



Université  
de Toulouse

# THÈSE

**En vue de l'obtention du  
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE**

**Délivré par :**

Institut National Polytechnique de Toulouse (INP Toulouse)

**Discipline ou spécialité :**

Systèmes Industriels

**Présentée et soutenue par :**

Sébastien Truptil

**le :** lundi 24 janvier 2011

**Titre :**

Etude de l'approche de l'interopérabilité par médiation dans le cadre d'une  
dynamique de collaboration appliquée à la gestion en crise

---

## JURY

Chihab Hanachi	Professeur Université de Toulouse 1	Examineur
Jean-Paul Pignon	Senior Expert Thalès	Examineur
Hervé Pingaud	Professeur Ecole des Mines d'Albi	Directeur de thèse
Frédéric Bénaben	Enseignant-chercheur Ecole des Mines d'Albi	Examineur

---

**Ecole doctorale :**

Systèmes (EDSYS)

**Unité de recherche :**

Centre de Génie Industriel, Université de Toulouse - Mines Albi

**Directeur(s) de Thèse :**

Hervé Pingaud

**Rapporteurs :**

Jean Bézivin, Professeur émérite à l'Université de Nantes

Jean-Pierre Bourey, Professeur Ecole Centrale de Lille







# Remerciements

Trois années viennent de s'écouler et le moins que l'on puisse dire c'est que ça n'a pas été un long fleuve tranquille. Il a y eu des moments heureux et forcément des moments plus difficiles. C'est dans ces périodes là que l'on se rend compte de la chance que l'on a d'être bien entouré tant au niveau professionnel que personnel. A travers ces quelques lignes, je souhaite rendre à ces personnes une partie de l'affection que j'ai reçue.

Par qui commencer si ce n'est les personnes sans qui rien n'aurait commencé : *Frédéric Bénaben* et *Hervé Pingaud*. *Frédéric*, merci de m'avoir convaincu de faire la thèse, merci pour tes conseils tant personnel que professionnel, merci de m'avoir soutenu en toutes circonstances. Au final, je n'ai pas eu le sentiment d'avoir un encadrant mais plutôt le grand frère que je n'ai jamais eu, et je sais que quoi qu'il arrive nous resterons amis et que nous nous retrouverons soit autour d'un MacDo soit avec une canette de coca à la main. *Hervé*, Monsieur le Directeur, je suis toujours impressionné par ta carrure, ton charisme, ta faculté à prendre du recul et surtout par ton humanité. Tu as joué le rôle du père qui donne les grandes lignes à suivre, remet dans le droit chemin, donne des conseils avisés et surtout fait tout son possible dès que quelque chose ne va pas. Même si je ronchonçais à faire certaines choses, je suis heureux d'avoir eu la chance de travailler avec toi et je te souhaite bonne chance dans ta nouvelle carrière. Merci à vous deux.

Je remercie également l'ensemble des membres du centre de Génie Industriel de l'école des Mines d'Albi-Carmaux pour l'ambiance agréable de travail et particulièrement *Jacques Lamothe*, qui m'a rappelé au moins une fois par an qu'il ne faut jamais dire jamais (je ne parlerais pas ici d'un certain repas !), *Isabelle Fournier*, la maman du centre, pour ta disponibilité, ta gentillesse et tes coups de gueules souvent justifiés, mes collègues de bureau : *Jibed*, *Amadou*, *Olina*, *Cedrick*, *Paul*, *Saina* et l'ensemble des doctorants : *Aurélien*, *Meriem*, *Thomas*, *Aurélien*, *Carrine*, ... pour les pauses cafés et les discussions.

Et que serait cette thèse sans le projet ISyCri. Plus précisément, l'ensemble des partenaires du projet qui au final ressemblait plus à une grande famille que des collègues de travail. Merci à tous pour votre convivialité et les moments passés ensemble. J'espère avoir l'occasion de retravailler avec vous. Parmi l'ensemble de ces personnes je souhaite particulièrement remercier : *Nicolas Salatgé* pour la patience dont il a fait preuve lors de ses explications techniques et pour m'avoir fait découvrir deux mois avant la fin du projet que la version du BC-SOAP avait changé et que je devais reprendre une partie du travail ;). *Jean-Paul Pignon* et *Yves Guilloux* pour leur accueil toujours chaleureux et les discussions que nous avons pu avoir sur les solutions envisagées dans le cadre du projet grâce à leurs expériences tant sur les situations de crise que sur la gestion des projets. *Alexandre Blanc* pour ton amitié et le travail réalisé lors de la dernière ligne droite du projet pour réaliser une démonstration convaincante et jolie.

Je souhaiterais remercier mes deux rapporteurs *Jean-Pierre Bourrey* et *Jean Bézivin* pour avoir apprécié mon travail et avoir accepté de l'évaluer pendant les fêtes de fin d'années.

Merci à *Chibab Hanachi*, membre du projet ISyCri, d'avoir accepté d'être mon président de jury et pour l'ensemble des discussions que nous avons pu avoir autour de la flexibilité de workflow.

Que serait cette thèse sans les précieux moments passés en dehors du travail entre Albi, Orléans et Le Havre ?

Tout d'abord je tiens à remercier de tout mon cœur mes parents *Philippe* et *Francine* sans qui je n'aurais pu soutenir cette thèse. Merci pour votre soutien, pour le temps que vous avez accordé à mon éducation et pour m'avoir poussé à faire des études même si ça voulait dire laisser votre fils unique partir à 900km de chez vous. En tout cas, on peut dire que votre menace « fais des études ou l'on fait de toi un fleuriste » a fonctionné !!! ;)

Aurait-il été possible de ne pas citer la « coloc » ? Bien sûr que non. *Guy, Nioc, Remy, Chris, Damien* qu'aurait été ma vie albigeoise au quotidien sans vous ? Sans les soirées PES, les barbecues, les soirées poker, la piscine, le jardin et le panier de légumes, les apéritifs chez les voisins, ... enfin bref tout ce qui a rendu son départ difficile. Je tiens aussi à remercier les « demi-colocs » *Eric, Gillou, Laura, Vanessa, Nicolas D, Cédric, Pauline, Antho* pour l'ensemble des soirées que l'on a pu faire et dont certains déguisements resteront mythiques.

Que serait la vie albigeoise sans le quizz du mardi soir au sully et les foots du lundi et mercredi midi ? Un grand merci aux colocs, demi-colocs, les *Oliviers, Philippe, Arezky, Souleymane, les Auréliens* d'avoir partagé ces moments ensemble.

Comme on dit « last but not least », un grand merci à *Claire*. Tu m'as soutenu, aidé, parfois fait comprendre que j'avais tort, poussé dans mes derniers retranchements, souligné mes contradictions mais surtout tu as fait ton maximum pour que je sois bien. Tu es la reine de l'organisation, notamment pour les vacances ;). Vivement les prochaines étapes de la vie que l'on traversera et découvrira ensemble. En parlant de découverte, je voulais aussi te remercier de m'avoir fait découvrir avec l'aide de tes parents, *Claude<sup>2</sup>*, que je remercie chaleureusement, le Lot et les spécialités culinaires associées. Bien que n'ayant pas réussi à sortir le mot friton pendant ma soutenance, j'espère que je pourrais toujours en goûter ;)

Un grand MERCI à tous et à tous ceux que j'ai oublié. Et pour les nouveaux doctorants : *Nicolas BD, Olina, Anne-Marie, Sarah*, je voudrais vous donner le même conseil que m'a donné Cédric : « Quand vous verrez la lumière au bout du tunnel, priez pour que ce ne soit pas le train (Daniel Lemire) »

# Sommaire

<b>Introduction générale</b>	<b>1</b>
------------------------------	----------

## Chapitre I :

<b>Contexte et problématique</b>	<b>7</b>
----------------------------------	----------

I. La notion de crise.....	- 8 -
II. La gestion de crise.....	- 9 -
II.1 Vue d'ensemble de la gestion de crise.....	- 9 -
II.2 La réponse à la crise : caractéristiques et faiblesses.....	- 9 -
III. Problématique générale.....	- 12 -
III.1 Vers une problématique d'interopérabilité.....	- 12 -
III.2 Une approche basée sur la médiation.....	- 12 -
III.3 Vers une démarche de conception d'un médiateur adapté à la situation.....	- 14 -
IV. Concepts supports de l'ingénierie dirigée par les modèles.....	- 15 -
IV.1 Cellule de crise et notion de système de systèmes.....	- 15 -
IV.2 Le processus collaboratif de gestion de crise.....	- 16 -
IV.3 SoS et système de systèmes d'information.....	- 17 -
IV.3.1 Notion de système d'information.....	- 17 -
IV.3.2 Du médiateur au système d'information de médiation.....	- 18 -
V. Conception de système d'information de médiation.....	- 19 -
V.1 Interopérabilité dirigée par les Modèles.....	- 20 -
V.2 Mécanisme de transformation de modèles.....	- 21 -
V.3 Vue d'ensemble du projet MISE.....	- 22 -
V.4 Le projet MISE dans un contexte de réponse à une crise.....	- 24 -
VI. Le projet ISyCri.....	- 26 -
VI.1 L'axe conception du projet ISyCri.....	- 26 -
VI.2 L'axe exécution du projet ISyCri.....	- 27 -
VI.3 Synthèse sur le projet ISyCri.....	- 29 -
VII. Conclusion.....	- 30 -

## Chapitre II :

<b>De la situation de crise au processus de réponse</b>	<b>31</b>
---	-----------

Introduction du chapitre.....	- 32 -
I. Métamodèle de crise.....	- 33 -

I.1 La situation .....	- 33 -
I.1.1 Le système d'étude.....	- 34 -
I.1.2 Les caractéristiques de la crise.....	- 35 -
I.2 Le système de traitement .....	- 36 -
I.3 Le processus collaboratif.....	- 37 -
I.4 Liens entre les différentes parties du métamodèle.....	- 39 -
II. Démarche de déduction du processus collaboratif .....	- 40 -
II.1 Etape (2) : confrontation des différents modèles.....	- 42 -
II.2 Règles de transformation .....	- 43 -
II.2.1 Règles permettant de Propager les Liens Sémantiques .....	- 43 -
II.2.2 Règles permettant la Déduction des Services Utilisables.....	- 45 -
II.3 Etape (3) : construction du processus collaboratif.....	- 47 -
II.3.1 Récupération des choix de la cellule de crise.....	- 48 -
II.3.2 Construction du processus .....	- 49 -
III. Illustration de déduction d'un processus collaboratif .....	- 57 -
III.1 Présentation du contexte .....	- 58 -
III.2 Déduction des composants du processus collaboratif.....	- 60 -
III.3 Déduction de l'ordre d'exécution des services .....	- 63 -
IV. Conclusion .....	- 66 -

## Chapitre III :

### Du modèle technique de médiation au Déploiement 69

Introduction du chapitre.....	- 70 -
I. Obtention de l'architecture technique du SIM.....	- 75 -
I.1 Une double transformation de modèle.....	- 75 -
I.2 Métamodèle d'architecture technique .....	- 76 -
II. Création des SU et SA nécessaires au déploiement.....	- 79 -
III. Création du SU-BPEL.....	- 81 -
III.1 Le langage BPEL .....	- 82 -
III.2 Transformations de modèles .....	- 85 -
IV. Conclusion .....	- 94 -

## Chapitre IV :

### L'agilité du médiateur 97

Introduction du chapitre.....	- 98 -
I. Présentation de la notion d'agilité .....	- 99 -
I.1 Définition de l'agilité.....	- 99 -
I.2 Apporter de l'agilité aux processus .....	- 99 -
I.3 Apport d'agilité au SIM.....	- 100 -
II. Détection de variations .....	- 101 -



II.1 Présentation de la fiche de suivi .....	- 102 -
II.2 Vers un suivi en temps réel de la réponse à la crise .....	- 104 -
II.3 Traitement de l'information issue de la fiche de suivi .....	- 106 -
III.Réagir suite à une évolution .....	- 107 -
III.1 Agilité du processus de conception.....	- 107 -
III.1.1 Le processus de conception nominal.....	- 108 -
III.1.2 Processus de conception suite à une évolution de la cellule de gestion de crise.....	- 108 -
III.1.3 Processus de conception suite à un dysfonctionnement. ....	- 109 -
III.2 Réalisation d'une chaîne de conception flexible .....	- 110 -
III.2.1 Support technologique de cette flexibilité.....	- 110 -
III.2.2 Technologies utilisées.....	- 111 -
III.2.3 Identification des WebServices de conception.....	- 112 -
IV.Conclusion .....	- 114 -

## Chapitre V :

### Exemple de réponse à la crise 117

Introduction.....	- 118 -
I. Conception du SoS à t0 .....	- 118 -
I.1 Création du modèle de crise .....	- 118 -
I.2 Définitions des liens de correspondance sémantique .....	- 119 -
I.3 Priorité de traitement et choix des services .....	- 121 -
I.4 Déduction du processus collaboratif de réponse .....	- 127 -
I.5 Déduction de l'architecture technique du SIM.....	- 128 -
I.6 Déduction du processus au format BPEL.....	- 129 -
II. Mise en évidence de l'agilité au travers d'évolutions .....	- 130 -
II.1 Variation de la composition de la cellule à t1. ....	- 130 -
II.2 Evolution de la situation à t2. ....	- 130 -
III.Conclusion .....	- 133 -

### Conclusion générale 135

### Glossaire 141

### Références Bibliographiques 143

### Annexes 153

# Liste des illustrations

## Chapitre I :

Figure I- 1 : Structure d'une cellule de gestion de crise .....	11 -
Figure I- 2 : Plusieurs formes d'architecture du système de systèmes .....	14 -
Figure I- 3 : définition d'un système d'information selon (Morley, 2002).....	18 -
Figure I- 4 : Vue d'ensemble d'un système d'information de médiation (Bénaben et al., 2008).....	19 -
Figure I- 5 : modèle de référence d'une démarche MDI (Bourey et al., 2007) .....	20 -
Figure I- 6 : Principes de transformation de modèles (Truptil et al., 2010a) .....	21 -
Figure I- 7 : place du médiateur au sein de la collaboration. ....	22 -
Figure I- 8 : Vue d'ensemble du projet MISE.....	23 -
Figure I- 9 : Problèmes de correspondance entre le niveau logique et technique du projet MISE.....	25 -
Figure I- 10 : Vue d'ensemble du projet ISyCri.....	29 -

## Chapitre II :

Figure II- 1 : positionnement du chapitre vis-à-vis de l'approche de conception d'un système d'information de médiation - 32 -	
Figure II- 2 : modèle générique du risque (Sienou, 2009) .....	34 -
Figure II- 3 : parties système d'étude et caractéristiques de crise du métamodèle .....	35 -
Figure II- 4 : partie système de traitement du métamodèle de crise.....	37 -
Figure II- 5 : métamodèle de processus collaboratif.....	38 -
Figure II- 6 : lien entre SIM_task et Partner_task .....	39 -
Figure II- 7 : métamodèle de crise .....	40 -
Figure II- 8 : processus de transformation dans le but d'obtenir le processus collaboratif.....	42 -
Figure II- 9 : exemple de propagation de lien SameAs par transitivité.....	44 -
Figure II- 10 : exemple de propagation de lien Near par transitivité .....	45 -
Figure II- 11 : illustration des règles de déduction de sélection des services utilisables.....	47 -
Figure II- 12 : interface de choix pour la cellule de crise .....	49 -
Figure II- 13 : exemple de résultat de la fonction TrouverFinServicePreRequis(S1) .....	54 -
Figure II- 14 : modèle de crise.....	58 -
Figure II- 15 : résultats des règles II-10 et II-11.....	61 -
Figure II- 16 : résultat de la règle II-12 .....	61 -
Figure II- 17 : résultat suite aux règles II-13 et II-14.....	63 -
Figure II- 18 : processus collaboratif suite à l'algorithme Initialisation_INI_FIN .....	65 -
Figure II- 19 : processus Collaboratif obtenu à la fin de la démarche .....	66 -
Règles II- 1 : propagation du lien SameAs par transitivité.....	43 -
Règles II- 2 : propagation du lien Near aux instances équivalentes.....	44 -
Règles II- 3 : propagation du lien Near par transitivité .....	44 -
Règles II- 4 : déduction des risques présents en fonction des caractéristiques.....	46 -
Règles II- 5 : déduction des services interdits par un élément de la crise.....	46 -

Règles II- 6 : déduction des services permettant de réduire les conséquences de la crise .....	46 -
Règles II- 7 : déduction des services qui pourraient réduire des conséquences de la crise.....	46 -
Règles II- 8 : déduction des services permettant de prévenir les risques de la crise .....	47 -
Règles II- 9 : déduction des services qui pourraient prévenir des risques de la crise.....	47 -
Règles II- 10 : création des éléments obligatoires du processus .....	50 -
Règles II- 11 : création des partner pools et partner lanes.....	50 -
Règles II- 12 : création des Partner_tasks et SIM_tasks.....	50 -
Règles II- 13 : création des éléments du processus liés à un pré-requis d'un service .....	51 -
Règles II- 14 : création des éléments du processus liés à un pré-requis d'un service via les conditions..	51 -
Règles II- 15 : parallélisme des services, produit par des partenaires différents, ne traitant pas le même problème n'impactant pas le même composant système d'étude .....	52 -
Règles II- 16 : parallélisme des services sélectionnés pour traiter le même problème.....	52 -
Règles II- 17 : parallélisme des services nécessités par un autre .....	52 -
Tableau II- 1 : tableau de description des services des gendarmes.....	59 -
Tableau II- 2 : tableau de description des services de la Croix Rouge.....	59 -
Tableau II- 3 : tableau de description des services du SAMU.....	60 -
Tableau II- 4 : tableau de description des services du SDIS.....	60 -
Tableau II- 5 : déroulement de l'algorithme Initialisation_INI_FIN.....	64 -

## Chapitre III :

Figure III- 1 : positionnement du chapitre vis-à-vis de l'approche de conception d'un système d'information de médiation.....	71 -
Figure III- 2 : exemple de relation entre les différents éléments d'un ESB basé sur la norme JBI.....	73 -
Figure III- 3 : description de la transformation de modèle.....	76 -
Figure III- 4 : métamodèle d'architecture technique.....	79 -
Figure III- 5 : vue d'ensemble de la démarche de conception des SAs et SUs.....	81 -
Figure III- 6 : extrait du métamodèle BPEL réalisé à partir de (OASIS, 2007).....	84 -
Figure III- 7 : sous-partie du processus collaboratif déduit au chapitre II.....	85 -
Figure III- 8 : principe de gestion de l'information au sein du BPEL .....	89 -
Figure III- 9 : exemple de processus collaboratif .....	90 -
Figure III- 10 : structure d'un flow en BPEL .....	91 -
Figure III- 11 : exemple de transformation de sous-séquences en BPEL.....	94 -
Tableau III- 1 : lien entre les différents éléments du bus.....	72 -
Tableau III- 2 : description de la structure des variables.....	86 -
Tableau III- 3 : illustration de la condition de recherche de l'élément final d'une sous-séquence .....	92 -
Algorithme III- 1 : création des sous-séquences .....	93 -

## Chapitre IV :

Figure IV- 1 : différents types de flexibilité selon (Schonenberg et al., 2008). .....	100 -
Figure IV- 2 : relations entre variations et approches de flexibilité (Truptil et al., 2010).....	101 -
Figure IV- 3 : exemple d'utilisation de la fiche de suivi.....	103 -
Figure IV- 4 : illustration de la limite d'utilisation du service de médiation seul.....	105 -
Figure IV- 5 : Transformation du processus collaboratif dans le but d'obtenir une mise à jour en temps réel du statut des services. ....	105 -
Figure IV- 6 : démarche de conception suite à une évolution de la situation. ....	108 -
Figure IV- 7 : déroulement de la démarche suite à une évolution de la cellule de crise.....	109 -
Figure IV- 8 : déroulement de la démarche suite à une évolution de dysfonctionnement. ....	109 -
Figure IV- 9 : architecture de la chaîne de conception.....	110 -
Figure IV- 10 : ensemble des technologies utilisées dans le cadre de la démarche .....	112 -
Figure IV- 11 : positionnement des travaux du projet sur le repère de (Schonenberg et al, 2008) .....	115 -
Figure IV- 12 : illustration d'un déroulement de réponse à la crise vue à travers l'ESB.....	116 -
Tableau IV- 1 : Eléments de la fiche de suivi permettant de déduire un type d'évolution.....	106 -
Tableau IV- 2 : Description des WebServices réalisés.....	113 -

## Chapitre V :

Figure V- 1 : modèle de crise à l'instant t0 .....	119 -
Figure V- 2 : tableau de correspondance sémantique des instances de risque.....	120 -
Figure V- 3 : tableau de correspondance sémantique des instances de conséquence.....	120 -
Figure V- 4 : tableau de correspondance sémantique des instances de condition.....	121 -
Figure V- 5 : définition de l'ordre de priorité de traitement à t0.....	122 -
Figure V- 6 : choix des services pour le risque de contamination .....	123 -
Figure V- 7 : état de la première interface après sélection des services pour le risque contamination ..	124 -
Figure V- 8 : choix des services à utiliser pour traiter la conséquence de personnes blessées .....	124 -
Figure V- 9 : la première interface après la sélection des services pour les personnes blessées.....	125 -
Figure V- 10 : choix des services à utiliser pour traiter la conséquence d'incendie réservoir .....	125 -
Figure V- 11: état de la première interface après la sélection des services pour l'incendie moteur.....	126 -
Figure V- 12 : des services à utiliser pour traiter la conséquence d'engorgement .....	127 -
Figure V- 13 : processus collaboratif de réponse déduit à t0.....	128 -
Figure V- 14 : Modèle d'architecture technique déduit .....	129 -
Figure V- 15 : fichier BPEL correspondant au processus de réponse à t0 .....	129 -
Figure V- 16 : état des services à t2 .....	131 -
Figure V- 17 : état des problèmes à t2.....	131 -
Figure V- 18 : état des acteurs de la réponse à t2 .....	131 -
Figure V- 19 : modèle de crise à t2 .....	132 -
Figure V- 20 : processus de réponse à t2.....	133 -

# Introduction générale

Les répercussions de l'évolution de l'écosystème économique mondial ont obligé et obligent encore les organisations à adopter de nouveaux schémas de comportement, à modifier en profondeur leur structure et à s'ouvrir davantage à leur environnement. Ces répercussions nécessitent une nouvelle philosophie de collaboration de la part des organisations. Ces collaborations peuvent être caractérisées par de nombreux facteurs, tels que la nature des relations entre les partenaires, la topologie du réseau constitué (chaîne, étoile, point à point) ou encore l'objectif de la collaboration. A partir de l'ensemble de ces caractéristiques, cette nouvelle philosophie a un réel impact d'une part, sur la fréquence de la collaboration et d'autre part, sur la durée de mise en place et de validité de la collaboration. En effet, aujourd'hui les organisations ne collaborent plus nécessairement dans une optique de long terme, elles cherchent des moyens pour collaborer ponctuellement sur des horizons de temps moins longs en fonction des opportunités des marchés. La durée de collaboration étant réduite, les organisations se doivent d'acquérir des capacités d'adaptation et de réactivité toujours plus importantes. Ces capacités sont devenues aujourd'hui des facteurs clés de succès car elles soulignent la possibilité pour une organisation de répondre rapidement aux évolutions du marché.

Dans (Touzi, 2007), à partir des niveaux de collaboration proposés par (Bouzguenda, 2006), l'auteur propose des niveaux de maturité collaborative :

- **communicante** : capacité à échanger et à partager des informations. Ce niveau correspond à un échange simple de données entre les organisations. Il est donc nécessaire que les organisations disposent d'interfaces de communication élémentaires.
- **Ouverte** : capacité de partage des fonctionnalités et des services. Ce niveau correspond au partage et à la coordination des tâches. Les organisations réalisent des tâches dont dépendent leurs partenaires et réciproquement. Les organisations doivent donc pouvoir ouvrir leurs compétences vers l'extérieur et vers leurs partenaires potentiels, mais elles doivent également pouvoir accéder aux fonctionnalités externes qui seraient mises à leur disposition (Pingaud, 2003).
- **Fédérée** : capacité à travailler selon un comportement collectif. Les organisations poursuivent un objectif commun. Ce stade nécessite de l'organisation, la maîtrise de ses propres processus afin de pouvoir s'intégrer au processus collaboratif (il existe ici une connexion naturelle vers les certifications de type *ISO* ou *Capacity Maturity Model*).
- **Interopérable** : capacité d'appartenance transparente à une même entité (réelle ou virtuelle). Elle correspond à une aptitude à s'immerger au sein d'une entité plus vaste et à en devenir un composant actif.

Ces niveaux de maturité collaborative permettent de caractériser le niveau d'adaptation des organisations au sein d'une collaboration. Selon le niveau de maturité collaborative, certaines solutions technologiques sont requises. Le premier niveau (Communicante) est outillé par des moyens de communication tels que Internet, EDI (Electronic Data Interchange), etc. Les deuxième et troisième niveaux (Ouverte et Fédérée) peuvent être assurés par différentes approches telles que l'EAI (Enterprise Application Integration) ou encore la vision SOA (Service-Oriented Architecture). Le quatrième niveau (Interopérable) nécessite que les partenaires puissent répondre à des exigences d'interopérabilité que nous allons maintenant détailler.

L'interopérabilité est définie comme « la capacité pour un système d'échanger de l'information et des services dans un environnement technologique et organisationnel hétérogène » (IEEE, 1990). Même si nous verrons que d'autres définitions de l'interopérabilité permettent d'affiner le concept et de rentrer plus en avant dans les exigences qu'il impose, cette première vision permet avant tout de mettre évidence la

notion d'hétérogénéité. Cette hétérogénéité implique la présence de barrières à la mise en place d'une collaboration. Ces barrières, listées par (Chen *et al.*, 2008), peuvent être conceptuelles, technologiques ou organisationnelles. Afin de s'affranchir de ces barrières, différentes approches de mise en place de l'interopérabilité existent :

- L'approche **intégrée** qui correspond à l'utilisation par tous les tiers (partenaires, outils,...) du même format de données. Ce format n'est pas obligatoirement un standard mais il doit être connu et accepté par toutes les parties afin de créer et d'adapter les différents systèmes en jeu.
- L'approche **unifiée** qui correspond à la définition d'un format commun servant de modèle aux différents formats de données des partenaires. Ce format de données n'est pas utilisé directement mais permet de faire les correspondances entre tous les formats de données utilisés au sein du réseau collaboratif et constitue par là-même un pivot.
- L'approche **fédérée** qui signifie qu'aucun format de donnée n'est imposé. Chaque partenaire utilise alors ses propres méthodes de travail et l'interopérabilité est réalisée à la volée. Seule une base de connaissances définissant un vocabulaire de référence est partagée entre les tiers.

La première approche (intégrée) pouvant aller jusqu'à la refonte des organisations, peut être vue comme « totalitaire ». Cette approche n'est pas compatible avec l'idée de réactivité de la mise en place de la collaboration, contrairement aux deux autres approches. Dans le cas de ces deux autres approches les organisations restent indépendantes au niveau opérationnel et managérial puisque dans l'approche fédérée les organisations travaillent « à leur façon », alors que dans l'approche unifiée, elles doivent juste s'adapter au format de données de référence.

Si nous résumons ce qui qualifie un réseau d'organisations interopérables, il est important de retenir les indépendances managériales et opérationnelles, ainsi qu'un comportement émergent du réseau constitué d'organisations éventuellement distribuées et géographiquement réparties, et enfin, un développement évolutif suite à l'évolution des collaborations inter-organisations. Ces cinq critères correspondent à la définition d'un système de systèmes proposée par (Maier, 1998). Par conséquent, une organisation interopérable peut être vue comme une organisation capable de rejoindre un système de systèmes.

La question portant sur la capacité des organisations à être interopérable impacte nécessairement le domaine des Systèmes d'Information (SI). Même s'il existe de nombreuses définitions du concept de « système d'information », ce système peut être considéré comme un ensemble d'interactions entre processus, données et applications de l'organisation. Dans ce cas, le comportement, la réactivité et la dynamique de l'organisation reposent sur le système d'information au travers des processus qu'il supporte, des applications qu'il propose et des données qu'il gère. Par conséquent, il existe une forte dépendance entre l'organisation et son système d'information, nous considérons que la capacité d'une organisation à rejoindre un système de systèmes passe par celle de son système d'information.

Partant de ce constat, un axe de recherche a vu le jour avec pour ambition d'assister la définition d'un système de systèmes, puis de supporter la collaboration entre les différents partenaires. Afin de rester cohérent avec la notion de système de systèmes, le support de la collaboration se base d'une part sur les systèmes d'information tels qu'ils existent, et d'autre part sur une passerelle entre les différents systèmes d'information des partenaires. Cette passerelle fournit l'interopérabilité au moment où il y en a besoin, en traitant l'hétérogénéité des données, processus et applications des SI.

Cette ambition a donné naissance, en 2004, au projet MISE (Mediation Information System Engineering). Ce projet a été construit autour de deux notions clés : (i) l'émergence d'approches d'architectures de systèmes d'information, telles que SOA (*Service-Oriented Architecture*), qui apportent une

certaines simplifications des interconnexions entre applications et la facilitation de l'établissement et de la reconfiguration de processus. (ii) La notion de médiateur, promue par Wielderhold (Wiederhold, 1992), qui remplit les conditions de la passerelle désirée. Par conséquent, le but du projet MISE est de créer à partir des informations de la collaboration, un système d'information de médiation (SIM) support d'un médiateur, reposant sur les principes SOA. Ce SIM est en charge de coordonner les actions des partenaires ainsi que de gérer l'information au sein du système de systèmes.

La création du SIM repose sur une approche dirigée par les modèles (MDE) (Miller *et al.*, 2003) et consiste en quatre niveaux : le niveau métier (CIM, pour Computation Independent Model), le niveau logique (PIM, pour Platform Independent Model), le niveau technique (PSM pour Platform Specific Model) et enfin le niveau de déploiement de la solution. Le niveau CIM concerne les organisations, les objectifs, les processus et les responsabilités des partenaires. Il nécessite, au moins dans le cadre de l'approche MISE, un modèle décrivant les interactions entre les partenaires. Le niveau PIM correspond à un modèle d'architecture logique indépendante de toute solution technologique. Le modèle PSM quant à lui est la projection du modèle PIM sur une solution technologique. Enfin, le niveau de déploiement consiste à réaliser le SIM à partir des informations des étapes précédentes.

Si les niveaux PIM, PSM et de déploiement sont indépendants du domaine métier spécifique de la collaboration, ce n'est pas le cas du niveau CIM. C'est pourquoi le projet MISE a donné naissance à deux sous-projets, l'un appliqué à la collaboration inter-entreprises et l'autre appliqué au domaine de la gestion de crise. Depuis le début de cette introduction, nous avons souligné l'adéquation de ces travaux pour la collaboration d'organisations, elle est encore plus évidente dans le cas où la collaboration concerne plutôt des organisations en charge de répondre à une situation de crise. Une crise est par définition un phénomène évolutif et peut s'avérer géographiquement dispersée. Les partenaires de la réponse de crise ne collaborent que durant le laps de temps de la réponse et doivent par conséquent garder leur indépendance. Cependant, cette indépendance ne doit pas être au détriment de la cohérence globale de la réponse. Une réponse à la crise correspond parfaitement à une vision orientée système de systèmes.

Une approche MDE étant basée sur la plongée en abstraction, elle consiste à définir les transformations de modèle permettant de passer d'un niveau à un autre. Dans le contexte du projet MISE, les travaux liés à cette approche ont commencé avec la thèse de J. Touzi (Touzi, 2007) qui s'intéressait au passage du niveau CIM au niveau PIM. Ces travaux portaient du principe que les partenaires pourraient fournir le modèle CIM décrivant leur collaboration (dans le cas de MISE, il s'agit d'un modèle de processus collaboratif). Malheureusement, cette hypothèse s'est avérée trop ambitieuse en pratique (en particulier du fait des exigences portant sur le modèle CIM afin d'être exploitable pour la transformation de modèle au niveau logique). La thèse de V. Rajsiri (Rajsiri, 2009) s'est donc intéressée au niveau métier du projet MISE (dans un contexte inter-entreprises) en définissant un système de base de connaissances permettant aux partenaires de décrire la collaboration souhaitée afin d'obtenir automatiquement le modèle CIM. Le passage du PIM au PSM a été réalisé dans le cadre du master de W. Mu. Enfin le passage du PSM au déploiement a été en partie réalisé par A.M. Barthe dans le cadre d'une collaboration avec EBM WebSourcing. Les travaux de cette thèse viennent conclure la version 1.0 du projet MISE. Ils ambitionnent tout d'abord de définir une démarche complémentaire permettant l'obtention d'un modèle CIM spécifique à une réponse à une situation de crise (travaux réalisés dans le cadre du projet ANR ISyCri), d'autre part, ces travaux visent également à compléter, homogénéiser et outiller la phase de déploiement en y adjoignant la nécessaire agilité incontournable dans un contexte de gestion de crise. Ces trois composantes (modèle CIM d'une gestion de crise, outillage du déploiement et prise en charge de l'agilité du système de système obtenu) constituent les différentes briques des travaux de thèse que ce manuscrit propose de décrire.



Les considérations précédentes, portant sur l'évolution des collaborations ainsi que sur les caractéristiques des organisations selon le niveau de maturité de collaboration, nous ont amené à introduire le projet MISE, en tant que premier support des travaux dont il est ici question. Néanmoins, au delà de son inscription dans la dynamique MISE, cette thèse constitue la dorsale du projet ISyCri (projet ANR relevant de l'appel CSOSG – Concept, Systèmes et Outils pour la Sécurité Globale - ANR/2006/SECU/006), nous avons pris le parti d'expliquer l'ensemble des travaux dans le cadre d'une réponse à une situation de crise (même si nombreux éléments peuvent être appliqués à la collaboration inter-entreprises). Enfin, ces travaux étant conceptuellement liés à la vision MDE nous avons décidé d'organiser ce manuscrit de façon non conventionnelle en ne proposant pas de chapitre bibliographique à proprement parler mais une étude bibliographique diffuse au sein des différents chapitres avec certain complément en annexe. Par conséquent le manuscrit est organisé de la façon suivante :

- le chapitre I présente la problématique et donne une vision d'ensemble des travaux réalisés dans le cadre de cette thèse. Ce chapitre s'intéresse tout d'abord au contexte de l'étude, à savoir la gestion de crise. Nous reviendrons donc sur l'adéquation de l'utilisation d'un ou plusieurs médiateurs et de systèmes d'information de médiation pour la réponse à la crise. Suite à ces explications, nous détaillerons la démarche proposée par le projet MISE et soulignerons ses limites pour le contexte de réponse à une situation de crise. S'en suivra alors la présentation du projet ISyCri, qui a vocation à gommer ces limites. Enfin ce chapitre permet de positionner finement les travaux de cette thèse vis-à-vis des différents projets (passé ou en cours).
- Le chapitre II présente la première étape du projet ISyCri (*i.e.* la première brique de nos travaux), la déduction du modèle CIM en fonction des informations de la situation. Cette première étape consiste à aider la cellule de gestion de crise à créer le processus collaboratif de réponse à la crise. Cette création se décompose en plusieurs actions, certaines manuelles d'autres automatisées. Ce chapitre se conclut sur la présentation d'un exemple de déduction de réponse à une crise.
- Le chapitre III présente le passage du PSM au déploiement opérationnel du médiateur (en se basant sur la plateforme technologique choisie, l'ESB PEtALS). Après avoir donné une vue d'ensemble de cette étape, nous nous focaliserons sur la transformation de modèles permettant d'obtenir le fichier BPEL destiné à l'orchestration du workflow collaboratif. Ces éléments relèvent de la deuxième brique de nos travaux.
- Le chapitre IV présente la troisième brique, en s'intéressant à la notion d'agilité et à sa prise en compte lors de la définition du SIM (afin d'obtenir un SIM agile). En effet, que ce soit dans le domaine de la collaboration d'entreprises ou dans celui de la gestion de situation de crise, le système de systèmes sera amené à être modifiable en fonction des évolutions probables de la situation. Or, la démarche MDE ne permet que la seule définition d'un système de systèmes spécifiquement dédié à un instant  $t$  de la collaboration. Au cours de ces travaux de thèse, nous nous sommes donc intéressés à gommer les frontières entre *Design-Time* et *Run-Time* afin de permettre au médiateur (et donc au système de systèmes) de demeurer en permanence le plus cohérent possible avec la situation fluctuante. Ce chapitre explique tout d'abord notre positionnement vis-à-vis de la notion d'agilité, puis il met en avant les différents travaux réalisés dans le but d'obtenir cette capacité.
- Le chapitre V donne rendez vous au lecteur afin de lui présenter, du point de vue de la cellule de gestion de crise, l'utilisation de la démarche de conception du SIM à travers un exemple

basé sur un exercice réalisé en 2004 par la préfecture du Tarn. Cet exercice simule la collision entre un camion citerne et un train de voyageur sur un passage à niveaux à proximité de Marssac-sur-Tarn (81). En plus des voyageurs blessés et du début d'incendie du camion citerne, la cellule de gestion de crise craint de devoir faire face à une crise de type NRBC<sup>1</sup> puisque les panonceaux de signalisation de la citerne du camion ne sont pas conformes à la réglementation. C'est à partir de ces informations que la cellule de gestion de crise va devoir définir en urgence une réponse à la crise à partir des compétences des organisations disponibles, à savoir le SDIS<sup>2</sup>, la gendarmerie et la DDE<sup>3</sup>. Cependant, cette définition de la réponse à la crise sera remise en cause dans un premier temps, suite à l'arrivée du SAMU<sup>4</sup> en tant que partenaire actif de la cellule de gestion de crise puis dans un second temps, par la propagation de l'incendie suite à l'explosion du moteur du camion citerne.

*Imaginez un environnement qui permette de caractériser au plus vite une situation de crise puis qui guide et accompagne la cellule de gestion de crise vers l'obtention d'un processus de réponse efficace avant de fournir automatiquement l'outil logiciel à même d'orchestrer ce processus collaboratif, de manière transparente, entre les acteurs de la résolution de crise... C'est ce que les travaux présentés dans ce manuscrit ambitionnent de proposer et de décrire.*

---

<sup>1</sup> Nucléaire, Radiologique, Bactériologique, Chimique

<sup>2</sup> Services Départementaux d'Incendie et de Secours

<sup>3</sup> Direction départementale de l'Équipement

<sup>4</sup> Service d'aide médicale urgente

# Chapitre I :

## Contexte et problématique

<b>I. La notion de crise.....</b>	<b>8</b>
<b>II. La gestion de crise .....</b>	<b>9</b>
II.1 Vue d'ensemble de la gestion de crise.....	9
II.2 La réponse à la crise : caractéristiques et faiblesses .....	9
<b>III. Problématique générale .....</b>	<b>12</b>
III.1 Vers une problématique d'interopérabilité.....	12
III.2 Une approche basée sur la médiation.....	12
III.3 Vers une démarche de conception d'un médiateur adapté à la situation .....	14
<b>IV. Concepts supports de l'ingénierie dirigée par les modèles .....</b>	<b>15</b>
IV.1 Cellule de crise et notion de système de systèmes .....	15
IV.2 Le processus collaboratif de gestion de crise.....	16
IV.3 SoS et système de systèmes d'information.....	17
IV.3.1 Notion de système d'information .....	17
IV.3.2 Du médiateur au système d'information de médiation .....	18
<b>V. Conception de système d'information de médiation.....</b>	<b>19</b>
V.1 Interopérabilité dirigée par les Modèles.....	20
V.2 Mécanisme de transformation de modèles .....	21
V.3 Vue d'ensemble du projet MISE .....	22
V.4 Le projet MISE dans un contexte de réponse à une crise .....	24
<b>VI. Le projet ISyCri .....</b>	<b>26</b>
VI.1 L'axe conception du projet ISyCri .....	26
VI.2 L'axe exécution du projet ISyCri .....	27
VI.3 Synthèse sur le projet ISyCri .....	29
<b>VII. Conclusion.....</b>	<b>30</b>

# I. La notion de crise

L'actualité regorge d'événements mettant en péril des personnes, des biens, des entreprises ou encore l'économie mondiale. Citons par exemple le Tsunami de 2004 sur les côtes asiatiques, les feux de forêts en Grèce en 2009, les tremblements de terre en Haïti et en Chine en 2010, l'instabilité de l'économie mondiale depuis 2008, ou encore l'éruption du volcan islandais Eyjafjöll en 2010 qui provoqua d'énormes problèmes de transports aériens. Ces situations, appelées selon les contextes et les sources *calamité* (Ryfman, 1999), *incident* (Devlin, 2006) ou *événement majeur* (Lagadec, 1992), se produisent dans un environnement vulnérable (présence d'un volcan en activité, mouvement de plaques continentales) et évoluent selon une issue défavorable appelée *crise* (Tomasini *et al.*, 2004).

En chinois, le mot *crise* s'exprime par l'adjonction de deux idéogrammes qui signifient pour l'un *danger* et pour l'autre *opportunité* (Devlin, 2006). On trouve dans cette illustration à la fois la notion d'*événement*, dit déclencheur (opportunité), et la notion de *vulnérabilité* (danger) qui rend le système sensible à cet événement. Cette dualité se retrouve dans la plupart des acceptions d'une crise existant dans la littérature :

- **acception étymologique** : « le mot signifie originellement décision, jugement, puis par extension la phase décisive d'une maladie. Hors du domaine médical, le terme *crise* en est venu à désigner l'idée de trouble, les situations de déséquilibre profond, voire de désordre (économique, politique, géopolitique, climatique, etc.) » ([http://fr.wikipedia.org/wiki/Gestion\\_de\\_crise](http://fr.wikipedia.org/wiki/Gestion_de_crise)),
- **acception générale** : plus précisément une crise est définie par le Webster's New Collegiate Dictionary comme « un moment crucial et instable duquel résultera une évolution du système vers un état meilleur ou pire. »,
- **acception business** : elle définit la notion de crise comme « un état instable d'une organisation pouvant engendrer des résultats indésirables » (Devlin, 2006),
- **acception humanitaire** : une crise est « une calamité qui provoque des victimes et potentiellement des dégâts matériels de grande ampleur » (Ryfman, 1999).

(Lagadec, 1992) propose de définir la notion de crise comme une instabilité d'un « univers de référence », ce qui implique une diversité de nature de crise selon « l'univers de référence ». La nature d'une crise peut être *militaire, politique, économique, sanitaire, sociale, technique* ou *humanitaire* selon (Devlin, 2006).

Concernant la notion d'événement déclencheur, il en existe deux grandes catégories pouvant, associées à une spécificité du système, conduire à une crise (Tomasini *et al.*, 2004) :

- les événements dont la survenue est brutale : ouragans, attentats, tremblements de terre, inondations, accidents de transport, éruptions volcaniques, catastrophes industrielles, ...
- Les événements dont l'émergence est progressive : sécheresses, épidémies, famines, conflits, crises économiques, ...

Une crise est donc caractérisée par sa *nature*, son (ou ses) *événement(s)* déclencheur(s), la *vulnérabilité* particulière du système mais aussi et surtout les *problèmes* apparus avec cette crise (l'ensemble des faits négatifs constatés, inhérents à cette situation) et qui engendrent l'instabilité de l'univers de référence. La résolution de telles situations critiques, lorsqu'elles se produisent, passe donc par la mise en place, de la part d'acteurs probablement de natures et d'origines diverses et variés, de schémas de réponse destinés prioritairement à réduire ces problèmes (afin de ramener l'univers de référence dans un état stable et

acceptable, éventuellement différent de l'état initial). C'est cette gestion de crise qui fait l'objet de la section suivante.

## II. La gestion de crise

### II.1 Vue d'ensemble de la gestion de crise

D'après [http://fr.wikipedia.org/wiki/Gestion\\_de\\_crise](http://fr.wikipedia.org/wiki/Gestion_de_crise), la gestion de crise peut-être vue comme l'ensemble des modes d'organisation, des techniques et des moyens qui permettent à une organisation de se préparer puis de faire face à l'apparition d'une crise, avant d'en tirer les enseignements pour améliorer les procédures et les structures dans une vision prospective. Une gestion de crise est généralement décomposée en quatre phases, pouvant coexister, décrites succinctement ci-dessous (Altay *et al.*, 2005), (Beamon *et al.*, 2004), (Van Wassenhove, 2005), (Tomasini *et al.*, 2009) :

- **Prévention** : Cette étape consiste à traiter la vulnérabilité de l'écosystème par une gestion des risques permettant soit de diminuer la probabilité de survenue d'un événement déclencheur, soit de rendre le système robuste à sa survenue. Il s'agit finalement de faire en sorte d'éviter la crise.
- **Préparation** : Cette étape consiste à prendre en compte les retours d'expériences afin de développer des nouveaux processus de réponses, d'une part, et de mettre en place l'organisation et les infrastructures qui permettront la mise en œuvre de la réponse à la crise, d'autre part. Il s'agit finalement de se préparer à affronter la crise.
- **Réponse** : cette phase consiste, lors de l'apparition d'une crise, à définir ou ajuster puis à mettre rapidement en œuvre un processus de réponse faisant appel à des activités définies à l'étape précédente selon les caractéristiques de la crise. Par la suite, nous nous intéresserons plus particulièrement à cette étape.
- **Rétablissement** : Cette phase relève de l'instauration d'une situation stabilisée et acceptable provisoirement (car bien souvent optimale compte tenu des conditions et de la situation traversée), et de la gestion efficace de la transition vers le retour à la normale. Par exemple, cette phase peut concerner la reconstruction d'une zone dévastée, le suivi des personnes atteintes par la crise, ...

### II.2 La réponse à la crise : caractéristiques et faiblesses

La réponse à la crise est définie comme « l'ensemble des mesures prises pour résoudre des problèmes causés par la survenue d'une crise » (Devlin, 2006). Cette définition souligne le fait que la réponse à la crise ne se déroule évidemment que lors de la survenue d'une crise, ce qui signifie que la réponse à la crise doit s'organiser dans l'urgence lorsque l'événement déclencheur est survenu (la réponse à la crise fait suite à l'échec de la phase de prévention et se base sur la phase de préparation). Il est évident que diriger le travail de différentes organisations, intervenant dans l'urgence dans l'optique de résoudre la crise, est un travail extrêmement délicat. Cette difficulté est soulignée dans la définition de la réponse à la crise de (Lagadec, 1995) qui la caractérise comme « une situation où de multiples organisations, aux prises avec des problèmes critiques, soumises à de fortes pressions externes, d'après tensions internes, se trouvent projetées brutalement, et pour une longue durée sur le devant de la scène; projetées aussi les unes contre les autres ... ». Dans de telles conditions, il est important que les différentes organisations susceptibles d'intervenir dans le cadre d'une réponse soient préparées à réagir de façon cohérente dans l'urgence.

Pour anticiper les exigences de cette réaction, des listes de type d'actions susceptibles d'être mises en œuvre dans le cadre d'une réponse à une crise sont proposées. Dans le contexte des crises humanitaires, il s'agit principalement des activités de déclenchement de plans d'opérations de secours et de mise en place de centres d'opérations de secours (Altay *et al.*, 2005). Dans le cadre des crises civiles, une multitude de plans d'opérations de secours existent. Ils se différencient selon la nature des crises et sont déclinés selon le niveau de décision, l'organisation ciblée et l'échelle à laquelle ils sont mis en œuvre. On peut citer certains plans officiels français : le Plan Particulier d'Intervention (PPI), le Plan Rouge, le Plan blanc, les Plans de Secours Elaborés, ou encore le plan ORSEC. « Ces plans très en amont, même si on ne décline pas tout à l'avance, vont permettre à chacun, au moment où l'on sera dans la crise, de comprendre la crise et d'agir, même d'initiative, de façon cohérente par rapport à la crise elle-même » (Ecole militaire, 2007).

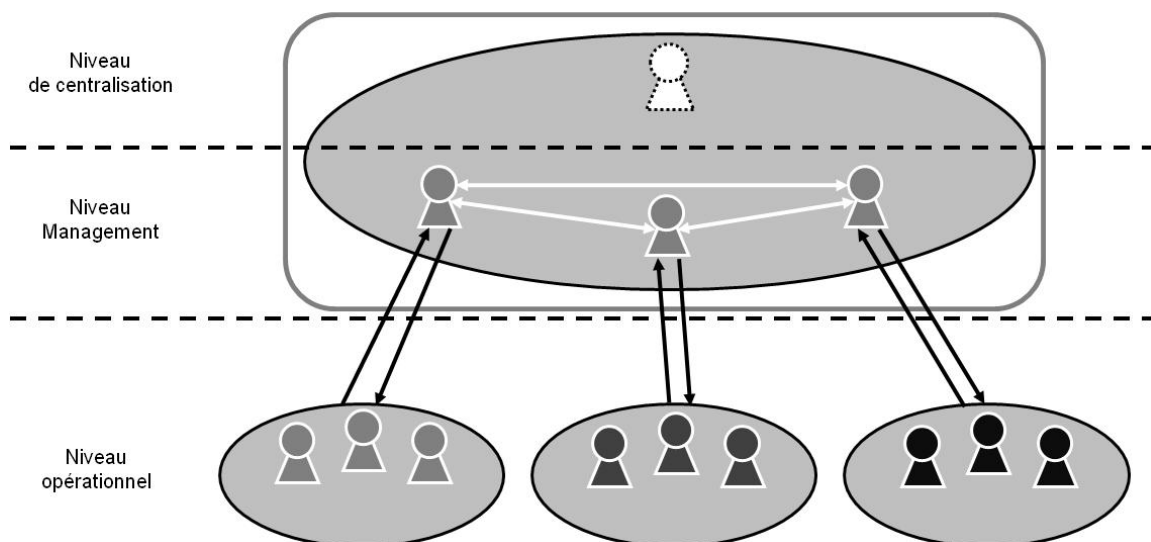
Bien que la création et l'utilisation des plans de secours constituent une exigence incontournable dans le cadre de la préparation à la gestion de crise, la mise en place de ces plans ne suffit pas lors de la réponse à la crise. Ce constat, dressé par une étude réalisée dans le cadre du projet Descartes (Aligne *et al.*, 2010) et basé sur (Aligne, 2009), (Dautun, 2004), (Ministère de l'intérieur, 2008), s'appuie en particulier sur les cas de coexistence de plusieurs plans. Par exemple, si certaines opérations de secours peuvent être communes à deux plans, d'autres, propres à l'un des plans, doivent s'intercaler dans l'ordonnancement des actions du premier. Qui plus est, ces plans peuvent mobiliser des ressources équivalentes (voire identiques) ou antagonistes. Enfin, ces plans peuvent être mis en œuvre à des échelles différentes (différentes zones d'actions par exemple). En outre, même dans le cas du déploiement d'un unique plan, la définition préalable et indépendante des responsabilités et des tâches des différents acteurs concernés (par exemple sous la forme de fiches réflexes) risque fortement d'entraîner des dysfonctionnements sur le terrain et d'entraver le déroulement efficace du plan. Chaque acteur risque effectivement de se focaliser sur la réalisation de ses propres missions, sans se préoccuper concrètement des interactions avec les autres acteurs, eux-mêmes à la poursuite de la réalisation de leurs missions (risques de gêne, possibilités de factorisation ignorées, omission de la nécessaire coordination de certaines actions, etc.). Il faut donc s'assurer de la cohérence globale des opérations de secours.

Cette coordination inter-organisationnelle est, par nature, un exercice délicat du fait de la variété des responsabilités engagées par des organisations différentes devant travailler ensemble dans l'urgence (Hansen-Glize, 2008). De plus, au jour d'aujourd'hui, les décisions reposent en grande partie sur l'expérience et l'intuition des acteurs. Par ailleurs, dans un tel contexte, la communication entre les acteurs n'est pas formalisée et l'information n'est que très peu partagée : les acteurs n'échangent que selon des schémas point à point, uniquement lorsque c'est nécessaire (en cas de blocage par exemple) ou défini dans le plan. La globalisation de l'information constitue donc un enjeu majeur.

