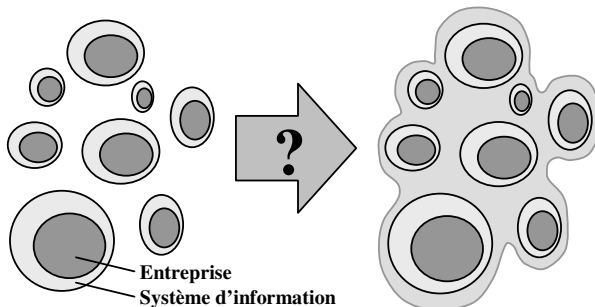


Figure 1. L'intégration des SI en un système de systèmes

Si nous pouvons partir du principe que le système d'information (ou plus exactement sa projection informatique) peut être vu comme *un ensemble de données, d'applications et de processus interagissant* (Morley,

2002), la notion d'interopérabilité des SI se décompose alors comme suit (Bénaben *et al.*, 2006) :

- Gestion des données,
- Management des applications,
- Pilotage des processus collaboratifs.

Notre hypothèse, reprise en partie de (Aubert, *et al.*, 2002) et discutée dans (Touzi, *et al.*, 2006) postule que les systèmes d'information des partenaires en présence ne peuvent, seuls, couvrir ces trois aspects (à moins de se caractériser par une uniformisation et une standardisation tant logique que technique trop réductrice pour être adaptée à notre étude) et doivent donc s'en remettre à un système chargée de la médiation de l'ensemble des SI pour en faire un SoS (cf. figure 2).

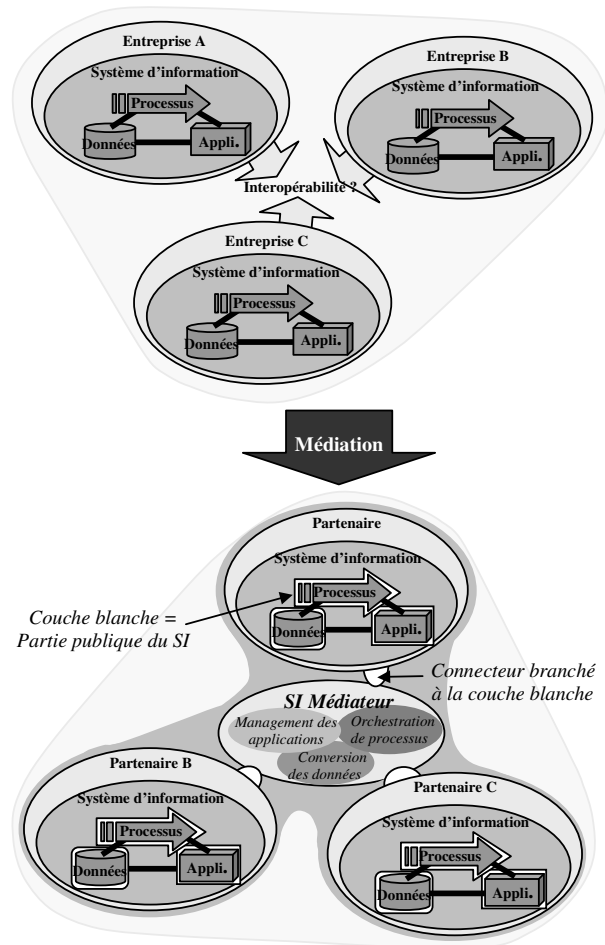


Figure 2. L'interopérabilité des SI par médiation

A ce postulat (interopérabilité par médiation), nous ajoutons le fait que les SI des partenaires concernés par la collaboration relèvent d'une architecture SOA (Service-Oriented Architecture). Ce point de vue implique que les partenaires potentiels soient en mesure de présenter leurs systèmes d'information sous la forme d'un ensemble de services, localisés et accessibles. Les trois niveaux de gestion de l'interopérabilité vus selon cet angle SOA nous permettent de proposer une première structuration du SIM (cf. figure 3) qui doit alors disposer de :

<sup>1</sup> INTEROP NoE (Network of Excellence), réseau de recherche européen travaillant sur l'interopérabilité des entreprises et de leurs applications. <http://www.interop-noe.org/>

- la connaissance sur les données des partenaires (dictionnaire ontologique et formats),
- l'annuaire des services des partenaires,
- les modèles de processus collaboratifs et le moteur de workflow pour les exécuter,

A ceci, nous pouvons également ajouter un ensemble de services propres au SIM lui permettant d'améliorer la collaboration (services *supports* tels que la sécurisation des échanges et des services *complémentaires* apportant une valeur ajoutée plus tangible).

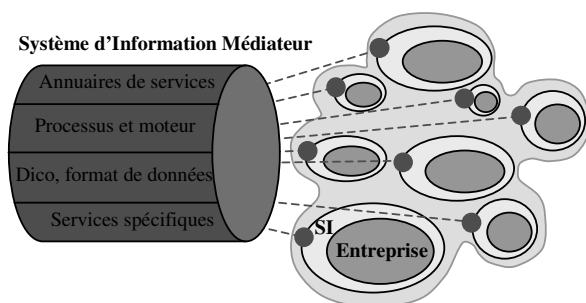


Figure 3. Première vision de l'architecture du médiateur

Cette architecture du *médiateur* permet d'envisager la collaboration selon le déroulement suivant :

- les partenaires définissent ensemble la dynamique de leur collaboration en construisant le modèle du *processus* de collaboration représentatif,
- chaque partenaire définit le « visage » qu'il veut présenter: ses *données* (en termes de sémantique et de formats) et ses *services* (en termes d'intérêt et de localisation) qui participeront à la collaboration,
- Les partenaires et l'entité responsable du *médiateur* définissent ensemble les services spécifiques que nécessite leur collaboration.

Le *médiateur* pourra alors être en charge de l'orchestration du processus collaboratif entre les différents partenaires. L'objectif fonctionnel principal de cette approche est de rendre l'existence du médiateur *transparente* : les partenaires transmettent ou reçoivent des *données* sans se préoccuper des formats, ils accèdent aux *services* conformément aux conventions de collaboration et le *processus* collaboratif s'exécute via le pilotage « discret » du médiateur.