Malheureusement la mise en place de cette coordination inter-organisationnelle est souvent difficile à réaliser du fait de l'hétérogénéité des différentes organisations qui œuvrent pour résoudre, ou au moins réduire, la crise. Ce manque de coordination est un point récurrent de dysfonctionnement des réponses à la crise dans le monde humanitaire. Un mémo, disponible en annexe A, revient sur les problèmes de coordination de la réponse à la crise suite au tremblement de terre en Haïti, « Haïti n'était pas prêt à accueillir cette affluence de bonnes volontés! Américains, Européens, Chinois,... ont fait preuve d'une générosité sans précédent. Porte-avions, hôpitaux de campagne, rations alimentaires, tentes, hélicoptères... ont été acheminés des quatre coins de la planète. Mais par manque d'organisation et de coordination, nous avons perdu du temps et de trop nombreuses vies humaines », Il est souligné aussi que ces erreurs avaient déjà été commises lors de la réponse à la crise suite au Tsunami en 2004 et que « Le monde humanitaire a besoin de préparation, d'encadrement, de coordination, de régulation, de structuration, de veille... Il lui faut une "intelligence humanitaire" pour anticiper et unifier son action ». En adéquation avec ce constat, la création d'une organisation mondiale suivant le modèle des casques bleus a été proposée avec l'objectif

d'assurer cette coordination. Ces casques « rouges » auraient « vocation à organiser et coordonner l'action des équipes de secours déployées, dans l'urgence, sur un théâtre de catastrophe naturelle » (<http://www.casques-rouges.org/fr/>).

Afin de mieux appréhender l'origine de ce manque de coordination intéressons-nous maintenant à l'organisation d'une réponse à une crise. Au-delà de la diversité des acteurs intervenant sur le terrain, il existe, quel que soit le type de crise, une structure particulière qui constitue par nature le meilleur support potentiel à l'activité incontournable de coordination de la gestion des opérations. En effet, dans ces contextes d'urgence et de situation critique, un collectif de personnes en charge de missions spécifiques, telles que l'échange ponctuel d'information, la répartition et la synchronisation d'activités et plus généralement l'organisation globale de la réponse, est généralement désigné et mis en place. L'autorité, l'influence et le rayonnement de cette structure sont incontestablement variables (de très faibles pour une crise humanitaire à très forts pour une crise prise en charge par les pouvoirs publics ou militaires). Ce collectif, appelé par la suite cellule de gestion de crise, est identifié comme l'organe de pilotage qui doit faire face à la situation. Dans ces travaux, nous ne nous intéresserons qu'au cas où cette cellule est unique (même physiquement distribuée), sachant que dans le cas où il y aurait plusieurs cellules de crise, on peut considérer que soit elles sont indépendantes et gèrent leurs missions propres (en s'appuyant éventuellement sur un couplage faible), soit elles sont fortement corrélées et dépendent d'une super-cellule qui deviendra alors l'objet de notre étude. Une telle cellule de crise est composée d'au moins un représentant par organisation active lors de la réponse à la crise, l'ensemble de ces représentants correspondant au niveau de management de la figure I-1. Ce représentant a plusieurs missions (i) constituer l'interface avec les opérationnels de l'organisation (niveau opérationnel de la figure I-1), répartis sur le site (*i.e.* remonter les informations du terrain et redescendre des consignes ou d'autres informations issues d'autres canaux), (ii) faire l'interface avec les autres organisations afin d'essayer d'échanger des informations et de coordonner leurs actions. Dans certain type de gestion de crise, un acteur est en plus en charge de l'autorité globale de la réponse (le préfet dans le cadre des crises civiles). Cet acteur est représenté par le niveau de centralisation de la figure I-1. Cette figure I-1 résume donc cette organisation de la cellule de crise.



**Figure I- 1 : Structure d'une cellule de gestion de crise**

## III. Problématique générale

### III.1 Vers une problématique d'interopérabilité

La description de la réponse à la crise de la partie précédente a mis en évidence les difficultés rencontrées dans le cadre de la définition d'une collaboration efficace entre les organisations durant le laps de temps où la crise est factuelle. Ces difficultés proviennent principalement du fait qu'il est nécessaire de coordonner le travail d'organisations hétérogènes, dans l'urgence et de manière dynamique et réactive afin de résoudre la situation de crise en prenant en compte son instabilité. Cette supervision de la réponse à la crise est une mission essentielle devant être assurée par la cellule de gestion de crise à travers ses décisions.

Cependant, la prise de ces décisions est un exercice délicat au sein d'une telle cellule de gestion de crise, en particulier du fait des deux considérations suivantes :

1. Chaque organisation (fonctionnellement indépendante) représentée au sein de la cellule de crise s'inscrit, certes, dans une démarche coopérative, mais apporte surtout ses propres compétences qu'elle entend gérer et mettre en œuvre selon ses propres procédures métier. Il faut par conséquent préserver son autonomie afin de lui permettre de déployer au mieux lesdites compétences. Pour cela, il faut traiter avec elle en acceptant la spécificité de ses informations, de ses actions et de ses missions. En effet, il n'est pas concevable d'imposer des standards opérationnels drastiques auxquels chacun devrait se conformer. Ainsi, chaque acteur doit demeurer à même de définir lui-même le niveau de granularité (des informations, des services et des missions) auquel il souhaite être perçu au sein de la cellule de crise (afin de ne pas interférer avec les procédures métiers en particulier). La pression liée à une adaptation potentielle doit être, et doit rester, minimale. C'est un gage d'efficacité.
2. Une crise est souvent un phénomène très dynamique, elle évolue au cours du temps dans un univers baigné d'incertitude. La capacité de prédiction avérée est en général assez faible. Ce constat induit un besoin de changer le spectre des compétences et/ou le champ des responsabilités des organisations au sein de la cellule de gestion de crise selon un mode très réactif. La composition et les rôles du collectif de la cellule de gestion de crise sont donc eux aussi évolutifs. Dans l'idéal, pour être pertinent, les organisations doivent quitter ou intégrer la cellule assez rapidement et à moindre effort (sans compter que ces évolutions peuvent également être subies). La cellule de gestion de crise se doit donc d'être un système de pilotage ouvert et configurable pour faire face à une situation qu'on sait changeante. Il faut impérativement qu'elle se dote des moyens de surmonter des obstacles organisationnels et techniques qui sont souvent sans commune mesure avec la gravité des situations engendrées par la crise.

Ces caractéristiques imposent à la cellule de gestion de crise d'être capable de faire interagir des organisations, nativement étrangères les unes par rapport aux autres, afin d'établir des comportements collectifs harmonieux et finalisés, sans avoir à modifier en profondeur leur structure ou leur comportement individuel. Cette capacité relève d'une problématique d'interopérabilité au sens de (Pingaud, 2009).

### III.2 Une approche basée sur la médiation

Cette partie explique les différentes méthodes de résolution d'une problématique d'interopérabilité et leur adéquation vis-à-vis des caractéristiques de la cellule de crise



- La méthode intégrée : cette méthode consiste à imposer une standardisation complète aux organisations de la gestion de crise va à l'encontre du caractère indépendant de chaque organisation. Il n'est par conséquent pas envisageable de considérer l'utilisation d'une telle méthode.
- une méthode d'unification et de «standardisation» : cette méthode consiste à unifier tous les échanges possibles entre les organisations autour de standards, que ce soit au niveau des informations utilisées, des opérations, des procédures métiers des différentes organisations. Comme exemple de standards, l'«Emergency Data Exchange Language» (EDXL) (OASIS, 2006) est une proposition de standardisation de l'ensemble des informations pouvant être échangées dans le cadre d'une situation de crise.
- une méthode de fédération ou de «non standardisation» : cette méthode doit permettre aux organisations de collaborer ensemble, tout en leur permettant de garder leur spécificité et leur hétérogénéité et surtout de se baser sur ce qui existe chez les différentes organisations.

La méthode d'unification n'est pas adaptée à un système de gestion de crise. Elle peut, d'une part, s'avérer coûteuse autant au niveau financier qu'au niveau délai de mise en place. En effet, l'utilisation de standards (même seulement en tant que pivots) au sein d'une organisation peut impliquer l'adaptation des outils, méthodes de travail et stratégies en place. D'autre part, lister l'ensemble des organisations pouvant être impliquées dans le cadre d'une réponse à la crise s'avère un exercice difficile, voire impossible, au vue de la multitude d'organisations publiques ou privées, identifiées ou non, susceptible d'intervenir. Bien qu'il soit envisageable de mettre en place cette méthode entre des organisations «incontournables» lors d'une réponse à une crise, tel que le SDIS<sup>1</sup> ou la gendarmerie pour les crises civiles, ces organisations devant ensuite faire l'interface avec les autres organisations, nous avons préféré ne pas retenir cette méthode et nous avons privilégié une méthode de fédération dans ces travaux de thèse.

L'approche par fédération semble plus attractive pour un système de réponse à la crise car elle offre une possibilité d'intégration d'organisations dans leur état actuel, qu'elles soient incontournables ou qu'elles interviennent plus occasionnellement. Cependant cette gestion de l'hétérogénéité des différentes organisations a un impact sur l'architecture du système de réponse à la crise. La structure basée sur des échanges de types point à point (cf. figure I-2, cas (a)) ne pourrait pas convenir à une cellule de réponse à la crise. En effet, la gestion de la traçabilité des échanges entre les organisations ainsi que le temps nécessaire à la mise en place d'une telle architecture ne pourraient pas être satisfaisants au final. Dans le même ordre d'idée, la distribution des compétences et le maintien de l'autonomie des organisations sont deux facteurs qui ne militent pas pour un pilotage trop centralisé dans lequel un système de contrôle serait amené à prendre une responsabilité hiérarchique couvrant à lui seul l'ensemble des opérations attendues (cf. figure I-2, cas (b)). Dans le cadre de ces travaux, nous avons donc opté pour une solution à mi chemin entre ces deux extrémités : **le concept de médiation**. C'est une déclinaison particulière de l'architecture de médiation promue par Wiederhold (Wiederhold, 1992) dans un contexte de recherche d'informations, avec l'idée de fédérer des ressources informationnelles distribuées. Notre architecture propage cette idée au niveau des connaissances sur les métiers. Le but est de savoir coordonner les activités des organisations en imposant une structure de contrôle dictée par des processus qui doivent être exécutés avec conformité. (cf. figure I-2, cas (c)).

---

<sup>1</sup> Services Départementaux d'Incendie et de Secours

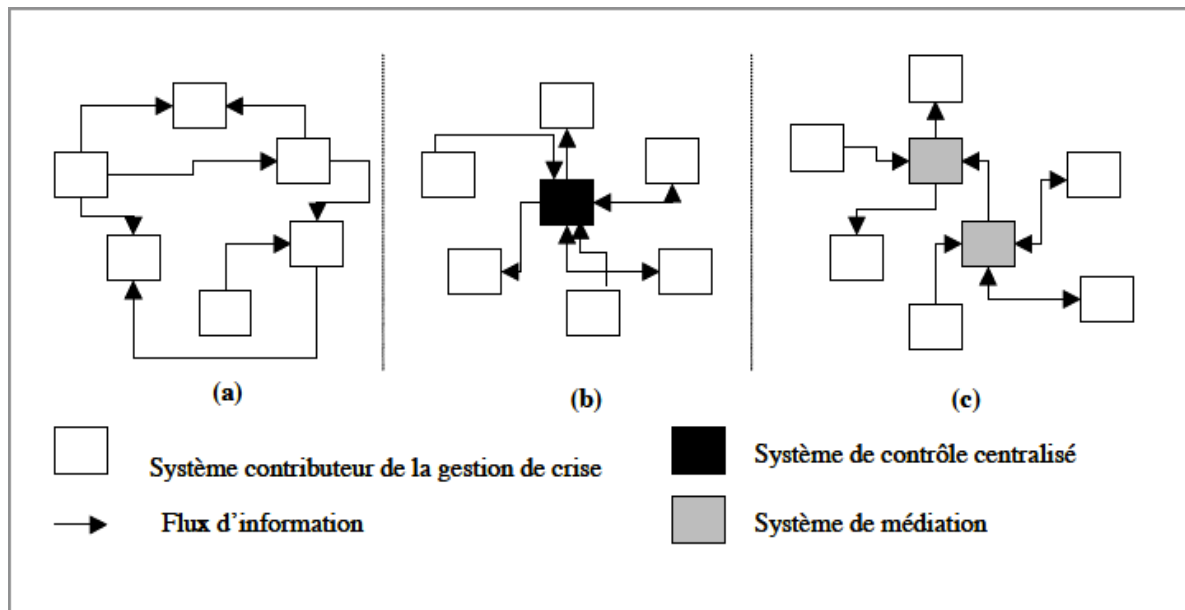


Figure I- 2 : Plusieurs formes d'architecture du système de systèmes

Ces systèmes de médiation, ou médiateurs, doivent réunir autour d'eux les systèmes contributeurs, *i.e.* les organisations. Ils doivent assurer la compatibilité entre les différents contributeurs ainsi que la coordination des leurs actions. Un médiateur doit donc orchestrer la succession des activités de la réponse à la crise tout en gérant les correspondances syntaxiques et sémantiques des informations échangées par les partenaires.

Sachant que d'une réponse à une autre, les compétences et les procédures des différentes organisations n'évoluent que très peu (ajout d'une nouvelle compétence, amélioration d'une autre suite à un retour d'expérience). Il en découle que seul(s) le (ou les) médiateur(s) change(nt) d'une situation à une autre. C'est donc principalement au médiateur de supporter les adaptations dues à la situation. De plus, dans un souci d'efficacité de la réponse à la crise, il est nécessaire que le médiateur soit parfaitement adapté à la situation et pilote les interactions entre les organisations. Ce pilotage global de la réponse, permettra de limiter la coexistence de plans de secours pouvant engendrer des problèmes supplémentaires.

### III.3 Vers une démarche de conception d'un médiateur adapté à la situation

Afin de configurer un (ou des) médiateur(s) devant supporter la collaboration entre les organisations, il est nécessaire de répondre aux interrogations principales suivantes :

- Comment définir les caractéristiques d'une crise, de façon collaborative et dans un langage compréhensible par l'ensemble des organisations intervenant dans le cadre de la réponse à la crise ?
- Comment définir l'ensemble des actions à mettre en œuvre dans le cadre de la réponse à la crise ? *i.e.* quelle organisation réalise quelle action pour quelles raisons ?
- Comment coordonner les actions des organisations ?
- Comment globaliser les informations et transférer l'information au bon format, au bon moment, à la bonne organisation ?
- Comment réagir face aux évolutions de la situation ?

L'objectif de ces travaux de thèse est de proposer une démarche de conception de médiateur répondant à ces cinq points. Cette démarche repose d'une part sur une démarche d'ingénierie dirigée par les modèles (IDM) permettant de proposer une solution technologique pour le médiateur et d'autre part, sur l'utilisation de deux langages spécifiques de domaine (DSL) définissant, pour le premier un langage de représentation de la situation et pour le second, un langage de description des compétences des organisations.

## IV. Concepts supports de l'ingénierie dirigée par les modèles

Dans la partie précédente, nous avons expliqué que l'objectif de ces travaux de thèse est de déterminer à partir des informations de la situation, un médiateur en charge de piloter la réponse à la crise, puis de proposer une solution technologique correspondant à la définition du médiateur. Ces travaux reposent sur un certain nombre de concepts que nous allons présenter dans cette partie.

### IV.1 Cellule de crise et notion de système de systèmes

Il existe plusieurs définitions du concept de système de systèmes (SoS pour System of Systems). Récemment, un ouvrage produit par des auteurs en ingénierie des systèmes (Luzeaux *et al.*, 2008) en a fait la synthèse et a proposé la définition suivante : « Un système de systèmes est un assemblage de systèmes pouvant potentiellement être acquis et/ou utilisés indépendamment, pour lequel le concepteur, l'acquéreur et/ou l'utilisateur cherche à maximiser la performance de la chaîne de valeur globale, à un instant donné et pour un ensemble d'assemblages envisageables ». On retrouve dans cette définition une forme très condensée des différences entre le concept de système de systèmes, et celui de système. Ces différences sont qualifiées par cinq critères discriminants proposés par (Maier, 1998) :

- (i) indépendance opérationnelle des éléments (pouvant être des systèmes),
- (ii) indépendance managériale des éléments,
- (iii) développement évolutif,
- (iv) comportement émergent,
- (v) distribution géographique.

Examinons ces différents critères dans le cadre d'une réponse à une crise. Les organisations doivent unir leurs efforts et interagir, avec deux finalités principales :

- comprendre la crise, la suivre et la caractériser,
- agir pour tenter de résoudre, ou tout au moins de réduire, la crise. Les actions entreprises par les systèmes contributeurs sont de nature préventive ou curative.

Cependant, ces acteurs ne sont pas nécessairement placés sous une seule et même autorité (critère (ii)). Si cela est vrai dans le cas d'une crise civile en France où les préfets de département, ou de région, sont en charge de l'autorité globale de la réponse et par conséquent de la responsabilité de déployer le plan selon des procédures prédéfinies (*i.e.* les différents plans), cela n'est pas aussi net dans le cas d'une crise humanitaire quand des acteurs divers tels que l'ONU, des institutions nationales et des organisations non gouvernementales multiplient les actions individuelles, sur une base de compétences dont la

complémentarité est à évaluer. Ils sont ainsi, de fait, en cogestion de crise, partageant les informations pour construire une coordination qui est conçue en situation sur le terrain (critère (i)).

En somme, l'hétérogénéité des organisations et leur autonomie de management sont généralement reconnues comme un fait réel pour ce type de situation. Chacun prend les responsabilités qui sont les siennes dans son champ de compétences. Par conséquent, la gestion de la crise peut être vue comme un ensemble de plusieurs systèmes autonomes qui doivent collaborer, à la fois pour la prise de décision et pour les opérations. Selon la crise et son degré d'expansion, différents systèmes répartis dans différentes régions peuvent interagir dans le but de la résoudre. Nous pouvons aussi considérer qu'il peut y avoir une distribution géographique des systèmes quand la crise concerne un territoire étendu, dépassant des limites d'un pays et de son gouvernement, tel que les crises humanitaires par exemple (critère (v)). La nature évolutive de la crise a des répercussions sur la composition de l'ensemble des organisations qui doivent intervenir ainsi que les opérations rendues pour la résoudre. La nature potentiellement évolutive d'une crise implique donc celle de la réponse qui lui est opposée (critère (iii)). Enfin, si les organisations se concertent afin de jouer sur la complémentarité de leurs compétences, la gestion de la crise ne peut qu'être bénéficiaire. Même si les retours d'expériences des crises passées, surtout dans le domaine humanitaire, soulignent le manque de coordination et de travail en commun des acteurs, la cellule de gestion de crise doit avoir un comportement émergent (critère (iv)) afin d'être plus efficace.

Nous pouvons donc conclure que la cellule de gestion de crise a, ou devrait avoir, toutes les caractéristiques spécifiques d'un système de systèmes.

## IV.2 Le processus collaboratif de gestion de crise

Dans cette partie, nous allons essayer d'expliquer l'utilité d'un processus collaboratif, non seulement pour orchestrer la succession des activités d'une réponse à une crise mais également pour prendre en charge la gestion des informations.

Il existe une multitude de définitions de la notion de processus, nous pouvons néanmoins citer celles qui nous semblent pertinentes dans le cadre de nos travaux :

- (Morley, 2002) définit un processus comme : « une organisation d'un ensemble finalisé d'activités effectuées par des acteurs mettant en jeu des entités ».
- (Vernadat, 1996) propose une autre définition portant sur la structure d'un processus : « un processus est un ensemble partiellement ordonné d'étapes exécutées en vue de réaliser au moins un objectif ».
- La norme ISO 9001 (2000) (ISO, 2000) définit un processus comme : « un ensemble d'activités corrélées ou interactives qui transforment des éléments d'entrée en éléments de sortie ».

Dans (Morley *et al.*, 2005), les auteurs définissent un processus comme « un plan d'ensemble indiquant comment les acteurs collaborent au moyen des informations gérées pour accomplir l'objectif ». Cette définition s'avère pertinente vis-à-vis de la dynamique de réponse à une crise puisqu'on y retrouve la notion de succession d'activités (« plan »), les échanges inter-organisationnels (« informations gérées »), le tout dans un contexte objectif (« accomplir l'objectif »). Néanmoins, dans notre cas particulier de gestion de crise, le fait que chaque organisation soit opérationnellement autonome et responsable de la seule exécution de ses propres activités, apporte à ce processus une particularité de contrôle partagée du processus (chaque organisation dispose d'une maîtrise partielle du processus au travers de la maîtrise de ses activités). On retrouve cette notion de « maîtrise partielle sur ce dernier » dans (Morley *et al.*, 2005) lorsque est évoqué le concept de processus collaboratif, ou de processus inter-organisationnel.

Pour affiner cette notion de processus collaboratif, nous nous référerons également à la définition de (Touzi, 2007) qui précise qu'un processus collaboratif est défini comme « un ensemble partiellement ordonné d'activités spécifiquement organisées chez les partenaires de la collaboration et localement exécutées chez ces derniers ». Cette définition est en parfaite adéquation avec notre contexte de système de systèmes puisque nous y retrouvons les notions d'indépendances managériale et opérationnelle.

Dans la suite de ces travaux, nous essaierons de définir un processus collaboratif pour chaque réponse à une crise en fonction de la situation rencontrée, des différents acteurs impliqués, des différentes procédures et plans tout en essayant d'éviter des problèmes d'interaction. Ce processus collaboratif sera ensuite le point d'entrée pour définir, concevoir et déployer le (ou les) médiateur(s).

### IV.3 SoS et système de systèmes d'information

Les considérations précédentes mettent en exergue la criticité du pilotage du processus collaboratif dans la mesure où ce dernier est responsable de l'invocation pertinente des activités des différentes organisations partenaires et de la juste circulation des informations. Le système de systèmes que constitue la cellule de gestion de crise doit donc reposer sur une couche fonctionnellement responsable de ce pilotage du processus collaboratif. Nous proposons dans nos travaux d'instancier cette couche de pilotage sous la forme d'un système d'information qui serait composé des systèmes d'information des partenaires et d'un (ou des) médiateur(s) de ces SI.

#### IV.3.1 Notion de système d'information

Selon (Bernus *et al.*, 1996) un « système d'information doit garantir que la bonne information est disponible au bon endroit au bon moment ». (Saadoun, 2000) propose que « la raison d'être d'un système d'information est l'accès au bon moment à la bonne information pour prendre la bonne décision ». Selon (Reix *et al.*, 1995) un système d'information est « un ensemble organisé de ressources matérielles (ordinateurs, documents papier, etc), personnelles (opérationnels, clients, décideurs, etc) et logicielles (applications informatiques, etc) qui permet d'acquérir, de traiter (générer, modifier, etc.), de stocker (mémoriser) et de transmettre des informations ». Ces trois définitions mettent en évidence un trait particulier des SI : ils sont l'interface entre la connaissance de l'entreprise (ses informations) et les acteurs internes à l'entreprise (personnels, machines, logiciels) mais aussi les acteurs externes à l'entreprise (clients, fournisseurs ou partenaires). Ce statut d'interface s'accompagne d'un caractère « actif » dans la mesure où le SI véhicule, stocke et communique les informations selon des canaux et des procédures préétablis et où il gère les droits des différents acteurs, utilisateurs de cette interface.

En outre, les définitions précédentes ne mettent pas en évidence une composante forte du concept de SI, à savoir sa capacité à constituer une dorsale fonctionnelle de l'organisation en étant le porteur de procédures et de comportements, inscrits (statiquement ou dynamiquement) dans ses entrailles. La vision proposée par (Morley, 2002) (Morley *et al.*, 2005) reprend cette notion en soulignant qu'un système d'information est la composition de deux sous-systèmes : le système de traitement de l'information (comprenant les acteurs, les données et les processus) et le système informatique (comprenant les ressources matérielles et logicielles, les bases de données et les applications), cf figure I-3.

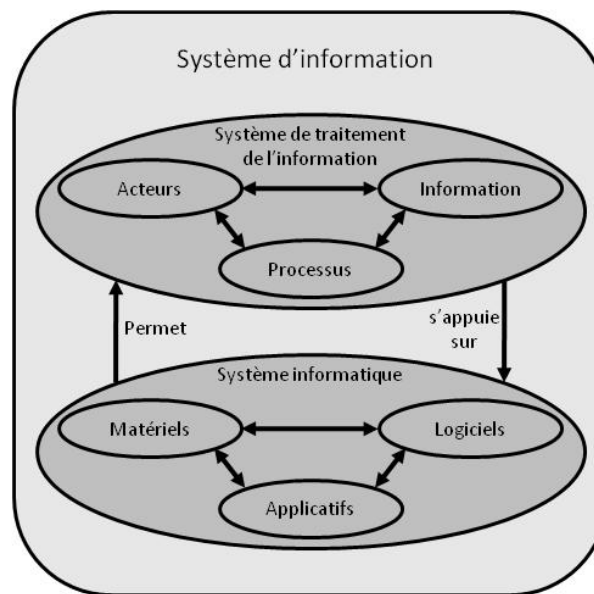


Figure I- 3 : définition d'un système d'information selon (Morley, 2002)

Au final, la vision que nous proposons des SI des organisations impliquées dans la cellule de gestion de crise relève des deux tendances précédentes : le SI est, d'une part responsable de la gestion des informations vis-à-vis des acteurs internes et externes de l'organisation et d'autre part en charge du pilotage (d'une partie) du comportement de l'organisation au travers des procédures et processus inscrits dans sa dynamique de fonctionnement.

#### IV.3.2 Du médiateur au système d'information de médiation

Les sections précédentes ont permis d'établir d'une part, que la cellule de gestion de crise pouvait s'apparenter à un système de systèmes et d'autre part que le pilotage du processus collaboratif, vecteur du déroulement harmonieux de la réponse à la crise, devait être supporté par un système d'information. La cellule de gestion de crise, en tant que système de systèmes est composée des représentants des différentes organisations impliquées (en charge des interactions entre la cellule et le terrain) et du (ou des) médiateur(s). Nous proposons donc que le système d'information de cette même cellule soit un système de systèmes d'information, composé des systèmes d'information des organisations et d'un système d'information représentatifs des fonctions de médiation : le système d'information de médiation (SIM), devant dérouler le processus collaboratif de réponse.

Nous ajoutons en outre l'hypothèse que les systèmes d'information des organisations concernées par la réponse à la crise relèvent d'une architecture SOA<sup>2</sup>. Ce point de vue implique que les organisations potentielles soient en mesure de présenter leurs systèmes d'information sous la forme d'un ensemble de services, localisés et accessibles. Si cette hypothèse peut sembler une contrainte forte, elle est grandement atténuée par le fait que, dans un contexte de gestion de crise, bon nombre de ces services correspondent en fait à des fonctions métiers qu'il suffit de représenter au sein du SI par l'intermédiaire de services d'interface permettant le dialogue indirecte avec ces fonctionnalités métier. Les spécificités de cette atténuation seront expliquées en détail en fin de ce chapitre.

Le SIM devient alors le garant de l'interopérabilité des SI des organisations en assurant à la fois l'exécution du processus collaboratif et la transmission de la bonne information au bon acteur au bon moment. Assurer l'interopérabilité revient, d'après (Chen *et al.*, 2003), à supporter les trois niveaux :

<sup>2</sup> Service-Oriented Architecture

données, applications et processus, en accord avec une sémantique métier convenue entre les organisations. L'interopérabilité des systèmes d'information des organisations, vue selon ces trois niveaux d'intervention, peut être décomposée comme suit (Bénaben *et al.*, 2008) :

- Gestion des données : ce point nécessite la connaissance des ontologies et des formats des différents partenaires afin d'assurer l'échange et le partage des données concernées par la collaboration, à moindre effort.
- Management des applications : cet aspect correspond au recensement (en termes d'identification fonctionnelle) et à la localisation (en termes de point d'accès) de l'ensemble des applications et services concernés par la collaboration.
- Pilotage des processus collaboratifs : ce troisième point relève de la compétence du SIM à disposer de modèles des processus collaboratifs et à en assurer le pilotage tout au long du partenariat.

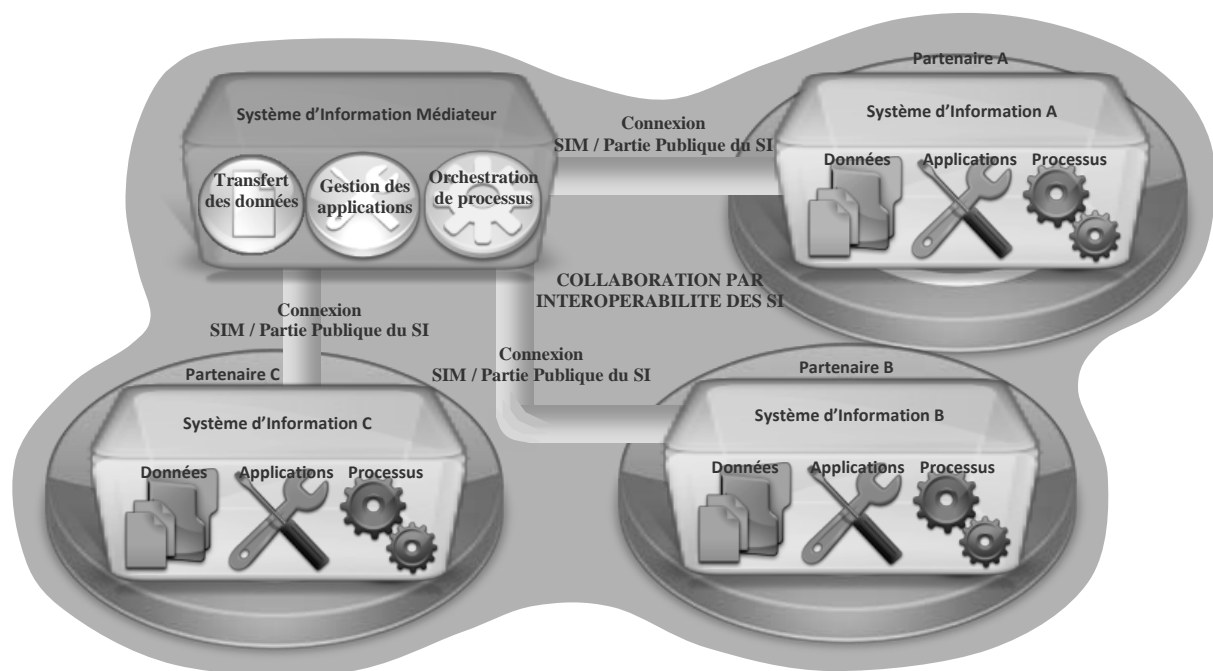


Figure I- 4 : Vue d'ensemble d'un système d'information de médiation (Bénaben *et al.*, 2008)

## V. Conception de système d'information de médiation

Il y a cinq ans, le projet MISE (Mediation Information System Engineering) a été lancé, par Hervé Pingaud et Frédéric Bénaben au centre de génie industriel de l'école des Mines d'Albi-Carmaux, dans le but de fournir une démarche outillée de support à l'interopérabilité des organisations et à l'établissement de réseaux collaboratifs, évolutifs et distribués.

## V.1 Interopérabilité dirigée par les Modèles

Le projet MISE s'inscrit dans la veine de l'interopérabilité dirigée par les modèles (MDI) (Bourey *et al.*, 2007) puisqu'ils ne se limitent pas à proposer une solution d'interopérabilité entre des organisations uniquement au niveau technologique mais aussi au niveau de la modélisation de la collaboration.

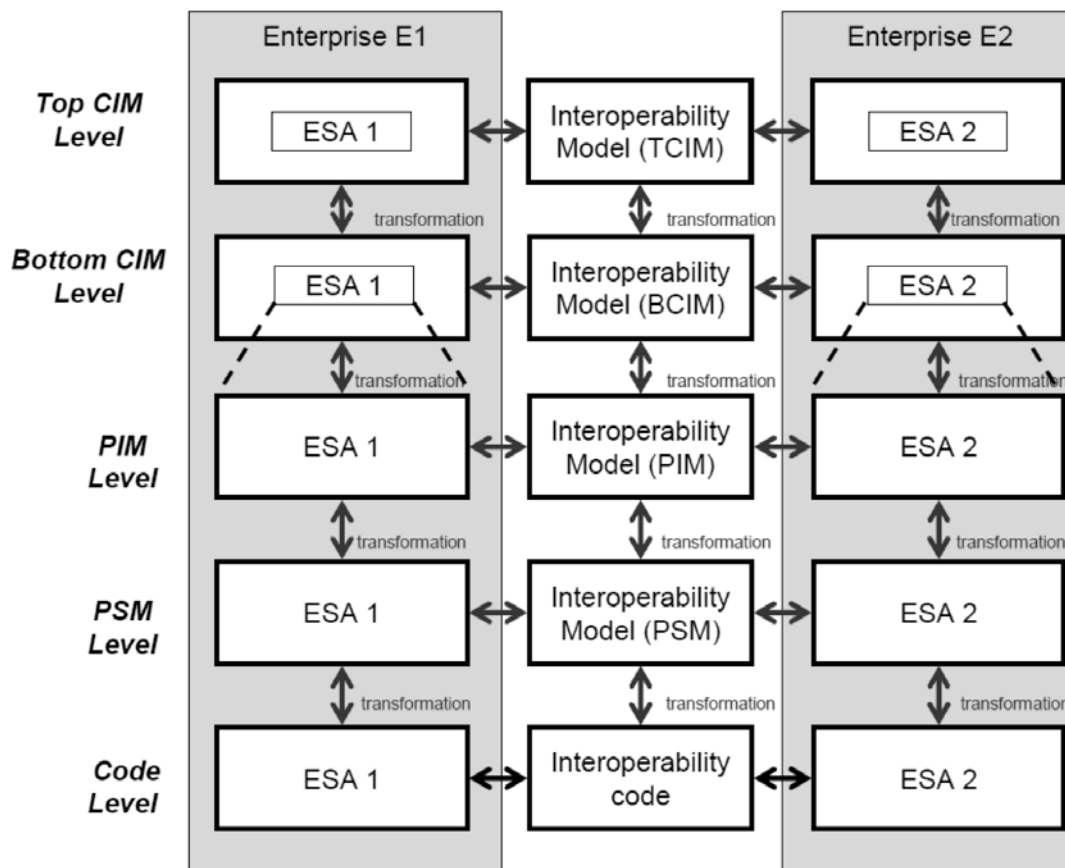


Figure I-5 : modèle de référence d'une démarche MDI (Bourey *et al.*, 2007)

La figure I-5 représente le modèle de référence d'une méthode MDI (NB : ESA signifie Enterprise Software Application). Cette méthode se base sur quatre niveaux directement inspirés des préceptes MDA (Model-Driven Architecture) (Miller *et al.*, 2003) dont le passage d'un niveau à un autre est réalisé grâce à une transformation de modèles.

1. Niveau métier : ce niveau ambitionne à définir les besoins en interopérabilité en fonction des situations. Ce niveau est décomposé en deux sous-niveaux :
  - niveau Top CIM : ce niveau vise à présenter la situation selon une approche holistique à un haut niveau d'abstraction et sans prendre en compte la moindre application logicielle,
  - niveau Bottom CIM : ce niveau représente la partie du CIM qui nécessite d'être implémentée sur une application logicielle sans être pour autant dédiée à une technologie ou à une application spécifique.
2. Niveau logique (PIM Level) : ce niveau décrit l'architecture de la solution permettant de répondre au besoin du modèle Bottom CIM. Cette architecture reste néanmoins indépendante de toute solution technologique.



3. Niveau technique (PSM Level) : ce niveau est le résultat de la projection de l'architecture du niveau PIM sur une technologie donnée.
4. Niveau déploiement (Code Level) : ce niveau est la réalisation technique d'une solution répondant au besoin défini au niveau métier.

## V.2 Mécanisme de transformation de modèles

Le mécanisme de transformation de modèle est à la base d'une approche IDM. Ce mécanisme de transformation repose sur l'utilisation de différents modèles, dont une taxonomie est présentée dans (Bézivin *et al.*, 2009). Dans cette partie, nous allons présenter succinctement le principe de transformation de modèle utilisé dans le cadre du projet MISE, une présentation plus détaillée de ce principe est disponible en annexe D.

Une transformation de modèle permet de transformer un modèle source, conforme à un métamodèle source, en un modèle cible, conforme à un métamodèle cible, en fonction des règles de mapping portant sur les éléments des deux métamodèles. Cependant, les règles de mapping ne peuvent être réalisées qu'à la condition que le métamodèle source et le métamodèle cible aient des concepts partagés. Par concepts partagés, nous ne sous-entendons pas que les mêmes concepts doivent apparaître selon la même syntaxe et sémantique au niveau des deux métamodèles, nous parlons de concepts pouvant au minimum être mis en relation. Cet ensemble de concepts partagés devient alors l'espace de définition des règles de mappings.

Les métamodèles source et cible n'étant pas limités à l'espace des concepts partagés, lors d'une transformation une partie du modèle source est potentiellement perdue. En effet, un modèle étant conforme à un métamodèle, le modèle source contient des éléments définis par la partie spécifique (*i.e.* non partagé) du métamodèle. Ainsi une partie des connaissances ne se retrouveront pas dans le modèle cible. Ces éléments devront être sauvegardés afin de ne pas perdre de connaissance. La réciproque de ce raisonnement implique que le modèle cible peut potentiellement nécessiter une étape d'enrichissement.

La figure ci-dessous (Truptil *et al.*, 2010a) synthétise, la définition d'une transformation de modèle que nous conserverons tout au long de ce manuscrit.

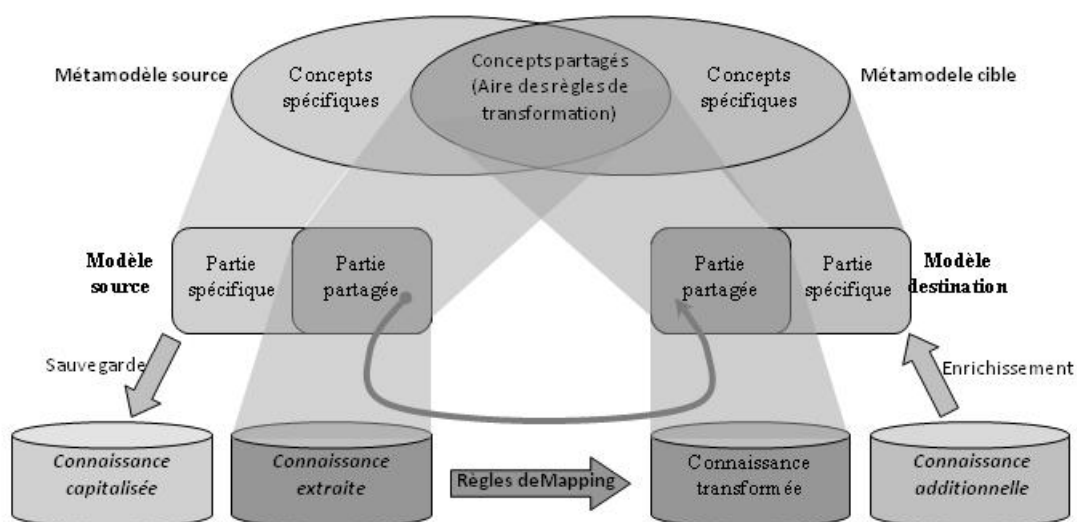


Figure I- 6 : Principes de transformation de modèles (Truptil *et al.*, 2010a)

Dans le cadre de ces travaux de thèse, nous avons choisi deux approches pour la réalisation de transformation de modèles en fonction de la nature endogène ou exogène de la transformation. Ce choix, détaillé en annexe E, abouti à l'utilisation du langage ATL (Jouault *et al.*, 2006) dans le cas d'une transformation exogène, et de l'utilisation d'ontologies et de moteur d'inférence pour le cas des transformations endogènes.

### V.3 Vue d'ensemble du projet MISE

Le but du projet MISE est de définir un médiateur entre différents partenaires. La figure I-6 représente positionne ce médiateur en charge du traitement de l'information et du support dynamique de la collaboration. Pour ce faire, ce médiateur repose sur un système d'information de médiation (SIM).

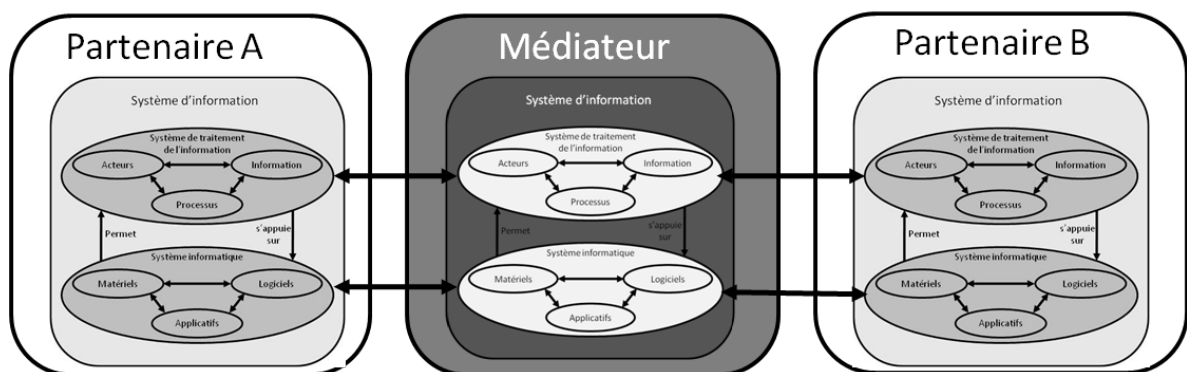


Figure I-7 : place du médiateur au sein de la collaboration.

L'approche conceptuelle proposée pour la conception du SIM vise à permettre le passage (le plus automatisé possible) du niveau métier au niveau de déploiement d'un SIM pertinent. Cette plongée dans les niveaux d'abstraction correspond à la démarche MDI et se base sur les quatre étapes :

1. Niveau métier : le modèle Top CIM correspond à une caractérisation de la situation collaborative. De cette caractérisation un processus collaboratif est défini, modèle Bottom CIM. Ce processus est au cœur des responsabilités de la collaboration. En effet, il représente les échanges aussi bien informationnels qu'opérationnels des différents partenaires. Dans le cadre de ce projet, Une partie restreinte du langage de modélisation de processus BPMN<sup>3</sup>, expliqué en détail au chapitre 3, a été utilisée pour représenter ce niveau.
2. Niveau logique : ce niveau décrit l'architecture logique du SIM indépendamment de la plate-forme. Ce modèle respecte la logique d'architecture du SIM définie dans (Touzi, 2007). Cette logique repose sur les préceptes SOA<sup>4</sup> et hérite de PIM4SOA (Benguria *et al.*, 2006). Cette architecture a été choisie car sa division en trois parties : service, information et processus est parfaitement adaptée à la décomposition précédente du rôle du médiateur.
3. Niveau technique : ce niveau fournit le modèle technique qui sera exécuté sur la plate-forme choisie, à savoir le bus de service (ESB) PEtALS et le langage Java. Ce modèle est utilisé dans le but, d'une part de configurer le bus de service, afin de pouvoir se connecter à l'ensemble des systèmes d'information des différents partenaires de la collaboration, et d'autre part de configurer le moteur d'orchestration du bus de service afin d'exécuter le processus.

<sup>3</sup> Business Process Modelling Notation

<sup>4</sup>Service Oriented Architecture

4. Niveau déploiement : ce dernier niveau se focalise sur la configuration de l'ESB PEtALS à partir des informations du niveau technique.

Ces quatre niveaux sont représentés sous la forme d'un « Double Y » présenté sur la figure I-7

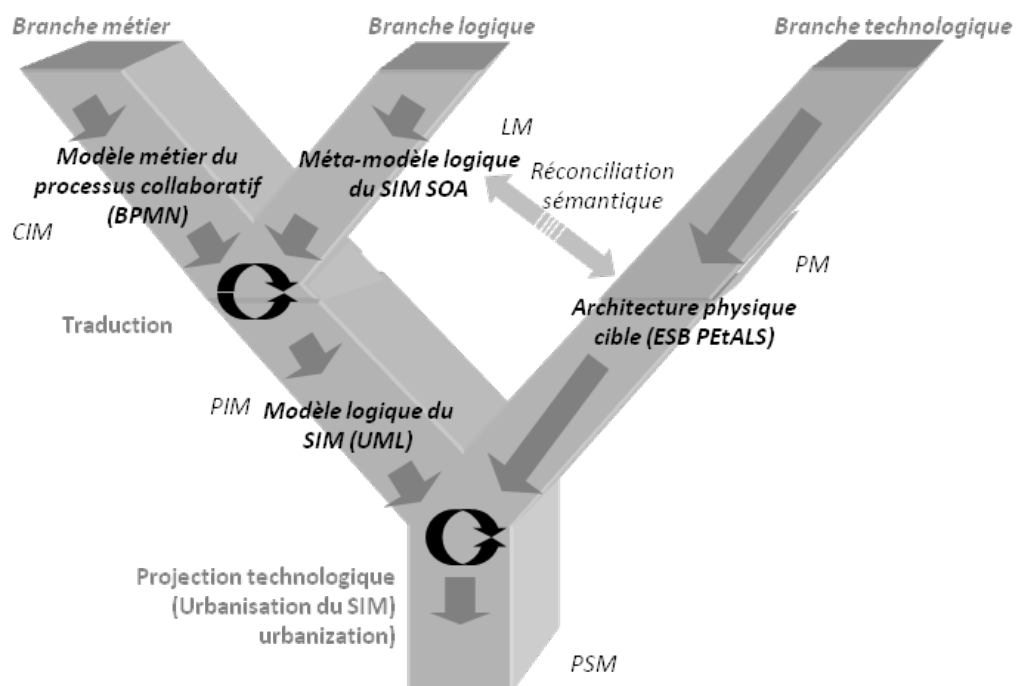


Figure I- 8 : Vue d'ensemble du projet MISE

Les travaux de ce projet ont commencé avec la thèse de J. Touzi (Touzi, 2007) qui s'intéressait au niveau logique en partant du principe que les partenaires pourraient fournir le CIM décrivant le processus caractéristique de la collaboration. Ces travaux ont abouti à une transformation de modèles, expliquée au chapitre 2, permettant d'obtenir le modèle PIM à partir de ce modèle CIM.

Malheureusement, l'hypothèse que les organisations étaient capables de fournir le modèle de processus collaboratif s'est avérée trop ambitieuse (en particulier du fait des exigences portant sur le modèle CIM afin d'être exploitable pour la transformation de modèle au niveau logique). La thèse de V. Rajsiri (Rajsiri, 2009) s'est donc intéressée au niveau métier en définissant un système de base de connaissances permettant aux partenaires de décrire la collaboration souhaitée afin d'obtenir automatiquement un modèle de processus collaboratif. Pour cela, deux bases de connaissances ont été utilisées : la première, résultat spécifique des travaux de V. Rajsiri (Rajsiri *et al.*, 2010), décrit les concepts et notions inhérents à la caractérisation d'une situation collaborative (partenaires, rôles, relations, topologies, etc.), la seconde, directement extraite du MIT Process Handbook (Malone *et al.*, 1999), décrit un très grand nombre de processus métiers standards ainsi que les activités qui les composent. Cette deuxième base de connaissance permet de disposer d'une base d'activités types du domaine industriel, exploitable dans un contexte de collaboration entre entreprises. V. Rajsiri a ensuite défini des règles de déduction entre ces deux bases de connaissances qui permettent de déduire un processus collaboratif à partir de la caractérisation de la collaboration.

Les travaux de Master de W. Mu (Bénaben *et al.*, 2010a) se sont intéressés, quant à eux, au niveau technique et au passage du modèle PIM d'architecture logique au modèle PSM d'architecture technique. Ce passage consiste à enrichir le modèle PIM d'informations techniques pertinentes.

La dernière étape de la démarche, consistant à déployer le SIM à partir du PSM, a été initiée lors du post-doctorat de J. Touzi puis poursuivi dans le cadre des travaux exposés dans ce manuscrit. La transformation entre le modèle PSM de l'architecture technique et le processus exécutable, au format BPEL<sup>5</sup>, sera présentée en détails dans le chapitre IV.

L'ESB<sup>6</sup> PEtALS a été sélectionné comme cible technologique de la démarche. PEtALS est un ESB open source du consortium ObjectWeb (<http://petals.objectweb.org>). Le choix de cette plate-forme se base sur plusieurs raisons :

- Il s'agit d'une solution technique basée sur les principes SOA.
- Une fois connecté au bus, l'ensemble des services d'un système d'information est connecté à l'ensemble des services des autres systèmes d'information.
- Un service d'orchestration, disponible au sein de PEtALS, permet d'exécuter un processus décrit en BPEL, *i.e.* la gestion de l'appel des services dans un ordre précis et selon certaines conditions.

## V.4 Le projet MISE dans un contexte de réponse à une crise

Les résultats du projet MISE ne sont pas totalement adaptés pour une collaboration en contexte de réponse à une crise. En effet, d'une part l'utilisation d'une base de connaissances de référence, tel que le Process Handbook (pour la déduction du modèle de processus collaboratif), pose problème en situation de gestion de crise, et d'autre part, une situation de crise est par définition un phénomène évolutif. Le SIM doit donc être susceptible d'être modifié dynamiquement en fonction de l'évolution de la situation. Dans cette partie, nous allons nous intéresser à détailler ces deux points faibles du projet MISE pour un contexte de gestion de crise.

La base de connaissances de référence (MIT Process Handbook) utilisée dans le contexte du projet MISE n'est pas adaptée en situation de crise. Tout d'abord, la nature même des activités et processus figurant dans cette base n'est pas compatible avec les problématiques de gestion de crise. L'ensemble des tâches et procédures dédiées au comportement des entreprises et celui des tâches et procédures dédiées au comportement des organisations impliquées dans une résolution de crise sont presque disjoints. Ensuite, l'utilisation de cette base de connaissance engendre également des problèmes de correspondance sémantiques, comme expliqué dans (Bénaben *et al.*, 2010). Ces problèmes de correspondances sémantiques se situent entre le processus déduit de la base de connaissance (donc générique) et l'implémentation technique du SIM (qui doit piloter un processus composé des services spécifiques des partenaires). Comme le résume la figure I-8, la résolution de ces problèmes sémantiques relèvent des points suivants :

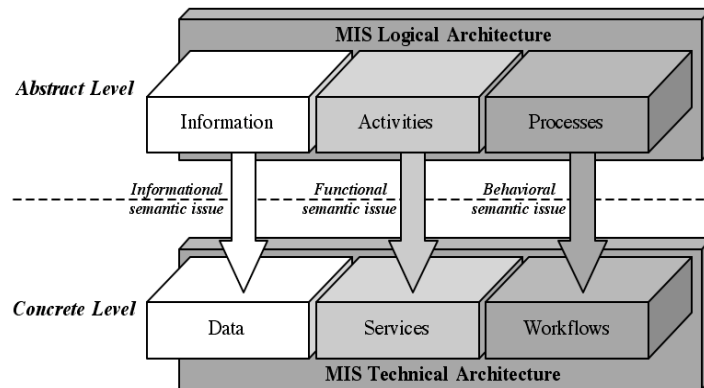
- Comment réconcilier les informations métiers avec les données d'entrées des services des systèmes d'information ?
- Comment assurer la correspondance entre les activités métiers, provenant de la base de connaissance, avec les services des différents systèmes d'information des différentes organisations?
- Comment obtenir le processus exécutable à partir du processus collaboratif, constitué d'activités génériques ?

---

<sup>5</sup> Business Process Execution Language

<sup>6</sup> Entreprise Service Bus

La figure suivante illustre ces considérations de problématiques sémantiques.



**Figure I- 9 : Problèmes de correspondance entre le niveau logique et technique du projet MISE**

Les deux premières questions posent un problème de correspondance « plusieurs à plusieurs », *i.e.* une ou plusieurs informations correspondent à une ou plusieurs données et une ou plusieurs activités correspondent à un ou plusieurs services des systèmes d'information. Le but est donc de réussir à mettre en place un mécanisme de réconciliation entre les activités métiers, avec leurs informations, et un ensemble de services techniques, avec leurs données. La troisième question correspond plutôt à un problème de traduction.