### 3. CONCEPTION DU MEDIATEUR

Nous avons présenté dans la section précédente le concept de SI Médiateur de la collaboration en tant que proposition de solution garante de *l'interopérabilité des SI des partenaires* (pourvu que ceux-ci respectent une architecture SOA). Nous allons à présent nous intéresser à la démarche de conception de ce système spécifiquement en charge de la médiation. Cette approche ambitionne d'accompagner une descente en abstraction au travers des couches traditionnelles de représentation des SI : couches *métier*, *logique* et *technique*, pour aboutir à la définition juste et adéquate du SI Médiateur.

#### 3.1 L'odyssée à travers les niveaux d'abstraction

Nous proposons de baser notre démarche de conception du SI Médiateur sur le cadre de conception que représente l'approche MDA. Cette approche de conception s'appuie sur deux points :

- l'utilisation de modèles successifs permettant de parcourir les différents niveaux d'abstraction nécessaires à la conception,
- la séparation des concepts technologiques du reste de la conception (vision métier du sujet, orientations et choix logiques concernant le systèmes). L'architecture physique (plate-forme et technologie) intervient au plus tard dans la conception.

La figure 4 présente notre proposition de démarche de conception projetée sur le « Y » du cycle MDA :

- La vue métier (branche gauche du « Y ») avec :
  - notre vision du *Computational Independent Model* (CIM) en tant que modèle fonctionnel décrivant le processus collaboratif à mettre en œuvre,
  - le *Platform Independent Model* (PIM) sous la forme du modèle UML logique de SIM (SOA),
- La vue technique (branche droite du « Y ») avec :
  - Le *Platform Model* (PM), choix technologique capable de supporter le SIM. Cette modélisation relève de la responsabilité d'un industriel, dans notre cas, il s'agit de la société eBM WebSourcing, éditeur et fournisseur de solutions d'interopérabilité (basée sur la technologie d'*Enterprise service Bus*)
- La vue d'intégration (branche basse du « Y ») avec :
  - Une phase placée sous la responsabilité de l'architecte de SI capable de conduire la phase d'urbanisation de système d'information permettant la projection des orientations et choix logiques (PIM) sur le canevas technique (PM)
  - La génération du *Platform Specific Model* (PSM) résultant de cette convergence logique/technique.

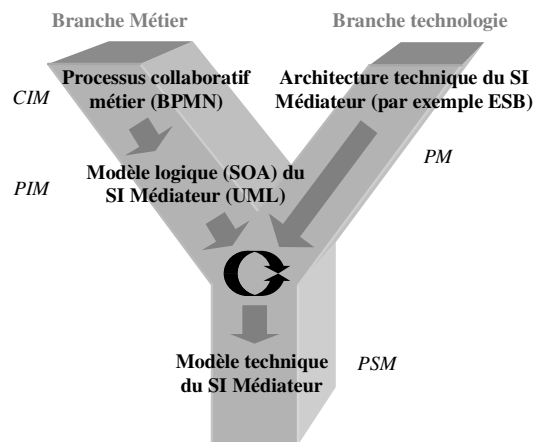


Figure 4. Approche MDA de conception de SI Médiateur

On constate que cette structuration de la conception du SI Médiateur selon une démarche MDA respecte la descente en abstraction citée précédemment puisqu'on retrouve bien les couches *métier*, *logique* et *technique*.

Néanmoins, une question se pose vis-à-vis de cette proposition : *S'il est courant et cohérent de s'orienter vers un modèle UML pour couvrir la vue logique de la solution au travers du PIM, qu'en est-il de la pertinence de l'utilisation d'un modèle de processus en BPMN pour le CIM ?* Nous proposons de discuter de cette interrogations dans la partie suivante de cette troisième section.

### 3.2 L'intérêt d'un modèle de processus exprimé en BPMN pour le niveau métier (CIM)

L'objectif du modèle situé au niveau de la couche métier (le CIM) est – de notre point de vue tout du moins – de fournir la connaissance formalisée nécessaire à la conception du modèle logique (le PIM). Partant de cet objectif, et puisque le PIM est quant à lui exprimé à l'aide du formalisme UML, la problématique peut alors être reformulée ainsi : *La connaissance contenue dans des modèles de processus en BPMN constitue-t-elle une « matière première » suffisante pour construire un modèle de SI Médiateur en UML ?*

C'est finalement la question de l'homogénéité de « l'équation » suivante qui est posée :

*modèle d'entreprise (BPMN) → modèle de SIM (UML)*

CIMOSA (Vernadat, 1996) identifie quatre points de vue des systèmes entreprises qui permettent de décrire la partie gauche de l'équation : vue fonctionnelle (processus), vue informationnelle (données, informations), vue organisationnelle (organigramme, structure), et vue des ressources (services, compétences, disponibilités).

D'après (Booch *et al.*, 2004) et (Roques, 2004), le formalisme UML permet une description de la partie droite de l'équation selon des vues complémentaires : architecturale (agencement des composants physiques), comportementale (description de la dynamique), fonctionnelle (identification des fonctions) et structurelle (agencement des composants logiques). L'équation peut alors être représentée par la figure 5.

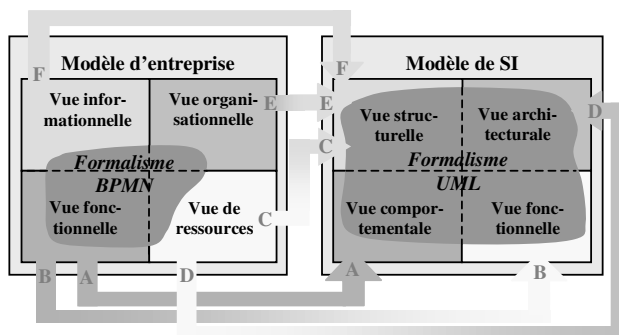


Figure 5. Correspondances entre modélisation d'entreprises et modélisation de SI

Avant d'entamer la réflexion sur la façon de réaliser telle ou telle liaison identifiée sur la figure précédente, nous allons commencer par statuer sur l'intérêt ou non des vues du modèle SI qui sont « au bout » de ces liaisons :

Est-il nécessaire que le PIM (représentant une vision logique du SI Médiateur) couvre toutes les vues du domaine de la modélisation des SI ?

Premièrement, il nous semble que la vue architecturale (couverte au moins partiellement par le formalisme UML et les diagrammes de composants ou de déploiement) est dédiée à une vision *technique* du système, et à ce titre ne fait pas partie du PIM. Nous proposons donc d'écarter la question de la liaison D.

Deuxièmement, il nous semble que la vue fonctionnelle du domaine de la modélisation UML (représentée par les diagrammes de cas d'utilisation) relève d'une excellente façon de formaliser le besoin et donc de fournir un CIM représentatif de la couche *métier*, ce qui n'est pas notre propos ici. Nous proposons donc également d'écarter la question de la liaison B. Notons à ce propos que cette liaison était partiellement couverte par BPMN même si la nature « comportementale » du formalisme BPMN (à opposer à une nature plus structurée d'un formalisme de représentation tel que SADT) ne lui permettait pas de la couvrir intégralement (il manque en particulier tout ce qui concerne la hiérarchie des fonctions).

A présent qu'il nous reste principalement les vues structurelle et comportementale à couvrir. Nous pouvons revenir sur les liaisons A, C, E et F :

- **Liaison A :** Cette liaison est globalement native dans la mesure où le formalisme BPMN apporte une vision formalisée permettant de construire la vue comportementale contenant les processus et scénarii.
- **Liaison C :** Cette liaison est partiellement couverte par le fait qu'un diagramme BPMN identifie les activités (métier) nécessaires au déroulement du processus, lesquelles peuvent permettre de lister les services métiers indispensables à la construction de la vue structurelle. En outre, nous proposons d'ajouter un recueil des services que chaque partenaire propose d'apporter au sein de la collaboration.
- **Liaison E :** Cette liaison ne peut être couverte que si le modèle de processus intègre des informations suffisantes sur l'organisation et la topologie du réseau de partenaires. Nous proposons pour ce faire de baser nos modèles de processus BPMN sur un méta-modèle permettant de formaliser la notion de processus collaboratif (voir §3.3).
- **Liaison F :** S'il est clair qu'une vision partielle des données manipulées par la collaboration peut apparaître au sein du modèle BPMN sous la forme de données échangées et de messages, cette représentation des objets métier manipulés par le réseau de partenaires demeure incomplète. Nous proposons donc d'ajouter un recueil des données que chaque partenaire apporte à la collaboration.
- **Complément :** Il semble difficile d'imaginer construire une description logique de SI Médiateur à partir d'un modèle de processus sans apporter un savoir sur le choix de son organisation logique. C'est pourquoi nous proposons de baser notre Si Médiateur sur un méta-modèle orienté-services (SOA).

La conclusion de cette étude est donc que, complétée à l'aide d'informations sur les services et les données mi-

ses en commun par les partenaires, il semble que le modèle de processus BPMN permette de construire un modèle UML de la vue logique du SI Médiateur. Nous allons discuter des méta-modèles de processus collaboratif et de SI Médiateur dans la sous-partie suivante.

### 3.3 Présentation des méta-modèles

Nous avons vu dans la partie précédente la nécessité de disposer de deux méta-modèles. Le premier permettant de définir l'organisation d'un processus collaboratif exprimé en BPMN (figure 6) et le second permettant de définir la structure logique d'un SI Médiateur respectant une architecture SOA et exprimé en UML (figure 7).

Le méta-modèle de processus collaboratif est composé les éléments suivants (**NB : SI Médiateur correspond à CIS en anglais**) :

- La classe *BPMN process* est une classe abstraite qui désigne un processus qui respecte le grammaire de langage BPMN, notre processus (la classe *collaborative process*) respecte aussi le grammaire BPMN (lien de spécialisation),
- *partner pool* désigne une *pool* d'un partenaire,
- *CIS pool* désigne la *pool* du SI Médiateur,
- *partner lane* désigne les *lane* d'une *partner pool*,
- *CIS lane* désigne les *lane* de la *CIS pool*,
- *message flow* permet de lier une *partner pool* avec une *CIS pool* et porte toujours au moins une *data*,
- *sequence flow* permet de lier des éléments BPMN appartenant à la *CIS pool* et peut véhiculer une *data*.
- *partner task* désigne une tâche d'une *partner pool*,
- *CIS task* désigne une tâche appartenant à *CIS pool*,
- *start event*, *intermediate event*, *end event*, *gateway* et *sub process* sont les éléments BPMN qui sont définis dans Définition1 de ce chapitre qui appartiennent uniquement à la *CIS pool*,
- *CIS component* est une classe abstraite désignant tout élément qui appartient à la *CIS pool*.

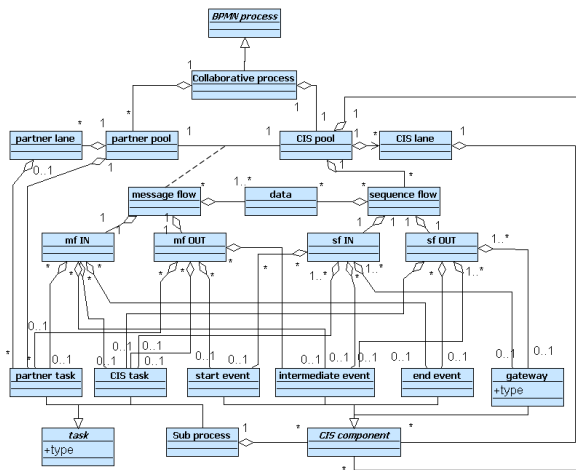


Figure 6. Méta-modèle de processus collaboratif BPMN

Afin de faciliter la gestion des liaisons entre éléments BPMN nous avons défini les éléments suivants :

- *mf IN* et *mf OUT* sont respectivement les extrémités d'entrée (in) et de sortie (out) d'un *message flow*.
- *sf IN* et *sf OUT* sont respectivement les extrémités d'entrée (in) et de sortie (out) d'un *sequence flow*.