Une autre limite du projet MISE, pour ce contexte de collaboration, est la gestion de l'évolution du SIM. En effet, la crise est par définition un phénomène évolutif imprévisible auquel il faut demeurer adapté. Qui plus est, le groupe d'organisations en charge de la réponse à la crise n'est pas figé (un partenaire peut rejoindre le réseau ou le quitter). Enfin, un dysfonctionnement lors du déroulement des activités peut remettre en cause l'ensemble de la réponse proposée. Le SIM doit donc pouvoir s'adapter à chacune des évolutions précédentes, qui peuvent nécessiter son évolution (voire sa complète redéfinition) à différents niveaux. Enfin, compte tenu du rôle central du modèle de processus collaboratif (qui cristallise les exigences recueillies au niveau métier et intègre la connaissance nécessaire aux niveaux logique et technique avant déploiement), il serait particulièrement intéressant d'ajouter une étape de vérification du processus collaboratif déduit avant de la mettre en place.

Pour résumer, les limites du projet MISE pour une collaboration en situation de crise sont donc :

- l'utilisation d'une base de connaissance de référence, telle que le Process Handbook,
- la robustesse du processus collaboratif déduit,
- la réconciliation sémantique,
- l'adaptation du SIM en fonction des évolutions possibles.

Afin de répondre à l'ensemble de ces limites, Le projet de recherche ISyCri<sup>7</sup>, correspondant à une instantiation du projet MISE, a été mené.

<sup>7</sup> Interopérabilité des Systèmes en situation de Crise

## VI. Le projet ISyCri

Afin de répondre aux limites du projet MISE dans le cas d'une collaboration en contexte de crise, le projet ISyCri (Interopérabilité des Systèmes en situation de Crise) a été construit selon deux axes : un axe de conception (design-time), consistant à produire un SIM correspondant à la situation et un axe d'exécution (run-time), consistant à prendre en compte les différentes évolutions possibles de la situation.

### VI.1 L'axe conception du projet ISyCri

L'axe de conception du SIM s'inspire de la démarche du projet MISE et par conséquent des principes du Model Driven Engineering (MDE) en traversant plusieurs niveaux d'abstraction (métier, logique et technique) et en s'appuyant à chaque niveau sur des modèles.

Seule la première partie de la démarche MISE a été modifiée afin de résoudre les limites de (i) spécificité industrielle du Process Handbook, (ii) correspondance sémantique entre activités métier et services techniques et (iii) vérification de la robustesse du processus. Ainsi, une fois le processus validé, le reste de la démarche de conception est héritée du projet MISE avec les transformations de modèles successives, au niveau logique comme expliqué dans (Touzi, 2007) et au niveau technique comme expliqué dans (Mu *et al.*, 2009). Enfin, la tâche de déploiement sera détaillée dans le chapitre 4 de ce manuscrit.

La première étape du projet ISyCri diffère de celle du projet MISE dans la mesure où elle ne repose plus sur une base de connaissance de référence mais sur les compétences et les services d'information réellement disponibles pour une réponse donnée. En considérant une crise particulière, cette première étape revient alors à déduire le processus collaboratif de réponse à partir des informations relatives, d'une part à la crise et d'autre part à la définition des compétences des organisations. Les informations recueillies sur la crise rassemblent à la fois, l'ensemble des problèmes à prendre en considération (personnes blessées, voie d'accès impraticable, risque d'explosion, etc.) et les caractéristiques de la sous-partie du monde impactée par la crise (les populations concernées, les sites naturels, les dangers et spécificités de la zone, etc.). Les informations sur les compétences des organisations, décrivent quant à elles, d'une part les aptitudes des organisations, au sens des compétences mobilisables (par exemple, pour le SDIS<sup>8</sup>, la faculté d'arroser une zone pour prévenir un risque d'incendie) et d'autre part, les renseignements « techniques » relatifs aux services du système d'information correspondant à cette compétence (il s'agit en particulier des informations permettant d'invoquer un service ou une application informatique correspondant à l'interface de telle ou telle activité métier). Comme évoqué précédemment, une hypothèse forte de nos travaux concerne le fait que pour chaque compétence opérationnelle d'un acteur, son système d'information puisse proposer un service informatisé permettant de l'invoquer. En effet, l'objectif du pilotage du processus collaboratif est de permettre l'appel successif, et à propos, de ces activités. Or, dans le cas où ce service est une application informatique (par exemple un service de météo ou un automate d'appel téléphonique), cette hypothèse est vérifiée nativement. Cependant, dans le cas d'un service plus « terrain », cette hypothèse est plus ambiguë. Pourtant, dans le cas d'un tel service terrain, nous proposons de déployer un service interface qui communiquera avec le représentant humain de l'acteur au sein de la cellule de crise. Cette interface sera un service simple dont l'objectif sera uniquement d'informer le représentant humain de la nécessité d'exécuter une telle action (par exemple « éteindre un incendie »), en lui fournissant les éventuelles informations nécessaires préalablement définies (par exemple : localisation, nature de l'incendie, superficie, etc.), puis d'attendre que ce même représentant humain confirme (après

---

<sup>8</sup> Services Départementaux d'Incendie et de Secours

réception des informations provenant du terrain) la bonne exécution de l'action (en fournissant éventuellement d'autres informations, nécessaires pour la suite de l'exécution du processus collaboratif). La réalisation d'un tel service d'interface est une tâche relativement aisée dans un contexte SOA, de plus, cette vision est parfaitement compatible avec le fait que les acteurs sont supposés pouvoir se présenter avec le niveau de granularité fonctionnelle qu'ils souhaitent (*cf.* section II.2, item 1). Cette hypothèse nous permet de plus de laisser de côté les problèmes de correspondance sémantique soulevés lors de la présentation du projet MISE.

Cependant, cette approche fait apparaître d'autres problèmes sémantiques. En effet, chaque organisation décrit ses compétences dans un langage qui lui est propre et potentiellement différent d'une part de celui utilisé par les autres organisations et d'autre part de celui utilisé pour caractériser la situation. Prenons un exemple simpliste mais illustratif. Lors d'une crise, un moteur prend feu, cette information est capitalisée sous la forme suivante : « incendie moteur ». En observant la description des compétences des acteurs, on remarque qu'aucune d'entre elles ne permet de résoudre ce problème spécifique. Néanmoins, le SDIS propose deux compétences : l'une consistant à éteindre un incendie de faible ampleur, l'autre un incendie de grande ampleur. Selon les circonstances, l'incendie moteur correspondra soit à un incendie de faible ampleur, soit à un incendie de grande ampleur. Par conséquent, le lien sémantique entre l'incendie moteur et un incendie de faible ou grande ampleur ne pourra se faire qu'au moment de la création du modèle de crise. Pour répondre à ce problème de création de lien de correspondance sémantique au moment où il est limitant, un embryon de réponse a été mis en place avec une interface utilisateur de création à la volée de liens d'équivalence ou de proximité entre les termes utilisés.

L'ensemble des informations recueillies sur la situation et sur les compétences des acteurs est ensuite injecté dans une base de connaissances, non pas de référence mais comportant uniquement les informations propres à la situation. Un ensemble de règles de déduction applicables sur cette base de connaissance permet alors de déterminer les services pouvant être utilisés dans le cadre de la réponse à la crise. Ces services sont ensuite organisés de façon à proposer un processus collaboratif, nécessitant une étape de vérification et de validation, afin de s'assurer que l'utilisation de ce processus collaboratif n'aura pas d'impact néfaste sur la crise. Cette évaluation de la robustesse du processus collaboratif est le rôle de l'AADE (approche anticipative dirigée par les effets), approche de vérification basée sur les effets résultant des travaux du LGI2P de l'École des Mines d'Alès (Daclin *et al.*, 2009) qui ne sera pas exposée dans ce manuscrit. Finalement, Le processus collaboratif validé correspond au modèle relatif à la couche « métier », le modèle CIM de la démarche. Les détails de cette étape métier seront fournis dans le chapitre II.

## VI.2 L'axe exécution du projet ISyCri

L'axe d'exécution du projet s'intéresse à l'adaptation du SIM en fonction des variations susceptibles de se produire durant la réponse à une crise. Selon le type de variations, l'impact sur le SIM ne sera pas le même. Il est donc nécessaire de déterminer les différents types d'évolutions possibles et leurs impacts sur le besoin d'évolution du SIM. Pour cela, nous avons décidé de nous baser sur les trois types de changement au sein d'un réseau industriel proposés dans (Pingaud, 2009) :

- Variations dans la mission d'un réseau,
- Variations dans la structure des systèmes contributeurs ou du réseau,
- Variations du comportement dans les échanges au sein du réseau.

Reprenant cette classification, nous la transposons aux évolutions possibles dans le cadre de la réponse à une crise. Nous proposons de regrouper les deux derniers types de changements et de ne retenir que deux types d'évolutions pour les situations de crise :

- Des évolutions de la situation elle-même : les caractéristiques perçues de la crise, et en particulier les problèmes à traiter, ne sont plus les mêmes et nécessitent un traitement différent.
- Des évolutions de la réponse à apporter : le pilotage de la résolution de la situation de crise doit s'adapter suite à (i) une évolution de la composition de la cellule de crise (arrivée, départ, etc.), (ii) un dysfonctionnement lors de l'exécution d'un service (il faut impérativement interrompre et tenir compte du dysfonctionnement sans poursuivre aveuglément le déroulement du processus) ou (iii) du fait de la définition initiale partielle du processus de réponse (choix retardé).

En fonction des évolutions, le SIM ne sera évidemment pas modifié de la même façon. En effet, lors d'une évolution de la situation elle-même et des caractéristiques perçues, l'ensemble de la chaîne de conception doit être ré-exécutée et un nouveau SIM doit être déduit. Dans le cas d'une évolution de la réponse, plusieurs modifications peuvent être envisagées : la plus lourde d'entre elles, consiste à recommencer la chaîne de conception au niveau de la déduction du processus collaboratif, une autre se propose de redemander l'exécution d'une ou plusieurs tâches, enfin la dernière consiste à compléter la définition partielle de la réponse. Nous en déduisons qu'afin de réaliser l'adaptation d'un SIM en fonction des évolutions, il est nécessaire de répondre à ces deux questions : comment détecter les évolutions ? Comment modifier le SIM en fonction de l'évolution ?

La première question revient à déterminer si la réponse à la crise en cours d'exécution est toujours adaptée à la situation. Pour cela, il est nécessaire d'avoir, à tout moment, d'une part toutes les informations provenant du terrain afin de pouvoir détecter si la situation a changé et d'autre part, une vision actualisée et précise sur l'avancement du processus de réponse. Pour cela, trois outils ont été créés. Les deux premiers servent à la remontée d'informations, le troisième résume ces informations et aide aux détections d'évolutions :

- **EarlyWarning** : Remontée des informations réelles de la situation. Cet outil a été développé afin que chaque organisation puisse relayer à l'ensemble des personnes de la cellule de crise, une nouvelle information sur la situation provenant des acteurs sur site. Il s'agit d'une interface particulière et autonome, accessible depuis chacun des SI des acteurs de la cellule de crise, et lui permettant à tout moment, pour peu qu'il soit informé d'un changement par les opérationnels avec lesquels il est en liaison, de saisir cette information particulière afin qu'elle soit affichée et stockée indépendamment des autres activités du processus collaboratif (on peut assimiler cet outil à un service sophistiqué de discussion, parallèle à l'avancement du processus). Cet outil permet donc à la cellule de crise de détecter les évolutions de la situation.
- **Service de médiation** : Suivi en temps réel de l'avancement du processus de réponse. Ce service, présenté en détails dans le chapitre IV et V de ce manuscrit, a été conçu dans un premier temps dans le but de pouvoir transférer au bon format les informations entre services. Se basant sur une table de correspondance, pour l'instant statique, il transforme les éléments de sortie d'un service en élément d'entrée d'un autre. Ce service est donc utilisé avant chaque action d'une organisation, ce qui lui confère la fonctionnalité secondaire (en plus de son rôle d'interprète) de permettre de savoir si les actions sont en cours, en attente ou finies, voire mal exécutées.
- **Fiche de suivi** : Recueil des informations et aide à la détection des évolutions. Cet outil, présenté en détail dans le chapitre V, peut être assimilé à une main courante qui s'actualise automatiquement. En



effet, les informations sur le statut des actions des organisations y sont résumées. En fonction de ces statuts, l'état théorique des problèmes de la crise est déduit. L'état théorique d'un problème est « résolu » si l'ensemble des actions devant le traiter est fini, « en cours » si au moins une activité a commencé, et « présent » si aucun service n'a commencé. Cet état théorique est alors comparé à l'état réel du problème provenant des informations du terrain par l'intermédiaire de l'outil EarlyWarning. Cette comparaison permet de détecter des écarts témoignant d'éventuels dysfonctionnements lors du déroulement de la réponse à la crise.

### VI.3 Synthèse sur le projet ISyCri

La figure I-9, ci-dessous, résume la vue d'ensemble du projet ISyCri. On retrouve les différents niveaux d'abstraction (métier, logique, technique et déploiement) d'une ingénierie dirigée par les modèles. Concernant le niveau métier, l'action de déduction du processus de réponse englobe à la fois la déduction du processus collaboratif puis l'étape de validation de ce dernier. Le passage d'un niveau à un autre est permis grâce à une transformation de modèles. Cet axe de conception permet de déployer un SIM. Ce SIM est représenté au niveau de l'exécution.

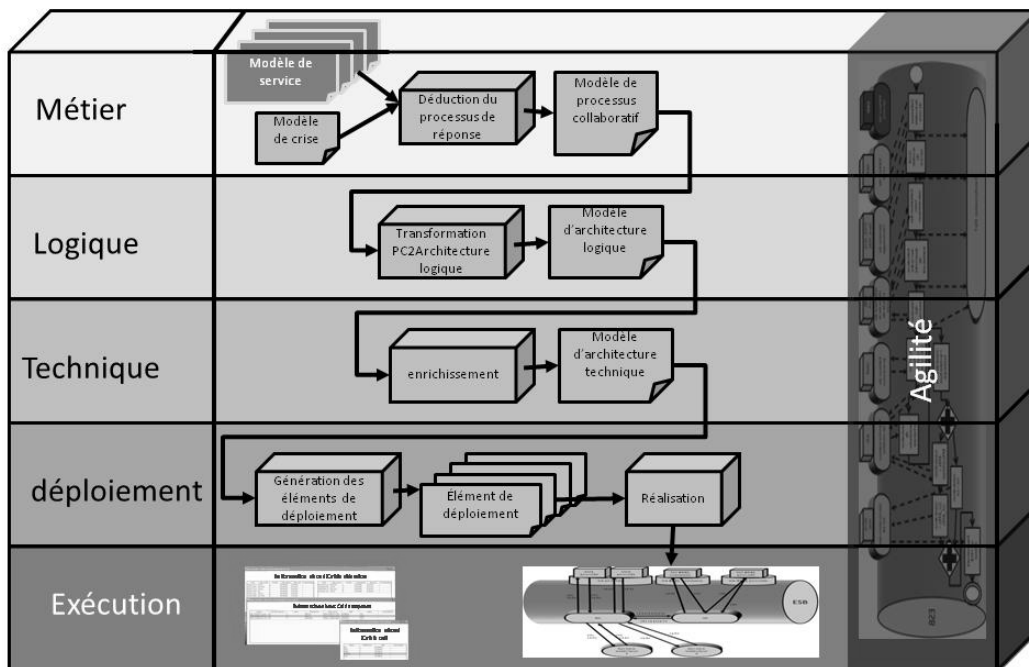


Figure I- 10 : Vue d'ensemble du projet ISyCri

En parallèle à l'exécution du SIM, une fiche de suivi est utilisée comme tableau de bord afin de détecter d'éventuelles évolutions lors de la réponse. Cette fiche de suivi est mis à jour d'une part, par les informations transmises du terrain grâce à EarlyWarning et d'autre part, par l'avancement de la réponse à la crise. En fonction des évolutions détectées, il peut s'avérer nécessaire d'arrêter le processus en cours ou de le définir partiellement. Ces deux facultés reposent sur un moteur d'orchestration, développé dans le cadre du projet, et sur le protocole ContractNet (Faure *et al.*, 2010). D'autres évolutions peuvent impliquer de redémarrer la chaîne de conception à différents niveaux. Il est donc nécessaire que la chaîne de conception offre cette souplesse d'exécution. Afin de permettre cette agilité, les différents outils développés à chacun des niveaux de la chaîne de conception (résultant des thèses de J. Touzi et V. Rajsiri, du post-doc de J. Touzi et du Master de W. Mu), ont, pour les besoins du projet, été implantés, sous forme de WebServices, au sein d'un ESB PEtALS, offrant ainsi la souplesse et la modularité inhérentes à ce type d'architecture. Le moteur d'orchestration se charge alors de l'appel successif des différents outils,

conformément à la structure de la chaîne de conception, préalablement décrite en BPEL. Cet ESB de conception est représenté sur la droite de la figure I-9.

## VII. Conclusion

Dans ce chapitre, nous avons voulu atteindre deux objectifs principaux : le premier concerne la délimitation de l'espace dans lequel se positionnent nos travaux, le deuxième concerne la présentation des problématiques à traiter et une présentation des solutions que nous proposons.

Le domaine de la gestion de crise et plus précisément de la réponse à la crise est particulièrement intéressant pour des travaux portant sur l'interopérabilité. En effet, essayer de supporter l'interopérabilité entre les différentes organisations devant travailler ensemble dans l'urgence afin de résoudre, ou au moins réduire, une crise est un challenge intéressant pour deux raisons : (i) la coopération entre les organisations doit être rapidement instaurée, que les organisations se connaissent ou non et (ii) la crise est par définition un phénomène évolutif, ce qui peut remettre en cause à tout moment la coopération et nécessiter son adaptation. La première raison favorise la création du support d'interopérabilité *via* une démarche de fédération afin de pouvoir créer un système de systèmes garantissant l'indépendance des organisations. Ce système de systèmes repose sur des médiateurs dont le rôle est de coordonner les activités des organisations et de partager les informations. Ce système de systèmes donne ensuite naissance à un système de systèmes d'information suite à sa projection sur les systèmes d'information des différentes organisations. Le médiateur, quant à lui, donne naissance à un système d'information de médiation (SIM) dont le but est d'exécuter le processus collaboratif de réponse à la crise et de transformer les informations afin que la bonne information, au bon format, soit disponible au bon moment à la bonne organisation. Notre problématique se focalise alors sur la démarche de création de tels SI de médiation.

Des travaux ont été réalisés dans le cadre du projet MISE afin de pouvoir déduire d'une caractérisation d'une collaboration inter-entreprises, un SIM adapté à la situation. Néanmoins, dans le cadre du projet ISyCri, il était nécessaire d'adapter ces travaux afin de pouvoir les appliquer à une situation de crise. Les travaux de thèse décrits dans ce manuscrit relèvent de cette adaptation et portent sur plusieurs points :

- La déduction d'un processus collaboratif de réponse à partir des informations disponibles à la fois sur la situation de crise et sur le savoir-faire des différents acteurs : Ces travaux se distinguent de ceux réalisés dans le cadre du projet MISE (Rajsiri, 2009) car ils ne s'appuient pas sur une base de connaissance de référence du domaine et par conséquent générique. Ce premier point, décrit dans le chapitre II, répondra aux trois premières interrogations principales liées à la problématique, à savoir : Comment coordonner les actions des organisations sélectionnées afin de résoudre la crise à partir de la confrontation des informations liées, d'un côté à la description des caractéristiques de la crise, et de l'autre du savoir faire des organisations.
- La réalisation d'une transformation de modèle partant du modèle d'architecture technique du SIM vers un fichier BPEL, décrivant le processus collaboratif (au niveau déploiement) : Cette transformation constitue un résultat à la fois du projet MISE et du projet ISyCri (il s'agit d'un composant commun factorisé). Au niveau de cette transformation un mécanisme de gestion de l'information a été ajouté. Cette gestion vise à prendre en charge l'acheminement et la traduction des données afin de transmettre la bonne information, au bon format, au bon moment, à la bonne organisation. L'explication de cette transformation relève du chapitre I.
- Les réalisations d'une fiche de suivi et d'un service de médiation dans le but de permettre au SIM de s'adapter aux évolutions. Ce dernier point donnera lieu au chapitre IV.

## Chapitre II :

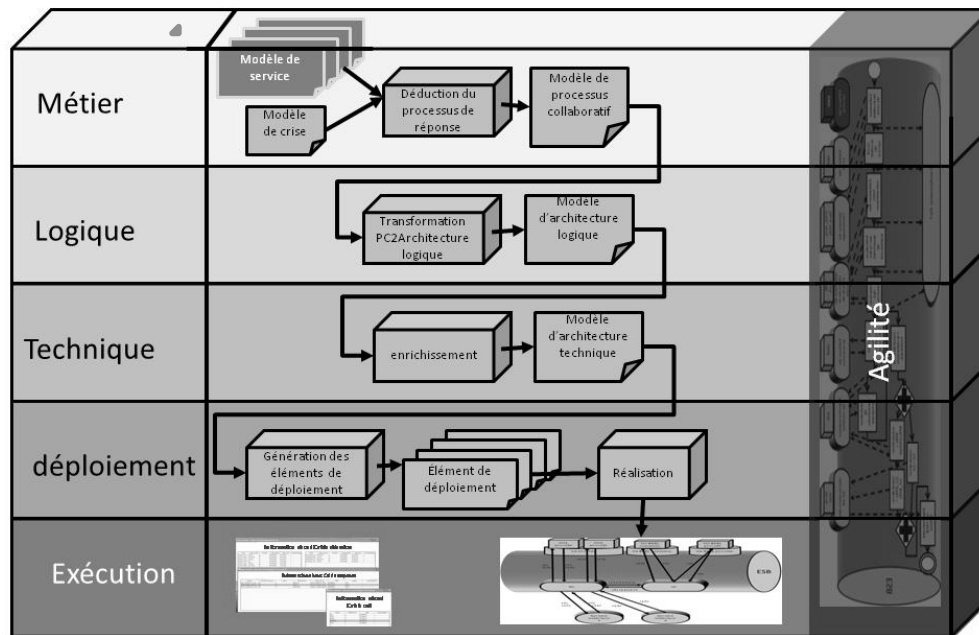
# De la situation de crise au processus de réponse

## *Caractérisation et obtention du CIM*

<b>Introduction du chapitre.....</b>	<b>32</b>
<b>I. Métamodèle de crise .....</b>	<b>33</b>
I.1 La situation.....	33
I.2 Le système de traitement .....	36
I.3 Le processus collaboratif .....	37
I.4 Liens entre les différentes parties du métamodèle.....	39
<b>II. Démarche de déduction du processus collaboratif .....</b>	<b>40</b>
II.1 Etape (2) : confrontation des différents modèles.....	42
II.2 Règles de transformation .....	43
II.3 Etape (3) : construction du processus collaboratif .....	47
<b>III. Illustration de déduction d'un processus collaboratif .....</b>	<b>57</b>
III.1 Présentation du contexte .....	57
III.2 Déduction des composants du processus collaboratif.....	60
III.3 Déduction de l'ordre d'exécution des services .....	63
<b>IV. Conclusion.....</b>	<b>66</b>

## Introduction du chapitre

Ce chapitre se focalise sur la première étape de déduction d'un système d'information de médiation en charge de supporter la collaboration entre les partenaires hétérogènes de la réponse à la crise. Cette étape, représentée par un cercle sur la figure II-1, consiste à déduire, à partir des informations recueillies à la fois sur la situation de crise et sur les caractéristiques des services métiers des partenaires, un processus collaboratif de réponse à la crise. Ce processus collaboratif correspond au niveau CIM de la démarche de conception du système d'information de médiation.



**Figure II- 1 : positionnement du chapitre vis-à-vis de l'approche de conception d'un système d'information de médiation**

La démarche de déduction du processus collaboratif de réponse à la crise repose sur l'utilisation d'une base de connaissance dont le contenu évolue en fonction des informations de la situation. Le but de cette démarche étant de déduire un processus de réponse, nous limiterons le contenu de la base de connaissance aux informations relatives à la situation permettant d'atteindre cet objectif, c'est-à-dire d'une part aux informations portant sur les savoir-faire des partenaires disponibles et d'autre part, aux informations décrivant la situation de crise. De plus, la nécessité de recueillir ces informations de façon structurée afin de permettre leur injection au sein de la base de connaissance, nous a conduit à représenter ces informations sous forme de modèles (cf. définition de la notion de modèle au chapitre II).

Pour cela, deux éditeurs graphiques ont été réalisés grâce à l'environnement GMF (Graphical Modeling Framework) (GMF, 2005). L'avantage de l'utilisation de cet outil est qu'il permet de créer une interface graphique de construction de modèle reposant sur un langage de modélisation défini spécifiquement pour le domaine considéré. Ces considérations relèvent pleinement du concept de DSL (Domain Specific Language) qui se trouve donc ainsi sollicité lors de cette étape de la démarche. Cet avantage nous assure de la cohérence entre les différents modèles. Comme détaillé dans (Truptil *et al.*, 2008), le premier éditeur permet de récolter les informations portant sur la crise. Il est utilisé au début de la phase de réponse à la crise contrairement au second éditeur qui est utilisé lors de la phase de préparation. Ce second éditeur permet de récolter les informations liées aux savoir faire d'un partenaire. Il peut donc exister au moins un modèle par partenaire.

La première étape de la démarche consiste donc à injecter l'ensemble des informations contenues dans les différents modèles (caractérisation de la crise et définition des fonctions métiers mobilisables) au sein de la base de connaissance. Ces différents modèles, construits à partir des mêmes DSL (dédié pour l'un à la couverture des informations portant sur la crise et pour l'autre à permettre la modélisation des caractéristiques des services métiers des participants) permettent alors d'instancier et de peupler une ontologie, afin de pouvoir manipuler ces informations, issues de sources hétérogènes (Grimm *et al.*, 2007). Les différents modèles dont il est question dans les lignes précédentes (relatifs à la situation ou aux services des acteurs disponibles), comme l'ontologie dédié au recueil de cette masse de connaissance repose sur un métamodèle de situation de crise. S'il est clair que les modèles de services, le modèle de caractérisation de crise, tout comme l'ontologie ne font référence ni de la même façon, ni même à des sous-parties similaires de ce métamodèle, il n'en demeure pas moins qu'ils en découlent néanmoins tous. Ce métamodèle, et c'est là un point très important de cette première étape de la démarche, ambitionne donc de couvrir les différents sous-domaines inhérents à l'obtention d'une vision pertinente de la globalité de la situation de crise (concernant à la fois la situation de crise, l'espace physique impacté, les partenaires mobilisables et leurs compétences, etc.).

Afin de présenter au mieux cette démarche de conception d'un système d'information de médiation, nous avons décidé de décomposer ce chapitre de la façon suivante : la première partie s'intéresse à décrire le métamodèle de crise, fruit du projet ISyCri (il ne s'agit pas à proprement parlé d'un résultat des travaux de thèse dont il est ici question). Cette description a pour objectif d'une part d'avoir une vue d'ensemble des concepts utilisés et, d'autre part, de poser les bases permettant la mise en place du processus de transformation. L'élaboration de ce processus de transformation, présentée dans la deuxième partie de ce chapitre, est décomposée en plusieurs étapes, dont certaines se basent sur l'exécution de règles de déduction. Ces règles sont de plusieurs natures : les premières servent à résoudre des problèmes sémantiques afin que l'ensemble des partenaires aient une compréhension mutuelle, alors que les secondes servent essentiellement à compléter le modèle de processus collaboratif. Cette construction est finalement illustrée à travers un exemple dans la dernière partie de ce chapitre.

*N.B.* : Dans ce chapitre est les suivants, les termes *gateway*, *pool*, *lane*, *sequence flow*, *message flow* ne seront pas traduits en français car ils relèvent d'un usage commun par les utilisateurs du formalisme BPMN.

## I. Métamodèle de crise

L'étape de la démarche de conception d'un système d'information de médiation que nous expliquons dans ce chapitre repose sur un métamodèle : le métamodèle de crise. Ce métamodèle, résultat du projet ISyCri, a été construit sur la base de connaissances récoltées à partir de l'étude de différentes crises et du management des risques. Ce métamodèle, représenté selon le formalisme UML par la figure II-7, est découpé en trois parties : la situation (composée du système d'étude et des caractéristiques de la crise), le système de traitement et enfin le processus collaboratif. Chacune de ces parties se base sur un point de vue différent du système, regroupant la crise ainsi que la réponse mise en place.

### I.1 La situation

Les informations portant sur la situation sont obligatoirement liées à la sous-partie du monde affectée par la crise. Plus précisément, une crise correspond à l'instabilité d'un « univers de référence » provoquée par la concrétisation de risques, pouvant être aussi appelés vulnérabilités du système. La récolte des informations liées aux risques est donc nécessaire, sans compter que la présence, ou l'absence, de certains risques aura une influence sur la composition de la réponse à la crise.

Il est difficile de donner une définition précise de la notion de risque tant les domaines d'applications, et par conséquent la littérature sur ce sujet, sont vastes et variés. Nous pouvons néanmoins citer la définition de (Gourc, 2006) appliquée au management de projet. Le risque y est défini comme : « la possibilité que survienne un événement dont l'occurrence entraînerait des conséquences (positives ou négatives) sur le déroulement de l'activité du projet ». Cette définition sert de base à la proposition du modèle générique du risque proposé par (Sienou, 2009) (figure II-2). L'auteur y considère le risque selon deux dimensions définies au sein d'un contexte particulier et reliées par une notion de causalité : une dimension événementielle (représentant les causes du risque) et une dimension état du système (représentant les conséquences du risque).

Le contexte du risque y désigne l'environnement de perception et de définition du risque. Cet environnement comporte des facteurs de risque capables d'influencer, de créer un événement concrétisant un risque. Ces événements risqués créent des conséquences sur l'environnement pouvant avoir un impact positif ou négatif sur celui-ci.

Dans cette définition, (Sienou, 2009) introduit la notion de contrôle correspondant à une option de traitement pouvant porter sur la cause ou la conséquence du risque.

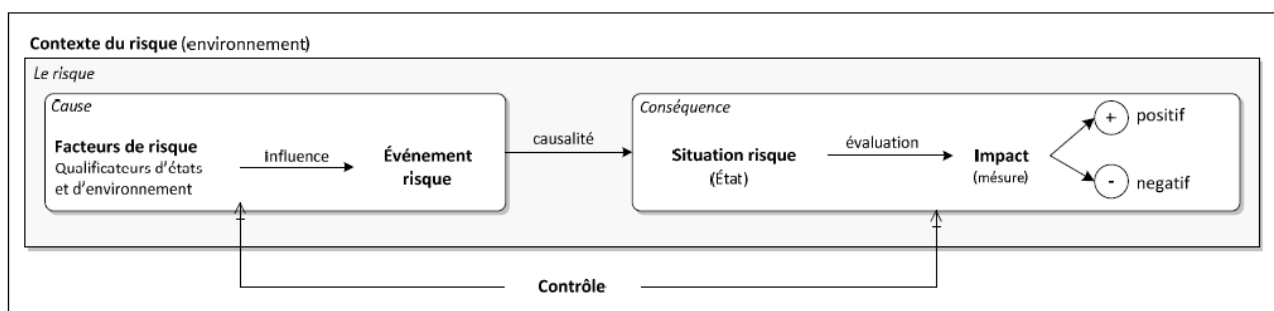


Figure II- 2 : modèle générique du risque (Sienou, 2009)

En nous inspirant de ces propositions, nous avons décidé de diviser les informations portant sur la situation de crise en deux parties : la partie système d'étude s'intéresse plus particulièrement à la notion de cause du risque, alors que la partie caractérisation de la crise s'intéresse aux conséquences sur l'environnement.

Afin d'éviter des problèmes de compréhension dans la suite de ce chapitre, nous entendons par risque : un élément né d'un facteur de risque, correspondant à une particularité de l'environnement, qui pourra voir son impact sur l'environnement se concrétiser suite à l'apparition d'événements (dits déclencheurs). Cette concrétisation donnera naissance à une ou des conséquences sur l'environnement.

### I.1.1 Le système d'étude

La partie système d'étude, orientée vers la cause de la crise, doit permettre de récupérer les informations liées aux facteurs de risques ou vulnérabilités de l'environnement.

Dans le cadre du projet ISyCri, nous avons décidé de définir les facteurs de risque comme des *caractéristiques*. Une *caractéristique* correspond à une particularité de l'environnement pouvant engendrer un ou plusieurs *risques*. Par exemple, une caractéristique d'instabilité sismique peut engendrer le risque de tremblement de terre. Lier à la notion de *risque*, la notion d'*événement* est vu comme l'instant de concrétisation du risque et donc le moment d'apparition de conséquences. L'environnement est donc impacté par ces conséquences, plus précisément ce sont les *composants systèmes d'étude (CSE)* caractérisant l'environnement qui sont impactés. Ces composants peuvent être regroupés en différentes catégories comme les *biens*, correspondant à ce qui est produit par les hommes, contrairement aux *sites*

*naturels*. Les *populations* concernent l'ensemble des groupes de personnes pouvant être impactés par la crise. Les *sociétés civiles*, quant à elles, représentent des entités sociales ayant une influence sur la situation, sur sa perception et sur son évolution, par exemple : les médias, les instances politiques, etc.

## I.1.2 Les caractéristiques de la crise

La partie du métamodèle portant sur les caractéristiques de la crise s'intéresse aux conséquences liées à la concrétisation des risques. On y retrouve les notions de *crise*, *conséquences*, *facteur de gravité* et *complexité*.

Une *conséquence* représente une concrétisation notable d'un risque créée par un événement. Un *facteur de gravité* est une caractéristique, ou facteur de risques, qui peut directement impacter la gravité de la crise. Par exemple, la présence de vent (respectivement pluie) est un facteur de gravité négatif (respectivement positif) pour une crise de type incendie de forêt. Un *facteur de complexité* est une caractéristique impactant directement la nature de la crise, par exemple une instabilité sociale ou politique peut amener une crise à changer de type, comme ce fut le cas lorsque la médiatisation de la canicule de 2003 transforma cette crise sanitaire en une crise devenue sociale.

La figure suivante présente la partie du métamodèle décrite dans les deux sections précédentes (II.1.1 et II.1.2). Il s'agit d'une sous-partie du métamodèle global du projet ISyCri.

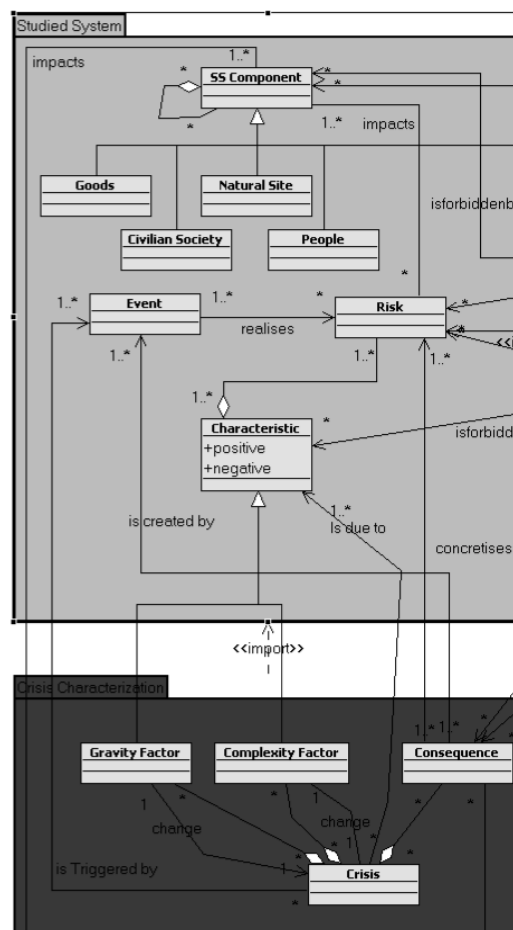


Figure II- 3 : parties système d'étude et caractéristiques de crise du métamodèle

## I.2 Le système de traitement

Cette partie du métamodèle vise à regrouper les informations nécessaires à la détermination d'une réponse à une crise. Pour rappel (cf. chapitre I), une réponse à la crise est définie comme « l'ensemble des mesures prises pour résoudre des problèmes causés par la survenue d'une crise » (Devlin, 2006). Par conséquent, la définition d'une réponse à la crise s'articule autour de la notion de « mesure prise ». Mesures qui s'illustrent concrètement au travers de l'invocation de *services*, correspondant aux services métiers présentés dans le chapitre II.

Nous allons voir que cette articulation de la réponse autour de la notion de *service* se retrouve au niveau de la partie système de traitement du métamodèle de crise. Pour cela, nous commencerons par caractériser les services. (Bakir, 2003) distingue classiquement deux types de services permettant de prendre en charge le traitement des risques :

- préventifs : services pouvant réduire la probabilité d'occurrence ou l'impact d'un risque. Ces services s'exécutent avant l'occurrence du risque,
- correctifs : services pouvant réduire la conséquence de la concrétisation d'un ou plusieurs risques.

Un *service* peut donc être réalisé dans le but de prévenir un risque ou de réduire une conséquence d'un tel risque. Pour cela, il utilise des *ressources* humaines ou matérielles. Il est cependant à noter que certains services nécessitent un état particulier du système d'étude ou du système de réponse afin d'être menés à bien, nous parlerons alors de *condition*. Les *conditions* représentent un état du système, elles peuvent être réalisées ou nécessitées par un service. Elles peuvent également être implicites et systématiques entre deux services (*i.e.* l'exécution du service A est une condition incontournable à l'invocation du service B). Dans ce cas, les liens de nécessités seront directement définis statiquement entre ces services.

Nous avons évoqué les conditions de réalisation d'un service sans évoquer les cas de conflit et d'interdiction. En effet, s'il est évident que certains risques et conséquences puissent interdire l'invocation de services incompatibles, il est également possible que l'exécution d'un service en interdise un autre. En effet, dans le chapitre I, nous avons introduit la possibilité que certains services puissent être antagonistes, notamment au niveau des ressources, mais pas seulement.

Cet ensemble de concept et relations, représenté par la figure II-3, servira de base à la déduction d'une dynamique organisée de réponse à une crise devant être supportée par un système d'information de médiation (SIM). Ce SIM, en plus de garantir l'interopérabilité entre les différents partenaires, doit également proposer ses propres services (support du fonctionnement collaboratif ou à valeur ajoutée dans la dynamique collective de réponse). Par conséquent, nous distinguerons deux types de *services* : les *services d'acteurs* et les *services de médiation*. Les services d'acteurs correspondent aux services métiers des partenaires, alors que les services de médiation correspondent aux services du SIM, tel qu'un service d'orchestration ou encore des services de coordination en charge d'invoquer les services des acteurs tout en gérant les informations.





- seul le médiateur est en charge de « dérouler » le processus collaboratif, la définition de la succession des activités doit donc être uniquement attribuée au médiateur (même si elle correspond pratiquement à une succession d'activités réalisées concrètement par les partenaires).
- Chaque acteur doit être représenté indépendamment. Les activités des différents acteurs ne peuvent pas être directement reliées, une transition par le médiateur est obligatoire (cette restriction, indissociable des fonctions d'orchestration et de traduction, est uniquement fonctionnelle et s'avère généralement complètement transparente).
- Chaque message envoyé et reçu par un acteur doit être traité afin de gérer l'information (en particulier la gestion des traductions et des formats), ce traitement se fait au niveau du médiateur.

Ce type particulier de processus s'apparente au processus défini et nommé processus collaboratif dans les travaux de thèse de (Touzi, 2007). Le métamodèle de la figure II-5, extrait du métamodèle de crise et défini par (Touzi, 2007), décrit ce processus collaboratif.

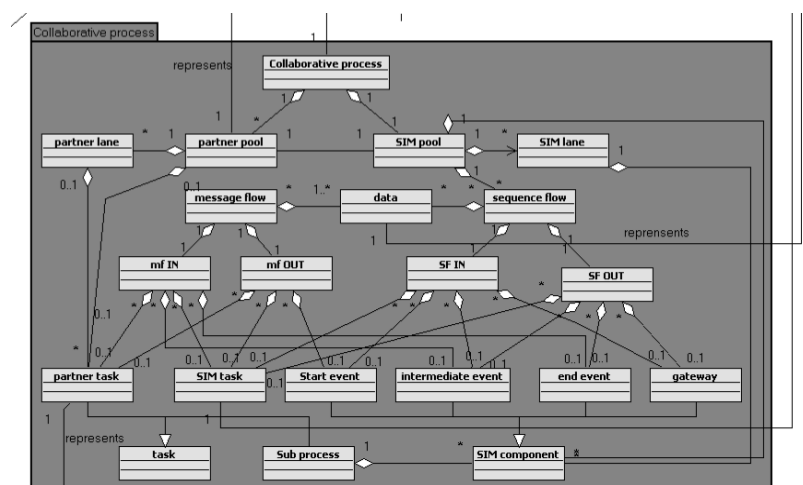


Figure II- 5 : métamodèle de processus collaboratif

Le processus collaboratif y est composé :

- d'un et unique pool, nommé *SIM\_pool*, représentant le médiateur. L'ensemble des opérateurs logiques du processus (*sequence\_flow*, *gateway*, *event*, etc.) définissant la dynamique de la succession d'activités ne peuvent appartenir qu'à cet unique pool.
- Chaque partenaire de la collaboration est représenté par un pool, ce pool ne peut contenir que des lanes, chacune représentant une division du partenaire, ne pouvant elles-mêmes contenir que des activités, nommées *Partner\_task*.
- Le service d'un partenaire, représenté par un *Partner\_task*, peut être mis en relation *via* des *messages flows* avec un service du médiateur, nommé *SIM\_task*. Ces *messages flows* modélisent le séquençement et le transfert d'information entre le partenaire et le médiateur. Un service du médiateur est utilisé afin de transformer l'information : (i) une première fois avant l'invocation d'un service d'un partenaire afin que les informations soient envoyées dans le bon format, puis (ii) une deuxième fois, lors de la récupération des informations reçues du service du partenaire dans le but de stocker l'information. La figure II-6 résume ce fonctionnement. Dans ces travaux de thèse, nous émettons l'hypothèse qu'il y a toujours un envoi et un retour d'informations entre le médiateur et le service du partenaire. Même si le service ne renvoie pas de nouvelles informations, nous supposons qu'il renvoie au minimum une information de fin.

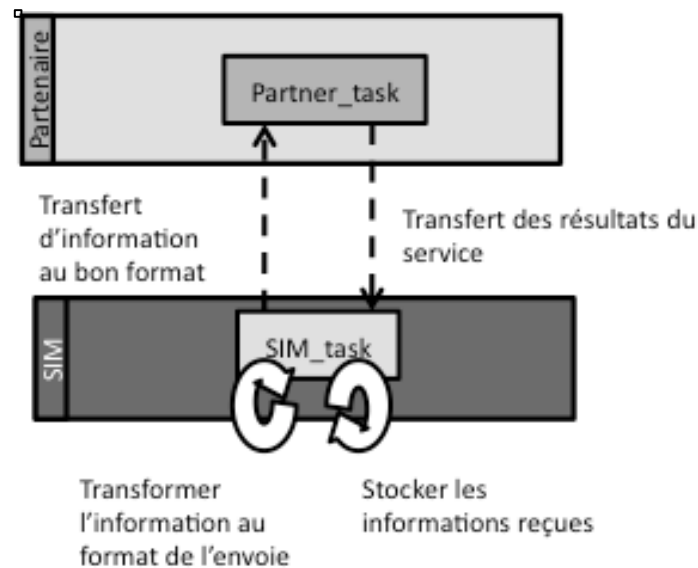


Figure II- 6 : lien entre SIM\_task et Partner\_task

## I.4 Liens entre les différentes parties du métamodèle

Les différentes composantes du métamodèle, que nous venons de décrire, sont reliées entre elles par des relations structurelles et sémantiques. Ces liens seront la base du processus de transformation permettant la déduction du processus collaboratif à partir des informations de la situation. Les liens entre les différentes parties du métamodèle sont les suivants :

- entre la situation  $S$  (système d'étude et caractérisation de la crise) et le système de traitement  $ST$  :
  - les services ( $\in ST$ ) sont reliés aux risques ( $\in S$ ) par des relations définissant qu'un service peut prévenir un risque, créer un risque ou être interdit par l'existence d'un risque.
  - Les services ( $\in ST$ ) sont reliés aux conséquences ( $\in S$ ) par des relations définissant qu'un service peut réduire une conséquence, ou être interdit par l'existence d'une conséquence.
  - Naturellement, un service ( $\in ST$ ) peut impacter des composants système d'étude ( $\in S$ ). Il peut, par exemple, détruire des biens ( $\in S$ ) ou déplacer des populations ( $\in S$ ). L'utilisation d'un service ( $\in ST$ ) peut aussi être interdite par un composant du système d'étude ( $\in S$ ). En effet, la présence de population ( $\in S$ ) peut empêcher l'utilisation de canadiers ( $\in ST$ ).
  - L'interdiction de l'utilisation d'un service ( $\in ST$ ) peut aussi être due à la présence d'une caractéristique ( $\in S$ ), d'un facteur de complexité ( $\in S$ ) ou d'un facteur de gravité ( $\in S$ ). Par exemple, la présence de média (facteur de complexité  $\in S$ ) peut interdire l'utilisation de service ( $\in ST$ ) pouvant risquer de produire une mauvaise image des secours.
- Entre le système de traitement  $ST$  et le processus collaboratif  $PC$  : ces liens déjà évoqués dans la partie de description du processus collaboratif sont les suivants :
  - chaque acteur de la réponse ( $\in ST$ ) est représenté par un Partner\_pool ( $\in PC$ ).
  - Selon la nature du service ( $\in ST$ ), ses liens avec le processus collaboratif diffèrent. Un service d'acteur ( $\in ST$ ) correspondra à un Partner\_task ( $\in PC$ ), alors qu'un service de médiation ( $\in ST$ ) correspondra à un SIM\_task ( $\in PC$ ).

Il n'existe pas de lien entre la situation  $S$  et le processus collaboratif  $PC$ , car le cheminement présenté dans la section précédente implique, d'une part, qu'il soit proposé de traiter la situation par le biais de la mise en place d'un système de traitement (ce qui explique la présence de liens entre  $S$  et  $ST$ ) et d'autre part, que ce système de traitement soit implémenté par l'intermédiaire de la mise en œuvre d'un processus collaboratif (ce qui explique la présence de liens entre  $ST$  et  $PC$ ). Ce mécanisme « en cascade » fait qu'il n'y a pas directement de lien entre  $S$  et  $PC$ , puisque leurs connexions implicites sont en fait le résultat d'une transitivité *via*  $ST$ . La figure suivante présente le métamodèle dans son ensemble.

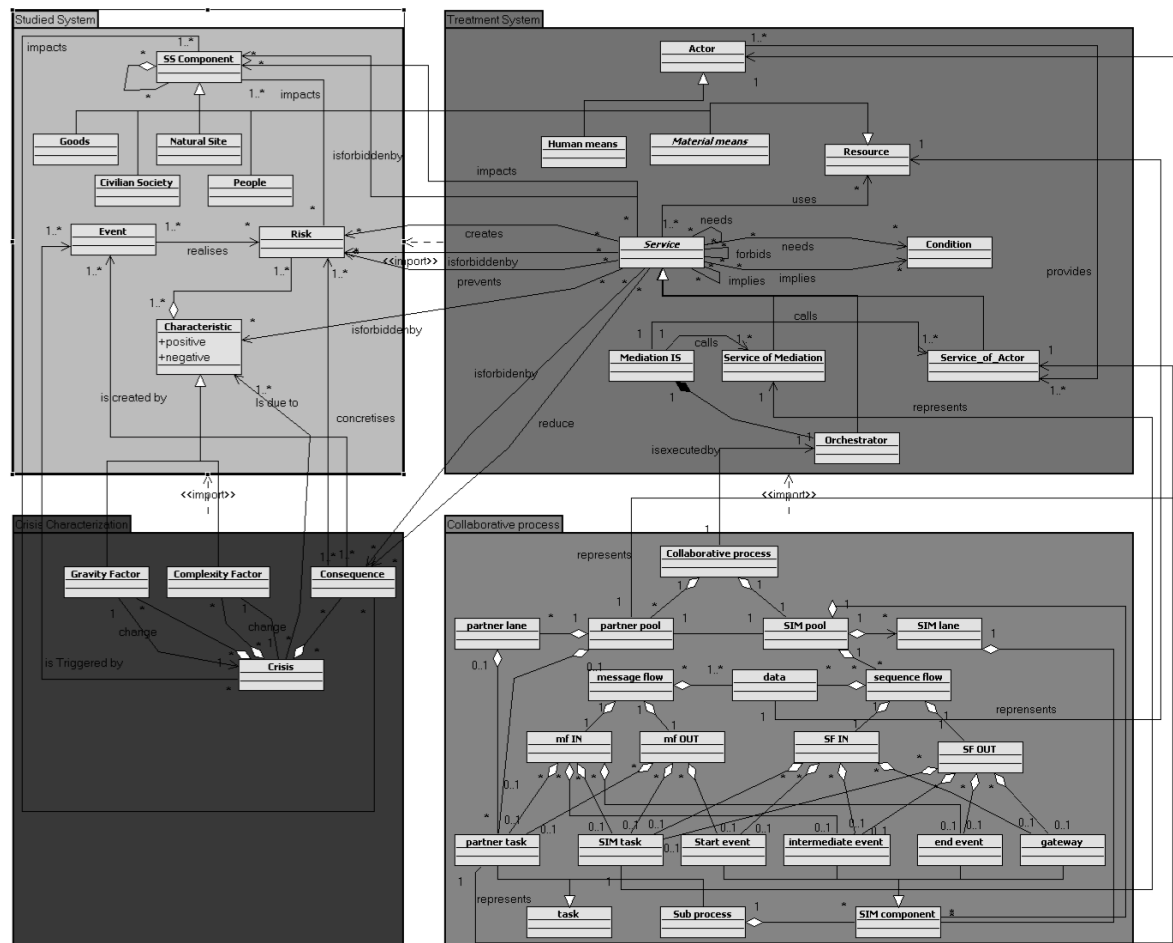


Figure II- 7 : métamodèle de crise

## II. Démarche de déduction du processus collaboratif

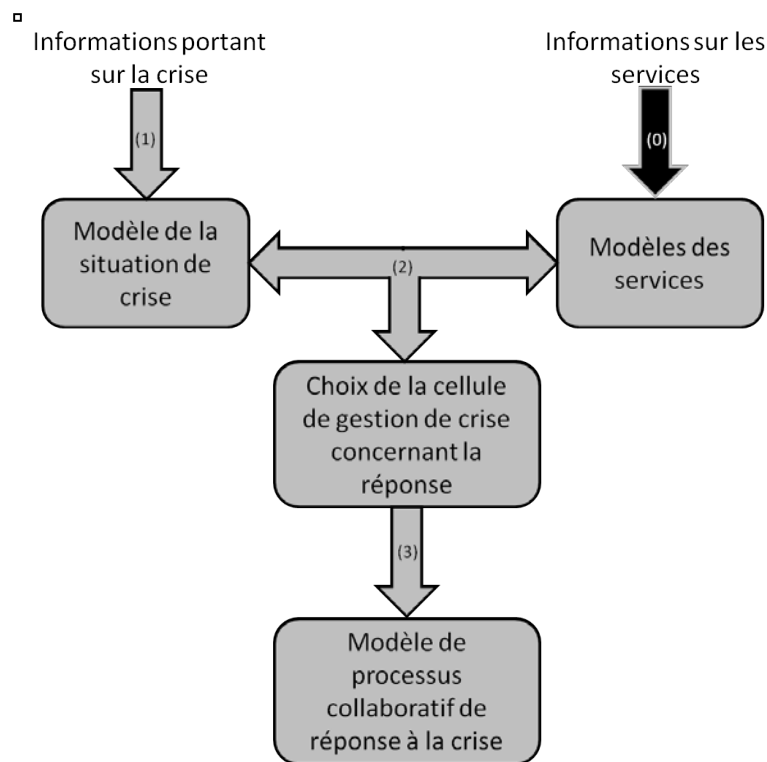
Cette partie s'intéresse à la succession de tâches ordonnées, appelée démarche de déduction, permettant la déduction du processus collaboratif à partir des informations recueillies sur la situation. Cette démarche de déduction, représentée par la figure II-8, doit dans un premier temps recueillir les modèles de description des services de chaque partenaire (dont la définition doit raisonnablement être effectuée avant l'occurrence de la crise), ainsi que le modèle de crise, afin de pouvoir confronter les informations capitalisées dans ces deux types de modèles et en déduire une proposition de processus collaboratif.

La première étape de cette démarche, représentée par l'étape (0) sur la figure II-8, devrait se dérouler lors de l'étape de préparation de la gestion de crise (cf. chapitre I), c'est-à-dire avant l'apparition d'une

crise. Cette étape consiste à récupérer, pour chaque partenaire potentiel d'une réponse à la crise, les informations sur ses services avant de les représenter sous forme de modèles respectant le métamodèle de crise. Il s'agit finalement d'instancier la partie système de traitement (*ST*) du métamodèle.

Cette démarche se poursuit suite à l'occurrence avérée d'une crise :

- **Etape (1)** : au fur et à mesure que des informations sur la situation remontent du terrain, elles sont recueillies et capitalisées dans un modèle de crise respectant la partie situation (*S*) du métamodèle (*i.e.* instanciant les parties système d'étude et caractéristiques de la crise). Une fois que ce modèle est jugé suffisamment riche, la démarche de déduction du processus collaboratif de réponse est lancée. Cette décision, si elle peut ici sembler tout à fait critique, ne l'est en fait que très peu au regard de la démarche complète proposée par ces travaux. En effet, dès lors que l'utilisateur estime qu'il a atteint un bon compromis entre la connaissance recueillie et le temps passé (*i.e.* que le fait de consacrer plus de temps au recueil d'information serait plus préjudiciable en termes de perte de temps que ce ne serait bénéfique en termes de gain de connaissance), ce qui ne constitue qu'une décision purement instinctive, il peut démarrer le mécanisme de déduction et donc de traitement de la crise. Pour autant, s'il s'avère que son jugement était erroné (ou qu'un revirement amène un surcroît massif de connaissance après sa décision de lancer la suite de la déduction), les mécanismes d'agilité (qui seront présentés dans le chapitre V), indissociables du support du traitement d'une situation de crise, permettront de pallier ces déconvenues. En synthèse, cette décision prise trop tôt n'aura théoriquement comme seule conséquence qu'une plus grande sollicitation des mécanismes d'agilité, alors que cette même décision prise plus à propos les sollicitera beaucoup moins. Si nous ne pouvons présager des conséquences sur le traitement de la crise et donc déterminer la meilleure approche (démarrer au plus tôt, quitte à s'adapter ou attendre une stabilisation de la connaissance), nous pouvons néanmoins affirmer que dans les deux cas, les mécanismes proposés permettront à la dynamique de traitement de coller à la situation, à son évolution et surtout à l'évolution de sa perception par la cellule de crise.
- **Etape (2)** : cette étape consiste à confronter les informations issues du modèle de situation (*S*) à celles des modèles de description des services (*ST*), afin de déduire l'ensemble des services pouvant être utilisés dans le cadre de la réponse à la crise.
- **Etape (3)** : cette étape a deux objectifs principaux : d'une part, (i) récupérer l'ensemble des choix faits par le personnel de la cellule de crise vis-à-vis des propositions du système concernant l'utilisation des services pour la réponse à la crise et d'autre part, (ii) interpréter ces choix afin de les traduire sous la forme d'un processus collaboratif représenté au format BPMN 1.0.



**Figure II- 8 : processus de transformation dans le but d'obtenir le processus collaboratif de réponse à la crise**

Comme expliqué dans l'introduction, les étapes (0) et (1) consistent à créer graphiquement des modèles en utilisant des éditeurs créés à l'aide de l'outil GMF. Dans le reste de ce chapitre, nous préférons mettre l'accent sur l'explication des deux autres étapes. Ces étapes doivent d'une part faire face à l'hétérogénéité des informations afin d'aider la cellule de gestion de crise à faire les choix nécessaires à la définition de la réponse à la crise et d'autre part, déduire de ces choix le processus collaboratif de réponse à la crise. Cette orientation est renforcée par le fait que lors du déroulement de l'exemple traité à la fin de ce chapitre, nous illustrerons chaque étape de la démarche et par conséquent les résultats obtenus par les étapes (0) et (1).

## II.1 Etape (2) : confrontation des différents modèles

Cette étape, consistant à confronter les informations issues des différents modèles, revient à réaliser une transformation de modèle endogène en utilisant des règles dites « sur place ». En effet, cette étape ayant pour but de déterminer l'ensemble des services utilisables pour implémenter la dynamique de réponse à la crise, il est nécessaire de confronter les informations des différents modèles et notamment les liens entre, d'une part, les services disponibles et, d'autre part, ce que l'on peut qualifier de « problèmes » de la crise (*i.e.* l'ensemble des risques et des conséquences recensés).

Cette étape est cruciale, dans la mesure où elle prend en charge la mise en correspondance d'éléments de connaissance issus du domaine de la problématique (les caractéristiques de la crise) avec des éléments de connaissance issus du domaine de la solution (les services mobilisables). Il est clair que le recueil et la description des services mobilisables devraient se faire par rapport aux caractéristiques de la crise (tel service étant adapté pour prévenir tel risque ou réduire telle conséquence), néanmoins, cette ambition s'avère évidemment complètement irréaliste dans la mesure où les services doivent être définis avant la modélisation de la situation de crise. Ainsi, s'il est envisageable qu'après un certain nombre d'itérations, la base de connaissance dispose de suffisamment de risques et de conséquences pour caractériser judicieusement les services en amont, puis de définir la situation de crise sur la base des mêmes instances

de risques et de conséquences (ce qui assurerait nativement la mise en correspondance entre les deux domaines), cette hypothèse n'est pas crédible initialement et les services doivent être définis par rapport à des risques et à des conséquences de natures « abstraites » uniquement identifiés pour permettre la description desdits services. La modélisation de crise, qui aura alors lieu ultérieurement, définira ses propres risques et conséquences (au moins tant que la base de connaissance ne sera pas assez copieuse). La problématique est donc clairement de réaliser le pontage sémantique entre les risques / conséquences utilisés pour définir les services et les risques / conséquences utilisés pour caractériser la crise (une problématique secondaire étant également de définir les proximités sémantiques entre les services eux-mêmes afin d'assurer une certaine souplesse à l'ensemble). Malheureusement, dans le cadre de ces travaux de thèse, nous n'avons pas mis au point de raisonneur sémantique, ceci faisant l'objet de travaux connexes (Boissel-Dallier *et al.*, 2009). Par conséquent, nous allons demander à l'utilisateur de définir certains liens de correspondance sémantique pour toutes instances des concepts de risques et conséquences (auxquels nous ajoutons le concept de condition). Notre système prend ensuite en charge la propagation et la divulgation dans toute la base de connaissance par le biais de règles de propagation des liens sémantiques (*cf.* II.2.1).

Nous avons ainsi prédéfini deux types de lien de correspondance sémantique : *Near* et *SameAs*. Le lien *SameAs* est un lien d'équivalence entre les instances des modèles. Ces liens traduisent généralement, pour un contexte précis, des synonymes exacts ou des écritures différentes d'un même terme (par exemple : « personnes en péril » et « personnes en danger »). Le lien *Near* est un lien de proximité entre les instances des modèles. Ce type de lien ne traduit pas d'équivalence entre deux entités (risque ou conséquence par exemple), mais le fait qu'un système de traitement mis en place pour traiter une telle entité pourrait être envisageable pour traiter une autre entité associée à la première par un lien *Near*. Par exemple, le traitement proposé pour contenir la boue issue de fortes précipitations pourrait être envisagé également pour gérer les eaux provenant d'une crue.

## II.2 Règles de transformation

Lors de cette étape, notre objectif est de déterminer l'ensemble des services utilisables dans le cadre d'une réponse à la crise. Pour cela, deux types de règles sont utilisées :

- RPLS : les Règles permettant de Propager les Liens Sémantiques entre les instances.
- RDSU : les Règles permettant la Déduction des Services Utilisables lors de la réponse à la crise.

### II.2.1 Règles permettant de Propager les Liens Sémantiques

Ces règles cherchent à propager les liens sémantiques définis entre les instances. Cette propagation se réalise par transitivité. Le langage SWRL, basé sur un schéma antécédent  $\rightarrow$  conséquent, est adapté pour écrire ces règles. Après la présentation des règles en SWRL, nous les illustrerons par un exemple.

#### *Propagation du lien SameAs par transitivité*

Cette règle propage le lien *SameAs*, c'est-à-dire que si deux instances ont un lien *SameAs* entre elles, toute nouvelle instance ayant un lien *SameAs* avec une des deux premières instances aura un lien *SameAs* avec l'autre. En SWRL, cette règle s'écrit de la manière suivante :

$$\text{Risque} (?x) \wedge \text{Risque} (?y) \wedge \text{Risque} (?z) \wedge \text{SameAs} (?x, ?y) \wedge \text{SameAs} (?y, ?z) \rightarrow \text{SameAs} (?x, ?z)$$

Règles II- 1 : propagation du lien SameAs par transitivité

La figure ci-dessous illustre cette règle, Les instances y sont représentées par des ellipses, les concepts par des rectangles et l'instanciation par un trait plein noir. Les liens *SameAs* entre les instances sont représentés par des flèches pleines. Les flèches de couleur noire représentent les liens définis nativement et la flèche grise représente le lien déduit grâce à la règle II-1.

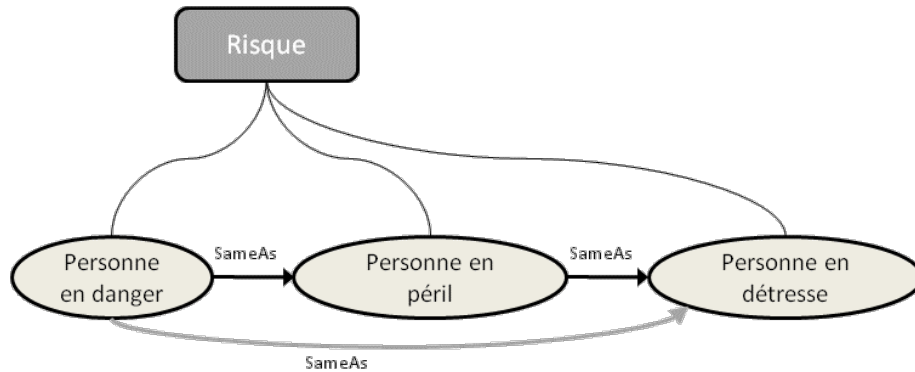


Figure II- 9 : exemple de propagation de lien SameAs par transitivité

#### Propagation du lien Near par transitivité :

La propagation des liens de type *Near* s'effectue à travers deux règles : la première, Règle II-2, propage le lien *Near* aux équivalents des instances, alors que la deuxième, Règle II-3, propage le lien *Near* par transitivité (nos travaux ne prennent pas en compte d'atténuation du lien *Near* par transitivité).

$$Risque (?x) \wedge Risque (?y) \wedge Risque (?z) \wedge SameAs (?x, ?y) \wedge Near (?y, ?z) \rightarrow Near (?x, ?z)$$

#### Règles II- 2 : propagation du lien Near aux instances équivalentes

$$Risque (?x) \wedge Risque (?y) \wedge Risque (?z) \wedge Near (?x, ?y) \wedge Near (?y, ?z) \rightarrow Near (?x, ?z)$$

#### Règles II- 3 : propagation du lien Near par transitivité

La figure II-10, basée sur les mêmes règles graphiques que l'exemple précédent, illustre le résultat de l'exécution de ces règles. Le lien *Near* entre *incendie de forêt* et *incendie de faible ampleur*, ainsi que celui entre *incendie de forêt* et *incendie de moteur*, sont le résultat de la règle II-2, alors que le lien entre *incendie de forêt* et *incendie de barbecue* est le résultat de la règle II-3.



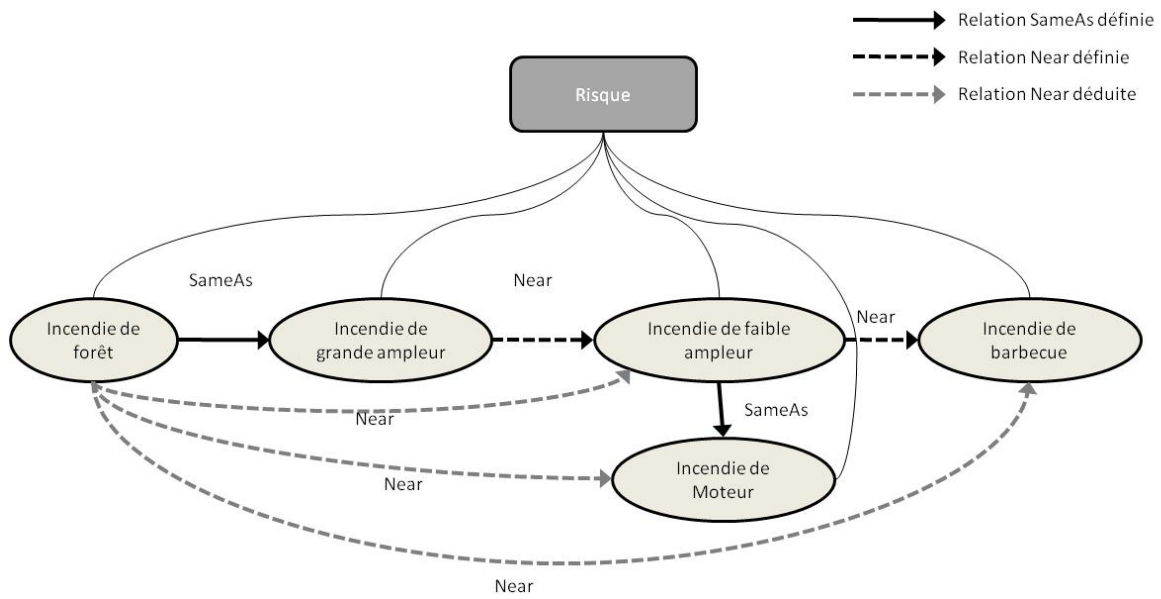


Figure II- 10 : exemple de propagation de lien Near par transitivité

## II.2.2 Règles permettant la Dédution des Services Utilisables

L'objectif de l'ensemble des règles suivantes est de déterminer l'ensemble des services pouvant être utilisés dans le but de résoudre un problème (risque ou conséquence de la crise). Afin de déterminer ces services, nous allons nous baser sur la définition des services ainsi que sur les liens de correspondance sémantique entre les problèmes. Différents cas de figure permettent de déterminer si un service est utilisable ou non lors d'une réponse à la crise. Le langage SWRL ne permettant pas l'utilisation d'opérateurs logiques de type négatif ou de type « ou », il a donc été nécessaire d'écrire une règle de déduction pour chaque cas de figure.

Afin de pouvoir mener à bien ce filtre de services, nous avons ajouté des relations :

- **agrégé** : correspond aux relations d'agrégation entre les différents concepts du métamodèle et représentées sur la figure II-7.
- **estComposéDe** : la crise a une relation *estComposéDe* avec chaque élément du modèle de crise.
- **estInterditPar** : cette relation est utilisée afin de sauvegarder l'interdiction de l'utilisation d'un service due à la présence d'un composant système d'étude, d'un risque ou d'une conséquence.
- **peutRésoudre** : cette relation entre un service et un problème (risque ou conséquence), permet de sauvegarder les résultats des règles de déduction. Cette relation veut dire qu'un service prévient exactement ce risque, ou réduit exactement cette conséquence.
- **pourraitRésoudre** : à l'instar de la relation précédente, cette relation est utilisée afin de sauvegarder le résultat d'un raisonnement ayant conduit à conclure qu'un service ne corrige pas exactement le problème, mais corrige un problème proche, il est donc envisageable de l'utiliser.

### Dédution des interdictions et compléments du modèle de crise

La règle suivante s'intéresse à sélectionner l'ensemble des risques qui peuvent être présents, bien que n'ayant pas été identifiés lors de la création du modèle de crise. Si une caractéristique, un facteur de gravité ou un facteur de complexité, est présent dans le modèle de crise, l'ensemble des risques qui lui sont nativement reliés doit être adjoint au modèle. Le langage SWRL permettant d'utiliser la notion d'héritage,

dans la règle ci-dessous, *Caractéristique(?z)* regroupe à la fois l'ensemble des instances de type caractéristique mais aussi celles de type facteur de complexité ou facteur de gravité.

$$Crise (?x) \wedge estComposéDe (?x, ?z) \wedge Caractéristique (?z) \wedge agrégé (?a, ?z) \wedge Risque (?a) \rightarrow estComposéDe (?x, ?a)$$

#### Règles II- 4 : déduction des risques présents en fonction des caractéristiques

La règle suivante s'intéresse à identifier l'ensemble des services interdits par un élément de la crise ou un de ces équivalents sémantiques :

$$Crise (?x) \wedge estComposéDe (?x, ?y) \wedge Service (?z) \wedge estInterditPar (?z, ?b) \wedge SameAs (?y, ?b) \rightarrow estInterditPar (?z, ?y)$$

#### Règles II- 5 : déduction des services interdits par un élément de la crise

##### Règles permettant de déterminer les services pouvant exactement corriger un problème

La règle suivante s'intéresse à sélectionner l'ensemble des services prévenant une conséquence de la crise ou toute autre conséquence qui lui serait sémantiquement équivalente :

$$Crise (?x) \wedge Conséquence (?y) \wedge Conséquence (?b) \wedge estComposéDe (?x, ?y) \wedge Service (?z) \wedge réduit (?z, ?b) \wedge SameAs (?y, ?b) \rightarrow peutRésoudre (?z, ?y)$$

#### Règles II- 6 : déduction des services permettant de réduire les conséquences de la crise

Par exemple, prenons le cas d'une collision entre un camion citerne et un train. Le moteur du camion citerne est en feu. Nous définissons ce feu comme étant équivalent à un incendie de faible ampleur (relation *SameAs* entre les deux conséquences). Or, un incendie de faible ampleur pouvant être réduit par l'action d'éteindre un incendie de faible ampleur, il peut être envisagé de faire appel à cette action pour éteindre l'incendie moteur.

La règle suivante sélectionne tous les services qui pourraient potentiellement réduire une conséquence de la crise, c'est-à-dire qu'ils ne réduisent pas exactement ce risque, mais un risque proche (relation *Near* entre les deux conséquences) :

$$Crise (?x) \wedge Conséquence (?y) \wedge Conséquence (?b) \wedge estComposéDe (?x, ?y) \wedge Service (?z) \wedge réduit (?z, ?b) \wedge Near (?y, ?b) \rightarrow pourraitRésoudre (?z, ?y)$$

#### Règles II- 7 : déduction des services qui pourraient réduire des conséquences de la crise

Reprenons notre exemple de collision entre le camion citerne et le train. La nature du contenu de la citerne du camion est inconnue. Par ailleurs, les conséquences incendie moteur et incendie de grande ampleur sont définies comme proches (lien *Near*). Deux actions peuvent être menées pour réduire la conséquence d'incendie de grande ampleur, la première consiste à l'éteindre en engageant des moyens plus importants que dans le cas précédent. L'autre consiste à arroser les alentours de l'incendie pour éviter son expansion. Ainsi, ces deux actions sont proposées comme pouvant résoudre la conséquence d'incendie moteur.

Les règles concernant la prévention des risques sont établies de manière totalement symétrique avec celles proposées précédemment pour le traitement des conséquences.

La règle suivante s'intéresse à sélectionner l'ensemble des services prévenant un risque de la crise ou tout autre risque qui lui serait sémantiquement équivalente :

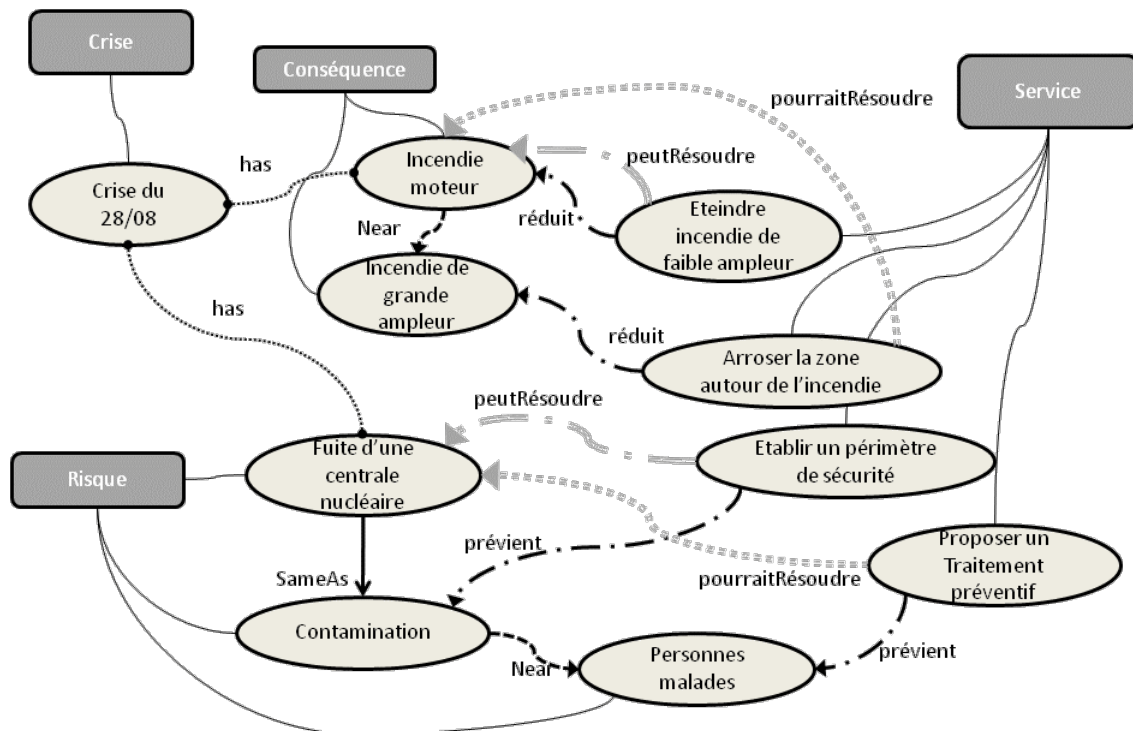
$Crise (?x) \wedge Risque (?y) \wedge Risque (?b) \wedge estComposéDe (?x, ?y) \wedge Service (?z) \wedge prévient (?z, ?b) \wedge SameAs (?y, ?b) \rightarrow peutRésoudre (?z, ?y)$

**Règles II- 8 : déduction des services permettant de prévenir les risques de la crise**

La règle suivante sélectionne tous les services qui pourraient potentiellement prévenir un risque de la crise, c'est-à-dire qu'ils ne préviennent pas exactement ce risque, mais un risque proche (relation *Near* entre les deux conséquences) :

$Crise (?x) \wedge Risque (?y) \wedge Risque (?b) \wedge estComposéDe (?x, ?y) \wedge Service (?z) \wedge prévient (?z, ?b) \wedge Near(?y, ?b) \rightarrow pourraitRésoudre (?z, ?y)$

**Règles II- 9 : déduction des services qui pourraient prévenir des risques de la crise**



**Figure II- 11 : illustration des règles de déduction de sélection des services utilisables**

L'illustration de la figure II-11 peut sembler décalée lorsque l'on regarde le résultat des règles de déduction et notamment le fait que donner un traitement préventif à une population pourrait prévenir le risque d'une fuite d'une centrale nucléaire. Cependant, de tels traitements peuvent réellement être utilisés. En effet, les organismes de secours distribuent des comprimés d'iode aux populations proches d'une centrale nucléaire afin de prévenir les dégâts potentiels d'une fuite.

### II.3 Etape (3) : construction du processus collaboratif

Cette étape est composée de deux sous-étapes. La première est destinée à proposer des alternatives aux membres de la cellule de crise afin de récupérer leurs réponses et leurs choix vis-à-vis des options présentées. Ces interactions portent, d'une part, sur le choix des services à utiliser afin de résoudre l'ensemble des problèmes de la crise et, d'autre part, sur la définition d'un ordre de priorité de traitement. Une fois ces choix validés, la deuxième partie de cette étape construit le processus collaboratif.

### II.3.1 Récupération des choix de la cellule de crise

Cette partie propose des alternatives aux membres de la cellule de crise afin de récupérer leurs réponses et leurs choix vis-à-vis des options présentées quant à la définition de la réponse à adopter. Ces choix sont de deux natures :

- le choix des services à utiliser pour prendre en charge le traitement (réduction ou prévention) de chaque problème (conséquence ou risque) identifiés pour la crise courante,
- les choix de priorité de prise en compte des problèmes identifiés pour la crise en cours afin de pouvoir disposer d'information concernant l'ordre de traitement des problèmes.

La définition d'un ordre de priorité de prise en compte des problèmes en situation de crise est une préconisation du retour d'expérience d'anciennes crises, telle que la crise qui fit suite à l'ouragan Katerina (Guilhou, 2009). Cet ordre de priorité servira de base à la définition de l'ordre de succession des services, concrétisé sous la forme d'un modèle de processus collaboratif.

La réalisation de ces choix s'effectue *via* une interface utilisateur. Cette interface est composée de deux types de fenêtres : (i) la première, représentée par la partie gauche de la figure II-12, liste l'ensemble des risques et conséquences de la crise courante et permet à l'utilisateur de préciser l'ordre de priorité de ces problèmes, vis-à-vis de son propre système de valeur. Afin de déterminer cet ordre de priorité, l'utilisateur doit ordonner les problèmes, grâce aux boutons « + » et « - », le premier de la liste étant, d'après lui, le problème le plus urgent à traiter. (ii) Une fois l'ordre de priorité établi, l'utilisateur a la possibilité d'associer un (ou des) service(s) à mettre en œuvre afin de résoudre chacun des problèmes (risques et conséquences) recensés pour la crise en cours. Pour cela, en cliquant sur le nom du problème dans la première interface, l'utilisateur ouvre le deuxième type d'interface, représenté par la partie droite de la figure II-12. Cette interface regroupe l'ensemble des services potentiellement utilisables afin de résoudre le problème sélectionné (ces services sont proposés automatiquement, suite aux inférences sur l'ontologie, selon les règles décrites précédemment). Ces services candidats sont repartis en deux tableaux. Le premier tableau liste l'ensemble des services pouvant résoudre exactement ce problème, alors que le second tableau liste l'ensemble des services qui résolvent un problème proche.

Nous avons décidé de lister l'ensemble des services sélectionnables en deux tableaux, car la pertinence d'utilisation d'un service résolvant exactement ce problème n'est pas la même qu'un service qui pourrait résoudre ce problème. Par conséquent, nous avons ajouté un indice de pertinence (appelé *confiance* sur l'interface) des services vis-à-vis du problème. Pour chaque service résolvant exactement le problème, cet indice est égal à 1. Pour chaque service qui pourrait résoudre ce problème, l'indice de pertinence est égal à  $1/x$ ,  $x$  étant le nombre minimum de relation *Near* entre l'élément du modèle de crise et l'élément résolu par le service. Ainsi, si l'atténuation de la relation *Near* entre concepts n'est pas prise en compte par notre système, en revanche, les conséquences d'une telle atténuation sur l'adéquation des services candidats se trouvent prises en charge, *a minima*, par ce mécanisme d'amortissement de la pertinence.

Dans le même ordre d'idée, pour chaque service nous ajoutons un indice d'accessibilité (appelé *accessibilité valeur* sur l'interface) du service. Cet indice décroît en fonction du nombre de pré-requis nécessaire à l'utilisation de ce service (*i.e.* suivant les conditions qui doivent être vérifiées avant son invocation et le nombre de services dont l'exécution doit lui être associée).

Enfin, pour chaque service un indice de dommages collatéraux est utilisé (appelé *collatéraux valeur* sur l'interface). Cet indice illustre les risques liés à l'utilisation de ce service (ce lien étant défini au niveau du modèle de définition des services). En effet, l'exécution d'un service peut créer un ensemble de risques, comme par exemple l'envoi de pompiers à l'intérieur d'un immeuble en flammes engendre des risques de blessures pour les pompiers. Cet indice prend la valeur à  $1/y$ ,  $y$  étant le nombre de risques créés par ce

service. Le calcul de cet indice est clairement insuffisant, il devrait prendre en compte la gravité du risque susceptible d'être créé. Cependant, il permet d'alerter la cellule de gestion de crise sur le choix de l'utilisation de ce service.

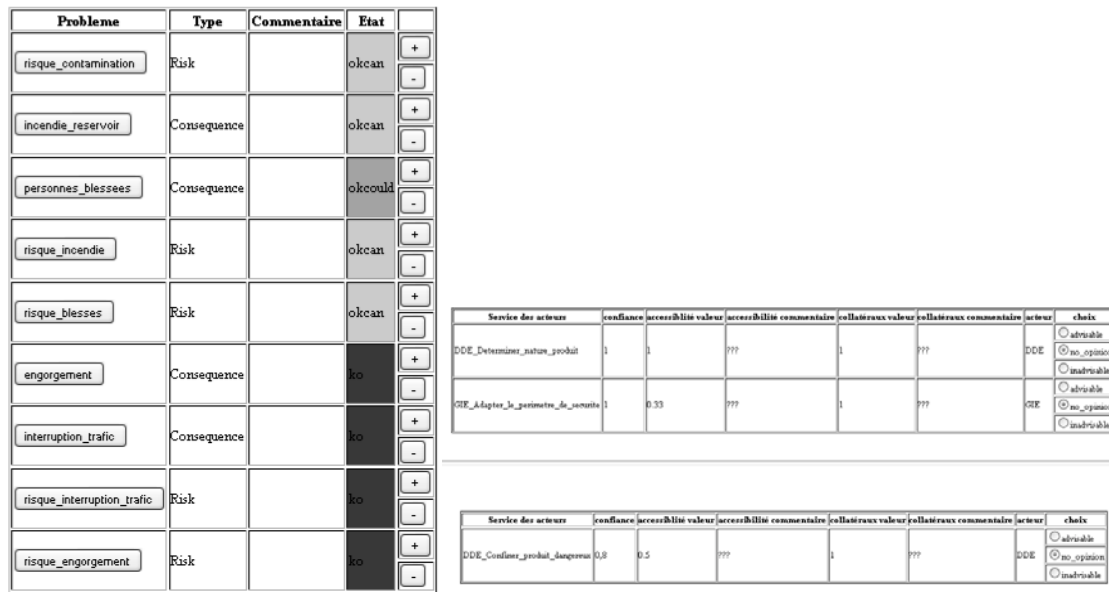


Figure II- 12 : interface de choix pour la cellule de crise

Lorsque la cellule de gestion de crise a choisi un ou des services à utiliser pour résoudre un problème, le statut de ce problème évolue sur la première interface. Si au moins un service sélectionné traite exactement ce problème, le statut du problème passe à « OKcan ». Dans le cas où les services sélectionnés ne traitent pas exactement ce problème, le statut du problème passe à « OKcould ». De plus, un service peut traiter plusieurs problèmes. A partir du moment où il est sélectionné, tous les problèmes qu'il peut traiter voient leur statut passer à « OKcan » ou « OKcould » en fonction de la pertinence.

Lorsque les choix de la cellule de crise (portant sur l'ordre de traitement des problèmes ainsi que sur le choix des services à utiliser pour la totalité ou une partie des problèmes de la crise) sont terminés, le système doit enregistrer ces informations complémentaires afin de les transmettre aux activités destinées à la construction du processus collaboratif. Ainsi, afin de sauvegarder les choix effectués concernant la sélection des services, l'attribut *état* d'un service *a* sélectionné prend la valeur « OK » et la relation *estUtiliséPour(a,b)* enregistre le fait que ce service *a* a été sélectionné pour réduire le problème *b*. Enfin, l'attribut *ordreTraitement* d'un risque ou d'une conséquence enregistre les choix concernant l'ordre de traitement de ce problème.

### II.3.2 Construction du processus

A partir des choix de la cellule de crise portant sur la définition de l'ordre de priorité de traitement des problèmes et sur la sélection des services à utiliser, il est possible de construire un processus de réponse à la crise. Cette création du processus se décompose en deux actions successives : (i) création de l'ensemble des éléments du processus collaboratif, (ii) définition de l'ordre de succession des services.

#### Création de l'ensemble des éléments du processus collaboratif

Cette partie présente les règles destinées à la construction des éléments du processus collaboratif. A l'origine, le langage SWRL (utilisé pour l'écriture des règles) ne permettait pas de créer de nouvelles instances, c'est pourquoi, pour la rédaction de ces règles particulières, nous avons fait appel à l'extension BuiltIn du langage SWRL. Cette extension est représentée dans l'écriture des règles par le préfixe « *swrlb*: ».

La règle suivante crée à la fois un *pool* et une *lane* pour le SIM, chacun étant obligatoire du point de vue du métamodèle de processus collaboratif. Nous traduisons cette obligation par le fait que pour chaque élément de type crise, un *pool* et une *lane* sont créés. De même, nous créons en même temps l'événement de fin et de début du processus qui seront nommés « start » et « end ». Il est important de noter que les noms des éléments créés nous sont imposés par l'exigence de compatibilité entre nos travaux et la transformation du processus collaboratif en architecture logique du SIM, proposée par (Touzi, 2007).

$$\text{Crise} (?x) \wedge \text{swrlx:makeOWLThing} (?y, ?x) \wedge \text{swrlx:makeOWLThing} (?a, ?x) \wedge \text{swrlx:makeOWLThing} (?b, ?x) \wedge \text{swrlx:makeOWLThing} (?c, ?x) \rightarrow \text{SIM\_Pool} (?y) \wedge \text{SIM\_Lane} (?a) \wedge \text{Start\_Event} (?b) \wedge \text{End\_Event} (?c) \wedge \text{name} (?b, \text{"start"}) \wedge \text{name} (?c, \text{"end"})$$

#### Règles II- 10 : création des éléments obligatoires du processus

Pour chaque acteur ayant au moins un service sélectionné (attribut *état* à « OK »), un *pool* et une *lane* sont créés.

$$\text{Acteur} (?x) \wedge \text{Service} (?a) \wedge \text{Fournit} (?x, ?a) \wedge \text{Etat} (?a, \text{« OK »}) \wedge \text{swrlx:makeOWLThing} (?b, ?x) \wedge \text{swrlx:makeOWLThing} (?c, ?x) \rightarrow \text{Partner\_Pool} (?b) \wedge \text{Partner\_lane} (?c)$$

#### Règles II- 11 : création des partner pools et partner lanes

Pour chaque service sélectionné, plusieurs éléments du processus collaboratif sont créés :

- un *Partner\_task*, image du service au niveau du processus.
- Un *SIM\_task*, service de coordination entre le médiateur et le service de l'acteur
- Deux *Messages\_Flows*, l'un allant du service de coordination au service de l'acteur, transférant les informations nécessaires à l'exécution du service, l'autre faisant le chemin inverse en récupérant le résultat du service.

$$\text{Service} (?x) \wedge \text{Etat} (?x, \text{« OK »}) \wedge \text{swrlx:makeOWLThing} (?a, ?x) \wedge \text{swrlx:makeOWLThing} (?c, ?x) \wedge \text{swrlx:makeOWLThing} (?b, ?x) \wedge \text{swrlx:makeOWLThing} (?d, ?x) \rightarrow \text{Partner\_task} (?a) \wedge \text{SIM\_task} (?b) \wedge \text{Message\_flow} (?c) \wedge \text{Message\_flow} (?d) \wedge \text{Mf\_IN} (?c, ?a) \wedge \text{Mf\_OUT} (?c, ?b) \wedge \text{Mf\_IN} (?d, ?b) \wedge \text{Mf\_OUT} (?d, ?a)$$

#### Règles II- 12 : création des Partner\_tasks et SIM\_tasks

Lors de la description du métamodèle, nous avons précisé qu'un service peut nécessiter un pré-requis. Les règles suivantes servent à déterminer l'ensemble des services nécessaires à la réalisation de ce pré-requis. Cette relation entre les services peut être définie directement (règle II-13), ou *via* des conditions (règle II-14). Une fois les services pré-requis listés, nous créons les éléments du processus les concernant, sur le même principe que celui correspondant à la règle II-12. Nous ne présenterons ici que les règles liées à la nécessité en « amont » (on cherche à trouver les services qui doivent être exécutés au préalable). Afin de retrouver les règles liées à la nécessité en « aval » (*i.e.* les services devant être exécutés *a posteriori*), il suffit de remplacer *nécessite* par *implique*.

Cependant, comment propager ces règles aux services remplissant la nécessité « amont » du service utilisé pour remplir une « amont » ? Afin de propager ces règles de façon récursive, le service remplissant la nécessité « amont » se voit attribuer la valeur « OK » pour *état*. Ainsi, les règles II-13 et II-14 peuvent lui être appliquées.

$$\text{Service} (?x) \wedge \text{Service} (?y) \wedge \text{Etat} (?x, \text{« OK »}) \wedge \text{nécessite} (?x, ?y) \wedge \text{swrlx:makeOWLThing} (?a, ?y) \wedge \text{swrlx:makeOWLThing} (?c, ?y) \wedge \text{swrlx:makeOWLThing} (?b, ?y) \wedge \text{swrlx:makeOWLThing} (?d, ?y) \rightarrow \text{Etat} (?y, \text{« OK »}) \wedge \text{Partner\_task} (?a) \wedge \text{SIM\_task} (?b) \wedge \text{Message\_flow} (?c) \wedge \text{Message\_flow} (?d) \wedge \text{Mf\_IN} (?c, ?a) \wedge \text{Mf\_OUT} (?c, ?b) \wedge \text{Mf\_IN} (?d, ?b) \wedge \text{Mf\_OUT} (?d, ?a)$$

**Règles II- 13 : création des éléments du processus liés à un pré-requis d'un service**

$$\text{Service} (?x) \wedge \text{Service} (?y) \wedge \text{Condition} (?z) \wedge \text{Condition} (?e) \wedge \text{Etat} (?x, \text{« OK »}) \wedge \text{nécessite} (?x, ?z) \wedge \text{SameAs} (?z, ?e) \wedge \text{implique} (?y, ?e) \wedge \text{swrlx:makeOWLThing} (?a, ?y) \wedge \text{swrlx:makeOWLThing} (?c, ?y) \wedge \text{swrlx:makeOWLThing} (?b, ?y) \wedge \text{swrlx:makeOWLThing} (?d, ?y) \rightarrow \text{Etat} (?y, \text{« OK »}) \wedge \text{Partner\_task} (?a) \wedge \text{SIM\_task} (?b) \wedge \text{Message\_flow} (?c) \wedge \text{Message\_flow} (?d) \wedge \text{Mf\_IN} (?c, ?a) \wedge \text{Mf\_OUT} (?c, ?b) \wedge \text{Mf\_IN} (?d, ?b) \wedge \text{Mf\_OUT} (?d, ?a)$$

**Règles II- 14 : création des éléments du processus liés à un pré-requis d'un service *via* les conditions**

### Définition de l'ordre d'exécution du service

Cette deuxième étape de la création du processus collaboratif s'intéresse à la déduction de l'enchaînement des services. Cette déduction se base sur l'ordre de traitement des problèmes défini par la cellule de crise. Si la déduction se limitait à appliquer ce seul ordre de priorité tel quel, le processus collaboratif serait parfaitement linéaire. Par conséquent, avant de commencer le traitement d'un problème, il serait nécessaire d'attendre que l'ensemble des problèmes ayant un ordre de priorité plus élevé soit résolu. Il est évident que dans une situation où l'urgence et la vitesse de réaction peuvent s'avérer primordiales, tout ce qui peut être parallélisé doit l'être. Néanmoins, dans un contexte où des vies humaines peuvent être en jeu, il est tout aussi prépondérant d'établir des critères de parallélisation qui soient rigoureux et fiables afin de s'assurer que la mise en parallèle ne puisse pas altérer la qualité de l'intervention. Lors de la déduction, nous allons par conséquent essayer de paralléliser au maximum les services. Pour cela, nous proposons trois règles :

- Règle II-15 : pour deux services sélectionnés, si : (i) ils ne sont pas réalisés par le même partenaire, (ii) ils réduisent des problèmes différents et que (iii) les problèmes pris en charge par ces services n'impactent pas le même composant système d'étude (bien, population, site naturel, société civile), alors ces deux services peuvent être réalisés en parallèle.
- Règle II-16 : si deux services sont sélectionnés pour résoudre le même problème (sans contrainte native de précédence ou de dépendance), alors ils peuvent être réalisés en parallèle. Ce choix de paralléliser ces services se base sur la possibilité que la cellule de crise ait décidé que la complémentarité des services soit nécessaire au traitement du problème.
- Règle II-17 : si un service nécessite ou implique la réalisation de plusieurs services, alors tous ces services sont réalisés en parallèle. Cette règle (présentée ci-dessous), ne traite que le cas de services directement liés par un lien de nécessité. D'autres règles, non présentées ici, sont nécessaires afin d'envisager l'ensemble des autres cas : le cas d'un lien de nécessité par une condition, celui d'un lien d'implication directe, ou encore celui d'un lien d'implication par une condition.

$$\text{Service} (?x) \wedge \text{Service} (?y) \wedge \text{Acteur} (?c) \wedge \text{Acteur} (?d) \wedge \text{produitPar} (?x, ?c) \wedge \text{produitPar} (?y, ?d) \wedge \text{differentFrom} (?c, ?d) \wedge \text{estUtiliséPour} (?x, ?a) \wedge \text{estUtiliséPour} (?y, ?b) \wedge \text{differentFrom} (?a, ?b) \wedge \text{impacte} (?b, ?e) \wedge \text{impacte} (?a, ?f) \wedge \text{differentFrom} (?e, ?f) \rightarrow \text{sameTime} (?x, ?y)$$

**Règles II- 15 : parallélisme des services, produit par des partenaires différents, ne traitant pas le même problème n'impactant pas le même composant système d'étude**

$$\text{Service} (?x) \wedge \text{Service} (?y) \wedge \text{estUtiliséPour} (?x, ?a) \wedge \text{estUtiliséPour} (?y, ?a) \rightarrow \text{sameTime} (?x, ?y)$$

**Règles II- 16 : parallélisme des services sélectionnés pour traiter le même problème**

$$\text{Service} (?x) \wedge \text{Service} (?a) \wedge \text{Service} (?b) \wedge \text{nécessite} (?x, ?a) \wedge \text{nécessite} (?x, ?b) \rightarrow \text{sameTime} (?a, ?b)$$

**Règles II- 17 : parallélisme des services nécessités par un autre**

Dans le cadre de nos travaux, nous n'avons pas réussi à finaliser la création du processus collaboratif uniquement avec des règles écrites en langage SWRL. Ceci est dû à l'impossibilité d'écrire des règles comportant de la négation avec l'interface proposée par Protégé (Protégé, 2000). Par conséquent, lorsque nous avons essayé de définir l'ordre d'exécution du processus, nous nous sommes heurtés à certaines difficultés. Par exemple, il pouvait être déduit qu'un service ait une relation d'antécédence avec l'ensemble des services situés en amont de sa position dans le processus, qu'ils appartiennent ou non à la même séquence. Par conséquent, l'exécution de ces règles se limite à créer l'ensemble des éléments du processus collaboratif et un algorithme, exécuté par un programme JAVA, construit l'ordre d'exécution du processus collaboratif et complète ainsi sa définition.

Une fois l'ensemble des éléments du processus construit, cet algorithme *Construction des séquences du processus collaboratif* (Algo II-2), est exécuté afin de finaliser la création du processus. Afin d'alléger la lecture de l'algorithme, nous introduisons les fonctions suivantes :

- *CréationSeqFlow(x,y)* : cette fonction crée un *SequenceFlow* entre l'élément  $x$  et  $y$ . Ce *SequenceFlow* définira une relation d'antécédence entre  $x$  et  $y$ , dont  $x$  sera la source et  $y$  la destination. Les *SequenceFlow* ne pouvant être utilisés qu'au sein de la *SIMPool*, ils ne peuvent être reliés qu'à des éléments de type *SIMTask*, *Gateway* ou *Event*. Par conséquent, si  $x$  ou  $y$  est un service, le *SequenceFlow* sera relié au *SIMTask* de ce service.
- *g1=CréationGateway()* : cette fonction permet la création d'un gateway ( $g1$  dans cet exemple d'appel de la fonction).
- *y=TrouverServiceParOrdre(x)* :  $x$  étant un service, cette fonction retourne un entier  $y$  qui correspond au rang d'exécution du service  $x$  dans le processus en fonction, d'une part, des choix de priorité faits par la cellule de crise (vis-à-vis des problèmes pour lequel il a été sélectionné) et d'autre part, des liens entre le service  $x$  et les autres services sélectionnés (parallélisation, précedence, etc.). Si  $x$  n'a pas été choisi pour résoudre un problème, mais qu'il est utilisé en tant que pré-requis, alors la fonction retourne -1.
- *SameTime(x)* :  $x$  est un service. Cette fonction applique un filtre au résultat des règles II-15, II-16, II-17 et renvoie l'ensemble des services respectant à la fois ces quatre conditions :
  - (i) ils sont exécutables en parallèle du service  $x$ .
  - (ii) Ils ont été sélectionnés pour un problème ayant un ordre de priorité supérieur ou égal à celui du problème pris en charge pour  $x$ .
  - (iii) Ils ne sont ni un pré-requis au service  $x$ , ni impliqués, directement ou indirectement, par le service  $x$ .



- (iv) S'ils peuvent par ailleurs être mis en parallèle d'autres services, il est nécessaire que les autres services pouvant être mis en parallèle aient été sélectionnés pour un problème ayant un ordre de priorité moins important que celui du problème ayant provoqué la sélection de  $x$ .

Les conditions (ii) et (iv) permettent de paralléliser au plus tôt les services. En effet, cette fonction sert à recenser l'ensemble des services pouvant être mis en parallèle de  $x$ . Pour cela, nous allons uniquement nous intéresser aux services devant initialement être exécutés après  $x$ , rôle de la condition (ii). Cependant, il est possible qu'un service  $a$  réponde à la fois aux trois premières conditions pour le service  $x$ , mais aussi pour un autre service  $y$ . Le service  $a$  peut alors être mis en parallèle de ces deux services. L'idée étant de mettre ce service  $a$  le plus tôt possible au niveau du processus, il sera mis en parallèle du service étant exécuté en premier, c'est le rôle de la condition (iv).

- **$INI(x)$**  : est une variable.  $x$  étant un service ayant la possibilité d'avoir plusieurs niveaux de services pré-requis, pouvant être vue comme une chaîne de services pré-requis. Cette variable sauvegardera le service initial de cette chaîne de pré-requis.
- **$FIN(x)$**  : est une variable.  $x$  étant un service ayant la possibilité d'avoir plusieurs niveaux de services impliqués, pouvant être vue comme une chaîne de services impliqués. Cette variable sauvegardera le service final de cette chaîne de pré-requis.
- **$NbPrerequis(x)$**  :  $x$  étant un service ou un gateway, cette fonction renvoie le nombre de services pré-requis à  $x$  ou 0 si  $x$  est un gateway.
- **$Prerequis^n(x)$**  :  $x$  étant un service,  $n$  étant un entier tel que  $0 < n \leq NbPrerequis(x)$ , cette fonction renvoie le  $n$ -ième service pré-requis à l'exécution de  $x$ .
- **$NbImplique(x)$**  :  $x$  étant un service, cette fonction renvoie le nombre de services devant être réalisés en aval de ce service ou 0 si  $x$  est un gateway.
- **$Implique^n(x)$**  :  $x$  étant un service,  $n$  étant un entier tel que  $0 < n \leq NbImplique(x)$ , cette fonction renvoie le  $n$ -ième service devant être réalisé en aval de  $x$ .
- **$TrouverFinServicePreRequis(x)$**  :  $x$  étant un service, cette fonction est utilisée lorsque le service  $x$  a plusieurs services en pré-requis. Cette fonction peut retourner deux choses : (i) un service, si les services pré-requis du service  $x$  ont ce service pré-requis en commun, (ii) un gateway, si les services pré-requis n'ont pas de service pré-requis commun, auquel cas la fonction crée en même temps un gateway (fourche, ouvrant sur les services pré-requis) et les Sequenceflows entre les services pré-requis du service  $x$ . L'exemple de la figure II-13 illustre l'utilisation de cette fonction. Dans le cas A, le service S1 a deux pré-requis : S2 et S3. S3 a un pré-requis S4, alors que S2 n'en a pas. La fonction génère donc deux gateways et six sequence flows. De plus, elle renvoie comme valeur résultat le premier gateway (fourche). Dans le cas B, basé sur le cas A, nous avons ajouté le fait que S4 et S2 ont le même pré-requis S5. Dans ce cas, la fonction génère les gateways et sept sequence flows, ainsi que la valeur S5 pour résultat.