Le méta-modèle de SI Médiateur SOA s'inspire fortement, dans sa structuration, des travaux sur PIM4SOA (Benguria *et al.*, 06). Nous considérons le SI Médiateur selon les trois vues différentes suivantes :

- **La vue services** : décrit les services (des partenaires et du SI Médiateur) utilisés dans la collaboration.
- **La vue informationnelle** : décrit les données échangées entre les différents partenaires.
- **La vue processus** : décrit l'enchaînement et la coordination des services en terme d'interactions et des flux de contrôle de processus.

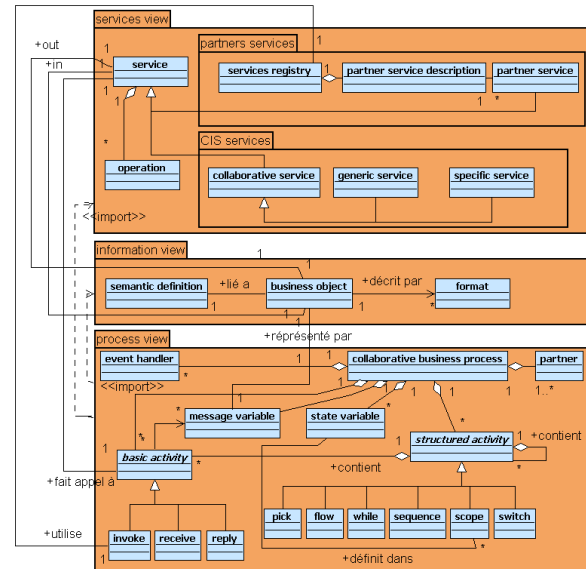


Figure 7. Méta-modèle de SI Médiateur SOA en UML

La vue service contient les éléments suivants :

- *service* est une classe abstraite qui désigne un service impliqué dans la collaboration.
- *operation* désigne une opération d'un service.
- Le package *Partners services* désigne la gestion des services des partenaires.
- *services registry* désigne un registre accessible par le SI Médiateur et qui gère un ensemble de fiches décrivant les services.
- *partner service description* désigne une fiche qui décrit un service : ses opérations et une présentation de son rôle dans la collaboration.
- *partner service* désigne un service offert par un partenaire dans la collaboration.
- Le package *CIS services* désigne la gestion des services du SI Médiateur.
- *collaborative service* est une classe abstraite qui désigne un service offert par le SIM.

- *generic service* désigne un service générique qui peut être utilisé dans un bon nombre de processus collaboratifs. Ce service existe déjà dans une bibliothèque de services gérée par le SI Médiateur.
- *specific service* désigne un service spécifique à concevoir pour une collaboration particulière.

La vue information contient les éléments suivants :

- *business object* désigne un objet échangé entre deux services.
- *format* décrit le format de message.
- *semantic definition* désigne une description sémantique de l'objet. Cette description permet d'identifier facilement l'objet.

La vue processus contient les éléments suivants :

- *collaborative business process* désigne le processus collaboratif géré par le SIC,
- *partner* désigne un partenaire de processus,
- *message variable* désigne une variable qui présente un message échangé dans le processus,
- *state variable* désigne une variable qui peut être utilisée au cours de l'exécution du processus. Cette variable peut prendre plusieurs valeurs suivant l'exécution du processus,
- *basic activity* est une classe abstraite qui désigne les classes de base dans un processus :
  - *receive* désigne l'attente d'un message,
  - *invoke* désigne l'appel à un service,
  - *reply* désigne la réponse à un appel de service,
- *structured activity* est une classe abstraite qui désigne les classes qui permettent de structurer le flux d'exécution de parties de processus :
  - *while* permet l'exécution d'une partie de processus en boucle jusqu'à la satisfaction d'une condition,
  - *pick* permet d'attendre la production d'un événement (arrivée d'un message, etc.) pour continuer l'exécution du processus,
  - *scope* permet de regrouper un ensemble d'activités. Ce regroupement peut avoir ses propres variables et ses propres gestionnaires d'événement,
  - *switch* permet de raisonner sur la valeur d'une variable pour le choix de l'exécution d'une branche d'activités parmi plusieurs,
  - *sequence* permet de lier directement deux parties dans le processus. L'exécution de la deuxième partie suit l'exécution de la première,
  - *flow* permet l'exécution en parallèle de plusieurs activités de processus,
- *event handler* permet de gérer la production d'un événement qui est associé à tout le processus ou uniquement à un *scope*.

A ceci, il convient d'ajouter les connexions entre les différentes vues :

- deux associations in et out entre *business object* et *service* reflètent le fait que les services communiquent entre eux en utilisant des messages contenant des objets métier,

- une association entre *service* et *basic activity* reflète le fait que l'appel à un service est assuré par l'exécution d'une activité de processus,
- une association entre *business object* et *message variable* reflète le fait qu'un objet métier de la vue informationnelle est représenté par une variable dans la vue de processus,
- une association entre *invoke* et *services registry* montre que l'appel d'un service de partenaire doit passer par l'annuaire qui gère les moyens d'accès à ces services.

La conception de ce méta-modèle, basé sur PIM4SOA, a également impliqué la société eBMWebSourcing, concernée par le développement du SIM (sur une base ESB). Ce partenariat a en particulier permis de prendre en compte leur besoin de nécessaire concordance entre le méta-modèle proposé et sa cible technologique à venir.

Dans la partie suivante, nous allons nous intéresser aux mécanismes de correspondance entre ces deux méta-modèles représentatifs des niveaux *métier* et *logique*.

#### 4. EXEMPLE ET OUTILLAGE

Les règles de traduction qui vont être présentées dans cette quatrième partie présentent les mécanismes de passage du niveau métier (processus BPMN) au niveau logique du SI Médiateur (modèle SOA UML). Les règles de traduction expriment les correspondances entre les éléments du modèle de processus et les éléments du modèle de SIM ainsi que les relations entre ces derniers. Ces règles sont structurées de la manière suivante :

- Les *règles de génération de base* sont les règles qu'on applique dans un premier temps pour assurer la génération directe des éléments du modèle de SIC (SOA). Cette première couche de règles définit les correspondances directes (*mapping*) entre les éléments des deux niveaux (BPMN-SIC). Nous présentons dans la suite ces correspondances en utilisant une modélisation graphique adaptée.
- Les *règles de liaison* sont les règles qu'on applique dans un second temps pour lier les éléments déjà créés par les premières règles. Ce deuxième type de règles est plus complexe que le premier. Ce sont des transformations indirectes des relations existantes entre plusieurs éléments du modèle de départ pour obtenir des relations dans le modèle d'arrivée.

##### 4.1 Règles de génération de base

Ces règles peuvent être décrites de plusieurs façons. Nous avons choisi de les structurer selon les trois packages à générer dans le modèle de SIC : le package *Services*, le package *Information* et le package *Processus*.

La figure 8 présente les règles de correspondances entre une sous-partie du méta-modèle de processus (nous retenirons les éléments concernées par ces règles) et les éléments du package *Services* du méta-modèle du SIM. Cinq règles définissent ces correspondances :

- **Rs1** : *CIS task* → *generic service* | *specific service*

Cette règle exprime la déduction des services proposés par le SIM à partir des tâches BPMN. Cette règle qui montre la relation *tâche* → *service*. Un service au sens « SOA » correspond à une fonctionnalité du système. Les services proposés par le SI Médiateur appartiennent à deux catégories : les services spécifiques à une collaboration particulière et les services génériques qui peuvent être utilisés dans plusieurs processus collaboratifs. Le choix d'utiliser une catégorie ou l'autre des services se fait au moment de la conception du processus collaboratif. Une annotation (*specific* ou *generic*) complète les tâches créés.

- **Rs2 : CIS lane** → *services category*  
 Cette règle exprime une organisation des services proposés par le SI Médiateur suivant des catégories de service. Dans ce cas la *CIS Pool* doit être composée de plusieurs *CIS Lane*. Une *CIS Lane* correspond alors à un attribut *service category* de la classe *collaborative service*.
- **Rs3 : partner task** → *partner service*  
 Cette règle ressemble à la règle Rs1 (*CIS task* → *service*) mais elle concerne la déduction des services des partenaires impliqués dans la collaboration. Un service d'un partenaire est représenté dans le processus par une tâche.
- **Rs4 : partner lane** → *entreprise division*  
 Cette règle exprime l'organisation des services selon les partenaires de la collaboration. L'attribut *entreprise division* de la classe *Partner service description* fait référence au partenaire qui détient le service. Il est déduit à partir de l'élément *Partner lane*.
- **Rs5 : services description** → *partner service description*

Cette règle montre le besoin de disposer d'un complément de description des services des partenaires. Cette description concerne l'implémentation des services (adresse, protocole d'accès).

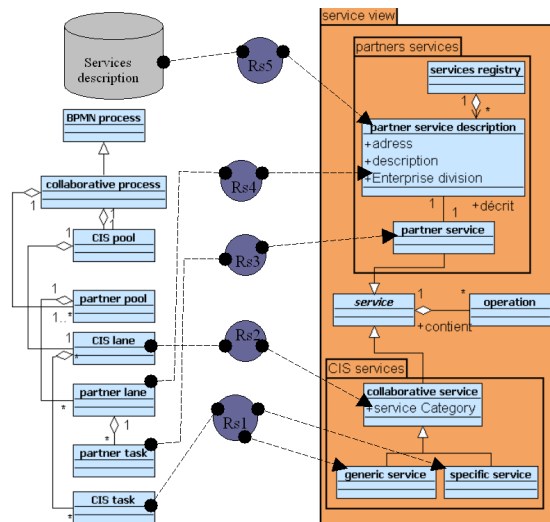


Figure 8. Règles de transformation pour la génération de la vue *service*

La figure suivante (figure 9) montre les règles de correspondances directes entre une sous-partie du méta-modèle de processus (nous retenons les éléments concernées par ces règles) et les éléments du package *Information* du méta-modèle du SIM. Comme nous l'avons vu précédemment, le processus BPMN ne peut pas contenir toute la connaissance nécessaire pour la génération de la vue informationnelle du SI Médiateur. Les règles présentées ci-dessous prennent en compte ce fait en ajoutant une source additionnelle de connaissance : *Information description*. Cette connaissance est nécessaire surtout pour la définition des structures des données utilisées par les services des partenaires (schéma de données).

- **Ri1 : data** → *business object*  
 Cette règle concerne l'élément *data* qui est associé aux éléments *message flow* (tout *message flow* est porteur d'une information *data*) et *sequence flow*. L'élément *data* se traduit par l'élément *business object* du méta-modèle du SIM.
- **Ri2 : Information description** → *format, semantic definition*  
 Cette règle montre le besoin d'une source additionnelle de connaissance sur les informations des partenaires pour déterminer les éléments *format* (structure d'un *business object*) et l'élément *semantic definition* (pour exprimer la sémantique de l'objet).