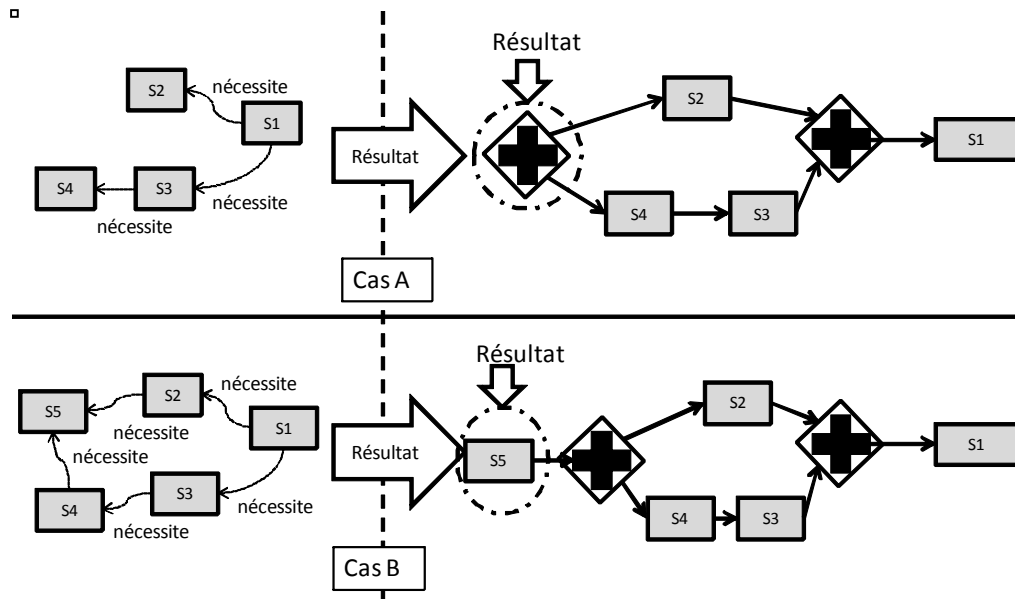


Figure II- 13 : exemple de résultat de la fonction *TrouverFinServicePreRequis(S1)*

- *TrouverFinServiceImplique(x)* :  $x$  étant un service, cette fonction est la même que la fonction *TrouverFinServicePreRequis(x)*, à la différence près qu'elle s'intéresse aux services requis en aval et non à ceux requis en amont.
- *Initialisation\_INI\_FIN()* : cette fonction sert à initialiser les variables de début et fin de séquence de chaque service sélectionné pour un problème. L'Algo. II-1 décrit cette fonction.

```

Nom : Initialisation_INI_FIN( )

Objectif : initialiser les variables INI et FIN pour chaque service
sélectionné

Début :

  Pour chaque  $x \in$  Service ayant  $\text{Ordre}(x) > 0$  Faire
    Service_debut  $\leftarrow x$ 
    Service_fin  $\leftarrow x$ 
    INI( $x$ )  $\leftarrow x$ 
    FIN( $x$ )  $\leftarrow x$ 
    Tant que  $\text{NbPreRequis}(\text{Service\_debut}) > 0$  Faire
      Si  $\text{NbPreRequis}(\text{Service\_debut}) = 1$ 
        CréationSeqFlow(Prerequis1(Service_debut),
Service_debut))
        INI( $x$ )  $\leftarrow$  Prerequis1(Service_debut)
        Service_debut  $\leftarrow$  Prerequis1(Service_debut)
      Sinon Si  $\text{NbPreRequis}(\text{Service\_debut}) > 1$ 
        Service_debut  $\leftarrow$  TrouverFinServicePreRequis
(Service_debut)
        INI( $x$ )  $\leftarrow$  Service_debut
      Fin Si
    Fin Tant que
  Tant que  $\text{NbImplique}(\text{Service\_fin}) > 0$  Faire

```

```

        Si NbImplique(Service_fin)=1
            CréationSeqFlow(Implique1(Service_fin),
Service_fin))
            FIN(x) ← Prerequis1(Service_fin)
            Service_fin ← Prerequis1(Service_fin)
        Sinon Si NbPreRequis(Service_fin) > 1
            Service_fin ← TrouverFinServiceImplique
(Service_fin)
            FIN(x) ← Service_fin
        Fin Si
    Fin Tant que
    Fin Pour
Fin

```

**Algo. II- 1 : fonction Initialisation des variables INI et FIN pour chaque service**

- *NbElementSansSeqFlowIn()* : est une fonction renvoyant le nombre d'éléments (*i.e.* SIM\_task ou Gateway), n'ayant aucun sequence flow en entrée.
- *ElementSansSeqFlowIn<sup>n</sup>()* : cette fonction renvoie le n-ième élément n'ayant pas de sequence flow en entrée.
- *TrouverOrdreMinServiceDeINI(x)* : cette fonction s'intéresse aux éléments  $x$  qui correspondent au début d'une chaîne de pré-requis, *i.e.*  $x$  est égal à la valeur d'une variable  $INI(y)$ ,  $y$  étant un service. Cependant, nous avons vu que des chaînes de pré-requis de différents services peuvent commencer par le même élément, notamment avec le résultat de la fonction *TrouverFinServicePreRequis(y)*. Cette fonction est utilisée afin de retourner l'ordre de traitement le plus faible (et donc le plus important) de l'ensemble des services ayant  $x$  comme élément initial.
- *NbElementSansSeqFlowOut()* : est une fonction qui renvoie le nombre d'éléments (*i.e.* SIM\_task ou Gateway), n'ayant aucun sequence flow en sortie.
- *ElementSansSeqFlowOut<sup>n</sup>()* : cette fonction renvoie le n-ième élément n'ayant aucun sequence flow en sortie.
- *TrouverOrdreMinServiceDeFIN(x)* : cette fonction est la même que la fonction *TrouverOrdreMinServiceDeINI(x)*, à ceci près qu'elle s'intéresse aux éléments finissant les chaînes d'implication.
- *ConstructionDesSéquencesDuProcessusCollaboratif()* : cette fonction permet de créer les liens entre les différents services sélectionnés en respectant l'ordre de traitement défini, ainsi que le parallélisme des services. Il s'agit donc de la fonction « finale » permettant la construction du processus collaboratif, sur la base de la prise en compte des diverses contraintes de parallélisme, de précédence et d'enchaînement. Cette fonction est décrite par l'Algo. II-2

```

Nom : Construction des séquences du processus collaboratif

Objectif : Construire les séquences du processus collaboratif à partir
des informations sur les services et l'ordre de traitement des problèmes

Début :

-- Initialisation des variables INI et FIN pour les services sélectionnés
Initialisation_INI_FIN()

Pour numOrdre = 0 à OrdreTraitementMaximum Faire

```

```

Service_en_cours ← TrouverServiceParOrdre(numOrdre)

Pour chaque x ∈ SameTime(x) Faire
    Si  $\exists n, m \in \mathbb{N}^2$  tel que PreRequisn(Service_en_cours) = PreRequism(x)
alors
    -- on vérifie que le couple (n,m) est le plus petit possible
    Tant que  $\exists k < n, 1 < m \in \mathbb{N}^2$  tel que PreRequisk(Service_en_cours) =
Prerequis1(x) faire
        n ← k
        m ← 1
    Fin Tant que
    g1 ← CréationGateway()
    CréationSeqFlow(PreRequisn(Service_en_cours), g1)
    CréationSeqFlow(g1, PreRequisn-1(Service_en_cours))
    CréationSeqFlow(g1, PreRequism-1(x))
    Sinon
    g2 ← CréationGateway()
    CréationSeqFlow(g2, INI(Service_en_cours))
    CréationSeqFlow(g2, INI(x))
    INI(Service_en_cours) ← g2
    INI(x) ← g2
    Fin Si
    -- on fait le même traitement pour les activités de fin
    Si  $\exists n, m \in \mathbb{N}^2$  tel que Impliquen(Service_en_cours) = Impliquem(x)
alors
    -- on vérifie que le couple (n,m) est le plus petit possible
    Tant que  $\exists k < n, 1 < m \in \mathbb{N}^2$  tel que Impliquek(Service_en_cours) =
Implique1(x) faire
        n ← k
        m ← 1
    Fin Tant que
    g3 ← CréationGateway()
    CréationSeqFlow(g3, Impliquen(Service_en_cours))
    CréationSeqFlow(Impliquen-1(Service_en_cours), g3)
    CréationSeqFlow(Impliquem-1(x), g3)
    Sinon
    g4 ← CréationGateway()
    CréationSeqFlow(FIN(Service_en_cours), g4)
    CréationSeqFlow(FIN(x), g4)
    FIN(Service_en_cours) ← g4
    FIN(x) ← g4
    Fin Si
Fin Pour

```

```

Fin Pour
-- On crée les séquences flows manquants pour finaliser le processus.
Si NbElementSansSeqFlowIn() = 1 Faire
    CréationSeqFlow(Start, ElementSansSeqFlowIn1())
Sinon
    Soit Sin un ensemble d'entier
        Pour i=1 à NbElementSansSeqFlowIn() Faire
            Sin ← Sin ∪ {TrouverOrdreMinServiceDeINI
(ElementSansSeqFlowIni())}
            Fin Pour
            Tri croissant des éléments de Sin
            -- Soit Sin(i) l'élément i de l'ensemble
            -- On relie l'élément ayant l'ordre de traitement le plus important (le
plus faible) à l'élément start
            CréationSeqFlow(start, INI(TrouverServiceParOrdre(Sin(j+1)))
                Pour j = 1 à nombre d'éléments de Sin - 1 Faire
                    CréationSeqFlow( FIN(TrouverServiceParOrdre(Sin(j))),
INI(TrouverServiceParOrdre(Sin(j+1)))
                Fin Pour
Fin Si
Si NbElementSansSeqFlowOut() = 1 Faire
    CréationSeqFlow(ElementSansSeqFlowOut1())
Sinon
    -- il y a une erreur, car tous les services auraient dû être reliés entre
eux.
Fin Si
Fin

```

Algo. II- 2 : fonction construction des séquences du processus collaboratif

### III. Illustration de déduction d'un processus collaboratif

Dans cette section, nous allons illustrer la démarche de déduction d'un processus collaboratif de réponse à une crise en nous attardant particulièrement à l'étape de construction du processus (étape 3). Pour cela, nous allons très rapidement passer sur les premières étapes correspondant au recueil des services des partenaires (étape 0), à la création du modèle de crise (étape 1), aux définitions sémantiques ainsi qu'aux choix, faits par la cellule de crise, des services à utiliser pour chaque problème (étape 2).

#### III.1 Présentation du contexte

Nous avons décidé d'illustrer cette étape en nous basant sur un exercice réalisé par la préfecture du Tarn, le 24 février 2009. Ce choix se justifie par le fait que nous étions présents et avons pu suivre le

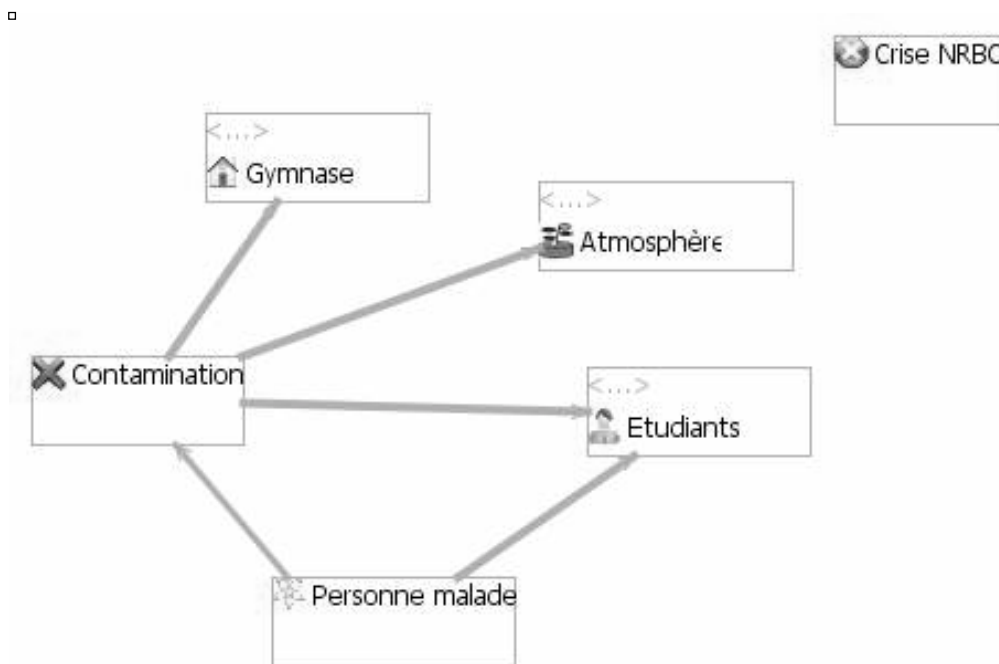
déroulement de la réponse, à la fois au niveau de la cellule de crise qu'au niveau de la mise en œuvre pratique de la réponse sur le site de l'événement.

Cet exercice, de type NRBC (Nucléaire, Radiologique, Bactériologique, Chimique), simulait une attaque terroriste et s'est déroulé sur le campus de l'école des Mines d'Albi-Carmaux. Le scénario de l'exercice est le suivant :

« Parmi un groupe d'étudiants participant à des activités sportives au gymnase de l'école, certains sont pris de malaise et de quintes de toux. Les pompiers appelés sur place afin de soigner ces étudiants décident, du fait de la présence d'une poudre blanche dans les vestiaires, de diagnostiquer une situation de crise. Cette décision est le point de départ de la réponse à la crise et déclenche ainsi la mise en place d'une cellule de crise ».

Lors de la mise en place de la cellule de crise, nous avons pu remarquer que les partenaires résumaient la situation sur un tableau blanc. Au niveau de la démarche, cette étape correspond à la création d'un modèle de crise. Ce modèle, présenté ci-dessous, contient les éléments suivants :

- Les **étudiants malades** (type *population*), impactés à la fois par le *risque* de **contamination** (type *risque*) et la *conséquence* désignée par le terme **personne malade** (type *conséquence*).
- Les **étudiants** (type *population*), soumis au *risque* de **contamination** (type *risque*).
- L'**atmosphère** environnante (type *site naturel*) et le **gymnase** (type *bien*), sont soumis au *risque* de **contamination** (type *risque*).



**Figure II- 14 : modèle de crise**

L'ordre de priorité des problèmes défini par la cellule de crise est le suivant :

1. risque de contamination,
2. conséquence de personne malade.

Concrètement, la cellule de crise a décidé que la prévention du risque de contagion était prioritaire vis-à-vis de la prise en charge des étudiants intoxiqués. Lors de la phase de préparation à la gestion de crise, réalisée avant l'occurrence de cette crise, les modèles de description des services, proposés par chaque partenaire, ont été définis (disponible en annexe B). L'exécution des règles de déduction RDSU a permis

de filtrer l'ensemble des services utilisables pour les problèmes identifiés. Cependant, de par la simplicité de cet exemple au niveau des règles RDSU, nous avons préféré insister sur la construction du processus collaboratif à travers cet exemple. Nous allons donc directement présenter le résultat des choix.

La cellule de crise a fait les choix suivants :

- pour prévenir le *risque* de **contamination**, les *services* suivants ont été choisis :
  - le service **Maintenir un périmètre de sécurité** du partenaire Gendarmerie,
  - le service **Décontaminer** du partenaire SAMU.
- Pour traiter la *conséquence* de **personne malade**, les *services* suivants ont été choisis :
  - le service **Soigner** du partenaire SAMU,

Pour des raisons de lisibilité, nous présentons uniquement une description, sous forme de tableau, des services qui seront utilisés par la réponse aux vues des choix de la cellule de crise. Néanmoins, les modèles de description des services sont disponibles en Annexe B.

#### Partenaire : Gendarmerie

**Tableau II- 1 : tableau de description des services des gendarmes**

Service	Abréviation	Prévient	Réduit	Nécessite	Implique
Définir un périmètre de sécurité	Def. Per. Sec.			Information Météo (condition)	
Mettre en place le périmètre de sécurité	Met. Pla. Per. Sec.			Def. Per. Sec.	
Enlever le périmètre de sécurité	Enl. Per. Sec.				
Maintenir le périmètre de sécurité	Main. Per. Sec.	Risque de contamination		Met. Pla. Per. Sec.	Enl. Per. Sec.

#### Partenaire : Croix Rouge

**Tableau II- 2 : tableau de description des services de la Croix Rouge**

Service	Abréviation	Prévient	Réduit	Nécessite	Implique
Mettre en place un poste médical avancé	Met. PMA				Poste médical installé (condition)

#### Partenaire : Samu

**Tableau II- 3 : tableau de description des services du SAMU**

Service	Abréviation	Prévient	Réduit	Nécessite	Implique
Soigner	Soin		Personne malade	Decon.	
Décontaminer	Decon.	Risque de contamination	Personne contaminée	Equ. Med. ; personne évacuée et poste médical installée	
Equiper les médecins	Equ. Med.				

### Partenaire : SDIS

Tableau II- 4 : tableau de description des services du SDIS

Service	Abréviation	Prévient	Réduit	Nécessite	Implique
Equiper les Pompiers	Equ. Pom.				
Préparer pour l'évacuation	Pré. Evac.			Equ. Pom	
Apporter Premiers Soins	App. PS.		Personne blessée		
Evacuer	Evac.			Pré. Evac. et App. PS.	Personne évacuée (condition)

## III.2 Dédution des composants du processus collaboratif

Dans cette partie, nous allons exécuter l'ensemble des règles de déduction. Pour cela, nous expliquerons et illustrerons leur résultat *via* un modèle de processus, qui évoluera au fur et à mesure de l'avancement de l'exécution des règles.

Les deux premières règles, les règles II-10 et II-11, créent l'ensemble des éléments de base du processus collaboratif :

- un *pool SIM*, contenant les éléments :
  - événement de départ nommé *Start*,
  - événement de fin nommé *End*,
- Un *pool* par partenaire nommé *Gendarmerie, SDIS, SAMU, Croix Rouge*.

La figure suivante illustre ce résultat.



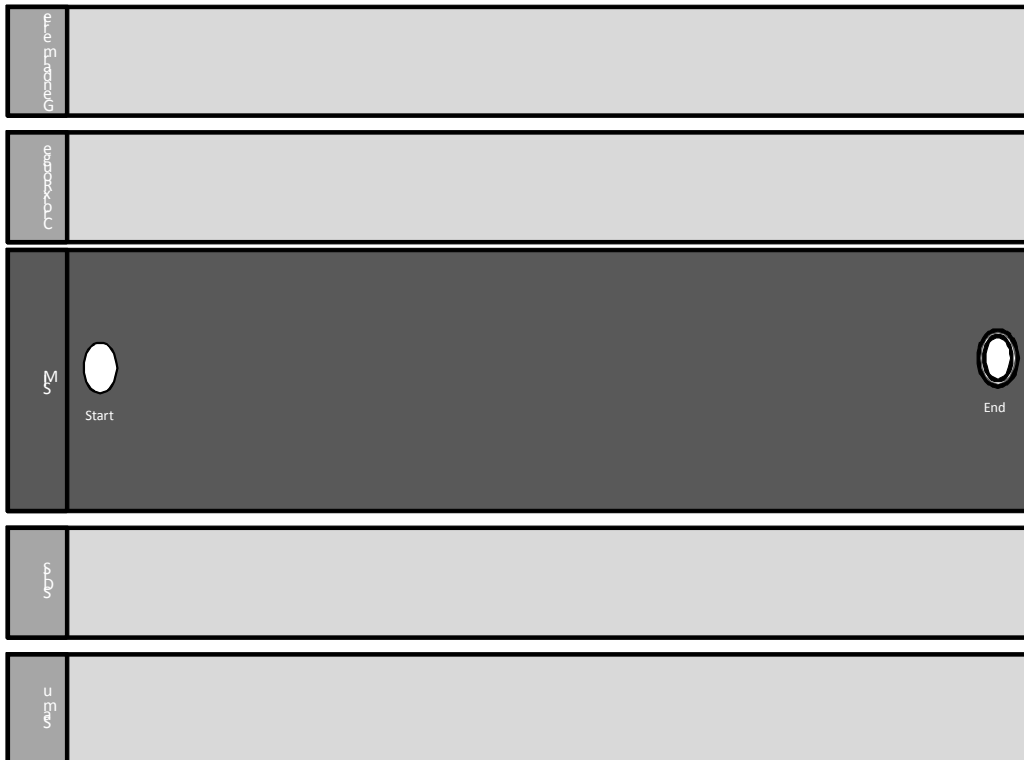


Figure II- 15 : résultats des règles II-10 et II-11

Pour chaque *service* sélectionné, *i.e.* *Maintenir périmètre de sécurité*, *Décontaminer* et *Soigner*, un *Partner\_Task*, un *SIM\_Task* et deux *messages flows* sont créés *via* la règle II-12. La figure II-16 résume le résultat de cette règle. On y retrouve les trois *services* choisis, représentés par leur abréviation, ainsi que les *SIM\_tasks* dont le nom est la concaténation de « ST\_ » et de l'abréviation du *service* qui correspond.

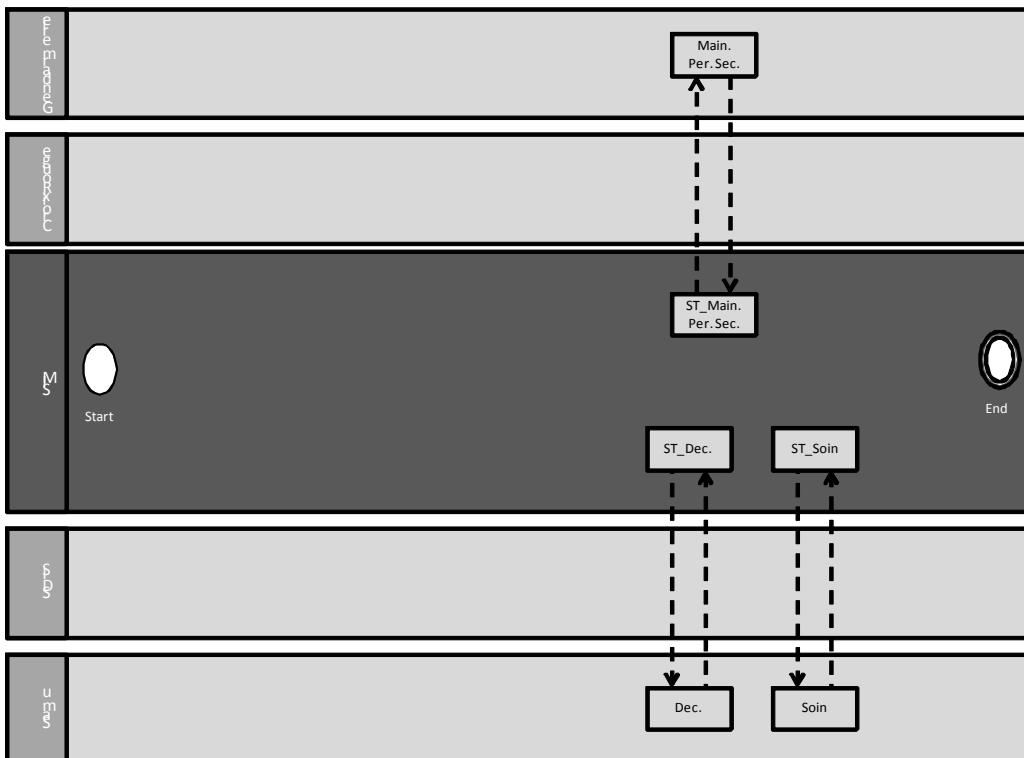


Figure II- 16 : résultat de la règle II-12

Les prochaines règles s'intéressent aux conditions de réalisation de ces *services*, la règle II-13 déduit l'ensemble des *services* définis comme directement pré-requis, alors que la règle II-14 déduit l'ensemble des *services* déduit comme pré-requis *via* une condition. Dans notre exemple, les résultats de ces règles sont les suivants :

- la règle II-13 :
  - le *service Maintenir périmètre de sécurité* a besoin comme pré-requis du *service Mettre en place le périmètre de sécurité*, qui a lui-même besoin du *service Définir le périmètre de sécurité*.
  - Le *service Soigner* a besoin en pré-requis du *service Décontaminer*, qui a lui-même besoin du *service Equiper Médecins*.
- La règle II-14 :
  - le *service Décontaminer* a besoin en pré-requis, d'une part de la *condition Poste médical installé* réalisé par le *service Mettre en place un poste médical avancé* et d'autre part, de la *condition personne évacuée* réalisée par le *service Evacuer*.
  - Le *service Définir le périmètre de sécurité* a besoin de la *condition informations météo*, aucun *service* ne permet de réaliser cette *condition*. Cependant, un *service de médiation*, nommé *SIM\_Météo*, permet de la réaliser.

Suite à l'exécution de la règle II-14, la règle II-13 donne de nouveaux résultats :

- Le *service Evacuer* a besoin en pré-requis des *services Préparer pour l'évacuation*, ainsi que *Apporter les premiers soins*. Ces deux services ayant besoin du même *service* pré-requis *Equiper les pompiers*.

Finalement, le résultat de ces règles est représenté par la figure II-17.

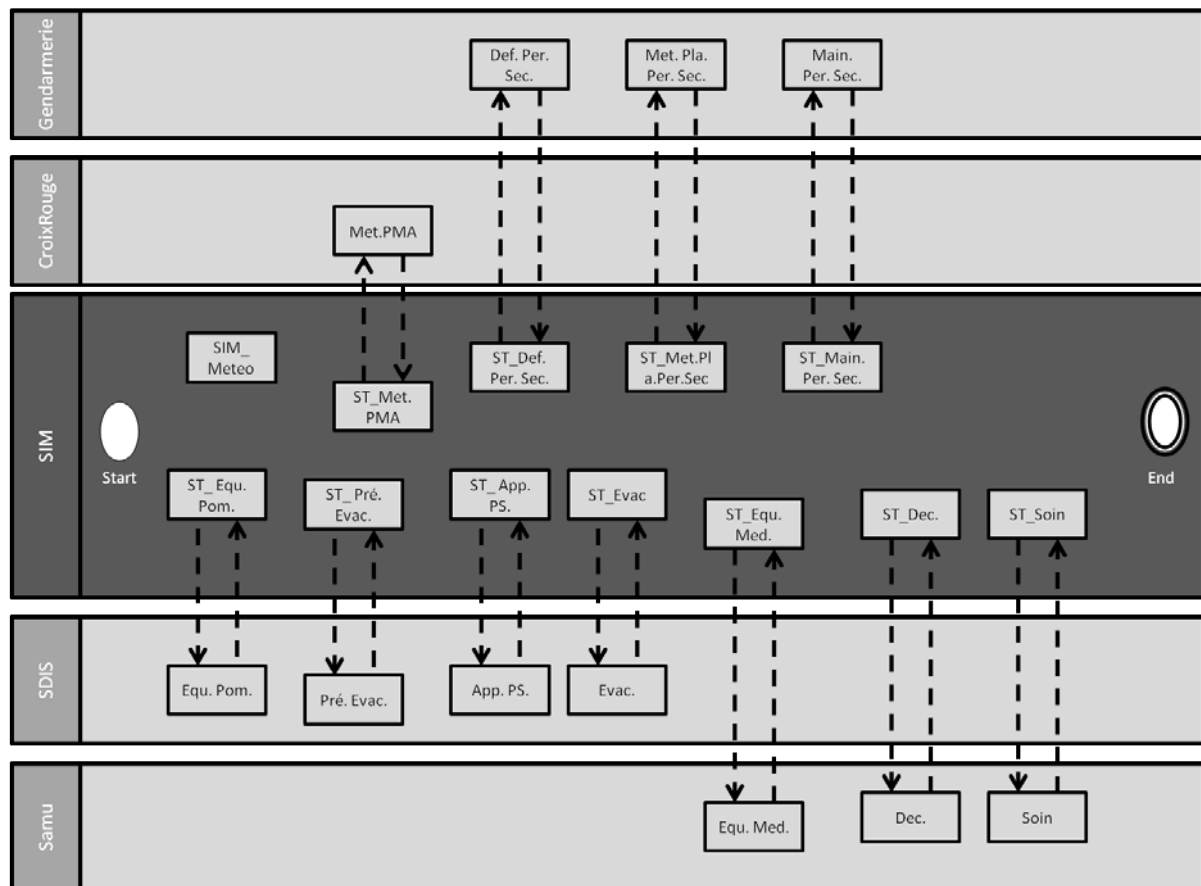


Figure II- 17 : résultat suite aux règles II-13 et II-14

### III.3 Dédution de l'ordre d'exécution des services

La partie précédente a permis de créer les éléments du processus collaboratif. Dans cette partie, nous allons nous intéresser à l'exécution des règles permettant la définition de l'ordre d'exécution de ces services. Pour rappel, l'ordre d'exécution ne peut être défini qu'au niveau du pool SIM (cf. partie processus collaboratif du métamodèle). Ainsi, lorsque nous parlerons d'un lien de précedence entre deux services, il s'agira en fait des liens de précedence entre les SIM\_tasks (dans le Pool SIM) reliées aux Partner\_task de ces services (dans les Pools des partenaires).

Une première vague de règles de déduction est exécutée, afin de déterminer un premier ensemble de services pouvant être exécutés en parallèle :

- la règle II-15 déduit que les couples de *services* suivants peuvent être exécutés simultanément : (*Maintenir en place un périmètre de sécurité, Soigner*).
- La règle II-16 déduit que les couples de *services* suivants peuvent être exécutés simultanément : (*Maintenir en place un périmètre de sécurité, Décontaminer*).
- La règle II-17 déduit que les couples de *services* suivants peuvent être exécutés simultanément : (*Mettre en place un poste médical avancé, Equiper Médecins*) ; (*Mettre en place un poste médical avancé, Evacuer*) ; (*Equiper Médecins, Evacuer*) ; (*Apporter premiers soins, Préparer pour l'évacuation*).

Une fois ces règles de déduction exécutées, l'algorithme *ConstructionDesSéquencesDuProcessusCollaboratif()*, (cf. Algo II-2), est exécuté. La première étape de ce processus consiste à initialiser, pour chaque service, les valeurs de ses variables INI et FIN. Cette étape se réalise en suivant l'algorithme *Initialisation\_INI\_FIN()*.

Pour illustrer son fonctionnement, nous allons le dérouler pour les services *Maintenir le périmètre de sécurité*, *Evacuer* et *Soigner*. Le tableau II-5 résume le déroulement de l'algorithme. Chaque ligne correspond à l'évolution des valeurs des variables au début de chaque itération de la boucle.

**Tableau II- 5 : déroulement de l'algorithme *Initialisation\_INI\_FIN***

Service_début	NbPrerequis (Service_ début)	INI(x)	FIN(x)	TrouverFinServicePreRequis (Service_Debut)
Maintenir Périmètre de sécurité	1	Maintenir Périmètre de sécurité	Maintenir Périmètre de sécurité	
Mettre en place un périmètre de sécurité	1	Mettre en place un périmètre de sécurité	Maintenir Périmètre de sécurité	
Définir un périmètre de sécurité	1	Définir un périmètre de sécurité	Maintenir Périmètre de sécurité	
SIM_météo	0	SIM_météo	Maintenir Périmètre de sécurité	
Evacuer	2	Evacuer	Evacuer	Equiper les pompiers (cas A de la figure II-13)
Equiper les pompiers	0	Equiper les pompiers	Evacuer	
Soigner	1	Soigner	Soigner	
Décontamination	3	Décontamination	Soigner	Gat 4 (cas B de la figure II-13)
Gat 4	0	Gat 4.	Soigner	

A la fin de cet algorithme, le processus collaboratif obtenu est le suivant :

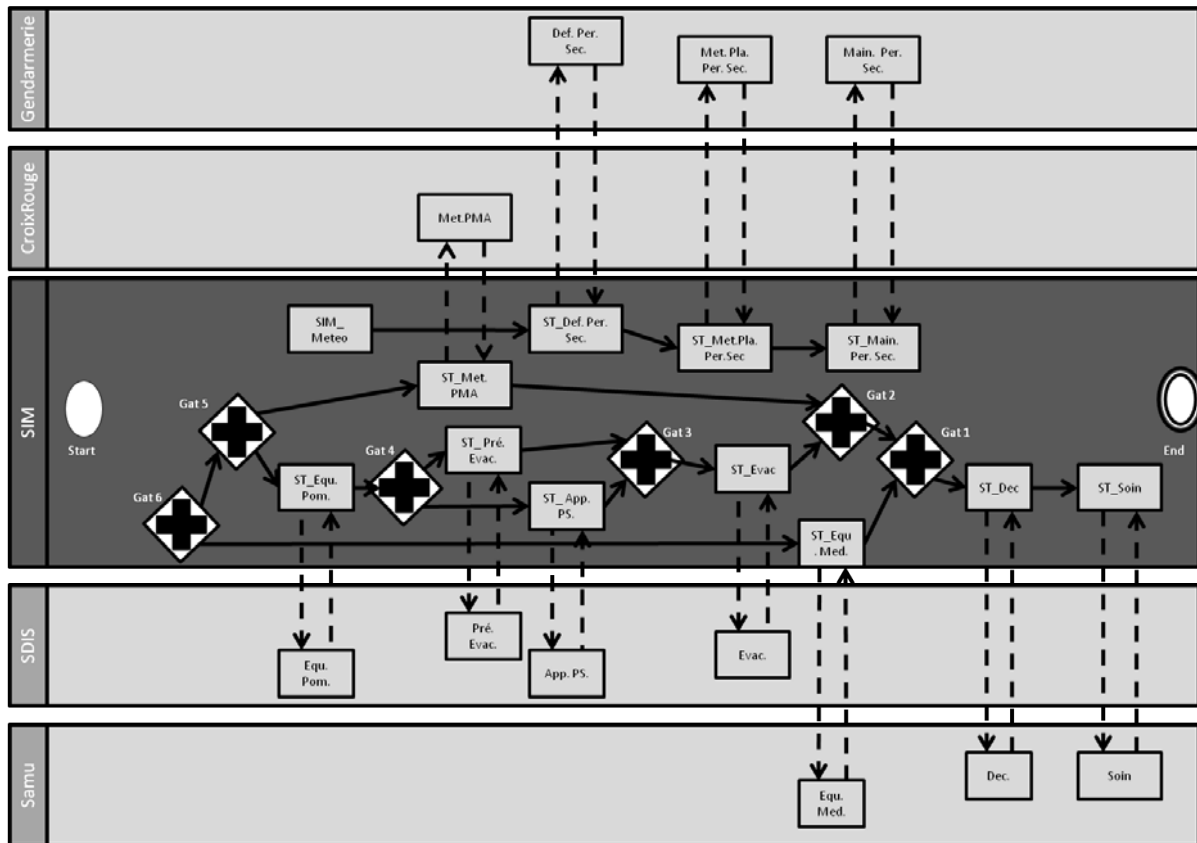


Figure II- 18 : processus collaboratif suite à l'algorithme *Initialisation\_INI\_FIN*

La suite de l'algorithme *ConstructionDesSéquencesDuProcessusCollaboratif* traite les mises en parallèle en se basant sur les résultats de la fonction *SameTime()*. Dans notre exemple, la fonction *SameTime(Maintenir le périmètre de sécurité) = {Soigner}*, alors que l'application de cette fonction aux autres services ne donne que des résultats vides. Ce résultat provient de l'application des quatre conditions de la fonction *SameTime* (expliqué en II.4.2) sur le résultat des règles de déduction II-15, II-16 et II-17.

Il n'y a par conséquent qu'une itération au niveau de l'algorithme. Les deux services n'ayant pas de pré-requis commun, deux gateways sont ajoutés, un avant INI(*Maintenir le périmètre de sécurité*) et INI(*Soigner*) l'autre après FIN(*Maintenir le périmètre de sécurité*) et FIN(*Soigner*).

Enfin, il reste à créer les séquences flows manquants :

- comme il n'y a qu'un seul gateway qui n'a pas de séquence flow entrant, un séquence flow est ajouté entre l'événement de départ *Start* et le gateway *g8*,
- comme il n'y a qu'un seul gateway qui n'a pas de séquence flow sortant, un séquence flow est ajouté entre l'événement de fin *End* et le gateway *g7*.

Finalement, la figure II-19 illustre le processus collaboratif obtenu.

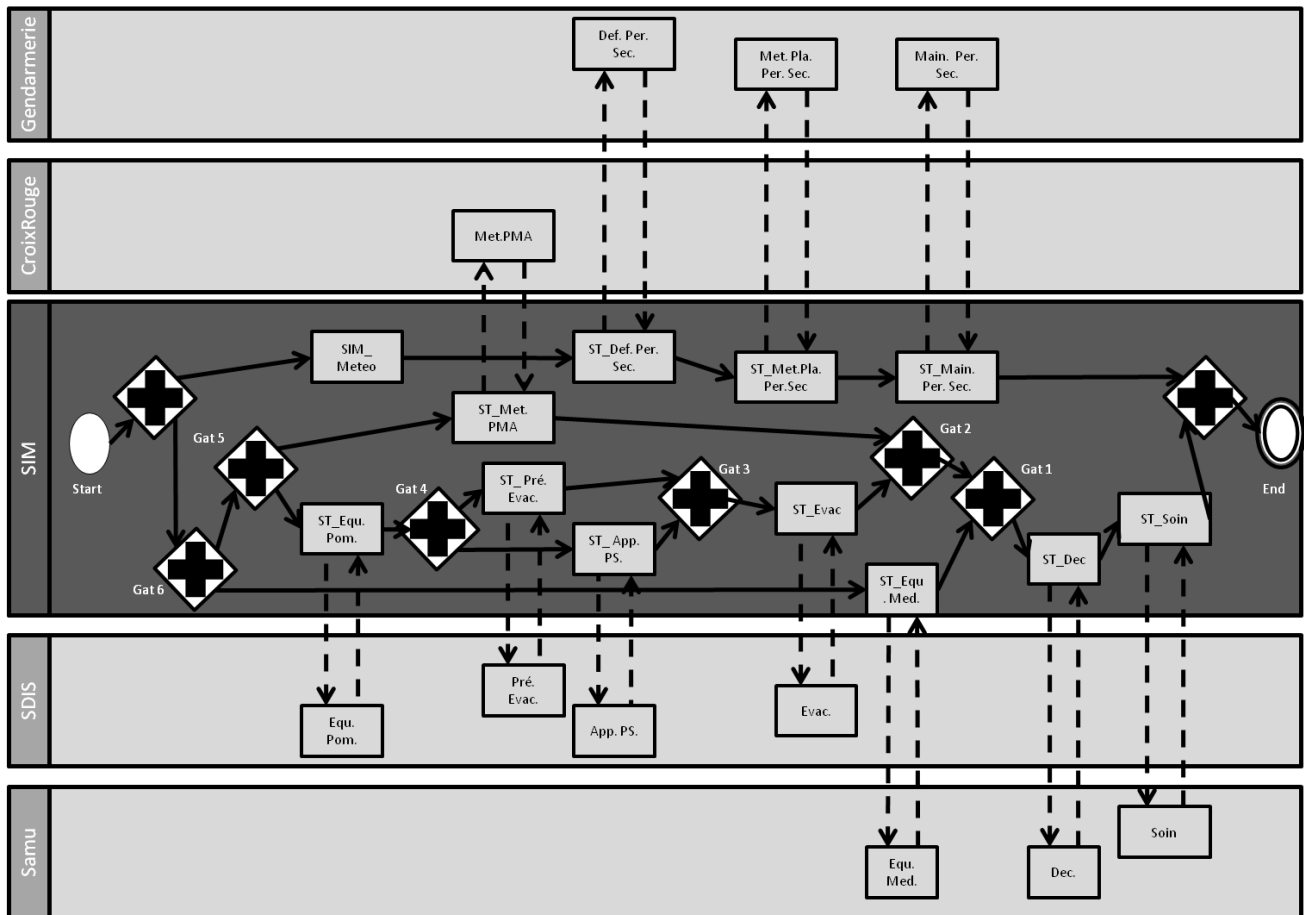


Figure II- 19 : processus Collaboratif obtenu à la fin de la démarche

## IV. Conclusion

Ce chapitre, concernant la déduction d'un processus collaboratif de réponse à une crise à partir des informations de la situation et du savoir-faire des partenaires, a été présenté en deux grandes parties, chacune correspondant à une composante principale d'une transformation de modèle. Cette transformation endogène se base sur le métamodèle de crise, présenté dans la première partie de ce chapitre. Ce métamodèle permet de faire le lien entre les différents points de vue de la crise : la situation, le système de traitement et le processus collaboratif. Ce lien entre les différents points de vue de la crise est la base de la définition des règles de transformation permettant la déduction du processus collaboratif. Ces règles, présentées dans la deuxième partie de ce chapitre, peuvent être divisées en deux vagues successives. La première vague est destinée à assurer une compréhension mutuelle entre les différents partenaires en résolvant des problèmes de nature sémantique au niveau des instances manipulées. Dans le cadre de ces travaux de thèse, les liens sémantiques entre les différentes instances doivent être définis manuellement. Cette solution est clairement insuffisante et nécessiterait de mener une exploration de solutions envisageables afin de faciliter cette définition. Cette exploration fait l'objet de travaux connexes aux nôtres (Boissel-Dallier *et al.*, 2009). La deuxième vague de règles, quant à elle, permet de construire un processus collaboratif de réponse à partir des choix réalisés par la cellule de crise (on peut noter que ces choix pourraient être guidés par un outil d'aide à la décision, ce qui est le cas dans le projet IsyCri, mais cet outil ne relève pas des travaux de la thèse dont il est ici question et a été conçu par le LGI2P de l'Ecole

des Mines d'Alès, partenaire du projet). L'ambition cachée derrière cette construction est de définir la coordination des services inter-partenaires.

Ce processus collaboratif déduit est la première pierre de notre édifice de système de systèmes d'information support de la réponse à la crise. En effet, ce modèle de processus collaboratif, qui correspond au niveau CIM de la démarche, pourra ensuite être transformé en modèle d'architecture logique de système d'information de médiation grâce à la transformation réalisée dans le cadre des travaux de thèse de J. Touzi (Touzi, 2007). Ce modèle d'architecture logique est ensuite transformé en modèle d'architecture technique. Cette transformation consiste à rajouter d'une part la correspondance entre service « métier » et service « technique », d'autre part, les informations caractéristiques des services « techniques » (cf. Annexe E). Cette architecture technique est ensuite utilisée afin de configurer un Entreprise Service Bus (ESB). Cette configuration passe notamment par la génération d'un fichier BPEL (Business Process Executed Language), objet d'étude du prochain chapitre.





# Chapitre III :

## Du modèle technique de médiation au déploiement

### *Obtention d'un médiateur opérationnel*

<b>Introduction du chapitre</b> .....	<b>70</b>
<b>I. Obtention de l'architecture technique du SIM</b> .....	<b>75</b>
I.1 Une double transformation de modèle.....	75
I.2 Métamodèle d'architecture technique.....	76
<b>II. Création des SU et SA nécessaires au déploiement</b> .....	<b>79</b>
<b>III. Création du SU-BPEL</b> .....	<b>81</b>
III.1 Le langage BPEL.....	82
III.2 Transformations de modèles .....	85
<b>IV. Conclusion</b> .....	<b>94</b>

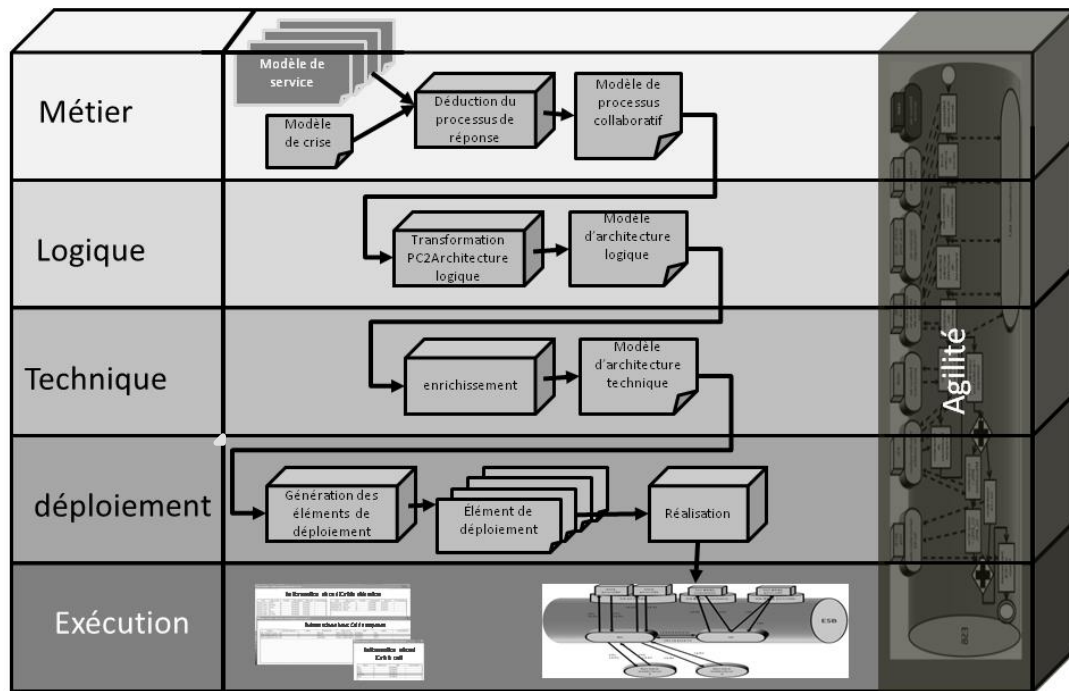
## Introduction du chapitre

Avant de présenter l'objectif de ce chapitre, nous tenons à revenir sur la notion de service (utilisé dans ce chapitre et dans le chapitre précédent). Au cours d'une démarche dirigée par les modèles, une plongée en abstraction est effectuée. Ainsi, les services du chapitre II correspondent à des services métiers des partenaires, décrivant un de leurs savoir-faire, et les services de ce chapitre correspondent à des services techniques du système d'information des partenaires. Le lien entre un service métier et un service technique est dans notre cas d'étude défini par les hypothèses posées au chapitre I, *i.e.* un service métier correspond à une opération d'un service technique (qui peut n'avoir potentiellement qu'une seule opération).

Ce chapitre a pour but de répondre à la question suivante : comment déployer un système d'information de médiation (SIM) spécifiquement dédié à supporter efficacement et de façon optimale la dynamique comportementale d'un réseau de partenaires en charge de la gestion d'une situation de crise ? Ce déploiement doit s'opérer à partir du modèle d'architecture technique du SIM, obtenu suite aux diverses transformations opérées depuis la caractérisation de la situation de crise. Cet objectif correspond à la dernière étape de la démarche, représentée par un cercle sur la figure III-1. Comme précédemment expliqué en section IV-2 du chapitre I, ce SIM est en charge (i) de la gestion des données, (ii) du management des applications (nommées services par la suite) et (iii) du pilotage de processus collaboratif. Afin de remplir ces objectifs, le SIM doit être capable d'une part de gérer l'hétérogénéité (tant du point de vue des services que des données) des différents systèmes d'information dont il doit assurer l'interopérabilité, et d'autre part, de dérouler la succession d'invocations des services définie par le processus collaboratif.

Face à ce constat, notre choix de solution technologique support du SIM s'est naturellement porté vers l'utilisation d'un Enterprise Service Bus (ESB) pour deux raisons principales :

- lors de la description des ESB (*cf.* annexe E), il a été souligné qu'un ESB permettait une connexion indirecte entre différents services issus de technologies et de philosophies hétérogènes. Cette faculté est le point d'entrée de la gestion de l'hétérogénéité des systèmes d'information. En effet, par définition un ESB doit pouvoir communiquer avec des services, quels que soient leurs protocoles de communication (SOAP, http, etc.), *i.e.* il doit prendre en charge la transformation des messages reçus dans les formats d'un service source (SOAP par exemple), vers des messages correspondant aux formats d'un service destination, (http par exemple).
- Un ESB offre des services qui lui sont propres, appelés *service engine* (SE), pouvant avoir des fonctionnalités complémentaires et intéressantes, telles que la prise en charge de l'orchestration des invocations des différents services (suivant l'ordre préétabli, par exemple, par l'intermédiaire d'un processus collaboratif).



**Figure III- 1 : positionnement du chapitre vis-à-vis de l'approche de conception d'un système d'information de médiation**

L'objectif de ce chapitre relève donc principalement de la question de la configuration d'un ESB. Un ESB étant avant tout un support technologique, la configuration de ce dernier dépend bien évidemment du produit choisi. Dans ce chapitre, nous nous limiterons à définir une démarche de configuration pour l'ESB open source PEtALS, proposé par la société PEtALS Link (anciennement nommé EBM WebSourcing), partenaire du projet ISyCri. PEtALS est basé sur la norme JBI (Java Business Integration).

La norme JBI, reposant sur le standard JSR208 (JSR 208, 2005), est une référence dans le domaine des ESB. En effet, plusieurs ESB, open source ou propriétaire, sont basés sur ce standard (on peut notamment citer Open ESB, proposé par Sun, ou encore ServiceMix, créé par Apache) ou sont au moins conformes à cette norme (on peut également citer Sonic Software ou plus récemment Mule, lancé par Mule source).

La norme JBI décrit un certain nombre de types de composants que nous allons essayer de définir vis-à-vis des besoins requis par le SIM. Face au problème de l'hétérogénéité des services de différents systèmes d'information, notamment au niveau des protocoles de communication, la norme JBI propose deux types de composants :

- le *service engine* (SE) est un service interne, directement embarqué dans le bus (pleinement compatible avec le mode de communication de l'ESB). Il est en charge de réaliser certains traitements, tels que la transformation de fichier XML ou encore de remplir le rôle de moteur de workflow.
- Le *binding component* (BC) peut être vu comme le décrypteur d'un protocole de communication. Ce BC est alors en charge de traduire les messages sortants (de l'ESB vers un service utilisant le protocole de communication) et entrants (de ce service vers l'ESB). Le BC permet donc de lier au bus des services externes utilisant un même protocole.

Pour réaliser une tâche particulière, ces composants ont besoin de récupérer les informations précises sur le traitement à effectuer. Ces informations peuvent être de deux natures : (i) soit des informations

internes qui expliquent comment réaliser une tâche, (ii) soit des informations externes qui permettent de demander la réalisation de la tâche par quelque chose ou quelqu'un d'extérieur à l'ESB. La norme JBI propose de sauvegarder ces informations dans des *services unit* (SU) pouvant être de deux types en fonction de la nature des informations qu'ils contiennent : les SU *provide* contiennent les informations permettant la réalisation de la tâche depuis un composant interne à l'ESB (BC ou SE) et les SU *consume* contiennent, quant à eux, les informations qui permettent de demander la réalisation de la tâche depuis l'extérieur.

Pour chaque service externe au bus qui doit être contacté *via* un BC (donc depuis l'intérieur), il existe un et un seul SU *provide* contenant la description de la partie publique de ce service, notamment son point d'accès (son adresse) et ses structures de messages (entrées et sorties). En effet, même si un partenaire a défini plusieurs parties publiques à un même traitement (partie privé du service), du point de vue du bus, il existe autant de service que de partie publique. Par conséquent, le bus ne peut contacter un service externe que d'un et un seul moyen. Il en est de même pour le SU *consume*.

Concernant les services internes (SE), il existe autant de SU *provide* que de tâches devant être réalisées par le SE. Ceci permet de définir les tâches indépendamment des autres et de n'installer au niveau du bus que les tâches nécessaires à un instant t. Il en est de même pour les SU *provide*.

Grâce à cette description, nous pouvons noter les points suivants :

- Un SU peut être lié soit à un BC, soit à un SE et peut avoir un rôle de *consumer* ou un rôle de *provider*.
- Un BC est mis en relation avec chaque SU décrivant chacun des services externes utilisant le protocole de communication traité par le BC.
- Un SE est potentiellement mis en relation avec une multitude de SU (un par tâche, que le SE est à même de remplir).

Au niveau du bus, il est donc nécessaire de sauvegarder le lien entre les SU et les composants (BC ou SE), afin de pouvoir exécuter correctement les tâches désirées. Dans ce but, la norme JBI propose l'utilisation de *services assembly* (SA). Les SA ne présentent pas de valeur ajoutée d'un point de vue extérieur à l'ESB. Ils sont également de deux types (*consume* ou *provide*). Ils servent d'une part à faire le lien entre un SU et un composant (un BC ou un SE) et d'autre part, ils permettent de rassembler l'ensemble des SU de même type (*consume* ou *provide*) et liés à un même composant afin de ne pas surcharger le bus (*i.e.* des SU de mêmes type, *consume* ou *provide*, correspondant à diverses tâches d'un même SE ou des SU de même type, *consume* ou *provide*, correspondant à des services externes utilisant un même protocole de communication et donc reliés à un même BC).

Le tableau III-1 résume les liens entre les SU, SA et les composants et les services extérieurs du bus. Chaque colonne représente un type de SU ou SA. La première ligne rappelle le rôle de ces différents éléments. Les lignes suivantes indiquent le nombre de SU ou SA lié à l'élément de la ligne (SE, BC ou service extérieur)

**Tableau III- 1 : lien entre les différents éléments du bus**

	SU <i>provide</i>	SU <i>consume</i>	SA <i>provide</i>	SA <i>consume</i>
Rôle	Sauvegarder les informations de réalisation de la tâche	Permettre l'exécution de la tâche par une demande extérieure	Faire le lien entre les SU <i>provide</i> et les composants	Faire le lien entre les SU <i>consume</i> et les composants

Service extérieur	1	0 ou 1	0 dédié (mais 1 SA <i>provide</i> contient le SU <i>provide</i> )	0 dédié (mais 1 SA <i>consume</i> contient son éventuel SU <i>consume</i> )
BC	0 dédié mais n en relation concernant les n services extérieurs utilisant le protocole de communication	0 dédié mais n en relation concernant les n services extérieurs utilisant le protocole de communication	Au moins 1 en relation. Eventuellement n si on ne met qu'un SU par SA	Au moins 1 en relation. Eventuellement n si on ne met qu'un SU par SA
SE	n en relation, chacun décrivant le déroulement d'une tâche	n en relation, chacun décrivant la façon d'appeler l'exécution d'une tâche		

La figure III-2 illustre un exemple des relations entre les différents éléments lors du déroulement d'un workflow. Soit A un élément extérieur au bus. A demande l'exécution du workflow *via* le SU-BPEL *consume*. Il est alors demandé au SE, correspondant au moteur d'orchestration, de dérouler ce workflow. Le SE va donc récupérer la description du workflow, contenue au sein du SU-BPEL-*provide*. Ce workflow, décrit sur la partie droite de la figure, consiste à invoquer trois opérations, deux appartenant au service extérieur 1 et une appartenant au service extérieur 2. Une fois cette description récupérée, le SE contacte le BC afin d'exécuter l'opération. Le BC va alors récupérer les informations contenues dans le SU1-*provide*, afin de demander l'exécution de l'opération11 du service extérieur 1. Une fois l'opération terminée, son résultat est envoyé au SE *via* le BC. Le même type de traitement s'effectue pour toutes les opérations à effectuer dans le cadre du workflow. Une fois le workflow terminé, le résultat final (correspondant au message de sortie du workflow) est retourné à A *via* le SU-BPEL *consume*.