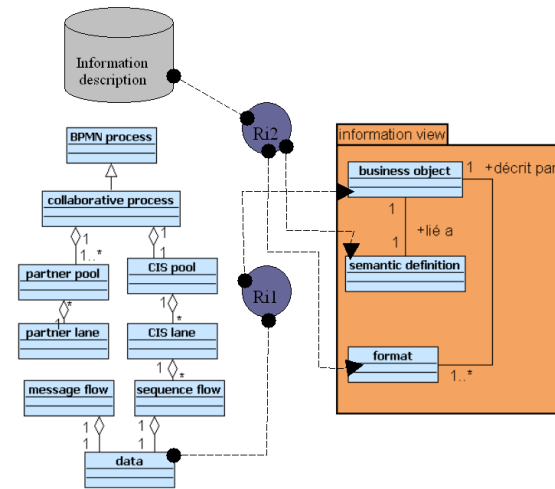


Figure 9. Règles de transformation pour la génération de la vue *information*

La figure suivante (figure 10) présente les règles de correspondance directes entre une sous-partie du méta-modèle de processus (nous retenons les éléments concernés par ces règles) et les éléments du package *Processus* du méta-modèle du SIM. Comme nous l'avons précisé le package *Processus* de ce méta-modèle s'inspire fortement du méta-modèle du langage BPEL qui présente le standard actuel en matière d'exécution de processus composé de services-web. Une partie des règles de transformation décrites dans cette partie sont adaptées d'une part des documents officiels sur le langage BPMN (BPMI, 04) (qui expliquent en partie la traduction vers

BPEL) et d'autre part, des travaux de recherche sur la transformation BPMN - BPEL (Ouyang et al., 06) :

- **Rp1** : *partner lane* → *partner*  
 Cette règle concerne la déduction de l'élément *partner* du modèle du SIM à partir de l'élément *partner pool* du modèle de processus collaboratif.
- **Rp2** : *data* → *message variable*  
 Cette règle permet d'avoir une variable qui présente une donnée échangée dans le processus collaboratif.
- **Rp3** : *sequence flow* → *sequence*  
 L'opérateur *sequence* désigne dans BPEL une succession logique de l'exécution de deux parties de processus. En BPMN un *sequence flow* permet, d'une manière similaire, d'assurer l'enchaînement de deux parties de processus BPMN.
- **Rp4** : *gateway* → *pick* | *flow* | *switch* | *flow* + *switch*  
 Selon le *gateway*, un élément différent est généré :
  - s'il s'agit d'une *parallel gateway* alors c'est une classe *flow* qui est générée pour exprimer une exécution en parallèle des différentes branches liées à la *parallel gateway*,
  - s'il s'agit d'une *data based inclusive gateway* alors c'est une classe *flow* qui est généré, associée avec une classe *switch* pour chacune des branches liées à la *data based inclusive gateway*,
  - s'il s'agit d'une *event based exclusive gateway*, c'est une classe *pick* qui est générée pour exprimer le raisonnement sur la production d'un évènement pour la poursuite de l'exécution du processus,
  - s'il s'agit d'une *data based exclusive gateway* alors c'est une *switch* qui est générée pour exprimer le raisonnement sur la valeur d'une donnée pour la poursuite de l'exécution d'un processus.
- **Rp5** : *message flow* + *intermediate event* → *receive* | *invoke*  
 Une analyse préalable de l'*intermediate event* est nécessaire avant de décider si on génère un *receive* ou un *invoke*. Il faut en effet voir si le *message flow* lié est entrant ou sortant :
  - s'il s'agit d'un *message flow* entrant par rapport à l'*intermediate event* alors on génère un *receive* car il y'a réception d'un nouvel évènement à partir d'un éventuel service.
  - s'il s'agit d'un *message flow* sortant par rapport à l'*intermediate event* alors on génère un *invoke* car il y'a production d'un nouvel évènement vers un autre service.
- **Rp6** : *start event* → *receive*  
 Un *start event* est un évènement qui désigne la réception d'un message pour commencer le processus. Un *start event* se traduit alors par un *receive*.
- **Rp7** : *CIS task* | *partner task* → *basic activity*  
 Les tâches BPMN *partner task* et *CIS task* se transforment naturellement vers des *Basic Activity* dans le méta-modèle du SIC. Selon la valeur type des éléments *partner task* et *CIS task* on décidera de la transformation :
  - Sous la forme d'un *receive* si le type est *Receive*
  - Sous la forme d'un *reply* si le type est *Service*
  - Sous la forme d'un *invoke* si le type est *Send*.

- **Rp8** : *end event* → *invoke*  
 Un *end event* est un évènement qui désigne l'envoi d'un message pour signaler la fin de processus. Un *end event* se traduit alors par un *invoke*.
- **Rp9** : *sub process* → *scope*  
*scope* désigne dans BPEL une partie de processus qui peut être composée d'un ensemble d'éléments : *basic activity*, *gateway*, *message variable*, etc. un sous-processus se transforme alors en *scope* dans le modèle du SIM.

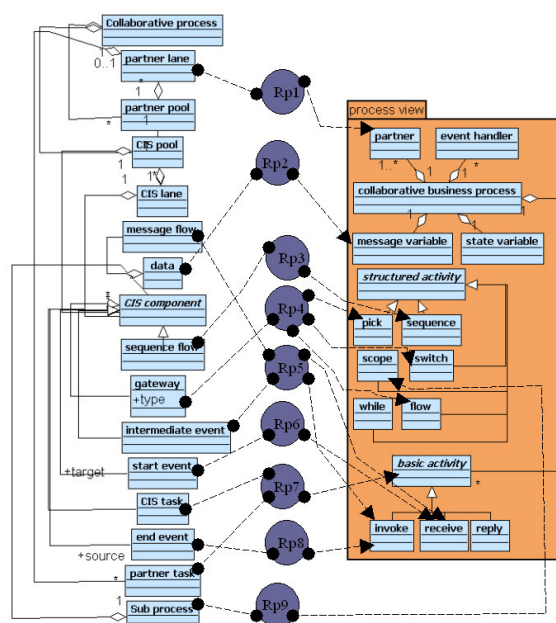


Figure 10. Règles de transformation pour la génération de la vue *processus*

#### 4.2 Règles de liaison

Ces règles interviennent une fois que le modèle cible du système d'information collaboratif est peuplé par des éléments résultant des règles de génération de base. Les règles de liaison ont pour objectif de lier (par des associations UML) des classes du modèles (qu'elles appartiennent à un même package ou à des packages différents). Ces règles illustrent la transformation d'un lien logique de dépendance entre des éléments du modèle de départ (inclusion, liaison, généralisation, spécification, etc.) vers un lien de dépendance entre les éléments du modèle d'arrivée. Avant d'identifier les différentes règles de liaison, nous définissons la fonction  $Y = Equivalent(X, pa)$  ou  $X$  est un élément qui appartient au modèle BPMN,  $Y$  est un élément équivalent qui appartient au modèle du SIM et  $pa$  désigne le package cible de la transformation. Par exemple :

$CIS\ service = Equivalent(CIS\ task, services)$   
 $basic\ activity = Equivalent(CIS\ task, processus)$

Dans la suite, nous définissons l'ensemble des règles de liaison qui permettent de finir la structuration de notre modèle de système d'information collaboratif.

- **RI 1 : gestion des sequence**  
Afin de pouvoir modéliser le séquençement de l'exécution des activités dans la vue processus, cette règle permet de lier un élément *sequence* généré par application de la règle **Rp3** à deux éléments du package processus.
- **RI 2 : gestion de la relation entre service (package service) et business object (package informations)**  
Cette règle permet de lier les *service* de la vue services avec les *business object* de la vue information.
- **RI 3 : gestion de la relation entre basic activity (package processus) et Service (package services)**  
Cette règle permet de lier les deux vues *informations* et *services* en montrant que les interactions entre services sont représentées par le biais d'activités dans la vue processus.

### 4.3 Traitement d'un exemple

A titre d'illustration, nous présentons dans cette dernière sous-section de la partie 4 un exemple de processus collaboratif (correspondant à une plateforme d'achat assisté) et sa traduction en un modèle logique de SI Médiateur qui permettrait d'assurer cette collaboration. Ce résultat a été obtenu par implémentation des règles de traduction au sein d'une maquette de traducteur dont l'architecture technique générale est basée sur trois outils « logiciels libres » qui s'exécutent tous dans l'IDE **Eclipse** : l'outil **Intalio Designer** permet en utilisant une palette de symboles BPMN, la modélisation de processus collaboratif, entrée de notre démarche de transformation. L'outil **Atlas Transformation Language (ATL)** nous permet de récupérer la définition exploitable du modèle de processus (format XML) et de générer un modèle UML du SI Médiateur (fichier .uml), résultat de l'exécution des règles de transformation que nous avons définies et implantées. Enfin, l'outil **TOPCASED** nous permet de visualiser le modèle généré dans un éditeur de diagramme de classe UML et d'intervenir facilement sur ce modèle.

L'exemple de processus est fourni en figure 11.

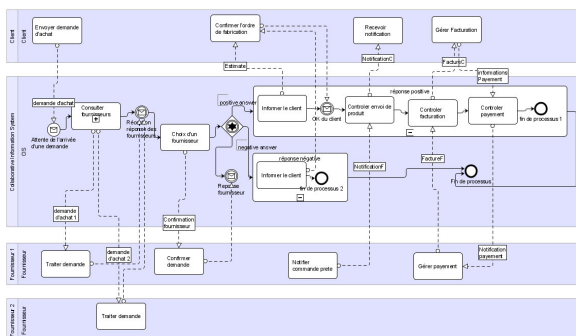


Figure 11. Un exemple de processus collaboratif

Le modèle de SI Médiateur généré par la traduction (via une maquette de traducteur) est présentée en figure 12.

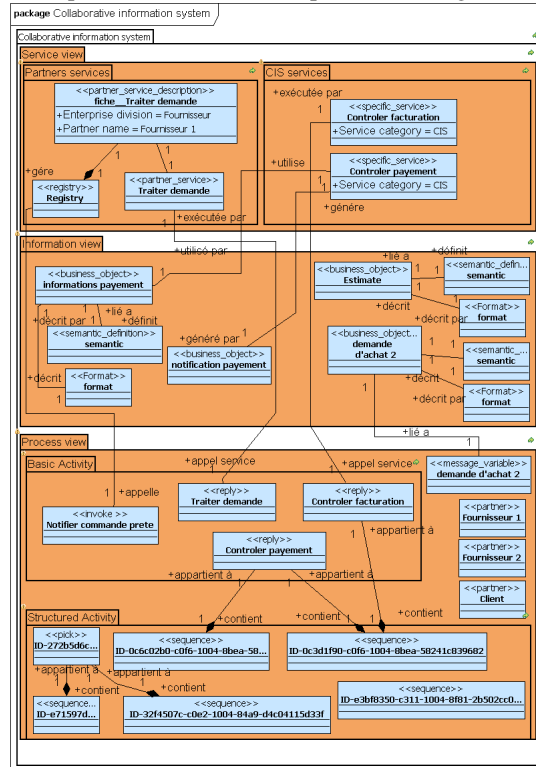


Figure 12. Modèle de SI Médiateur généré par traduction

Nous retrouvons dans cette figure 12 la structure du SI Médiateur respectant une vision SOA. Cependant, pour des raisons de lisibilité, cette figure ne présente qu'une vision partielle du diagramme de classes UML généré : chacun des packages qui composent ce modèle contient en réalité beaucoup plus d'éléments.

Les règles présentées dans cette dernière partie relèvent, dans leur construction, de deux facteurs complémentaires. Le premier concerne la position de pivot du modèle CIMOSA (Kosanke, 1996), sur lequel les différentes vues de la modélisation d'entreprise, reliées entre elles, permettent d'anticiper les relations de correspondances entre constituants de l'entreprise (élément de processus et composant de SI). Le second relève plus empiriquement d'un apport d'expertise qui ne peut être négligé.

## 5. PERSPECTIVES ET CONCLUSION

La question de l'adaptation des entreprises à la nécessaire mise en collaboration avec d'autres partenaires constitue un enjeu majeur de la survie des acteurs industriels. Il nous a semblé pertinent d'aborder cette thématique selon l'angle de l'interopérabilité des systèmes d'information et plus particulièrement, de la **conception d'un SI Médiateur**, capable de supporter la collaboration entre SI des partenaires (pourvu que ceux-ci soient orientés-services).

S'il est clair que ces travaux se positionnent à un niveau syntaxique, prenant finalement peu en compte les notions sémantiques, intrinsèques au concept même de collaboration, il semble pour autant intéressant de proposer un cadre de support de l'interopérabilité de systèmes d'information, au sein duquel les avancées sur le plan de l'interopérabilité « fine » et de la gestion ontologique des collaborations pourront se positionner et évoluer d'une façon stable et presque « sécurisée ».