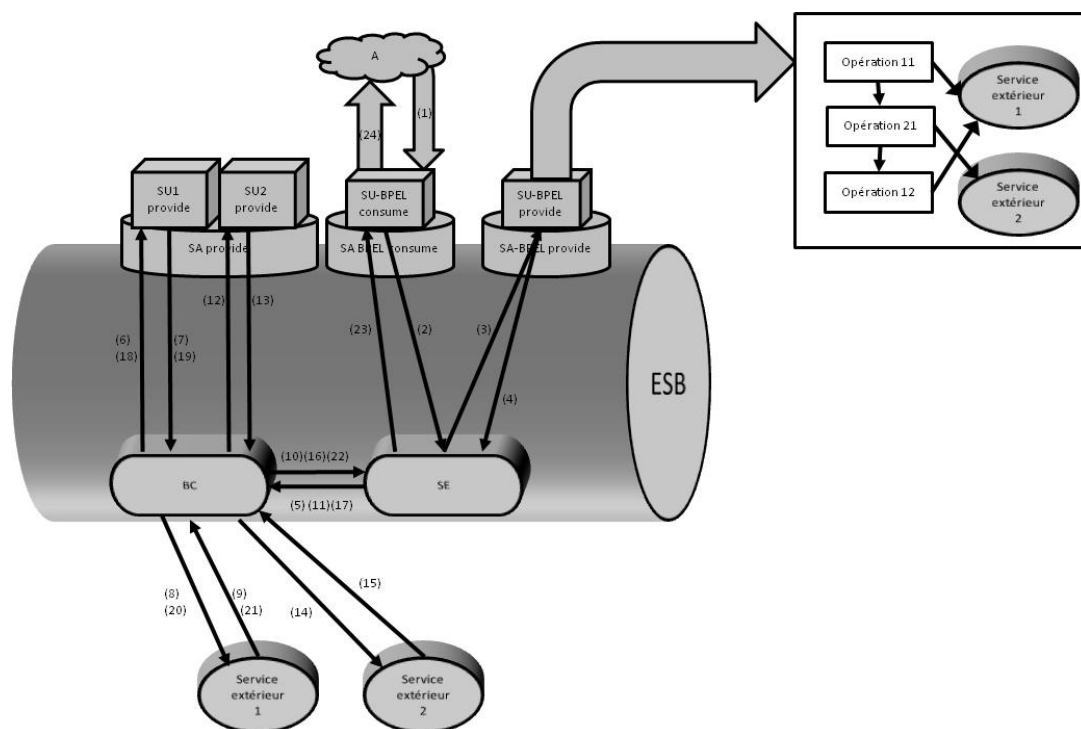


Figure III- 2 : exemple de relation entre les différents éléments d'un ESB basé sur la norme JBI

La liste ci-dessous résume l'ensemble des composants nécessaires à la configuration d'un ESB basé sur le standard JBI :

- les *service unit* (SU) *provide* : il existe un SU *provide* par service à invoquer (que le service soit interne ou externe). Un SU est composé d'un fichier *jbi.xml* (basé sur le fichier WSDL du service, lequel décrit l'ensemble des informations liées à la partie publique du service, cf. annexe E), permettant de le décrire de façon unique vis-à-vis de l'ESB. De l'ensemble de ces SU *provide*, certains servent à définir un workflow qui sera exécuté par un moteur d'orchestration, appelé SE-BPEL. Ces SU seront appelés par la suite SU-BPEL.
- Les *service unit* (SU) *consume* : un SU *consume* est un point d'accès sur le bus qui permet l'exécution, par quelque chose d'extérieur au bus, d'un traitement défini par un SU *provide*.
- Les *binding component* (BC) : chaque BC est utilisé pour transformer, de façon bilatérale, les messages entre l'ESB (environnement JBI) et tous les services externes qui se basent sur le protocole de communication pris en charge par le BC.
- Les *service engine* (SE) : chaque SE est un service interne proposé par l'ESB. La description des actions qu'il doit exécuter est définie et stockée dans un (ou des) SU *provide*. De l'ensemble des SE, nous distinguerons le moteur d'orchestration en l'appelant SE-BPEL.
- Les *service assembly* (SA) : permettent de faire le lien entre les SU et les composants du bus. Un SA peut regrouper différents SU, à condition qu'ils utilisent le même composant et qu'ils aient le même rôle (*consume* ou *provide*). Un SA peut donc par conséquent être aussi de rôle différent.

Ayant une vision plus claire des différents éléments nécessaires à la configuration d'un ESB basé sur JBI, la question principale de ce chapitre concerne la création de l'ensemble des SU et SA en fonction de la caractérisation d'une collaboration. En effet, les services des partenaires ne seront présents que sous la forme de services externes. Par conséquent, les SE seront, dans notre cas, principalement des services de médiation (qui, s'ils doivent être définis par nos soins, devront néanmoins être fournis nativement avec l'ESB, hors de toute la déduction dynamique dont il est ici question). Les BC, quant à eux, doivent également être fournis nativement avec l'ESB (*i.e.* hors de toute déduction dynamique) afin de proposer les différents protocoles nécessaires à l'intégration fonctionnelle des éventuels services externes partagés par les partenaires de la collaboration (voire des services de médiation, qui pourraient éventuellement être externes). En conclusion, les SE, comme les BC, ne relèvent pas de l'activité de transformation dont il est ici question, alors que la création ainsi que le déploiement sur le bus des SU, et par conséquent également des SA, sont spécifiques à la collaboration considérée et vont évoluer d'une collaboration à l'autre, puisque l'ensemble des services à invoquer dépend de la réponse à la crise choisie.

Si ce chapitre s'intéresse au déploiement opérationnel du SIM (sous la forme d'un ESB) à partir d'un modèle d'architecture technique, il est indispensable de revenir sur les mécanismes qui permettent l'obtention de ce modèle d'architecture technique. Nous allons donc procéder à une description rapide des étapes intermédiaires entre le résultat du chapitre II (à savoir le modèle de processus collaboratif) et le point d'entrée du chapitre III (à savoir le modèle d'architecture technique), car même si ces résultats ne relèvent pas de nos travaux, il est indispensable d'éclaircir les mécanismes sollicités afin de démontrer la complétude de la chaîne de conception de SIM :

- le chapitre II a présenté la construction, à partir d'un modèle de caractérisation de la crise, d'un modèle de processus collaboratif dédié à la résolution de cette situation en se basant sur les

fonctionnalités (services) disponibles et invocables chez les différents partenaires. Il s'agit du niveau **métier**.

- Les travaux de thèse de Jihed Touzi (Touzi, 2007) permettent de réaliser la transformation d'un modèle de processus collaboratif (respectant le métamodèle utilisé dans le chapitre II) en un modèle d'architecture logique de médiation basé sur les préceptes SOA. Il s'agit du niveau **logique**.
- Les travaux de post-doctorat de Jihed Touzi et de Master de Wenxin Mu (Bénaben et al., 2010) permettent de réaliser la transformation d'un modèle d'architecture logique de médiation orientée services en un modèle d'architecture technique de médiation orientée services. Il s'agit du niveau **technique**.

Ainsi, la deuxième partie de ce chapitre revient globalement sur les travaux de Jihed Touzi et de Wenxin Mu afin (i) de décrire ces résultats du projet MISE, (ii) de mettre en évidence la continuité de l'ensemble de la démarche du projet ISyCri et (iii) de faciliter la compréhension des travaux exposés dans ce manuscrit, et en particulier la génération des fichiers de déploiement du SIM.

La troisième partie de ce chapitre s'attache ensuite à décrire la création des SUs et SAs et plus particulièrement la création de ceux liés au moteur d'orchestration. Nous illustrerons ensuite la configuration de l'ESB en nous basant sur le processus collaboratif de réponse à la crise NRBC obtenue au chapitre II.

## I. Obtention de l'architecture technique du SIM

### I.1 Une double transformation de modèle

Contrairement au projet MISE, le projet ISyCri s'appuie sur le fait que pour tout service « métier » d'un partenaire, il existe une opération d'un service « technique » au sein de son système d'information, image de ce dernier (*cf.* chapitre I). Cette hypothèse va nous permettre d'obtenir le modèle d'architecture *technique*, point d'entrée de l'étape de déploiement, à partir du modèle de processus collaboratif, grâce à une transformation de modèle. Cette transformation est en fait réalisée en deux temps : la première étape, basée sur les résultats des travaux de thèse de (Touzi, 2007), relève d'une transformation exogène ayant comme métamodèle source le métamodèle de processus collaboratif (Touzi, 2007) et comme métamodèle cible un métamodèle d'architecture *logique* de médiateur (Touzi, 2007), inspiré du modèle PIM4SOA (Benguria et al., 2006). Cette première transformation de modèle permet d'obtenir, à partir du modèle de processus collaboratif, un modèle d'architecture *logique* du médiateur. Cependant, cette architecture logique ne comportant pas d'informations liées à la partie publique des services techniques, elle n'est pas suffisante en tant que point d'entrée pour la construction de l'ensemble des SUs et SAs. La nécessité d'enrichir ce modèle à l'aide des informations manquantes a donc donné naissance au sujet de Mastère de recherche de Wenxin Mu. Ces travaux ont amené à la création du métamodèle d'architecture *technique* du médiateur (appelé UMLT), ainsi qu'à la définition de règles d'enrichissement du modèle, complémentaires des règles proposées dans (Touzi et al., 2009).

La figure III-3 résume cette double transformation de modèle. Le métamodèle source correspond au métamodèle de processus collaboratif, présenté au chapitre II. Le métamodèle cible, appelé métamodèle d'architecture *technique*, est basé sur le métamodèle d'architecture *logique* (Touzi, 2007) auquel des attributs et des classes de description de messages ont été ajoutés afin de récolter les informations techniques nécessaires à la création des SAs et SUs.

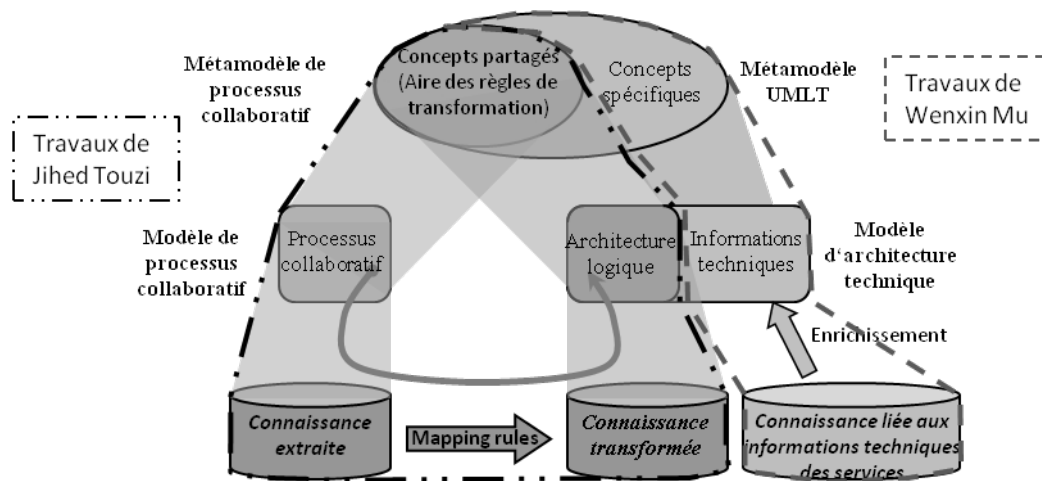


Figure III- 3 : description de la transformation de modèle

Cette section a surtout pour vocation de permettre au lecteur de comprendre la jonction qui existe entre, d'une part, le modèle de processus collaboratif déductible sur la base de la démarche présentée dans le chapitre II et, d'autre part, le modèle d'architecture *technique* de médiateur qui va être utilisé pour poursuivre la chaîne de déploiement du SI de médiation. La continuité de cette chaîne est évidemment critique pour la crédibilité des travaux présentés dans ce manuscrit. Les maillons existants, issus de travaux autres que les nôtres ayant été présentés globalement, le métamodèle d'architecture technique va maintenant être détaillé.

## I.2 Métamodèle d'architecture technique

Ce métamodèle est composé de trois parties : la vue service, la vue information et la vue processus.

La vue service, partie supérieure de la figure III-4, décrit les services rendus disponibles d'une part par les partenaires de la collaboration et d'autre part par le SIM lui-même. Les concepts utilisés dans cette partie du métamodèle sont définis ci-dessous :

- la classe « *service* » est une classe abstraite, qui désigne un service impliqué dans la collaboration.
- La classe « *opération* » désigne une opération offerte par un service. Elle correspond à un service du processus collaboratif. Il est nécessaire d'y renseigner les messages d'entrées et de sorties.
- Le package « *Partners services* » désigne la partie du modèle dédiée aux services des partenaires.
- La classe « *services registry* » désigne un annuaire accessible par le SIM constitué, pour chaque service, d'une fiche de description de ce service.
- La classe « *partner service description* » est utilisée afin de renseigner les informations de la partie publique du service, telles que l'adresse du point d'accès ou encore le lien vers le WSDL du service.
- La classe « *partner service* » désigne un service offert par un partenaire dans la collaboration. Ce service est constitué d'opérations.



- Le package « *SIM services* » désigne la partie du modèle dédiée aux services du SIM.
- La classe « *collaborative service* » est une classe abstraite, qui désigne un service offert par le SIM.
- La classe « *generic service* » est une classe qui désigne un service générique, qui peut être utilisé dans bons nombres de processus collaboratifs. Ce service existe déjà dans une bibliothèque de services gérée par le SIM.
- La classe « *specific service* » est une classe qui désigne un service spécifique à concevoir pour une collaboration particulière.

La vue information, partie médiane de la figure III-4, décrit les données impliquées dans la collaboration et gérées par les différents services. Ces données sont échangées entre les services sous forme de messages. Les concepts qui y sont définis sont les suivants :

- la classe « *business object* » désigne un message échangé entre deux services. Il est composé de « *business element* », chacun décrivant le format et le nom d'un élément d'un message.
- La classe « *semantic definition* » désigne une description sémantique du message. Cette description permet d'identifier facilement un message.

La vue processus, partie basse de la figure III-4, décrit l'orchestration des appels aux services en décrivant les différentes interactions entre les services. Les concepts liés à cette vue sont listés ci-dessous :

- la classe « *collaborative business process* » caractérise le processus collaboratif devant être exécuté par le SIM. L'ensemble des informations nécessaires à l'exécution du processus par un SE d'orchestration y est aussi stocké sous forme d'attributs.
- La classe « *partner* » désigne un partenaire de processus.
- La classe « *message variable* » désigne une variable qui présente un message échangé dans le processus.
- La classe « *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.
- La classe « *basic activity* » est une classe abstraite qui désigne les classes de base dans un processus :
  - la classe « *receive* » désigne l'attente de l'arrivée d'un message,
  - la classe « *invoke* » désigne l'appel à un service,
  - la classe « *reply* » désigne la réponse à un appel de service.
- La classe « *structured activity* » est une classe abstraite, qui désigne les classes qui permettent de structurer le flux d'exécution de parties de processus :
  - la classe « *while* » permet l'exécution d'une partie de processus en boucle jusqu'à la satisfaction d'une condition,
  - la classe « *pick* » permet d'attendre la production d'un événement (arrivée d'un message, etc.) pour continuer l'exécution du processus,
  - la classe « *scope* » permet de regrouper un ensemble d'activités. Ce regroupement peut avoir ses propres variables et ses propres gestionnaires d'événements, d'après la transformation

de modèle de (Touzi, 2007), un « *scope* » est l'image sur l'architecture logique d'un sous processus du processus collaboratif,

- la classe « *switch* » permet de tester la valeur d'une variable afin de déterminer la branche d'activités qui doit être exécutée parmi plusieurs proposées,
- la classe « *sequence* » permet de lier directement deux parties du processus. L'exécution de la deuxième partie suit l'exécution de la première,
- la classe « *flow* » permet l'exécution en parallèle de plusieurs activités de processus.
- La classe « *event handler* » permet de gérer la production d'un événement qui est associé à tout le processus ou uniquement à un « *scope* ».

Les connexions entre les différentes vues présentées sont indispensables pour assurer une compréhension efficace de tout le modèle de SIM. La figure III-4 représente le métamodèle complet de notre Système d'Information de Médiation en version SOA et les connexions existant entre les différentes vues. Ces connexions sont détaillées ci-dessous :

- les deux associations « *in* » et « *out* » entre « *business object* » et « *service* » reflètent le fait que les services communiquent entre eux *via* des messages contenant (et représentant) des objets métier.
- L'association « *call* » entre « *service* » et « *basic activity* » met en évidence le fait que l'exécution d'une activité de processus sera assurée dans le workflow par l'appel d'un service.
- L'association « *represent* » entre « *business object* » et « *message variable* » illustre le fait qu'un objet métier de la vue informationnelle est représenté par une variable dans la vue processus.
- L'association « *need* » entre « *invoke* » et « *services registry* » montre que l'appel d'un service d'un partenaire doit passer par l'annuaire qui gère les moyens d'accès à ces services.

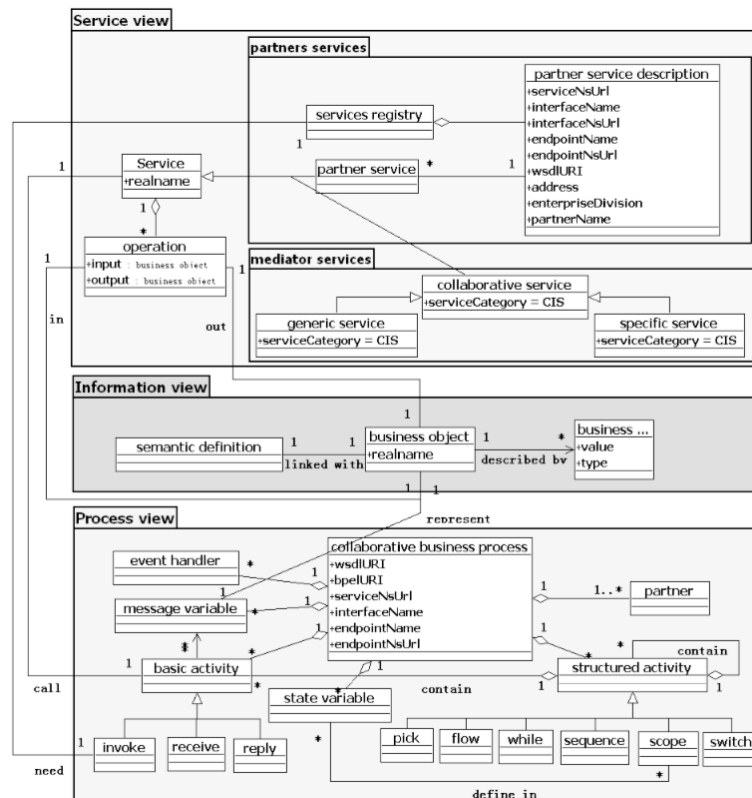


Figure III- 4 : métamodèle d'architecture technique

## II. Création des SU et SA nécessaires au déploiement

En introduction de ce chapitre, il a été expliqué que la réalisation de la dernière étape de la démarche de conception d'un SIM se concrétise par le besoin de créer différents SU et SA en fonction des BC et SE disponibles. Des deux rôles liés aux SU et SA (*provide* et *consume*), seul le rôle *provide* nous intéresse. Par conséquent, par la suite lorsque nous parlerons de SU ou SA, nous parlerons de ceux ayant un rôle *provide*. Ce choix provient de notre volonté de nous focaliser sur la réalisation d'un SIM exécutable. Dans le cadre de ces travaux, deux types de SU et SA doivent être pris en compte :

- **SA et SU liés aux services** : ces SU et SA doivent permettre de décrire les services (externes ou internes au bus) et en particulier leurs entrées/sorties, spécificités d'accès (adresses, etc.). Les éléments constituant un SU sont :
  - un premier fichier XML (de type *jbi.xml*) contenant, entre autres, les informations de point d'accès et l'éventuel protocole de communication pour un service externe.
  - un deuxième fichier XML (de type WSDL), descriptif des informations (contrat de service) liées au service concerné.

Les éléments constituant un SA sont :

- le(s) SU précédent(s) associés, soit aux services utilisant le protocole géré par le BC associé à ce SA, soit aux tâches implémentées par le SE associé à ce SA.
- Un fichier XML (de type *jbi.xml*), faisant le lien entre le(s) SU contenu(s) dans le SA et, soit le BC en charge de transformer les messages au format interprétable par les services

dépendant de ce BC et dont le(s) SU sont (est), par conséquent, dans le SA concerné, soit le SE implémentant les tâches représentées par le(s) SU contenu(s) dans ce SA.

- **SA et SU liés aux processus collaboratifs** : ces SU et SA particuliers, notés SU-BPEL et SA-BPEL, doivent permettre la configuration du moteur d'orchestration SE-BPEL. Le SA-BPEL permet de faire le lien entre le moteur d'orchestration et le SU-BPEL, lequel contient explicitement la description du processus. Ce SU-BPEL est composé des éléments suivants :
  - un fichier XML (de type *jbi.xml*) contenant, entre autres, les indications de point d'accès.
  - L'ensemble des fichiers WSDL, descriptifs des services invoqués (de type *s.wsdl*).
  - Le fichier WSDL, descriptif des entrées/sorties du processus (de type *process.wsdl*). Ce fichier contient le nom du processus, les messages d'entrées et de sorties du processus.
  - Un fichier WSDL, qui sert à faire le lien entre les demandes d'invocation au sein du processus et les fichiers WSDL des services invoqués (de type *processartifact.wsdl*). Ce fichier décrit l'ensemble des intervenants du point de vue du moteur d'orchestration.
  - Un fichier BPEL, descriptif de la succession d'invocations des services à réaliser (de type *process.bpel*). Ce fichier repose sur les fichiers *processartifact.wsdl* et *process.wsdl*.

Le SA-BPEL est composé de :

- Du SU-BPEL précédent correspondant à la description du processus collaboratif.
- Un fichier XML (de type *jbi.xml*), faisant le lien entre le(s) SU-BPEL contenu(s) dans ce SA-BPEL et le (ou les) SE-BPEL en charge de son (leur) orchestration.

Tous ces éléments constituent les couches basses de la figure III-5.

Afin de créer cet ensemble de fichiers et d'éléments nécessaires à la configuration du l'ESB PEtALS, différents éléments, issus ou non des étapes amont de notre démarche « model-driven », constituent les sources d'information entrantes :

- certaines sources sont globalement indépendantes des étapes amont de la démarche. Il s'agit principalement des éléments « mobilisables » : le moteur d'orchestration à utiliser, l'ensemble des BC et SE disponibles, l'ensemble des fichiers WSDL de ces différents services. La liste de ces éléments « mobilisables » est sauvegardée dans un fichier d'initialisation. Ce fichier permet, lors de la phase de déploiement de cette étape, de créer les SU et SA en fonction de ces éléments.
- En revanche, certaines sources sont quant à elle pleinement relatives au résultat des étapes amont de la démarche et en particulier au modèle de processus collaboratif construit : il s'agit en fait de l'ensemble des informations embarquées dans le modèle d'architecture technique, respectant le métamodèle UMLT (modèle qui correspond au modèle PSM de la démarche).

Ces deux types de sources constituent les parties « *éléments nécessaires* » et « *modèle* » de la couche haute de la figure III-5.

La partie « *modèle* » (*i.e.* le modèle PSM issu de la chaîne de déduction) sert de base à diverses transformations de modèles (réalisées grâce à l'outil ATL et expliquées dans la suite de ce chapitre) afin d'obtenir les fichiers *process.bpel* et *processArtifact.wsdl*. Ces fichiers, associés aux éléments issus de la partie « *éléments nécessaires* », sont ensuite utilisés, non seulement pour remplir le SU-BPEL, mais également pour créer les différents SUs et SAs. Cette étape spécifique (création des SUs et SAs) repose sur des travaux

effectués en parallèle aux nôtres par A.-M. Barthe dans le cadre d'une collaboration avec notre partenaire PEtALS Link et qui n'ont pas encore fait l'objet de publication à ce jour. L'annexe G décrit la démarche de création des SAs et SUs mise en place.

Ces tâches de transformation de modèles et de création de fichiers correspondent à la partie intermédiaire de la figure III-5.

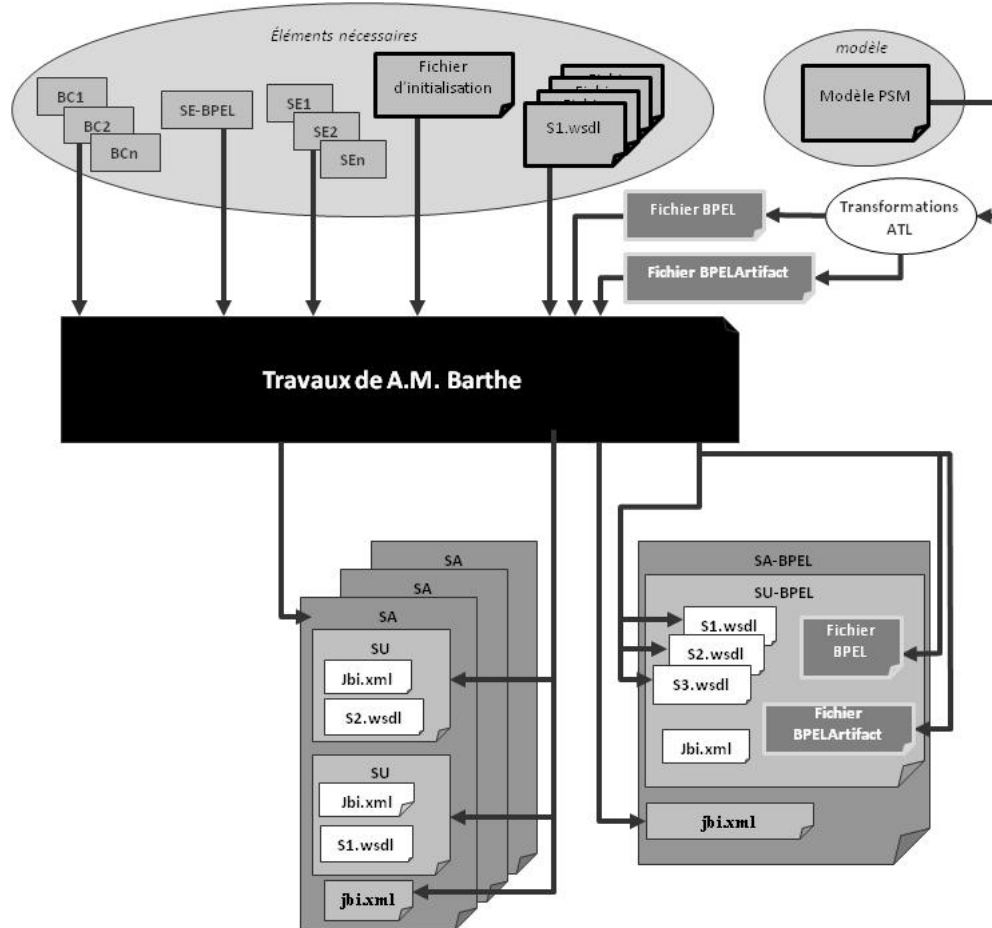


Figure III- 5 : vue d'ensemble de la démarche de conception des SAs et SUs

L'ensemble des différents SU, SA, SU-BPEL et SA-BPEL est entièrement généré par les travaux de A.-M. Barthe. La génération de ces derniers prend en entrée l'ensemble des « éléments nécessaires », ainsi que les fichiers BPEL et BPELArtifact. Ces « éléments nécessaires » et les fichiers BPEL et BPELArtifact sont au même niveau d'abstraction, correspondant au niveau du déploiement, alors que le modèle PSM est à un niveau d'abstraction différent. C'est donc la transformation du modèle PSM vers les fichiers BPEL et BPELArtifact qui correspond à une descente en abstraction et par conséquent à une transformation de modèle verticale. Par conséquent, la suite de ce chapitre se focalisera sur ces transformations.

### III. Création du SU-BPEL

Cette partie porte sur la création des éléments nécessaires à la constitution du SU-BPEL (*i.e.* l'utilisation du modèle PSM pour générer les fichiers *process.bpel* et *process.Artifact.wsdl*). Afin de décrire au mieux cette étape, nous présenterons le langage BPEL, avant de détailler les deux transformations de modèles permettant d'obtenir les fichiers *process.bpel* et *process.Artifact.wsdl*.

## III.1 Le langage BPEL

### III.1.1 Vue d'ensemble

Le langage BPEL (ou BPEL4WS pour *Business Process Execution Language for Web-Services*) a initialement été conçu pour supporter et favoriser l'utilisation de web-services *via* des processus multi-organisation (OASIS, 2007). (Andrews et al., 2003) explique que BPEL4WS constitue un langage de spécification formelle de processus métier et de protocoles d'interaction métier. Le langage BPEL est un langage expressif (Wohed et al., 2003) supporté par de nombreuses plateformes telles que PEtALS, Oracle BPEL Process Manager, IBM Websphere Application Server Entreprise et Microsoft BizTalk Server 2004. Malheureusement, le langage BPEL est loin d'être intuitif. En effet, sa syntaxe écrite, basée sur XML, le place à un niveau différent de celui d'un langage de représentation graphique (tel que peut l'être BPMN). L'intérêt d'une démarche dirigée par les modèles prend ici tout son sens, puisqu'il est évidemment plus simple de définir graphiquement un processus puis de le transformer en un fichier exécutable, que de rédiger directement ce code exécutable décrivant le workflow. Par ailleurs, il est encore plus simple de remonter encore d'un degré d'abstraction et de caractériser une situation collaborative afin de se voir proposer le modèle graphique de processus (*cf.* chapitre II), qui pourra ensuite être traduit en code BPEL.

A ce stade, on est en droit de se poser une question importante : compte-tenu de la proximité conceptuelle et sémantique entre BPMN et BPEL, pourquoi ne pas s'orienter vers un passage direct de l'un à l'autre (*i.e.* du modèle du processus collaboratif de gestion de crise au fichier exécutable représentatif du workflow) au lieu d'effectuer ce « détour » par les architectures logique puis technique ? (Ouyang et al., 2008), à l'issue d'une étude sur les différentes méthodes, techniques et approches de transformation de modèles BPMN en fichiers BPEL, explique que de telles transformations ne satisfont pas les propriétés suivantes :

- complétude : les informations contenues dans un modèle BPMN ne sont pas suffisantes pour construire un BPEL exécutable.
- Automatisation : il n'est pas possible de créer le processus BPEL de manière complètement automatisée, sans intervention humaine.
- Lisibilité : le langage BPEL est, pleinement et nativement, dédié à une interprétation informatisée (par des machines), ce qui en contrepartie le rend peu lisible par un humain. Par conséquent, la possibilité de modifier ou d'affiner (par exemple pour définir les mécanismes de transformation de données), *a posteriori*, le comportement décrit par un fichier BPEL se trouve grandement compliquée.

L'utilisation d'intermédiaires tels que les architectures logique et technique permet de répondre partiellement à ces trois critères :

- complétude : les compléments adjoints à l'architecture logique lors de sa transformation en architecture technique (*cf.* travaux de W. Mu) permettent d'enrichir le modèle de l'ensemble des informations nécessaires à l'invocation des services. Ce complément permet de générer un fichier BPEL complet sur le point en termes d'invocation des services.
- Automatisation : le fait d'avoir décomposé en plusieurs étapes permet de fractionner la démarche et de cerner très précisément la connaissance manquante qu'il est nécessaire d'apporter au modèle BPMN. Ainsi, si la démarche ne peut de toute façon pas être totalement automatisée, tout ce qui peut l'être s'intègre dans un déroulement où seules les informations à compléter sont demandées à l'utilisateur.

- Lisibilité : tout d'abord, la simple existence de tels modèles UML intermédiaires, dédiés à la description du médiateur, assure la possibilité de consulter, modifier ou corriger cette architecture avant d'exécuter les étapes de déploiement automatisé. Ces intermédiaires « réhaussent » le niveau de contrôle et d'intervention humain en offrant un panel de modèles, à des niveaux d'abstraction différents, au réajustement de l'utilisateur (là où une traduction de BPMN vers BPEL ne permet que de travailler sur le modèle BPMN).

Les autres apports de cette démarche fractionnée relèvent essentiellement du fait que cette architecture intermédiaire constitue un pivot intéressant pour la gestion du cycle de vie du SIM. En effet, ce modèle permet, du fait des caractères universel, visuel et formel d'UML, de constituer un support de maintenance et de capitalisation quant à la structuration organique du médiateur (le modèle BPMN et le fichier BPEL étant, respectivement, incapable de décrire l'architecture du SIM et difficile d'accès).

Cependant, bien que l'architecture technique soit composée de l'ensemble des informations nécessaires d'une part à l'invocation des services et d'autre part à l'ordre d'enchaînement des invocations (ordre hérité du processus collaboratif), elle ne comporte pas les informations liées à la gestion de l'information (*i.e.* le stockage, le transfert des informations d'un message en un autre). De par le manque de lisibilité du BPEL, il n'est pas envisageable de demander aux utilisateurs de renseigner ces informations directement au niveau du BPEL.

Par conséquent, en parallèle à la démarche de conception, nous avons mis en place une définition de la correspondance des informations des différents messages. Cette définition est écrite au sein d'un fichier XML (qui sera exploité par les services de médiation comme expliqué plus loin dans ce chapitre), où il est noté de façon statique que la valeur de l'élément *x* du message *X* doit être copiée puis collée dans la valeur de l'élément *y* du message *Y*. Ce transfert d'informations d'un message à un autre, bien que simpliste, est suffisamment illustratif pour démontrer la faisabilité de la définition, hors BPEL, de la gestion de l'information.

### III.1.2 Métamodèle du langage BPEL

Le langage BPEL a comme ambition de pouvoir supporter la modélisation de deux types de processus : les processus exécutables ou les processus abstraits (Andrews et al., 2003). Un processus abstrait est un protocole métier, spécifiant le comportement des messages entre les différents intervenants sans révéler leur comportement interne. Un processus exécutable spécifie l'ordre d'exécution de différentes activités, les partenaires concernés, l'échange de messages, ainsi que les problèmes et les erreurs d'exécution pouvant être rencontrés.

Nos travaux ont porté sur la création de processus exécutables, seule la partie du langage BPEL concernant cette partie est donc présentée. En outre, la gestion des exceptions étant assurée par le moteur d'orchestration maestro, réalisé par le partenaire du projet PETAALS Link, elle ne sera pas abordée.

Le langage BPEL est centré autour de la notion d'activités (« *Activity* »). Une activité peut être primitive ou structurée. Les activités primitives peuvent concerner l'invocation d'opérations de web-service (« *invoke* »), la réception d'information depuis une source externe (« *receive* ») ou l'envoi d'information à une source externe (« *reply* »).

Les activités structurées servent à définir des structures complexes du processus. Les séquences définissent l'ordre d'exécution. Les instructions « *switch* » permettent de matérialiser la séquence à réaliser en fonction d'une condition. Les instructions « *while* » permettent de répéter une séquence tant qu'une

condition (associée au « *while* ») n'est pas vérifiée. Enfin, les instructions « *flows* » sont utilisées pour paralléliser la réalisation de séquences.

Par ailleurs, le formalisme BPEL permet également de manipuler des données *via* l'instruction « *assign* ». Cette dernière permet la copie de valeur, mais également l'utilisation d'expressions plus complexes, telles que les manipulations de chaîne de caractères. Les activités primitives (*invoke*, *reply*, *receive*) utilisent des variables, définies au début du processus. Ces variables sont utilisées afin de transmettre ou recevoir les informations d'un service. Par conséquent, chaque variable a besoin d'avoir la structure (sauvegardée dans l'attribut *messageType* ou *type*) du message d'entrée ou de sortie d'un service. Les types des messages sont eux-mêmes retrouvés grâce aux importations de fichiers WSDL (« *Import* »).

La figure III-6 montre cette partie du métamodèle de BPEL, définie à partir des correspondances entre le métamodèle du formalisme BPEL (OASIS, 2007) et les composants du processus collaboratif, tels qu'ils figurent dans notre métamodèle d'architecture technique.

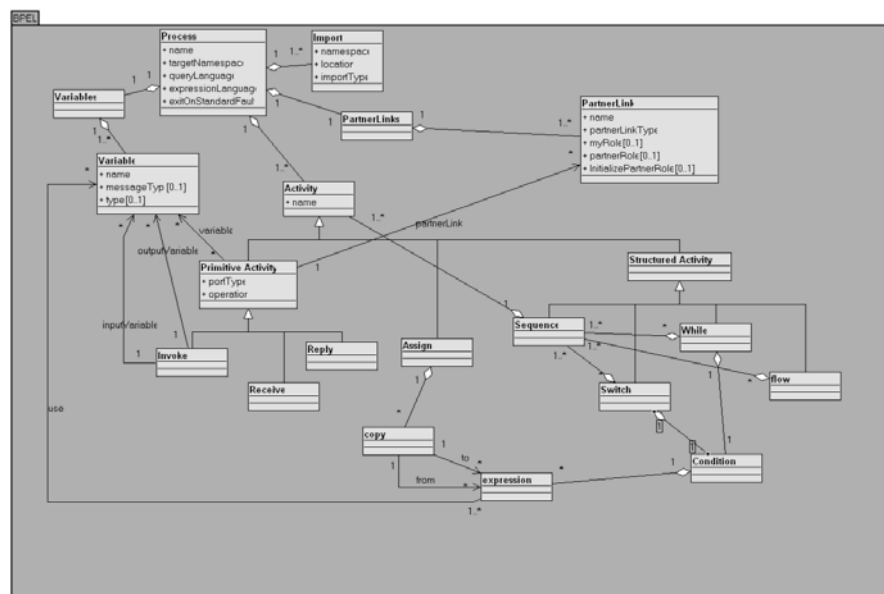


Figure III- 6 : extrait du métamodèle BPEL réalisé à partir de (OASIS, 2007)

Cette figure expose un concept que nous n'avons pas encore introduit : les *partnerLinks*. D'après (BPEL norme), la notion de *partnerLink* est utilisée dans le but de différencier les acteurs du workflow. Chaque acteur sera représenté par un *partnerLink*, *i.e.* chaque « Pool » du processus collaboratif donnera naissance à un *partnerLink*. Un acteur peut proposer un ensemble de services. Au sens du BPEL, chaque service d'un acteur correspond à un type de fonctionnalité qu'il peut assurer, représenté par la notion de rôle. Par conséquent, pour chaque service d'un acteur, un rôle sera créé. Deux types de rôle sont distingués, faisant écho aux rôles *consume* et *provide*. Le premier type de rôle, *myRole*, est attribué au service du partenaire en charge des relations extérieures au BPEL, notamment la réception du message d'entrée et l'envoi du message de fin du workflow. Ce type de rôle est équivalent au *consume* pour les SU. Le deuxième type de rôle, *partnerRole*, est attribué au service exécuté au cours du workflow. Ce type de rôle est équivalent au *provide* pour les SU.

Par conséquent, dans le cadre de nos travaux, seul le *partnerLink* correspondant au SIM pourra avoir un rôle de type *myRole*.



## III.2 Transformations de modèles

Les transformations de modèles permettant d'obtenir les modèles *process.bpel* et *processArtifact.wsdl* reposent sur le principe général de transformation de modèles défini en annexe D. Au niveau de ces transformations, nous avons noté deux catégories de règles de transformation :

- les *règles de génération de base*. Elles permettent la génération directe des éléments du modèle *process.bpel* et *processArtifact.wsdl*. Ces règles définissent les liens entre les concepts du métamodèle source (métamodèle d'architecture technique) et ceux du métamodèle cible (métamodèle BPEL).
- Les *règles à valeur ajoutée de reconstruction de l'ordre d'invocation des services et de gestion de l'information*. Ces règles permettent d'une part de construire les éléments du modèle BPEL, *via* les règles de génération de base, selon l'ordre d'exécution défini préalablement par le modèle de processus collaboratif et d'autre part, d'assurer la gestion de l'information tout au long de l'exécution de la réponse à la crise.

La suite de cette partie traite plus particulièrement des règles les plus intéressantes : celles à valeurs ajoutées. Les règles de génération de base sont quant à elles décrites en annexe F.

### III.2.1 Gestion de l'information au sein du BPEL

Afin d'expliquer la gestion de l'information que nous souhaitons mettre en place, nous allons nous baser sur une sous-partie du processus collaboratif déduit lors du chapitre II.

La figure III-7 illustre cette sous-partie, centrée sur la gestion du périmètre de sécurité.

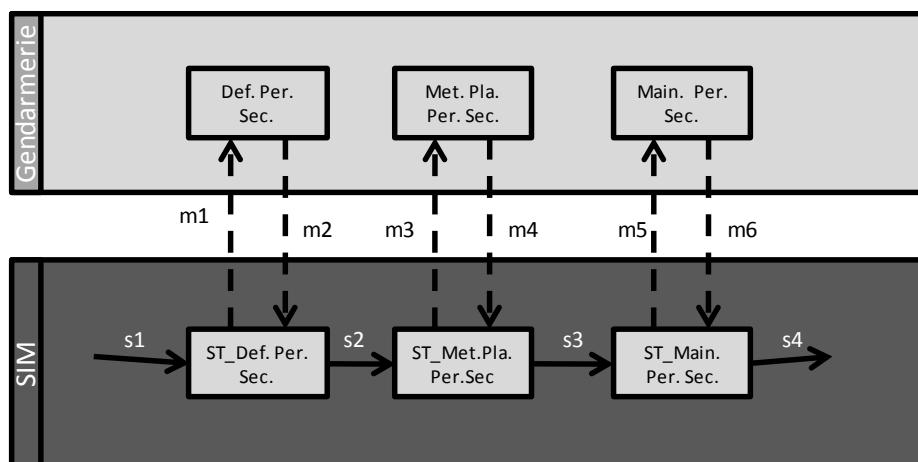


Figure III- 7 : sous-partie du processus collaboratif déduit au chapitre II

Cette partie de processus est composée de trois services hébergés sur le système d'information de la gendarmerie, de trois services de médiation ainsi que de six *message flows*, nommés m1 à m6, et enfin de quatre *sequence flows*, nommés s1 à s4.

Par la suite, les *sequence flows* seront vus comme des variables de type chaîne de caractères, alors que les *message flows* seront vus comme des structures composées d'éléments dépendant des entrées et des sorties des services.

Le tableau III-2 décrit la structure de ces différentes variables.

Tableau III- 2 : description de la structure des variables

Variable	structure	Variable	structure
m1	vitesse du vent : valeur numérique unité : par défaut nœud	s1	cc : chaîne de caractères
m2	centre : position gps diamètre : longueur unité : unité de distance par défaut en mètre	s2	cc : chaîne de caractères
m3	point initial : position gps distance à couvrir : longueur unité : unité de distance par défaut en mètre	s3	cc : chaîne de caractères
m4	nombre d'hommes déployés : entier	s4	cc : chaîne de caractères
m5	hommes en place : entier centre de la zone : position gps limite de la zone : longueur	m6	besoin : chaîne de caractères

Comme nous l'avons déjà évoqué, le transfert d'une information d'un message à un autre se base sur un fichier statique où les liens de correspondance entre les différents éléments des différents messages sont définis. Cette partie formalise le fonctionnement de ce transfert de l'information.

Afin d'expliquer la manipulation de l'information au sein de l'ESB, nous avons pris le parti de nommer chaque élément constituant un *message flow* de la façon suivante : *nom\_du\_message.nom\_de\_élément*. Par exemple, l'élément *m3.point\_initial* correspond à la valeur de l'élément *point\_initial* du *message flow m3*.

Dans le cadre de nos travaux, la manipulation de l'information se limite à la correspondance et à la continuité d'un concept au travers des différents messages qui le portent (autrement dit, il s'agit principalement de copier-coller une valeur). Par exemple, l'élément *m3.point\_initial* prendra la valeur *m2.centre* renvoyée par le service *Def. Per. Sec.*. Dans ce cas, le service de médiation *ST\_Met.Pla.Per.Sec* devra copier la valeur de l'élément *centre* de *m2* vers l'élément *point\_initial* de *m3*.

Cette manipulation est clairement insuffisante, idéalement, il serait nécessaire de pouvoir transformer, fusionner, découper une (ou des) information(s). Cependant, ce point particulier de la réconciliation sémantique des données relève essentiellement d'un objectif de démonstration de faisabilité, nous nous contenterons donc de cette correspondance primitive, tout en en reconnaissant les limites. Les principes de cette correspondance sont enregistrés dans un fichier XML. Dans le cas de notre exemple, ce fichier de « matching » serait le suivant :

- m2.centre ↔ m3.point initial,
- m2.diamètre ↔ m3.distance à couvrir,
- m2.unité ↔ m3.unité,
- m4.nombre d'homme déployé ↔ m5.homme en place,

- m2.centre ↔ m5.centre de la zone,
- m2.diamètre ↔ m5.limite de la zone.

Il est important de constater que ces correspondances de données ne se limitent pas aux relations entre une entrée d'un service et une sortie d'un service réalisé immédiatement avant (relativement à l'ordre d'exécution décrit dans le processus collaboratif). Ces correspondances portent plutôt potentiellement sur les entrées d'un service et toutes les sorties de tous les services réalisés au préalable (toujours relativement à la séquence d'exécution décrite dans le modèle du processus collaboratif). Le service de médiation doit donc récupérer l'ensemble des sorties des services déjà réalisés et fournir les valeurs du (ou des) message(s) d'entrée du service courant. Notre objectif étant de définir un service de médiation générique, il est nécessaire de se poser les questions suivantes :

- comment accéder aux résultats des services déjà réalisés ?
- Comment récupérer les valeurs du résultat du service de médiation afin de créer le message d'entrée du service suivant ?

L'idée est de sauvegarder les résultats de l'ensemble des services au niveau des *sequence flows*. Par exemple, le *sequence flow* s3 correspondra à la concaténation de s1, m2, m4. Afin de retrouver plus facilement les valeurs des éléments des messages précédents et la structure du message à créer, nous avons choisi de concaténer les informations selon une certaine logique : la première partie de la chaîne de caractère sauvegarde le nom du *sequence flow* actuel (afin de permettre de trouver l'activité suivante *via* la définition du processus collaboratif). La deuxième partie sauvegarde la (ou les) chaîne(s) de caractères issue(s) du (ou des) précédent(s) *sequence flow(s)*. En effet, dans le cas des *gateways*, il est possible d'avoir plusieurs *sequence flows* précédents. Enfin, la dernière partie sauvegarde la valeur des différents éléments du message renvoyé par le service précédent. Pour s3, ce serait par exemple m4 en tant que message de sortie du service Met.Pla.Per.Sec. La méthode de création des *sequence flows* est donc la suivante :

*Soit i le nombre de séquence flows précédents. Dans certain cas de gateway,  $i > 1$ .*

*Soit sfj le jème séquence flow précédent.*

*Soit m le message résultat du service précédent.*

*Soit ek le kème éléments du message m,  $k > 0$ .*

*sf.cc ← concaténation ('[Process]', '{SeqFlow}', 'sf') -- ici sf.cc correspond à la valeur de l'attribut cc de sf. cc prend donc comme valeur la chaîne de caractères résultat de la concaténation de l'ensemble des éléments écrit entre ' '.*

*Pour j=1 à i faire*

*sf.cc ← concaténation (sf.cc, sfj.cc)*

*Fin Pour*

*sf.cc ← concaténation ('sf', '[m]')*

*Pour k = 1 à nombre d'éléments du message faire*

*Sf.cc ← concaténation(sf.cc, '{ek}', 'm.ek')*

*Fin Pour*

En supposant que  $s1.cc = \text{« début »}$ , le résultat pour  $s3$  serait le suivant :  $s3.cc = [\text{Process}]\{\text{SeqFlow}\}s3$   
 $[\text{Process}]\{\text{SeqFlow}\}s2$   $[\text{Process}]\{\text{SeqFlow}\}s1$  début  $[m2]\{\text{centre}\}$ gymnase  $\{\text{diamètre}\}100$   $\{\text{unité}\}$ mètre  
 $[m4]\{\text{nombre d'homme}\}10$ .

Intéressons nous maintenant à la deuxième question. Pour cela, commençons par rappeler que le but du service de médiation est de générer les valeurs des éléments du message d'entrée d'une opération d'un service. Cette génération se base d'une part sur l'ensemble des informations générées par les opérations précédentes et d'autre part, sur le modèle de processus collaboratif ainsi que le fichier de correspondance, définissant les liens entre les éléments des messages.

Du fait d'une part de la spécificité des variables représentant un message d'entrée d'un service et d'autre part, de l'unicité du type du résultat de l'opération du service de médiation, il est impossible pour le service de médiation de créer directement la variable correspondant au message d'entrée de l'opération du service suivant. Il est donc nécessaire de passer par une variable intermédiaire. Cette variable, nommée *res* par la suite, est de type chaîne de caractères. Le service de médiation attribut à *res* une valeur permettant de déduire le message d'entrée de l'opération suivante. Cette attribution est faite de la façon suivante, afin de permettre de retrouver la relation entre un élément et la valeur de cet élément : « *res*  $\square$  '[nom de message d'entrée de l'opération]' '{élément1 du message}' 'valeur de l'élément1' '{élément2 du message}' 'valeur de l'élément2' ... '{élémentn du message}' 'valeur de l'élémentn' . »

Par exemple, après l'exécution du service ST.Met.Pla.Per.Sec. *res* a pour valeur : «  $[m3]\{\text{point initial}\}$ gymnase $\{\text{distance à couvrir}\}100$  $\{\text{unité}\}$ mètre ».

Pour chaque élément du message d'entrée, il suffit alors de récupérer sa valeur au sein de la variable *res* et de la copier au sein de la variable correspondant à ce message d'entrée. Cette action est réalisée par l'expression suivante :

*Pour chaque élément e du message m faire*

$m.e \leftarrow \text{substring-before}(\text{substring-after}(res, \{e\}), \{'\})$

*Fin Pour*

La figure III-8 présente un exemple générique de cette gestion de l'information. Afin de simplifier la lecture de la figure, nous avons intentionnellement masqué certains éléments du BPEL, tels que les attributs *partnerLinks* et *Rôle* pour les *invoke*, etc.

La partie gauche de la figure III-8 représente une sous-partie d'un processus collaboratif (alors que la partie droite représente son image en BPEL). Cette sous-partie est composée d'un service de médiation, d'un service de partenaire, des éléments *mi* et *mf* représentant respectivement le message d'entrée et le message de sortie du service et enfin de deux éléments *si* et *sf*, l'un représentant le *sequence flow* initial (contenant l'ensemble des valeurs des messages de sortie des services précédents) et l'autre le *sequence flow* final faisant suite à l'exécution du service.

La première action du processus BPEL caractéristique de cette sous-partie correspond à l'exécution du service de médiation. Ce service prend en entrée la valeur de *si*, puis génère, en se basant sur le fichier de correspondance et la définition du processus collaboratif, le message *mi* sous la forme d'une chaîne de caractères sauvegardée au sein de la variable *res*. En supposant que le message *mi* ne soit constitué que d'un seul élément *e*, la copie de la valeur de *e* (déduite par le service de médiation) est collée dans l'élément *e* de la variable *mi* par l'instruction écrite dans le premier *Assign* (cf. fichier BPEL de la figure III-8).

Suit ensuite l'invocation d'un service d'un partenaire dont le résultat est sauvegardé dans la variable *mf*. Nous supposons que le message *mf* n'est également constitué que d'un seul élément *e*.