Comme nous l'avons présenté en partie 3 de cet article, ces travaux se positionnent dans la branche *métier* du « Y » représentatif de l'approche de conception MDA. Plus précisément, il s'agit d'une transition de la couche *métier* vers la couche *logique*. Nous souhaitons revenir sur ce positionnement en discutant en particulier des travaux connexes que nous menons actuellement et qui se positionnent également sur ce « Y ».

Ainsi, la génération du CIM (modèle métier) servant de base à la génération du PIM, constitue également une question cruciale : il est finalement peu courant de rencontrer un groupement d'entreprises souhaitant collaborer qui soient capable de formaliser leur collaboration jusqu'à être en mesure de fournir un modèle du processus collaboratif sur lequel ils souhaitent baser leur réseau. Dans (Rajsiri *et al.*, 07) nous présentons une démarche ontologique de construction de ce type de processus collaboratif (en BPMN et qui respectent le méta-modèle présenté en figure 6) à partir de la connaissance dont le groupe de partenaires dispose sur eux-mêmes et sur la collaboration à venir. Ces travaux contribuent particulièrement aux avancées sur le plan de la prise en compte sémantique du concept de collaboration et s'inscrivent dans la continuité des résultats présentés dans cet article .

De même, la réalisation d'un (ou plusieurs) PM susceptibles de s'interfacer efficacement (lors d'une phase d'urbanisation) avec le PIM proposé par une traduction telle que celle que nous présentons dans cette article, n'est pas une tâche triviale. Le projet **JOnES** (ANR/RNTL2005) s'intéresse à la définition d'une architecture technique d'Enterprise Service Bus (*ESB*) respectant une structure SOA et particulièrement adapté à une convergence vers le type de PIM que notre approche de traduction peut générer.

Enfin, le projet **ISyCri** (ANR/CSOSG2006) s'intéresse à la conception de système d'information médiateur dans un contexte de crise (bénaben *et al.*, 07) et en particulier à la faculté du PSM généré de présenter une faculté **d'agilité** en étant à la fois **réactif** (la définition de ce type de SI Médiateur doit être extrêmement rapide compte-tenu de la situation dramatique) et **flexible** (la situation de crise est par définition instable et il est crucial que le SI Médiateur s'adapte à l'évolution de l'environnement).

L'ensemble de ces travaux doit, à terme, constituer une concrétisation de l'approche MDA dans un contexte de support de l'intégration de partenaires. Cette ambition se base sur notre conviction que l'interopérabilité des SI constitue une clef essentielle à ce verrou de l'intégration mais également sur la « contrainte minimale » que nous avons identifiée : la nature SOA des SI des partenaires. Une étape suivante de nos travaux serait alors de nous intéresser à la façon de permettre aux acteurs de la collaboration de faire apparaître leur SI comme « orientés-services » (en les modifiant selon une démarche à définir ou en les équipant de connecteurs jouant le rôle d'interface SOA). L'unité et la cohérence de nos travaux nous semble donc satisfaisante même si leur incomplétude globale ouvre encore de nombreuses perspectives.

## REFERENCES

- Aubert, B., A. Dussart, 2002. *Systèmes d'Information Inter-Organisationnels*, Rapport Bourgogne du CIRANO.
- Bénaben, F., J.P. Pignon, C. Hanachi, J.P. Lorre, et V. Chapurlat, 2007. Interopérabilité des systèmes en situation de crise, Workshop Interdisciplinaire sur la Sécurité Globale, Troyes.
- Bénaben, F., J. Touzi, V. Rajsiri, H. Pingaud, 2006. Collaborative Information System Design. In *AIM 2006 Information Systems and Collaboration: State of the Art and Perspectives*, GI-Edition Lecture Notes in Informatics, p. 281-296.
- Benguria, G., X. Larrucea, B. Elveseater, T. Neple, A. Beardsmore, M. Friess, 2006. A Platform Independent Model for Service Oriented Architectures. 2<sup>ème</sup> conférence internationale sur l'interopérabilité, IESA'06, Bordeaux, France, mars 2006.
- Booch G., I. Jacobson, J. Rumbaugh, 2004. *UML 2.0 Guide de référence*. CampusPress, Paris.
- BPMI, 2004. *Business Process Modeling Notation (BPMN)*, Version 1.0 - May 3, 2004.
- Konstantas, D., J.-P. Bourrières, M. Léonard, N. Boudjlida, 2005. Preface de : *Interoperability of Enterprise Software and Applications*, Actes de INTEROP-ESA'05, Genève, Suisse, 21-25 février 2005, Springer-Verlag, p. v-vi.
- Kosanke, K., F. B. Vernadat, M. Zelm, 1996. CIMOSA: Industrial Applications of Enterprise Modelling. Actes de ASI'96, Toulouse, 02-06 juin 1996, pp 273-280.
- Maier, M. W., 1998. Architecting principles for systems-of-systems, *Systems Engineering*, Volume 1, Issue 4, Wiley InterScience, p. 267-284.
- Morley, C., J. Hugues, B. Leblanc, 2002. *UML pour l'analyse d'un système d'informations*, 2<sup>ème</sup> édition, Dunod.
- Ouyang, C., W. Van Der Aalst, M. Dumas, A. Hofstede, *Translating BPMN to BPEL*, Technical report - BPM group of Queensland University of Technology Brisbane.
- Rajsiri, V., J.P. Lorre, F. Benaben and H. Pingaud, 2007. Cartography for Designing Collaborative Process, *Enterprise interoperability: New challenges and approaches II*, Springer edition.
- Roques P., 2004. *UML 2 par la pratique*. Eyrolles, Paris.
- Vernadat F., 1996. *Enterprise modelling and integration*, Chapman & Hall.
- Vernadat, F., 2006. Interoperable enterprise systems: architectures and methods. Conférence plénière de INCOM'06, St Etienne, France, 17-19 mai 2006.