Reste enfin à construire la valeur de *sf*. Cela commence par la sauvegarde du nom du *sequence flow* (qui permettra de déterminer le message d'entrée de l'opération suivante), représenté par la première partie de la deuxième expression. Ensuite, la valeur de *si* est sauvegardée afin de ne pas perdre d'information, cette opération est représentée par la partie intermédiaire de la deuxième expression. Enfin, les informations du message *mf* sont ajoutées, dernière partie de la deuxième expression.

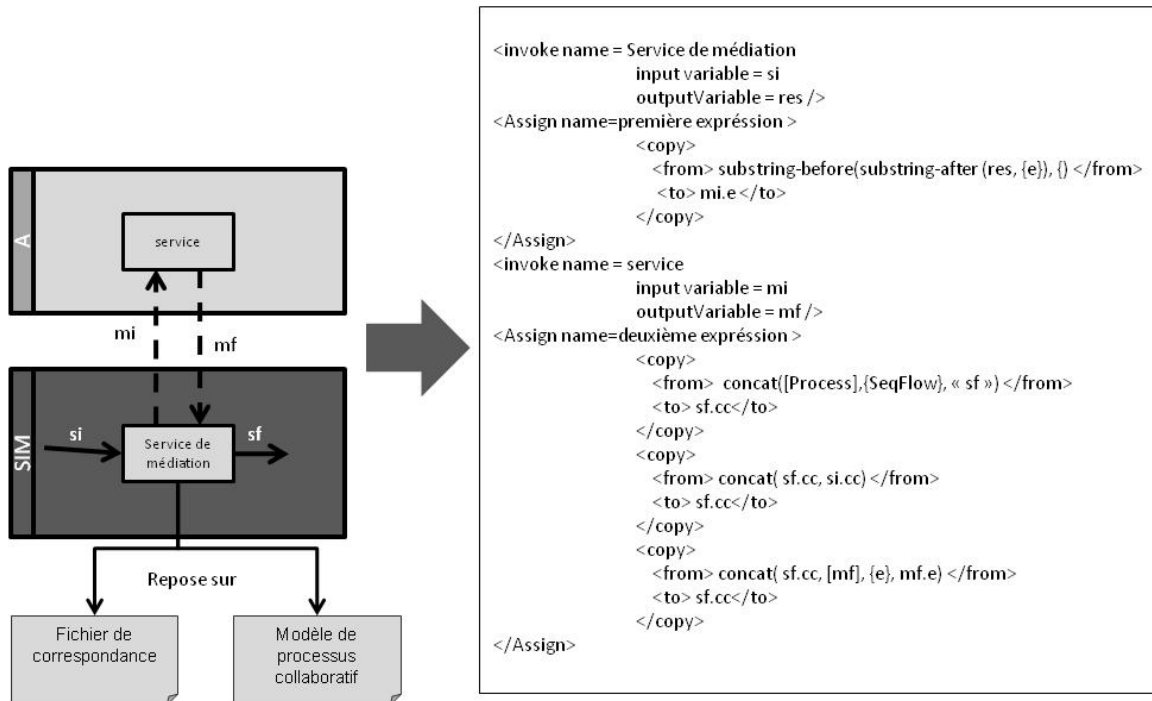


Figure III- 8 : principe de gestion de l'information au sein du BPEL

### III.2.2 Reconstruction de l'ordre d'invocation des services

Dans un fichier BPEL, si au sein d'une séquence deux sous-séquences doivent être réalisées en parallèle, elles sont définies à l'intérieur de la séquence l'une à la suite de l'autre. Cette particularité est le point de départ de notre démarche de reconstruction de l'ordre d'invocation.

Cette démarche est basée sur différentes fonctions :

- trouverActivitéSuivante(x) : où x est un *service*, cette fonction permettant de trouver le *service* ou le *gateway* suivant.
- TrouverActivitésSuivantes(y) : où y est un *gateway*, cette fonction permettant de trouver l'ensemble des *services* et/ou des *gateways* suivant un *gateway*.
- TrouverGatewayFin(x) : où x est un *gateway*, cette fonction permettant de déterminer le *gateway* associé, fermant l'ensemble des séquences parallèles ouvertes par x.
- DeterminerSousSéquence(x,y) : cette fonction est utilisée dans le cas où l'on a un *gateway fork*. Elle consiste à déterminer l'ensemble des éléments d'une sous-séquence à partir de l'élément initial de la sous séquence, x, et de l'élément final de la sous-séquence, y.

La complexité de ces fonctions n'étant pas similaire, nous ne les présenterons pas de la même façon. Les fonctions évidentes, telles que la fonction TrouverActivitéSuivante(x) consistant à déterminer l'élément suivant (correspondant à la destination du *sequence flow* ayant pour origine l'élément x transmis en

paramètre), ainsi que la fonction *TrouverActivitésSuitantes(y)* renvoyant l'ensemble des éléments suivant un *gateway* (correspondant chacun à l'activité initiale d'une sous-séquence), ne seront pas détaillées.

Nous expliquerons plus en détail les autres fonctions avant de les illustrer à travers l'exemple du chapitre II, représenté par la figure III-9.

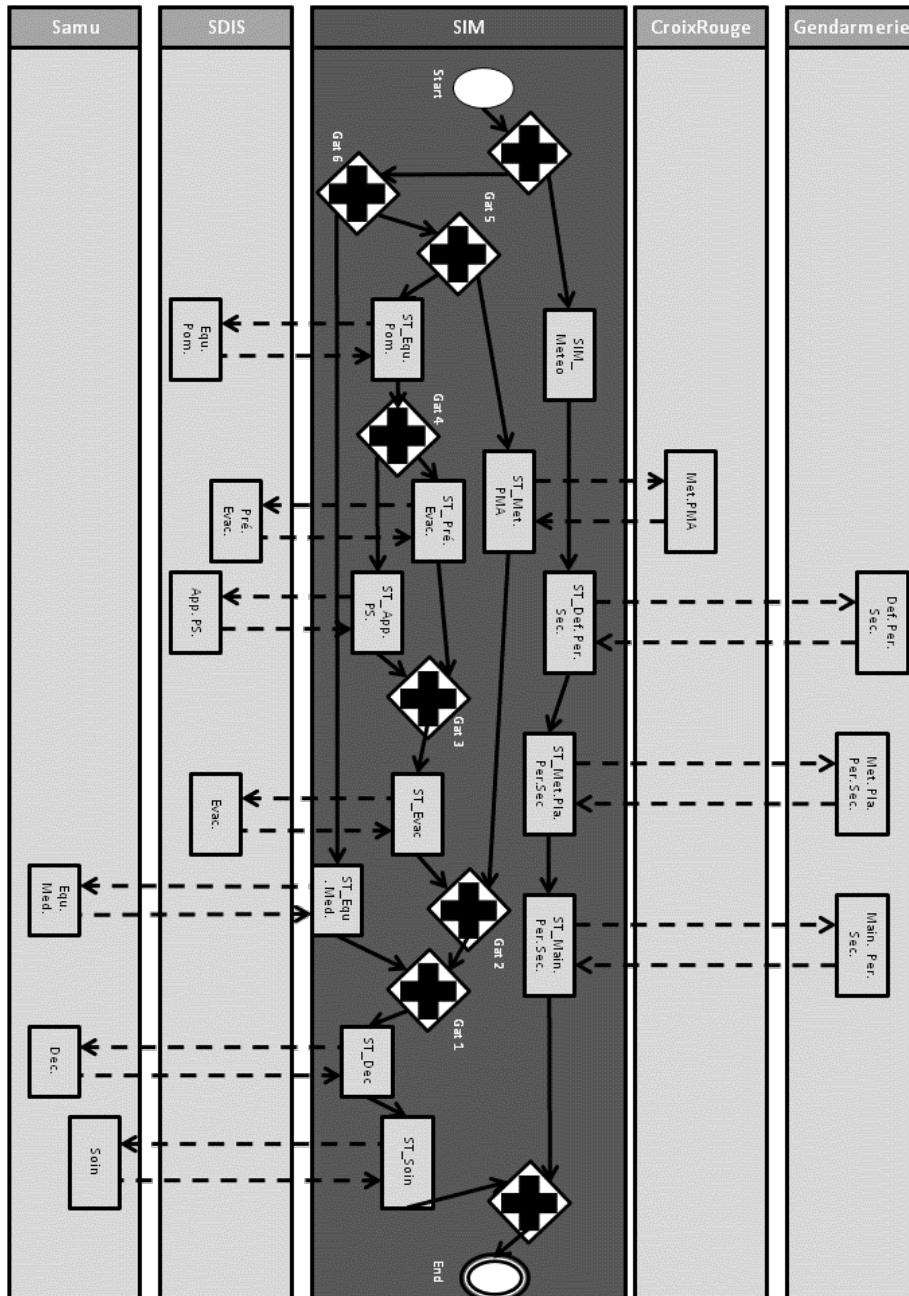


Figure III- 9 : exemple de processus collaboratif

Dans cet exemple, nous pouvons remarquer qu'il existe des sous-séquences parallèles à l'intérieur de sous-séquences parallèles (par exemple, l'une des branches parallèles situées entre les *gateways* Gat5 et Gat2 contient elle-même deux branches parallèles entre les *gateways* Gat 4 et Gat3). Ce point nous permet d'illustrer notre démarche. Commençons par les sous-séquences parallèles suivant le *gateway* Gat4. Les deux séquences sont constituées chacune d'une invocation d'un service, avant de se terminer par le *gateway* Gat3. Avec la même figure que la figure III-9, la figure III-10 représente une partie simplifiée du BPEL généré.

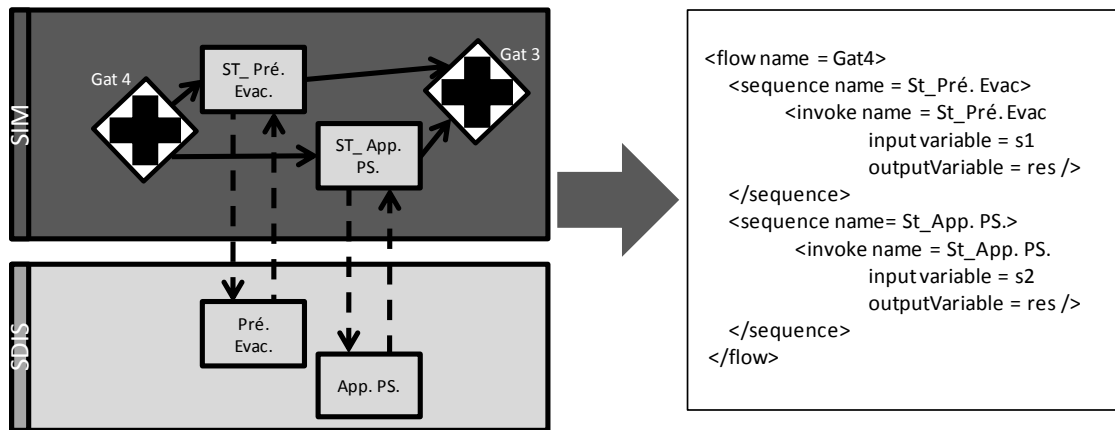


Figure III- 10 : structure d'un flow en BPEL

Dans le code BPEL, la représentation d'un service de médiation, du service de partenaire et des expressions sur les variables ont été remplacées par une simple invocation du service de médiation, alors que certains éléments complémentaires devraient y figurer en plus des éléments suivants (dans la section « *invoke* ») : la référence à l'opération, la référence au service, la référence au rôle du *partnerLink*. Cette représentation a pour but de simplifier et mettre l'accent sur la structure du BPEL. Cet exemple permet d'illustrer le fait qu'un *flow* est composé de deux sous-séquences parallèles définies l'une après l'autre. Par défaut, nous avons choisi de nommer le *flow* de la même manière que le *gateway* et chaque séquence porte le nom de son premier élément.

Ce cas simple va nous servir de base pour expliquer notre démarche concernant la gestion de plusieurs sous-séquences. Compte tenu de l'écriture d'un *flow* en BPEL, nous avons décidé de définir une séquence après l'autre. Ce positionnement nous permet de définir une fonction récursive qui permet de traiter une sous-séquence d'une sous-séquence de la même manière. Ce choix implique de connaître l'élément final de la sous-séquence, afin de pouvoir finir la fonction. Cet élément final sera déterminé par la fonction *TrouverGatewayFin(x)* ou *x* est un *gateway*.

Dans le cadre du projet, les *gateways* déduits au niveau du processus collaboratif ne peuvent être que de type « *merge* » ou « *fork* », mais pas les deux à la fois. Du fait de ce constat, la recherche de l'élément final d'une sous-séquence se base sur le nombre de *sequence flows* arrivant et sortant des *gateways*. En effet, une sous-séquence se termine lorsque le nombre total de *sequence flows* arrivant à l'ensemble des *gateways* est égal au nombre total de *sequence flows* sortant de l'ensemble des *gateways*. Pour illustrer ce positionnement, le tableau III-3 ci-dessous reprend l'exemple de la figure III-9. Les colonnes illustrent le nombre de *sequence flows* arrivant à un *gateway* et partant d'un *gateway* en fonction du *gateway* d'origine représenté en ligne.

Par exemple, prenons le cas où l'on cherche l'élément de fin d'une sous-séquence qui fait suite au *gateway* 5. Le *gateway* 5 a une *sequence flow* arrivant et deux sortants, par conséquent le nombre de *sequence flows* arrivant = 1 et le nombre de *sequence flows* sortant = 2. Ensuite, quelle que soit la branche parcourue, la recherche s'arrêtera lorsque le nombre total de *sequence flows* arrivant = nombre total de *sequence flows* sortant. Nous allons donc parcourir les deux branches pour l'illustrer. Tout d'abord en prenant la branche ST.MET.PMA, les éléments de la branche sont parcourus jusqu'à ce qu'il y ait un *gateway*, dans notre cas le *gateway* 2. Les nouveaux nombres de *sequence flows* sont alors calculés. Il arrive deux *sequence flows* et un *sequence flow* part du *gateway* 2, par conséquent le nombre total de *sequence flows* arrivant = 3 et le nombre total de *sequence flows* sortant = 3. Le *gateway* 2 termine bien la sous-séquence démarrée au niveau du *gateway* 5.

Dans le cas de la branche commençant par ST. Equ.Pom., le *gateway* suivant rencontré est le *gateway* 4. Ce *gateway* a un *sequence flow* arrivant et deux sortants. Par conséquent, le nombre total de *sequence flows* arrivant = 2 et le nombre total de *sequence flow* sortant = 4. Ensuite, quelle que soit la branche prise, suite au *gateway* 4, le *gateway* suivant est le *gateway* 3. Le *gateway* 3 a deux *sequence flows* entrant et un sortant. Par conséquent, le nombre total de *sequence flows* arrivant = 4 et le nombre total de *sequence flows* sortant = 5. Le *gateway* suivant est le *gateway* 2. Par conséquent, le nombre total de *sequence flows* arrivant = 6 et le nombre total de *sequence flows* sortant = 6. C'est donc bien le *gateway* 2 qui arrête les sous-séquences.

**Tableau III- 3 : illustration de la condition de recherche de l'élément final d'une sous-séquence**

Gateway	Gat 6	Gat 5	Gat 4	Gat 3	Gat 2	Gat 1	Gat 7
Gat 8	Arrivant : 2 Sortant : 4	Arrivant : 3 Sortant : 6	Arrivant : 4 Sortant : 8	Arrivant : 6 Sortant : 9	Arrivant : 8 Sortant : 10	Arrivant : 10 Sortant : 11	Arrivant : 12 Sortant : 12
Gat 6		Arrivant : 2 Sortant : 4	Arrivant : 3 Sortant : 6	Arrivant : 5 Sortant : 7	Arrivant : 7 Sortant : 8	Arrivant : 9 Sortant : 9	Arrivant : 11 Sortant : 10
Gat 5			Arrivant : 2 Sortant : 4	Arrivant : 4 Sortant : 5	Arrivant : 6 Sortant : 6	Arrivant : 8 Sortant : 7	Arrivant : 10 Sortant : 8
Gat 4				Arrivant : 3 Sortant : 3	Arrivant : 5 Sortant : 4	Arrivant : 7 Sortant : 6	Arrivant : 9 Sortant : 7

Une fois déterminé l'élément de fin des sous-séquences, il est possible de générer de façon récursive l'ensemble des éléments de la sous-séquence. Pour cela, nous avons utilisé l'algorithme III-1, représenté ci-dessous, en prenant comme paramètre l'élément initial de la sous-séquence (correspondant à l'un des éléments suivant le *gateway*), et l'élément final de la sous-séquence (résultat de la fonction `TrouverGatewayFin(x)`). Le déroulement de cet algorithme est simple, il consiste à transformer tous les éléments de la sous-séquence l'un après l'autre. Selon le type de l'élément de la sous-séquence à transformer, un élément du BPEL est généré. Cette génération est représentée par les différentes fonctions de génération d'éléments (`GenerateSequence`, `GenerateInvoke`, etc.). Ces fonctions sont détaillées en annexe F. En outre, lorsque l'élément de la sous-séquence à transformer en BPEL est un *gateway*, l'algorithme est exécuté de façon récursive.

Nom : `_ DeterminerSousSéquence`

Objectif : créer les sous-séquences du processus au niveau du BPEL

Paramètres :

`ElementDébut` : élément initial de la sous-séquence,

`ElementFin` : élément terminant la sous-séquence, généralement un *gateway*,

Variables :

`ElementEnCours` : élément de la sous-séquence à générer



ListeElementsInitiaux : liste d'éléments suivant un gateway, ListeElementsInitiaux[i] correspond au ième élément de la liste et NbListeElementsInitiaux correspond au nombre d'éléments de la liste.

Corps de l'algorithme :

Début :

//ajout des séquences du processus

ElementEnCours ← ElementDébut

**GenerateSequence**(ElementEnCours)

Tant que (ElementEnCours ≠ ElementFin)

Si (TrouverTypeElement(ElementEnCours) = « invoke » )

**GenerateInvoke**(ElementEnCours)

Sinon si (TrouverTypeElement(ElementEnCours) = « receive »)

**GenerateReceive**(ElementEnCours)

Sinon si (TrouverTypeElement(ElementEnCours) = « reply »)

**GenerateReply**(ElementEnCours)

Sinon si (TrouverTypeElement(ElementEnCours) = « flow »)

**GenerateFlow**(ElementEnCours)

ListeElementsInitiaux ← TrouverActivitésSuivantes(ElementEnCours)

Pour i = 0 à NbListeElementsInitiaux faire

DeterminerSousSéquence(ListeElementsInitiaux[i],  
TrouverGatewayFin(ElementEnCours))

FinPour

Sinon si (TrouverTypeElement(ElementEnCours) = « while »)

**GenerateWhile**(ElementEnCours)

ListeElementsInitiaux ← TrouverActivitésSuivantes(ElementEnCours)

Pour i = 0 à NbListeElementsInitiaux faire

DeterminerSousSéquence(ListeElementsInitiaux[i],  
TrouverGatewayFin(ElementEnCours))

FinPour

Sinon si (TrouverTypeElement(ElementEnCours) = « switch »)

**GenerateSwitch**(ElementEnCours)

ListeElementsInitiaux ← TrouverActivitésSuivantes(ElementEnCours)

Pour i = 0 à NbListeElementsInitiaux faire

DeterminerSousSéquence(ListeElementsInitiaux[i],  
TrouverGatewayFin(ElementEnCours))

FinPour

Fin si

ElementEnCours ← TrouverActivitéSuivante (ElementEnCours)

Fin tant que

Fin

**Algorithme III- 1 : création des sous-séquences**

La figure III-11 représente le résultat de cette fonction pour une sous-partie du processus de la partie IV-10, qui commence par le *gateway* Gat5. Dans cet exemple, nous voyons que lors de la génération de la séquence ST\_Pré.Evac., deux sous-séquences sont présentées suite au *gateway* Gat4. L'algorithme est donc appelé de façon récursive, afin de générer les sous-séquences suite au *gateway* 4. Le résultat de cet appel, correspondant à l'exemple illustré par la figure III-10, se retrouve au sein du BPEL de la partie droite de la figure III-11.

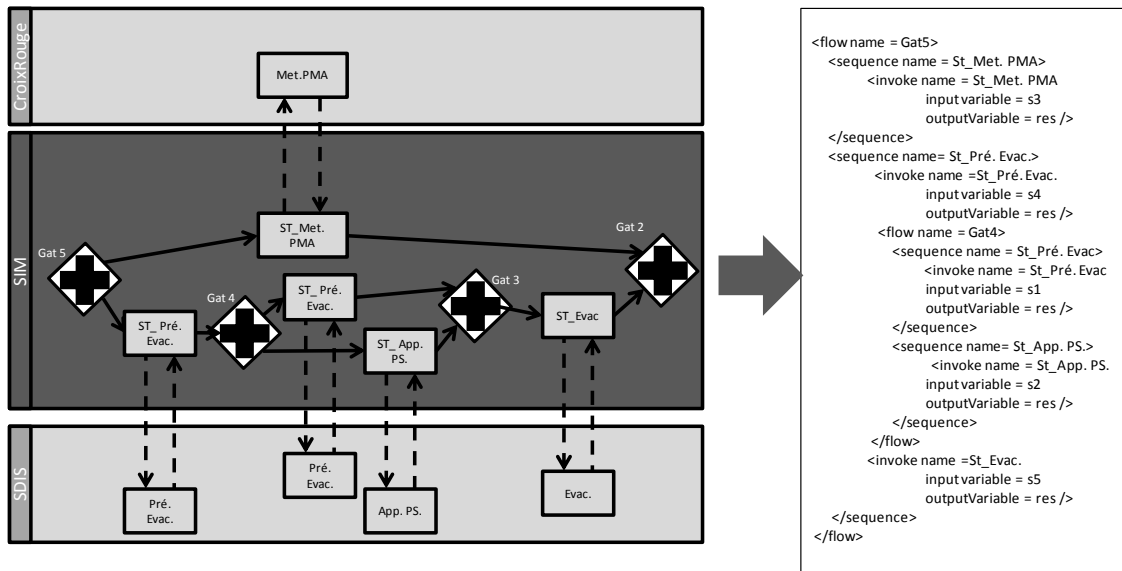


Figure III- 11 : exemple de transformation de sous-séquences en BPEL

## IV. Conclusion

Ce chapitre explique la dernière étape de la démarche de conception du système d'information de médiation. Cette étape consiste à configurer l'ESB PEtALS à partir du modèle d'architecture technique déduit de la caractérisation de la collaboration et des services concrets mis à disposition pour cette collaboration. De façon plus précise, La configuration de l'ESB PEtALS passe par la création de SUs et SAs regroupant les informations nécessaires à l'invocation des services avec les composants internes à l'ESB en charge de ces invocations. Au sein de ces SUs et SAs, deux nous intéressent particulièrement : le SU-BPEL et le SA-BPEL. En effet, ils sont en charge de sauvegarder l'ordre d'invocation des services des partenaires en suivant l'ordre défini par le processus collaboratif.

Cet ordre est retranscrit au sein d'un fichier BPEL. Le passage du processus collaboratif au fichier BPEL n'étant pas directement possible, notamment à cause de la gestion de l'information au cours du déroulement du processus, sa génération à partir du processus collaboratif a été réalisée grâce à deux transformations de modèles, dont une réalisée dans le cadre de cette thèse.

En plus de cette transformation, un service de médiation ainsi que des règles de gestion de l'information au sein du BPEL ont été mises en place afin de permettre cette gestion de l'information. Toutefois, la gestion de l'information que nous proposons est bien évidemment insuffisante en elle-même, puisqu'elle permet uniquement de faire des affectations de valeurs de type « un à un » (*i.e.* de copier-coller des valeurs sans pouvoir les transformer). Cependant, au sein du processus, l'information est transférée au bon format, au bon moment. L'ajout d'une couche de gestion dynamique de la sémantique au sein des services de médiation, à la place de l'utilisation du fichier de correspondance statique, permettrait une gestion de l'information de façon plus satisfaisante. Cette piste constitue une perspective à ce sujet de thèse.

Enfin, l'apparition du langage BPMN 2.0, qui a vocation à devenir un langage permettant de supporter l'exécution au même sens que le langage BPEL, peut remettre en cause les travaux présentés dans ce chapitre. Cependant, les éditeurs ayant investi dans un moteur d'orchestration BPEL, referont-ils un nouveau moteur d'orchestration pour le BPMN2.0 ou essaieront-ils d'adapter leur moteur afin de supporter le langage BPMN2.0 ? Dans le deuxième cas, les travaux de ce chapitre peuvent constituer une base de travail.



# Chapitre IV :

## L'agilité du médiateur

*Une exigence en situation de crise*

<b>Introduction du chapitre.....</b>	<b>98</b>
<b>I. Présentation de la notion d'agilité .....</b>	<b>99</b>
I.1 Définition de l'agilité .....	99
I.2 Apporter de l'agilité aux processus.....	99
I.3 Apport d'agilité au SIM.....	100
<b>II. Détection de variations .....</b>	<b>101</b>
II.1 Présentation de la fiche de suivi.....	102
II.2 Vers un suivi en temps réel de la réponse à la crise.....	104
II.3 Traitement de l'information issue de la fiche de suivi.....	106
<b>III. Réagir suite à une évolution.....</b>	<b>107</b>
III.1 Agilité du processus de conception.....	107
III.2 Réalisation d'une chaîne de conception flexible .....	110
<b>IV. Conclusion.....</b>	<b>114</b>

## Introduction du chapitre

Dans le chapitre I, l'agilité du SIM est apparue comme une exigence afin de pouvoir répondre aux possibles évolutions d'une situation de crise. En s'inspirant des propositions figurant dans (Pingaud, 2009), deux types d'évolutions de telles situations collaboratives peuvent être distingués :

- Evolutions de la situation elle-même : les caractéristiques perçues de la crise, et en particulier les problèmes à traiter, ne sont plus les mêmes et nécessitent une nouvelle réponse à la crise.
- Evolutions de la réponse à apporter : le pilotage de la résolution de la situation de crise peut être amené à évoluer suite à (i) une évolution de la composition de la cellule de crise (arrivée, départ, etc.), (ii) un dysfonctionnement lors de l'exécution d'un service (nécessitant impérativement une interruption afin de tenir compte du dysfonctionnement sans poursuivre aveuglément le déroulement du processus) ou (iii) du fait d'une possible définition initiale partielle du processus de réponse.

Ces évolutions ont bien évidemment une répercussion sur le système de traitement et par conséquent sur le système de systèmes d'information support de la réponse à la crise. Cependant, selon le type d'évolution le système de systèmes d'information ne doit pas évoluer de la même manière. En effet, suite à une évolution l'adéquation de la réponse à la crise vis-à-vis de la situation est remise en cause à différents niveaux. Dans le cas d'une évolution de la situation, telle qu'une sur-crise, ce seront les informations portant sur la description de la situation qui seront remises en cause alors que cette remise en cause portera, soit sur la définition, soit sur l'exécution de la réponse à la crise dans le cas d'une évolution de la réponse à apporter.

Cette observation, point d'entrée de ce chapitre, impacte inévitablement la démarche de conception du système de systèmes d'information support de la réponse à la crise. Cette remise en cause intervient à différents niveaux de la chaîne de définition du système de médiation (en tant que pierre angulaire du SdS). En effet, suite à une évolution de la situation, c'est le modèle de crise qui devra être modifié afin de correspondre à la nouvelle perception de la situation. Concernant l'évolution de la réponse, la construction du processus collaboratif peut être remise en cause au niveau (i) de l'ensemble des services disponibles (*i.e.* l'ensemble des modèles de ces services), (ii) de la sélection des services choisis pour résoudre chaque problème (*i.e.* les risques ou les faits identifiés), (iii) de la définition nativement incomplète ou parcellaire du processus. Enfin, dans certain cas de dysfonctionnement, il est envisageable de conserver le même modèle de processus collaboratif en demandant au participant de recommencer certaines activités.

Par conséquent, en fonction de l'évolution, il est nécessaire, d'une part, de pouvoir recommencer le mécanisme de déduction de la réponse à la crise au niveau le plus pertinent, d'autre part, de pouvoir agir sur la réponse en cours, que ce soit pour l'arrêter ou redemander l'exécution de certaines tâches.

Ce chapitre est construit de la manière suivante : Nous essaierons tout d'abord d'expliquer notre position vis-à-vis de la notion d'agilité puis sa prise en compte au sein de nos travaux. Une capacité d'agilité n'étant requise que lorsqu'une modification doit être apportée, une solution de détection de ces changements (au sens de la prise en compte de tels événements par le SIM et sa chaîne de conception), par l'intermédiaire de l'adjonction d'un tableau de bord a été réalisée. Cette solution est présentée dans la deuxième partie de ce chapitre. Suite à cela, nous nous attarderons sur la solution que nous avons mise en place. Cette solution, basée sur la décomposition de la chaîne de conception en services informatisés (eux-mêmes intégrés à un ESB), adresse directement les questions de rebouclage de la chaîne de conception à différents niveaux. Pour autant, cette approche de l'agilité ne permet cependant pas la conception partielle du processus ni la possibilité de l'interrompre, de le relancer, de le recommencer, voire de sauter un

service. Ces deux derniers points ont été pris en charge par d'autres partenaires du projet ISyCri avec l'utilisation du ContractNet (protocole contractnet proposé par l'IRIT (FAURE *et al.*, 2010)) et du moteur d'orchestration Maestro (proposé par EBM-WebSourcing et rebaptisé depuis SE-BPEL).

## I. Présentation de la notion d'agilité

### I.1 Définition de l'agilité

La notion d'agilité est apparue avec l'évolution du tissu industriel, évolution caractérisée comme « la transformation d'une structure cristallisée en un environnement fluide » (Bénaben *et al.*, 2007) ou encore comme « la transformation d'une construction statique de lego® vers une un organisme vivant » (Luzeaux *et al.*, 2008). Ces métaphores reflètent l'évolution des collaborations inter-entreprises. Elles ne se fondent plus sur une optique à long terme avec des collaborations stables et difficiles à établir. Elles se fondent aujourd'hui sur des collaborations ponctuelles, opportunistes, rapides à mettre en place et à dissoudre.

De nombreuses définitions du concept d'agilité s'accordent sur le fait qu'une entreprise peut être qualifiée d'agile si elle peut *s'adapter à l'évolution* du contexte dans lequel elle intervient. Certains auteurs comme (Badot, 1998) parlent plutôt de *reconfiguration du système* pour satisfaire un besoin d'adaptation. D'autres auteurs, comme (Kidd, 1994), (Lindberg, 1990) et (Sharifi *et al.*, 1999) parlent plutôt d'un besoin de *flexibilité, réactivité* ou encore d'*adaptabilité*. Dans le domaine de la logistique, où l'agilité est définie comme « l'habilité à répondre à des changements à courts termes » (Sheffi, 2004), elle est différenciée de l'adaptation en fonction du temps de réponse à un changement (Mc Cullen *et al.*, 2006).

Suite à ces considérations, nous conserverons comme vision de l'agilité sa double composante : *réactivité* et *flexibilité* (la première composante ayant trait à la vitesse d'adaptation et la seconde à l'amplitude de cette adaptation). Si nous reportons ce double objectif au niveau des systèmes d'information et plus précisément du système de systèmes d'information, nous nous rendons compte que l'agilité ne peut pas porter sur une modification de la partie « système informatique » (au sens de (Morley *et al.*, 2005)) puisqu'une modification de l'implémentation des services techniques ou des fonctionnalités de l'environnement d'exécution poserait d'évidents problèmes de délai. Ainsi, que ce soit par rapport à la *réactivité* ou à la *flexibilité*, seule la partie « système de traitement de l'information » (toujours au sens de (Morley *et al.*, 2005)) peut être modifiée. Ce « système de traitement de l'information » reposant en grande partie sur les processus, nous nous proposons de concentrer nos efforts sur les travaux, approches et solutions concernant l'agilité des processus.

### I.2 Apporter de l'agilité aux processus

Cette partie de l'étude s'inspire principalement de travaux sur la gestion de l'incertitude dans les systèmes de workflow. Nous citerons en particulier le workflow inter-organisationnel lâche (Andonoff *et al.*, 2007) ou encore l'approche ADEPT<sub>flex</sub> (Reichert *et al.*, 1998) ou des travaux dans une veine apparentée (van der Aalst, 1998). (Schonenberg *et al.*, 2008) ont récemment proposé une synthèse de ces travaux avec une taxonomie des approches de la flexibilité dans les systèmes de workflow. Cette taxonomie, schématisée sur la figure II-6, identifie quatre grandes classes d'approches :

- *Flexibilité par conception (Flexibility by design)* : Il s'agit d'apporter de la flexibilité au niveau de la conception du processus en incluant des chemins d'exécution alternatifs. La sélection du chemin le plus approprié sera effectuée au moment de l'exécution. Par exemple, l'approche

proposée par (Rueppel *et al.*, 2007) consiste à définir un certain nombre de processus de réponse possibles concernant les réponses d'une crise de type inondation en Allemagne.

- *Flexibilité par déviation (Flexibility by deviation)* : il s'agit d'apporter de la flexibilité au moment de l'exécution en permettant de modifier l'ordre d'exécution des activités sans les changer (*i.e.* permettre d'annuler, recommencer ou sauter une tâche). Par exemple, Le système FLOWer (van der Aalst *et al.*, 2005) permet d'utiliser ce type de flexibilité.
- *Flexibilité par définition partielle ou sous-spécification (Flexibility by underspecification)* : il s'agit de définir partiellement le processus au niveau de la conception et de le compléter au moment de son exécution. Cette flexibilité se base sur des tâches ou sous-processus abstraits ne pouvant être définis précisément, mais pouvant être identifiés (*i.e.* il existe un certain nombre de tâches ou sous-processus disponibles mais ce n'est que lors de l'exécution que le choix sera fait). Ce type de flexibilité est mis en œuvre dans le système YAWL (van der Aalst *et al.*, 2004).
- *Flexibilité par changement (Flexibility by change)* : il s'agit de modifier la définition du processus en cours d'exécution, par le biais d'insertion ou de suppression de tâches. Ce type de flexibilité est mis en œuvre dans le système ADEPT<sub>flex</sub> (Reichert et Dadam, 1998).

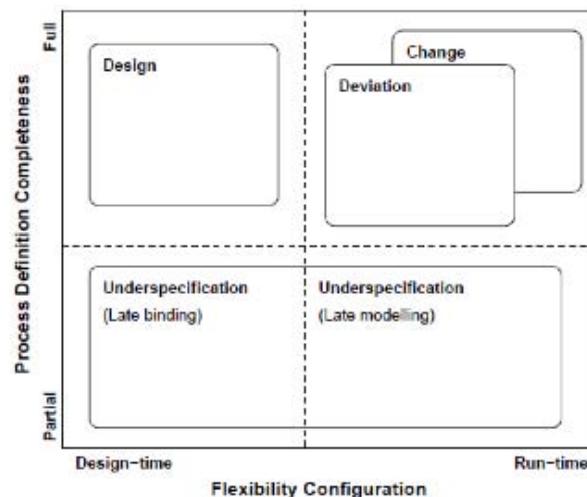


Figure IV- 1 : différents types de flexibilité selon (Schonenberg *et al.*, 2008).

### I.3 Apport d'agilité au SIM

Comme évoqué en introduction de cette partie, nous avons décidé de décliner, dans le cadre de nos travaux, la notion d'agilité en retenant deux dimensions largement partagées :

- **la réactivité** : se concentre sur le rapport au temps. Nous allons faire l'hypothèse que ce temps est un délai court qui n'accepte des modifications qu'au seul niveau de la configuration du SIM. La réactivité peut être définie comme la capacité d'un SIM à être rapidement opérationnel suite à une demande de modification, conséquence d'une évolution.
- **la flexibilité** : se concentre sur le rapport à l'amplitude du changement que le SIM peut mettre en œuvre. La flexibilité peut être définie comme la capacité du SIM à engager des évolutions pertinentes quels que soient les écarts entre son comportement initial (*i.e.* avant l'événement déclencheur de la nécessité d'évoluer) et le comportement escompté (*i.e.* après l'adaptation du système).



La nécessité de flexibilité du SIM étant dûe aux évolutions possibles de la situation, il est intéressant de mettre en relation les classes d'approche de flexibilité avec les types d'évolution :

- Une *variation du comportement dans les échanges au sein du réseau* peut impliquer l'arrêt, la relance d'une activité ou encore la modification du cours des événements lors de l'exécution, cette nature de variation amène à développer une *flexibilité de type déviation et/ou par changement*.
- Une *variation dans la structure des systèmes contributeurs ou du réseau* de même qu'une *variation dans la mission d'un réseau* amènent à modifier le SIM et à intervenir sur la *flexibilité de conception*. Cependant, dans le cadre de nos travaux, nous ne concevons pas la flexibilité par conception comme la définition de l'ensemble des chemins possibles à priori. Nous avons décidé de définir un chemin que nous proposons de modifier en cours d'exécution, suite à une variation.
- Concernant la *flexibilité par spécification partielle*, elle peut être utilisée dans le cas d'une *variation de la structure des systèmes contributeurs ou du réseau*, ainsi que lors d'une *variation du comportement dans les échanges au sein du réseau*.

La figure suivante illustre ces correspondances entre approches de flexibilités et types de variations.

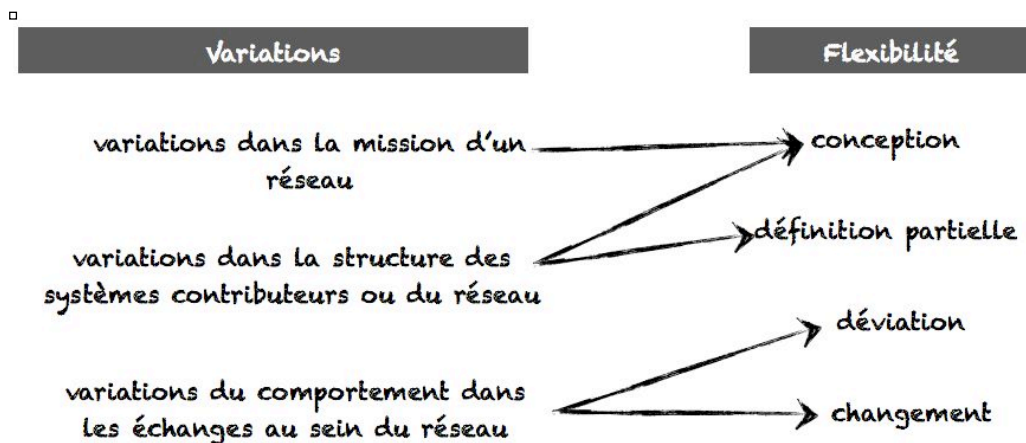


Figure IV- 2 : relations entre variations et approches de flexibilité (Truptil et al., 2010).

## II. Détection de variations

Nous avons souligné le fait qu'un enjeu critique concerne la détection des événements significatifs d'une évolution conséquente de la situation (dans l'optique de déclencher justement l'application des principes d'agilités évoqués). Ainsi, avant de présenter l'implémentation des outils permettant de prendre en charge les exigences de réactivité et de flexibilité inhérentes à notre vision de l'agilité, nous allons tout d'abord nous focaliser sur l'amélioration de la détection des changements significatifs, voire sur leur anticipation. Le tableau de bord présenté dans cette partie, que nous appellerons par la suite *fiche de suivi*, vise à prendre en charge cette fonctionnalité.

## II.1 Présentation de la fiche de suivi

Afin de permettre cette détection de changements ou d'évolutions significatifs, la fiche de suivi requiert une collecte d'informations sur plusieurs plans : l'évolution des caractéristiques de la crise, l'évolution du déroulement du processus collaboratif (et des activités atomiques réalisés par les services invoqués), ainsi que l'évolution du groupe de partenaires eux-mêmes. Cette opération s'apparente pour partie à une supervision de l'exécution des processus (« Business Activity Monitoring »). Dans la réalité opérationnelle de traitement d'une crise (par exemple une crise civile), l'ensemble de ces informations existe aujourd'hui sous la forme d'une trace écrite appelée *main courante* (éventuellement capitalisée par l'intermédiaire d'un outils national de stockage). Une *main courante* sert à contrôler le déroulement de la réponse à la crise. Elle contient l'ensemble des événements, qu'il s'agisse de compte-rendu de la bonne exécution d'un service ou bien de description d'événements non programmés dont il faut jauger les causes et les effets.

Pour la cellule de gestion de crise, la *main courante* remplit les fonctions suivantes : (i) vérifier le bon déroulement des services, (ii) fournir un compte-rendu de fin des services, (iii) noter les retards ou problèmes lors de l'exécution de services, tout en indiquant les raisons, (iv) noter tout nouveau risque ou conséquence qui nécessite d'être traitée, (v) noter l'arrivée ou le départ de tout partenaire, (vi) conserver la trace temporelle de la gestion de crise et des événements rencontrés. L'ensemble de ces informations est ensuite utilisé par la cellule de gestion de crise afin de prendre les décisions, à savoir : (i) anticiper à court terme l'évolution de la crise, notamment les problèmes possibles lors de la mise en exécution de la réponse, (ii) anticiper à moyen et long termes, *i.e.* essayer de prévoir l'évolution de la crise ainsi que l'apparition de nouveaux risques ou conséquences non souhaités.

La fiche de suivi a été bâtie autour de ces informations vitales afin d'en permettre la synthèse. La figure IV-3 montre un exemple de fiche de suivi reprenant l'exemple détaillé au cours du chapitre II. La réponse à la crise ayant déjà débuté, la situation est la suivante : les gendarmes ont fini de mettre en place le périmètre de sécurité, la Croix Rouge est en train de mettre en place les postes médicaux avancés alors que les pompiers sont entrés dans le gymnase afin de préparer les étudiants à l'évacuation.

Comme l'illustre la figure IV-3, la fiche de suivi est composée de quatre tableaux. Les deux premiers portent sur les risques et les conséquences de la crise. Ils synthétisent donc des informations liées à la caractérisation de la crise (suivi de l'état des problèmes). Les deux autres sont relatifs aux acteurs et aux services. Ils décrivent par conséquent le système de traitement en cours d'activité.

Les deux premiers tableaux, liés à la caractérisation de la crise, contiennent d'une part des informations qui peuvent être considérées comme statiques et d'autre part, des informations à vocation plus dynamiques dépendant de l'avancement de la réponse. La partie statique des informations présentées dans cette fiche de suivi relèvent tout d'abord de la liste des risques (tableau de gauche), et des conséquences (tableau de droite) diagnostiqués vis-à-vis de la situation de crise courante, mais également (et surtout) de détails sur ces risques et conséquences tels que l'ordre de priorité de traitement décidé par la cellule de crise (*cf.* chapitre II) ou encore le mode d'affectation des services curatif (pour une conséquence) ou préventifs (pour un risque) sélectionnés par la cellule de crise (« *okcan* », « *okcould* », etc.). La partie dynamique des informations relève quant à elle de l'évolution du statut d'un risque ou d'une conséquence en fonction de l'avancement de la réponse. Les statuts possibles sont les suivants :

- Présent : correspond à un risque (ou une conséquence) pour lequel (ou laquelle) aucun service sélectionné pour le (ou la) résoudre n'a débuté.
- En cours de Traitement : correspond à un risque (ou une conséquence) pour lequel (ou laquelle) au moins un service sélectionné pour le (ou la) résoudre n'est pas terminé.

- Résolu : correspond à un risque (ou une conséquence) pour lequel (ou laquelle) l'ensemble des services sélectionnés pour le (ou la) résoudre est terminé.
- Pas pris en compte : correspond à un risque (ou une conséquence) pour lequel (ou laquelle) aucun service n'a été sélectionné pour le (ou la) résoudre.

Cette définition des différents statuts dépend clairement de l'avancement du processus de réponse. Cependant, rien ne certifie que la fin de l'exécution d'un ensemble de services ne résolve effectivement le problème (risque ou conséquence) pour lequel ils ont initialement été sélectionnés. En effet, suite à un dysfonctionnement, détecté ou non, un risque ou une conséquence peut encore être concrètement présent alors que l'ensemble des services utilisés pour le (ou la) résoudre ont été exécutés. Afin de pouvoir détecter ce cas de figure, nous distinguons l'état *réel* et l'état *supposé* (désigné par le terme *déduit* dans le tableau de la figure IV-3) d'un risque ou d'une conséquence : l'état supposé est déduit automatiquement en fonction de l'état des services, *i.e.* lorsque les services utilisés pour résoudre le problème A sont terminés, l'état du problème A passe à *résolu* alors que l'état réel sera directement renseigné par un membre de la cellule de gestion de crise en fonction des informations qui lui proviendront du terrain.

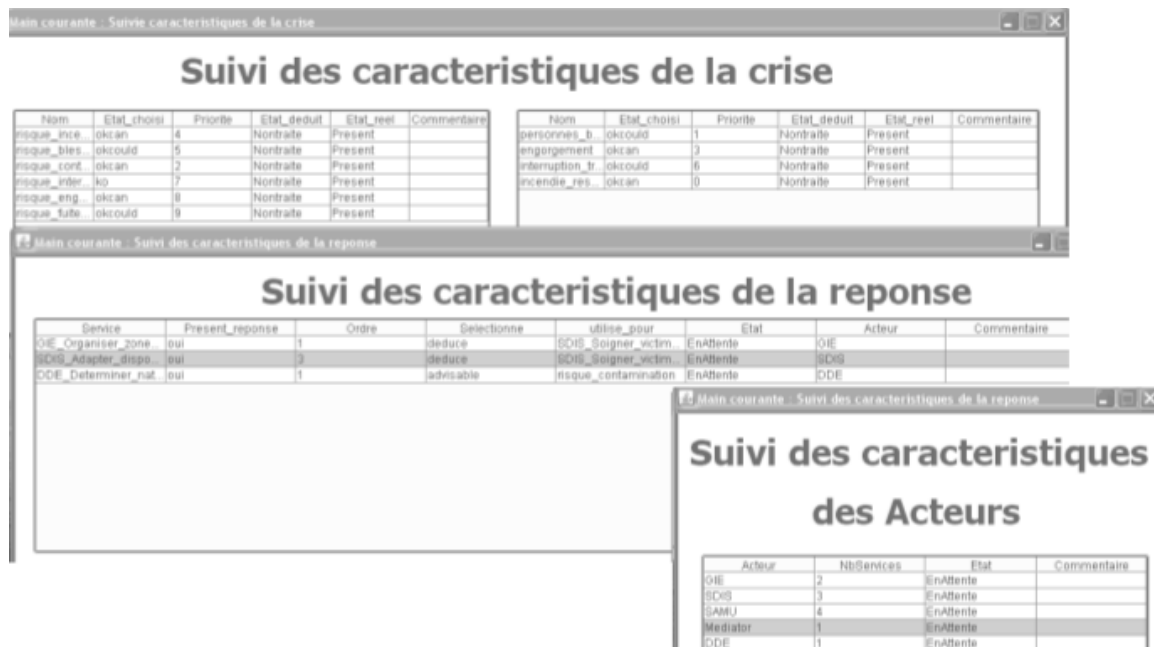


Figure IV- 3 : exemple d'utilisation de la fiche de suivi.

Concernant les deux autres tableaux (Acteurs et Services), ils sont destinés à fournir des informations sur les caractéristiques liées à la réponse à la crise. Le tableau concernant la structuration de la cellule de crise (*Suivi des caractéristiques des Acteurs*, partie inférieure droite de la figure IV-3), permet de détecter l'évolution structurelle de la composition du groupe d'acteurs. Un acteur peut être *présent*, *en attente* ou *indisponible* selon son statut au sein de la cellule. Le tableau portant sur les services (*Suivi des caractéristiques de la réponse*, partie centrale de la figure IV-3), recense plusieurs informations sur les services mobilisés : (i) Les informations statiques portant sur l'acteur en charge de leur réalisation ou encore la raison de leur présence dans le cadre de la réponse (a-t-il été choisi pour résoudre un risque ou une conséquence ou est-il nécessaire à l'exécution d'un autre service ? cf. chapitre II), (ii) les informations dynamiques quant à elles reposent sur le statut des services à l'instant considéré. En fonction de l'avancement de la réponse, un service peut être *en attente*, *en cours d'exécution*, *fini* ou *en dysfonctionnement* si les acteurs ont rencontré un problème lors de la réalisation du service.

Trois classes de diagnostic peuvent donc être identifiées à partir d'une surveillance régulière de ces tableaux, constitutifs de la fiche de suivi :

- si une nouvelle information entre dans un des deux premiers tableaux (*Suivi des caractéristiques de la crise*, parties *risques* et *conséquences*), la cellule de gestion de crise se trouve face à une évolution de la situation, et doit par conséquent modifier le modèle de crise, puis relancer la démarche de conception du système d'information collaboratif.
- si une différence est notée entre les états réel et supposé d'un risque ou d'une conséquence de la crise (toujours dans le tableau *Suivi des caractéristiques de la crise*, parties *risques* et *conséquences*), elle est significative d'une évolution de type dysfonctionnement. Plusieurs solutions sont alors envisageables pour la cellule de gestion de crise, elle peut redemander l'exécution d'un service ou rechercher de nouvelles compétences (en interne ou en externe) à même de prendre en charge le traitement de ce problème,
- Le dernier tableau en bas à droite permet de détecter des évolutions liées à la composition de la cellule de gestion de crise ou au champ de responsabilité des acteurs. Dans le cas d'une modification, le processus collaboratif doit être modifié.

## II.2 Vers un suivi en temps réel de la réponse à la crise

Lors de la description de la fiche de suivi, il a été mis en évidence que pour permettre la détection des dysfonctionnements, il est évidemment nécessaire de suivre en temps réel l'avancement du processus. Cependant dans le cadre de nos travaux, il n'est pas envisageable d'ajouter aux services des acteurs la possibilité de mettre à jour directement la fiche de suivi puisque c'est contradictoire avec la notion de système de systèmes. Il faudrait en effet pour cela que la fonction de mise à jour soit prise en charge par les services des acteurs, et ce, au sein de leur partie privée, seule à même de prendre en charge une nouvelle fonctionnalité, or, seule la partie publique des services du système d'information d'un contributeur est accessible (*cf.* chapitre I). En outre, afin que chaque service puisse prendre part à cette mise à jour, il faudrait également pouvoir ajouter cette fonctionnalité à l'ensemble exhaustif des services mobilisables. Pour d'évidentes raisons (on ne peut pas accéder à l'ensemble des services de tous les acteurs mobilisables), il n'est pas envisageable de disposer, au préalable, de la liste des services à enrichir de cette fonctionnalité de mise à jour de la fiche de suivi.

L'idée est donc de mettre au point un service interne au SIM permettant de mettre à jour la fiche de suivi en fonction de l'avancement du processus. Pour cela, il est nécessaire de connaître à chaque instant le(s) service(s) en cours de traitement afin de pouvoir suivre le déroulement du processus.

Suite à ce constat, nous nous sommes rendu compte que le service de médiation, présenté au chapitre II, répondait globalement à ces attentes : il est automatiquement invoqué avant l'exécution d'un service. De plus, le message d'entrée du service de médiation commence par le nom du *sequence flow* précédent l'activité. Il est donc possible à travers le service de médiation de retrouver l'activité qui débutera à sa suite (et donc d'identifier le service en cours d'exécution). Chaque invocation du service de médiation permet donc de connaître l'avancement du processus et ainsi de fournir les informations nécessaires au monitoring.

Dans le but de suivre en temps réel le déroulement de la réponse, il est nécessaire de savoir quand une activité se termine. Une première idée se base sur la fréquence d'invocation du service de médiation : dans le but de créer le message d'entrée d'un service d'acteur, le service de médiation est invoqué avant ce service et fait suite à un service précédent. Ainsi, dès l'invocation d'un service de médiation, le statut du

service d'acteur suivant ce service de médiation passe à *en cours* et celui du service d'acteur précédent à *fini*. Malheureusement, ce principe ne convient pas lorsque des services sont exécutés en parallèle. Afin d'illustrer notre propos, reprenons une sous-partie du processus collaboratif déduit au chapitre II et représenté par la figure IV-4. Dans le cas (a), lorsque le service de médiation, représenté par ST\_Pré. Evac, est invoqué il est possible de mettre à jour en temps réel le statut du service d'acteur « préparer pompier » à *fini* et celui de « préparer pour évacuation » à *en cours*. Dans le cas (b), pour que le service de médiation ST\_Evac soit invoqué, (*i.e.* pour lancer le service « évacuation ») il faut que les services « préparer pour évacuation » ainsi que « apporter les premiers soins » soient terminés. Par conséquent, lorsque le service « évacuation » commence, son statut passe à *en cours* et le statut des deux services (« préparer pour évacuation » et « appliquer les premiers soins ») passent simultanément leur statut à *fini*. Cependant rien ne certifie que ces deux services se soient terminés en même temps. Cette approche peut donc engendrer un décalage entre la fiche de suivi et le déroulement en temps réel de la réponse à la crise.

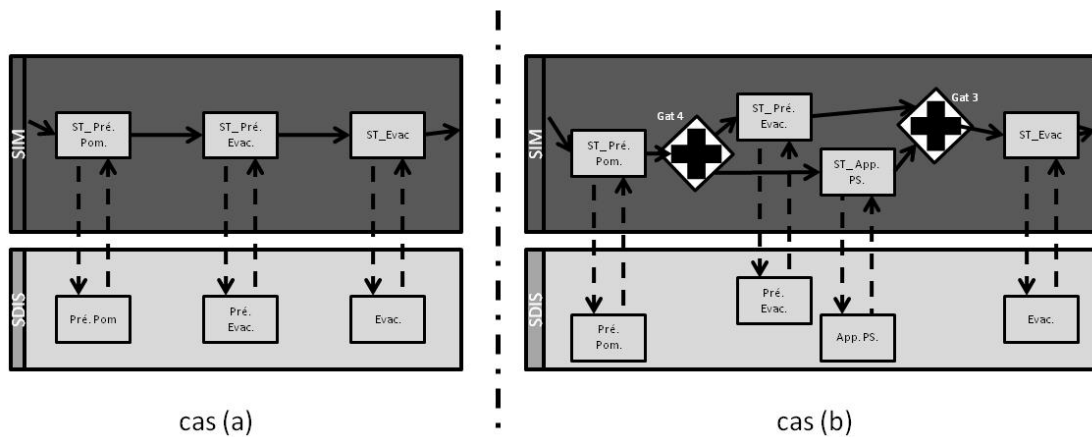


Figure IV- 4 : illustration de la limite d'utilisation du service de médiation seul.

Afin d'éviter ce décalage entre la mise à jour de la fiche de suivi et l'avancement réel du processus de réponse, il est nécessaire d'ajouter au processus collaboratif une autre invocation du service de médiation suite à un service d'acteur. Ce service de médiation n'a pas de valeur ajoutée concernant la gestion de l'information, il n'est utilisé que pour une seule raison : connaître le moment où un service se termine. Grâce à ce service, dans le cas (b) de la figure IV-4, le service « préparer pour évacuation » verra son statut passer à *fini* dès la fin du service, et ce, que le service « appliquer les premiers soins » soit *fini* ou *en cours*. Cependant, cette modification engendre un processus collaboratif différent. La figure IV-5 correspond à la nouvelle version du processus de réponse, précédemment représenté par le cas (b) de la figure IV-4.

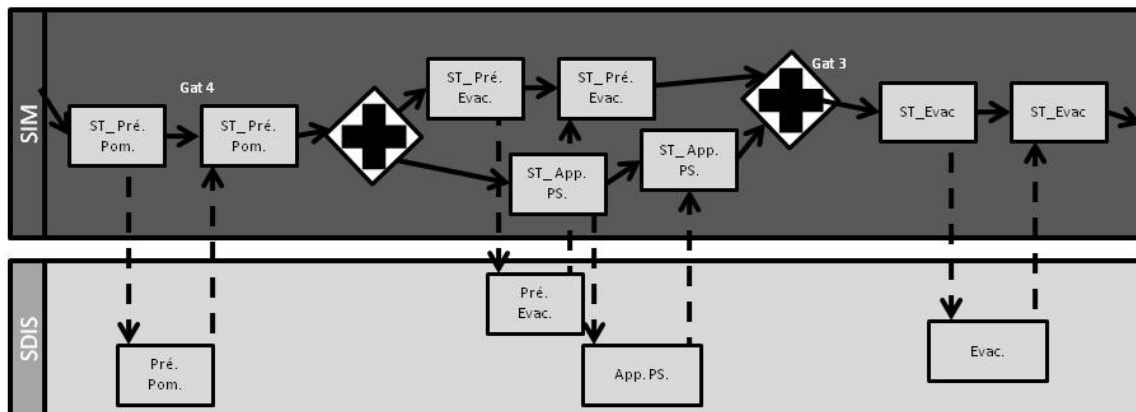


Figure IV- 5 : Transformation du processus collaboratif dans le but d'obtenir une mise à jour en temps réel du statut des services.

Ainsi, la partie automatisée du suivi, qu'on peut assimiler à une ébauche de monitoring, doit structurellement être assurée par les services du médiateur. En outre, afin d'améliorer la granularité et la précision des informations, une solution intéressante pourrait consister à séparer ces services de médiation (jusqu'alors théoriquement dédiés à faire l'interface avec un service technique) en services de médiation plus fins, en charge uniquement de l'émission ou de la réception d'un message. Cette granularité des services du SIM peut leur permettre de jouer également le rôle de capteurs destinés à renseigner sur l'avancement du projet.

## II.3 Traitement de l'information issue de la fiche de suivi

L'intérêt de l'utilisation de la feuille de suivi est de détecter l'apparition d'évolutions lors de la réponse à la crise. Bien que les évolutions puissent porter sur la situation elle-même ou sur les caractéristiques de la réponse apportée, la fiche de suivi permet principalement de détecter l'ensemble des évolutions liées à la réponse à la crise.

Le tableau IV-1 fait le lien entre les évolutions et les éléments de la fiche de suivi utilisés permettant de les détecter.

**Tableau IV- 1 : Eléments de la fiche de suivi permettant de déduire un type d'évolution.**

Evolutions	Eléments de la fiche de suivi
Evolutions de la situation	Tableaux de risques et de conséquences
Variation de la composition de la cellule de gestion de crise	Tableaux d'acteur
Variation liée à un dysfonctionnement	Tableaux de risques, conséquences et services (le tableau de service permet de déduire l'état supposé alors que les tableaux de problèmes renseignent l'état réel)
Variation liée à une définition partielle du processus de réponse	Pas pris en compte

Concernant les évolutions liées à des dysfonctionnements, leurs détections sont permises, d'une part, directement par un acteur lorsqu'une exécution de service est problématique ou présente un résultat indésirable, et d'autre part, par l'apparition de décalage entre l'état *déduit* et l'état *réel* d'un risque ou d'une conséquence. En effet, Si le décalage entre les deux états est le suivant : l'état *déduit* est à *résolu* et l'état *réel* est à *présent*, cela signifie que bien que le problème est toujours présent, et par conséquent à traiter. Pour autant, si le processus poursuit son exécution aucune action dédiée à ce problème ne sera mise en place au cours du reste de la réponse à la crise (puisque le statut théorique indique qu'il n'y a plus d'action en cours ou à venir sur ce sujet). Dans de telles circonstances, la réponse à la crise à besoin d'être modifiée. Malheureusement, si la fiche de suivi peut permettre de détecter ce type problème, elle n'apportera évidemment pas d'indication concernant les modifications à apporter à la réponse. Les leviers de modification relèvent de la possibilité de recommencer la démarche de conception du médiateur à différents niveaux : étape de choix fait par la cellule de gestion de crise, *i.e.* la définition des liens

sémantiques, la définition de l'ordre de priorité de traitement, la sélection des services pour chaque problème et enfin le regard critique porté sur le processus collaboratif déduit.

La cellule de gestion de crise est par conséquent en charge de choisir le levier à actionner et donc le niveau de modification en fonction de la situation. Ce point est primordial concernant nos travaux portant sur l'apport d'agilité. En effet il souligne de nouveau le besoin pour la cellule de gestion de crise de pouvoir reprendre la conception du SIM à n'importe quel niveau de modification.

### III. Réagir suite à une évolution

Nous avons vu précédemment que l'apparition d'évolutions peut remettre en cause la réponse à la crise en cours. Suite au diagnostic d'une évolution, dont la détection est aidée par la fiche de suivi, il est nécessaire de modifier le SIM. Dans cette partie, nous allons donc détailler les caractéristiques d'agilité que nous avons apporté au SIM et à sa conception dans le cadre du projet ISyCri. Concernant la capacité de *réactivité*, son acquisition n'est pas un obstacle majeur dans la mesure où la démarche de conception du système d'information de médiation est systématique et automatisée dès que le processus collaboratif est défini. En effet, puisque les systèmes d'information des systèmes contributeurs sont stables, le passage à l'acte consiste à actualiser l'adresse d'un ou plusieurs service(s) dès que la cellule de gestion de crise a validé le processus collaboratif actualisé. D'autre part, la conception de ce processus collaboratif est facilitée dès que la situation est caractérisée. En effet, lors de sa conception, seuls les choix faits par la cellule de gestion de crise ne sont pas automatisés. Cependant, la cellule de gestion de crise est aidée dans la prise de décisions, qui consiste à définir de liens sémantiques (« Near » et « SameAs ») puis l'ordre de priorité de traitement avant de choisir pour chaque problème les services à mettre en œuvre parmi l'ensemble hiérarchisé de ceux qui sont proposés. De plus, ces choix peuvent être réalisés rapidement tout en gardant une certaine sérénité dans la mesure où l'agilité fournit la possibilité de remettre en cause ces choix. On peut par conséquent en déduire un cercle vertueux : parce qu'on est agile, on peut aller plus vite et parce qu'on va vite on est réactif.

En revanche, la flexibilité induit un traitement beaucoup plus conséquent. Comme expliqué en introduction, nous nous sommes intéressés à mettre en œuvre une flexibilité par conception. Contrairement à la définition d'une flexibilité par conception proposée par (Schonenberg *et al.*, 2008), notre approche consiste à pouvoir reprendre la chaîne de conception du processus collaboratif. Pour cela nous allons commencer par comparer les différents scénarios de conception en fonction du type d'évolution.

#### III.1 Agilité du processus de conception

La démarche de conception proposée doit aboutir à un Système d'Information de Médiation reposant sur un unique processus collaboratif de réponse à la crise (correspondant donc à un scénario satisfaisant de résolution de la crise). Notre vision se base sur la supposition qu'il est difficile de connaître a priori l'ensemble des événements potentiellement susceptible de se produire, pouvant non seulement modifier la gravité de la crise mais aussi sa nature. L'inhérente instabilité et imprévisibilité des situations de crise en général, restreignent les approches, consistant à définir le processus de réponse ainsi que toutes ses alternatives possibles, à des domaines de crise bien particuliers identifiés et bien maîtrisés (qualitativement et quantitativement). Dans bien d'autres crises, ce degré d'anticipation n'est pas possible : qui pouvait anticiper en 2009 le basculement de la crise de la Grippe H1N1 en France d'une crise sanitaire à une crise sociale face aux mesures prises par les autorités (quantité de vaccins, de traitements, de masques) ? Ces considérations justifient, d'une part, la volonté affichée dans ces travaux de ne pas essayer de

«cartographeur» a priori, l'espace des situations de crise et l'espace de la gestion de crise, (ce qui nous semble utopique dans ces environnements à l'instabilité chronique) et d'autre part, l'incontournable nécessité de pouvoir remettre en cause la dynamique de réponse construite.

Afin de répondre à cet objectif de reconstruction du SIM en fonction de l'évolution, il est nécessaire de présenter les différentes alternatives, en fonction de l'évolution, au processus nominal de conception.

### III.1.1 Le processus de conception nominal

Qu'on se trouve au démarrage de la gestion d'une situation de crise (par exemple, quelques instant après un tremblement de terre) ou lors de l'une de ces évolutions majeures (par exemple, lorsqu'un incendie se déclare dans une usine de produit chimique affectée par le tremblement de terre), la démarche est initiée par la création (ou remise à jour profonde) du modèle de crise et se conclut par la configuration de l'ESB de médiation (en passant par l'ensemble des étapes décrites dans les différents chapitres de ce document).

Cette démarche est représentée par le processus (de conception du SIM) de la figure IV-6. On y retrouve une descente en abstraction selon une démarche d'ingénierie dirigée par les modèles, (métier, logique, technique, déploiement).

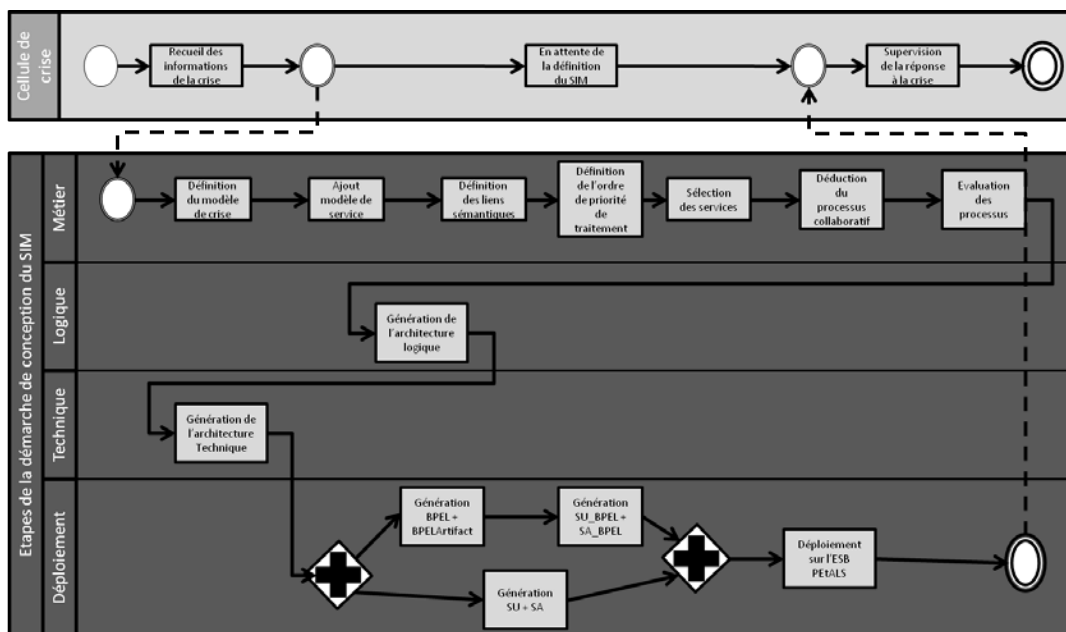


Figure IV- 6 : démarche de conception suite à une évolution de la situation.

### III.1.2 Processus de conception suite à une évolution de la composition de la cellule de gestion de crise

Lors d'une évolution de la composition de la cellule de gestion de crise (par exemple l'arrivée ou le départ d'un acteur), l'ensemble des compétences à disposition de la cellule de gestion de crise évolue. Cette évolution peut par conséquent remettre en cause les choix effectués par la cellule de crise (positivement ou négativement).

Comme représenté par la figure IV-7, ce type d'évolution modifiant l'ensemble des modèles de services, il est nécessaire de redéfinir les liens sémantiques avant de continuer la démarche de conception.



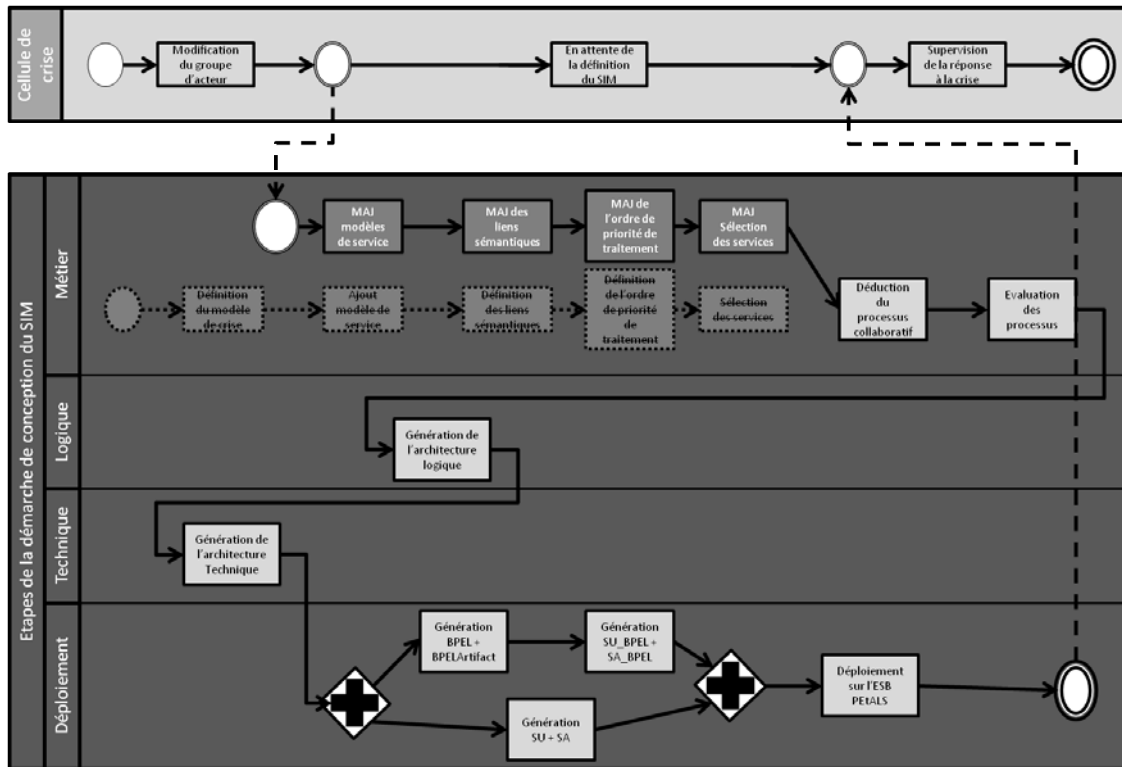


Figure IV- 7 : déroulement de la démarche suite à une évolution du groupe d'acteur de la cellule de crise.

### III.1.3 Processus de conception suite à un dysfonctionnement.

Suite à un dysfonctionnement dans le déroulement du processus de réponse à la crise (par exemple l'inefficacité d'une activité), le modèle de crise, l'ensemble des modèles de services ainsi que la définition des liens sémantiques restent inchangés. Dans ces conditions, ce sont la définition de l'ordre de priorité de traitement ainsi que la sélection des services qui doivent être réévalués, comme illustré sur la figure IV-8.

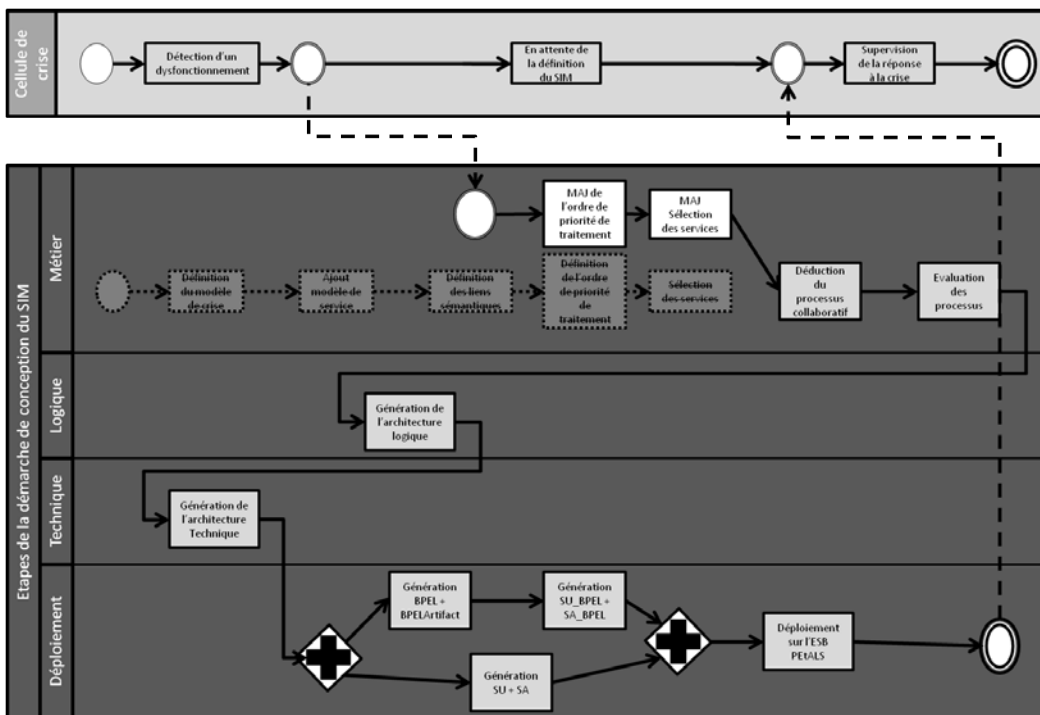


Figure IV- 8 : déroulement de la démarche suite à une évolution de dysfonctionnement.

## III.2 Réalisation d'une chaîne de conception flexible

Nous venons de voir qu'apporter de la flexibilité de conception à nos travaux revenait à permettre de s'engager dans la démarche de conception à différents niveaux du processus en fonction des évolutions apparues. Cette flexibilité du workflow de conception (design-time) reposant principalement sur une dynamique d'orchestration, nous avons décidé de construire notre chaîne de conception autour d'un ESB.

Cette orientation conceptuelle et technique exige de créer des WebServices pour toutes les étapes de la démarche de conception, et d'intégrer cette base de WebServices au sein d'un ESB qui sera en charge de leur invocation en fonction du processus de conception à dérouler.

### III.2.1 Support technologique de cette flexibilité

Dans le but de réaliser une démarche de conception flexible, nous avons réalisé un prototype en nous basant sur les résultats des différentes étapes de la démarche de conception d'un SIM. Etapes que nous avons détaillées à travers les chapitres précédents. Ce prototype, illustré par la figure IV-9, est structuré autour d'un ESB et est composé d'un ensemble de WebServices.

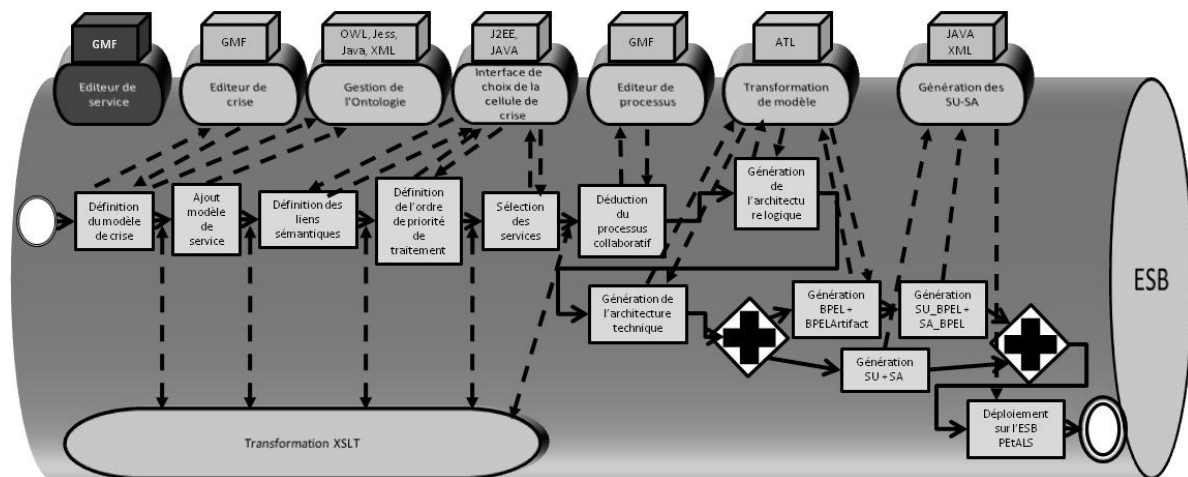


Figure IV- 9 : architecture de la chaîne de conception

Le processus nominal de conception, représenté au sein de l'ESB de la figure IV-9, fait appel à un certain nombre de WebServices. Ces WebServices permettent la réalisation des différentes étapes de la démarche de conception détaillée à travers les chapitres précédents. Par exemple, les transformations de modèle, notamment celle expliquée au chapitre III, sont réalisées par le WebService « Transformation de modèle ». Concernant la déduction du processus collaboratif, présenté au chapitre II, sa réalisation est en grande partie permise par les WebServices suivants :

- « Gestion de l'ontologie » : ce service permet d'ajouter de connaissance, d'exécuter les règles de déduction et d'extraire de la connaissance,
- « Interface de choix de la cellule de crise » : ce service permet l'aide et la sauvegarde des choix de la cellule de gestion de crise, que ce soit au niveau de la définition des liens sémantique, au niveau de la définition de l'ordre de priorité de traitement ou encore au niveau de la sélection des services pour résoudre un problème,
- « Editeur de processus collaboratif » : ce service permet de visualiser et modifier le processus collaboratif de réponse déduit.

Ces WebServices ainsi que ceux n'ayant pas été détaillés ci-dessus, repose sur des technologies différentes (représentées par un cube « technologique » positionné au-dessus de chaque WebService de la figure IV-9). Ainsi avant de détailler plus précisément les WebServices développés pour le prototype, nous commencerons par identifier les différentes technologies utilisées.

### III.2.2 Technologies utilisées

La démarche de conception du système d'information de médiation se base notamment sur trois types de modèles (de crise, de service, de processus collaboratif) respectant chacun une partie du métamodèle de crise. Ces trois modèles, représentés (parmi d'autres éléments) sur la ligne du milieu de la figure IV-10, résultent d'une activité de modélisation (création/modification de modèles) prise en charge par la cellule de gestion de crise. Il est donc nécessaire de proposer aux acteurs de cette cellule de gestion des crise un panel d'éditeurs et d'outils logiciels leur permettant de créer ces modèles de manière efficace et accessible, tout en respectant le métamodèle.

Pour créer ces éditeurs, représentés dans la partie supérieure de la figure IV-10, nous avons décidé d'utiliser l'outil GMF (Graphical Modeling Framework)(GMF, 2005). GMF est un plugin Eclipse libre. C'est un outil de métamodélisation et de configuration d'éditeurs de modèles qui permet non seulement de créer graphiquement des métamodèles (ou d'importer des métamodèles existants) mais également de construire des éditeurs graphiques dédiés à la création de modèles respectant les métamodèles définis. En outre, ces éditeurs permettent d'extraire, à l'issue d'une telle modélisation graphique, un fichier XML contenant l'ensemble des informations relatives au modèle construit.

Ainsi, l'ensemble des informations contenu dans un modèle et extrait au format XML, peut ensuite être injecté dans une ontologie, représentée dans la partie inférieure de la figure IV-10, créée avec Protégé (Protégé, 2000). Cette injection permet, grâce à l'utilisation de feuilles de style, de transformer les informations contenues dans le fichier XML au format OWL (Web Ontology Language).

Les règles de déduction, présentées en détails au chapitre II, peuvent alors être appliquées à cette connaissance grâce au moteur d'inférence JESS (cette action est également représentée dans la partie inférieure de la figure IV-10). Pour rappel, deux types de règles sont utilisées successivement : (i) celles permettant de filtrer les services pouvant être utilisés afin de résoudre un problème et (ii) celles permettant de construire le processus de réponse. L'application de cette deuxième vague de règles nécessite au préalable que les acteurs de la cellule de crise aient effectués les choix relevant des priorités de traitement et des affectations de services. Afin d'aider la cellule de gestion de crise à faire ces choix, nous avons souhaité créer une interface web dynamique. Nous avons donc opté pour une interface Web (qui figure dans la partie supérieure de la figure IV-10) réalisée en J2EE. Afin de pouvoir transférer les informations depuis l'ontologie vers cette interface et vice-versa, des fichiers XML et des feuilles de styles sont utilisés.

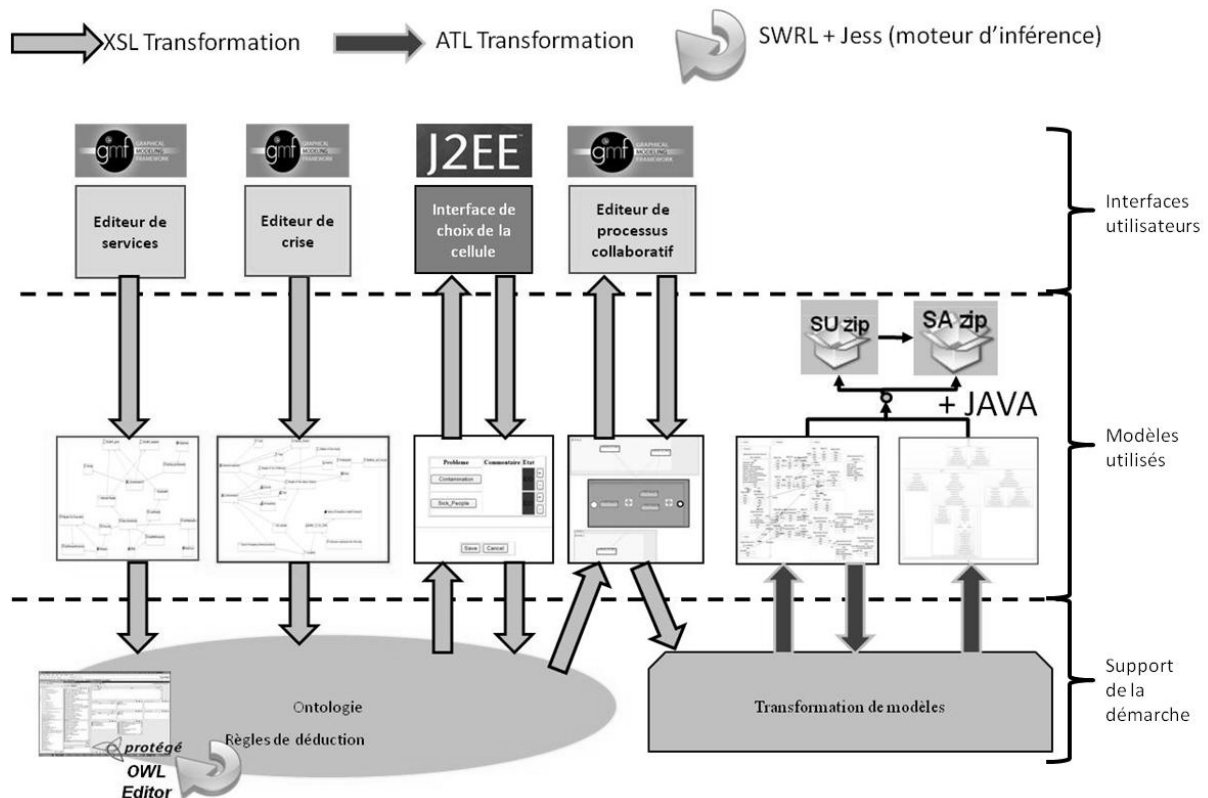


Figure IV- 10 : ensemble des technologies utilisées dans le cadre de la démarche

Concernant la réalisation de la transformation de modèles (concrètement, la série de transformations depuis le modèle de processus collaboratif jusqu'aux artefacts de déploiement du médiateur en passant par les modèles d'architecture logique et technique), le choix s'est porté sur l'utilisation du langage ATL. Celui-ci permettant de définir des règles de transformations au niveau des métamodèles (source et cible), il est particulièrement pertinent dans le cas d'une démarche dirigée par les modèles.

Enfin, les travaux de A.M. Barthe (qui prennent en charge le déploiement effectif du médiateur) sont basés sur le langage JAVA. Cependant afin de pouvoir réaliser cette dernière étape, il est nécessaire de fournir des fichiers XML contenant des informations liées au BC et SE disponibles (*i.e* les *éléments nécessaires* de la démarche présentée dans le chapitre III).

### III.2.3 Identification des WebServices de conception

Suite à l'implémentation de ces différentes applications correspondant aux différentes étapes de la chaîne de déduction (sur la base des technologies présentées), un point crucial concerne la compatibilité et l'intégration de ces applications au sein de l'ESB en charge d'héberger (et d'orchestrer) la démarche de conception du SIM (tout en prenant également en charge son agilité). Afin de satisfaire cet objectif, les applications réalisées ont été converties en WebServices afin d'être aisément intégrables au bus de services. Ces WebServices sont décrits dans le tableau IV-2. On y retrouve pour chaque service l'ensemble des opérations disponibles ainsi que leurs entrées et sorties. Au niveau de ce tableau, il faut noter qu'une opération de création a été ajoutée au WebService « Editeur de processus collaboratif ». Bien qu'elle ne soit pas utilisée dans le cadre de nos travaux, cette opération permet de commencer la démarche au niveau du processus. Ce choix a été fait dans le but de gagner en flexibilité.

Tableau IV- 2 : Description des WebServices réalisés

Service	Opération	Entrées	Sorties	Technologie de base
Editeur de Crise	Création de nouveau modèle		Modèle de crise	GMF
	Modification de modèle	Modèle de crise	Nouveau Modèle de crise	
Editeur de Service	Création de nouveau modèle		Modèle de service	GMF
	Modification de modèle	Modèle de service	Nouveau Modèle de service	
Gestion de l'ontologie	Ajout modèle de crise	Modèle de crise		OWL ; XML ; XSLT
	Ajout modèle de service	Modèle de service		
	Exécution de règle de déduction 1			Jess
	Extraction pour les choix de la cellule		Liste des problèmes et des services.	OWL ; XML ; XSLT
	Ajout des choix de la cellule	Choix de la cellule		
	Exécution règle de déduction 2			Jess
	Extraction du processus collaboratif			Modèle de processus collaboratif
Editeur de processus collaboratif	Création de nouveau modèle		Modèle de processus collaboratif	GMF
	Modification de modèle	Modèle de processus collaboratif	Nouveau Modèle de processus collaboratif	
Transformation de modèle	Transformation de modèle	Métamodèle source, Métamodèle Cible, Fichier de règle de transformation, modèle source	Modèle cible	ATL

	Ajout de connaissance	Modèle cible, Données à ajouter	Modèle cible enrichi	XML ; XSLT
Interface de choix de la cellule de crise	Définitions des liens sémantiques	Liste des risques, conséquences et conditions	Choix de la cellule	J2EE ; JAVA
	Définition de l'ordre de priorité et des services à utiliser	Liste des problèmes et des services.		
Génération des SU-SA	Création SU-SA	UMLT, Liste des composants, WSDLs,	SU, SA	JAVA, XML
	Création SU-BPEL SA-BPEL	UMLT, Liste des composants, WSDLs, BPEL, BPELArtifact	SU-BPEL, SA-BPEL	JAVA, XML

Dans un souci d'agilité et comme présenté en section III.1 de ce chapitre, l'orchestration de ces services de « Design-Time » est totalement flexible. Cette flexibilité est obtenue grâce à la décomposition de cette démarche en WebServices et à leurs intégrations au sein d'un ESB équipée d'un orchestrateur susceptible d'être interrompu (suite à la détection d'une variation significative remettant en cause le workflow courant de « Run-Time ») et d'être relancé sur la base d'un workflow de « Design-Time » pertinent (cf. les workflows de « Design-Time » présentés en section III.1 de ce chapitre). L'ensemble de ces WebServices a été réalisé et déployé sur l'ESB PEtALS. Ainsi l'ensemble des WebServices et l'ESB PEtALS constituent un prototype opérationnel d'atelier de conception d'un SIM dédié à la gestion de crise. Ce prototype est non seulement flexible, mais il est aussi évolutif. En effet, la décomposition de la démarche en WebServices permet de modifier plus simplement des étapes de la démarche. Par exemple, il suffit de modifier les WebServices liés à la déduction du processus collaboratif pour appliquer la démarche à un autre cas d'étude par exemple.

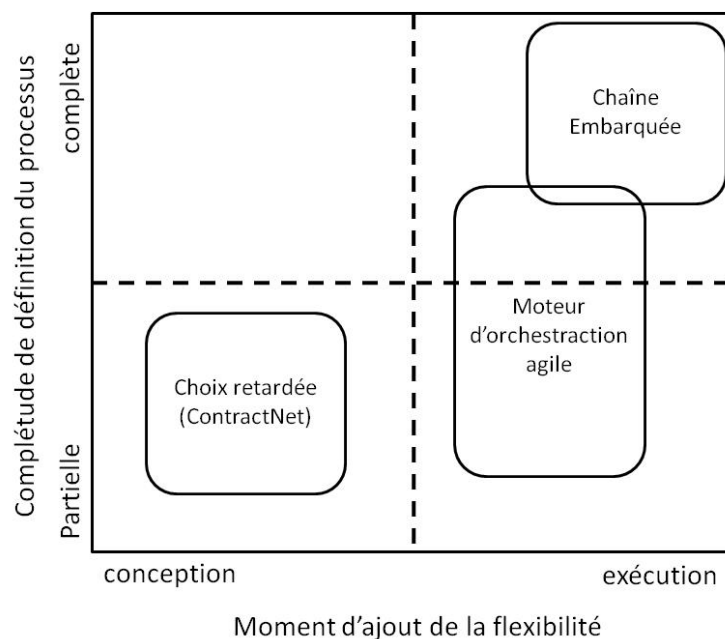
## IV. Conclusion

Une crise est par définition un événement évolutif. Il est donc nécessaire de fournir au système de systèmes d'information, support à la réponse à la crise, une caractéristique d'agilité. Cette caractéristique se décompose en deux sous caractéristiques : (i) réactivité et (ii) flexibilité.

La réactivité ne peut être atteinte que grâce à la cellule de crise. En effet, bien que la démarche d'ingénierie dirigée par les modèles soit semi-automatisée pour obtenir le processus collaboratif (s'appuyant en particulier sur (i) les déductions et présélections proposées à la cellule de crise et (ii) la construction automatisée d'un processus collaboratif) puis entièrement automatisée pour le déploiement du SIM, la durée de création de ce modèle est soumise aux temps de décision de la cellule de crise.

Concernant la flexibilité, des travaux ont été menés en parallèle au sein du projet. Ces travaux ont été menés sur l'apport de flexibilité par définition partielle du processus de réponse avec l'utilisation du ContractNet (faure *et al*, 2009), d'autres sur la création d'un moteur d'orchestration « agile », Maestro, permettant d'interrompre ou de recommencer l'exécution d'un workflow, voire d'en sauter une activité.

Enfin les nôtres porte sur la réalisation d'une chaîne de conception flexible. La figure IV-11 positionne ces travaux vis-à-vis du cadre proposé par (schonenberg *et al*, 2008).



**Figure IV- 11 : positionnement des travaux du projet sur le repère de (Schonenberg *et al*, 2008)**

La flexibilité par conception consiste à permettre de démarrer la démarche de conception à n'importe quelle étape. Pour ce faire, nous avons importé l'ensemble des différentes étapes sous formes de WebServices au sein d'un ESB. Cette importation est représentée par la partie basse de la Figure IV-12. Cette « mise en abîme » de l'orchestration (on cherche à construire un ESB orchestrant un workflow de gestion de crise en exécutant, sur un ESB un workflow de conception de SIM) permet d'en assouplir grandement le déroulement en profitant des spécificités de l'orchestration de services. Ainsi, en design-time, cette souplesse d'orchestration (qui permet de modifier le processus en cours d'exécution) apporte la possibilité de parcourir à nouveau tout ou partie de la chaîne de conception du SIM, alors qu'en Run-Time, elle apporte la possibilité au SIM de modifier son processus de gestion de crise.

L'apport d'une caractéristique d'agilité n'est pas suffisant en lui-même, encore faut-il déterminer les conditions de mobilisation de cette agilité. Nous avons donc mis en place un tableau de bord (la fiche de suivi), dans le but d'aider la cellule de gestion de crise à détecter l'apparition d'évolutions. Cette fiche de suivi est mise à jour lors du déroulement de la réponse à la crise, représentée par la partie supérieur de la figure IV-12, plus précisément à chaque début et fin de service.

En fonction des décisions prises par la cellule de crise suite à la consultation de la fiche de suivi, le moteur d'orchestration doit arrêter l'exécution de la réponse en cours et recommencer la démarche de conception à un certain niveau. Suite à la définition du nouveau système d'information de médiation, le moteur d'orchestration lance alors le nouveau processus. Ce processus est orchestré tant qu'il n'y a pas de nouvelle interruption (ou que la réponse à la crise n'est pas terminée et que la situation n'est pas revenue à un état stable et acceptable).

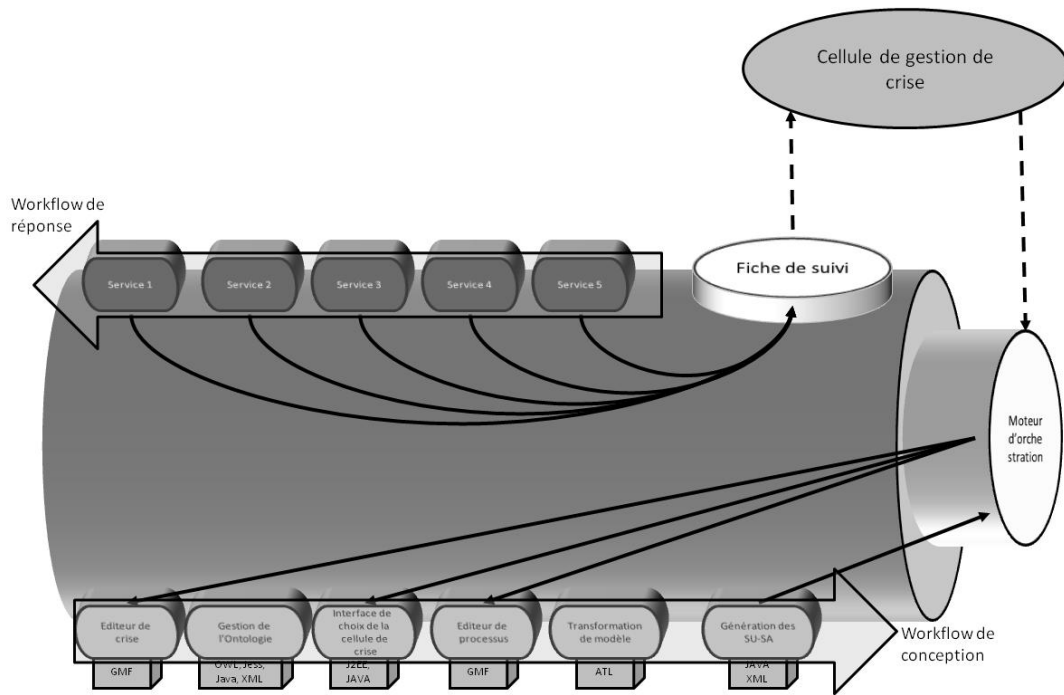


Figure IV- 12 : illustration d'un déroulement de réponse à la crise vue à travers l'ESB





# Chapitre V :

## Exemple de réponse à la crise

### *Mise en place d'un système de systèmes*

<b>Introduction</b> .....	<b>118</b>
<b>I. Conception du SoS à t0</b> .....	<b>118</b>
I.1 Création du modèle de crise .....	118
I.2 Définitions des liens de correspondance sémantique.....	119
I.3 Priorité de traitement et choix des services.....	122
I.4 Dédution du processus collaboratif de réponse.....	128
I.5 Dédution de l'architecture technique du SIM.....	129
I.6 Dédution du processus au format BPEL.....	130
<b>II. Mise en évidence de l'agilité au travers d'évolutions</b> .....	<b>131</b>
II.1 Variation de la composition de la cellule à t1. ....	131
II.2 Evolution de la situation à t2. ....	132
<b>III. Conclusion</b> .....	<b>134</b>

## Introduction

Ce chapitre propose de présenter un exemple de démarche de conception d'un système d'information de médiation et par ce biais d'obtenir un système de systèmes de réponse à la crise. Cet exemple présente un but illustratif, ce qui implique que les modèles de services, présentés en Annexe B, ne sont pas exhaustifs et le scénario de réponse est fictif.

Afin d'illustrer au mieux cette démarche, le chapitre commencera par une présentation de la situation avant de poursuivre par la déduction du processus collaboratif et le déploiement du médiateur pour finir sur le traitements de deux évolutions lors de la réponse à la crise. Afin de distinguer ce qui relève de la cellule de gestion de crise de ce qui est transparent pour elle, deux polices d'écritures seront utilisées : celle-ci sera utilisée pour expliquer les choix et actions relevant de la cellule de gestion de crise. **Celle-ci sera utilisée d'une part pour donner un complément d'explication sur les outils, et d'autre part, pour expliquer les étapes transparentes pour la cellule de gestion de crise.**

Le théâtre de cet exemple, inspiré d'un exercice réalisé en 2004, est un passage à niveaux à proximité de Marssac-sur-Tarn (81). Sur ce passage à niveaux, un camion citerne est entré en collision avec un train de voyageur. De plus, les panonceaux de signalisation de la citerne du camion n'étant pas conformes à la réglementation, le produit, contenu dans la citerne, est inconnu. Suite à la collision, les gendarmes sont prévenus et ils constatent qu'un incendie s'est déclaré au niveau du moteur du camion et que des voyageurs du train sont blessés. C'est le début de la réponse à la crise

## I. Conception du SoS à $t_0$

Nous sommes à l'instant  $t_0$ , la cellule de gestion de crise décide de lancer la création du système de systèmes de réponse à la crise en sachant que le SAMU n'est pour l'instant pas disponible.

La cellule de gestion de crise réunit autour du préfet les agents de liaison des acteurs suivants :

- la gendarmerie nationale (GIE),
- les sapeurs pompiers (SDIS),
- la DDE,
- Le SAMU.

Chaque agent de liaison se présente avec le modèle de description de ses services ainsi que son propre système d'information et ses moyens de communications avec les agents sur le terrain.

### I.1 Création du modèle de crise

La cellule de gestion de crise commence par collecter les informations portant sur la situation afin de créer le modèle crise. Ce modèle de crise est représenté par la figure V-1. On y trouve l'événement déclencheur de la crise « *Collision train camion* » ainsi que les problèmes suivants :

- la conséquence « *incendie réservoir* » qui concrétise le risque « *risque incendie* » et qui impacte le « *camion* »,

- la conséquence « *personnes\_blessées* » qui concrétise le risque de « *risque\_blessés* » et qui impacte les « *victimes de la collision* »,
- la conséquence « *interruption trafic* » qui impacte le « *passage à niveau* » et la « *voie ferrée* » alors que la conséquence « *engorgement* » impacte la « *voie circulaire* » et le « *passage à niveau* »,
- le risque de « *contamination* » de la « *population sur zone* »,
- le risque de « *fuite de la citerne* » qui impacte les « *personnes sur zone* », le « *camion* » et le « *train de voyageur* ».

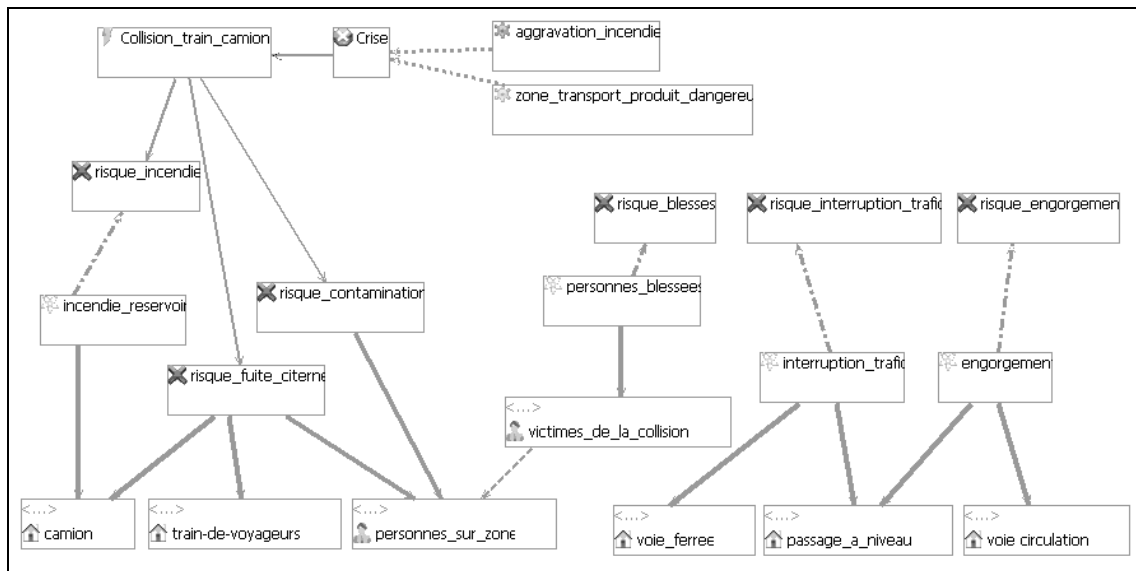


Figure V- 1 : modèle de crise à l'instant t0

Une fois créé, ce modèle de crise est injecté au sein de l'ontologie ainsi que l'ensemble des modèles de services des acteurs présents. Il est alors demandé à la cellule de gestion de crise de définir les liens de correspondance sémantique entre ces différents modèles. Pour cela l'ensemble des risques, conséquences et conditions est extrait de l'ontologie puis envoyé comme message d'entrée au service d'interface de choix de la cellule.

## I.2 Définitions des liens de correspondance sémantique

La cellule de la gestion de crise doit définir l'ensemble des liens de correspondance sémantique entre les différents éléments des modèles hétérogènes. L'interface, représentée par les figures V-2, V-3 et V-4, est alors proposée à la cellule de gestion de crise.

Cette interface est décomposée en trois tableaux chacun regroupant les instances d'un concept du métamodèle de crise. La figure V-2 représente le tableau concernant les risques alors que la figure V-3 représente le tableau concernant les conséquences et enfin la figure V-4 représente le tableau des conditions.

On retrouve en colonne des deux premiers tableaux (risques et conséquences) les éléments du modèle de crise. Les lignes, quant à elles, représentent les éléments des différents modèles de description des acteurs. Ainsi le lien de correspondance

entre un élément du modèle de crise et un élément d'un modèle de description des acteurs se réalise à l'intersection de la ligne et de la colonne concernées. Au niveau de cette intersection, l'utilisateur a trois boutons à sa disposition lui permettant de modifier le lien de correspondance sémantique.

Concernant le tableau pour les conditions, l'ensemble des conditions se retrouve à la fois en ligne et en colonne dans le but de pouvoir définir un lien de correspondance entre toutes les conditions. De plus, seul le lien de correspondance « SameAs » est possible entre deux instances de conditions.

Liste des risques

Risques	risque_incendie	risque_blessee	risque_contamination	risque_interruption_trafic	risque_engorgement	risque_fuite_citerne
Risque_fuite_produit_dangereux	no_opinion same_as near no_opinion	no_opinion same_as near no_opinion	near same_as near no_opinion	no_opinion same_as near no_opinion	no_opinion same_as near no_opinion	same_as same_as near no_opinion
Risque_possibilite_produit_dangereux	no_opinion same_as near no_opinion	no_opinion same_as near no_opinion	same_as same_as near no_opinion	no_opinion same_as near no_opinion	no_opinion same_as near no_opinion	near same_as near no_opinion
Risque_aggravation_etat_blessees	no_opinion same_as near no_opinion	near same_as near no_opinion	no_opinion same_as near no_opinion	no_opinion same_as near no_opinion	no_opinion same_as near no_opinion	no_opinion same_as near no_opinion
risque_victimes	no_opinion same_as near no_opinion	near same_as near no_opinion	no_opinion same_as near no_opinion	no_opinion same_as near no_opinion	no_opinion same_as near no_opinion	no_opinion same_as near no_opinion
Risque_de_massification_blessees	no_opinion same_as near no_opinion	near same_as near no_opinion	no_opinion same_as near no_opinion	no_opinion same_as near no_opinion	no_opinion same_as near no_opinion	no_opinion same_as near no_opinion
risque_de_population_autour_objet_dangereux	no_opinion same_as near no_opinion	near same_as near no_opinion	no_opinion same_as near no_opinion	no_opinion same_as near no_opinion	no_opinion same_as near no_opinion	no_opinion same_as near no_opinion
risque_explosion	near same_as near no_opinion	no_opinion same_as near no_opinion	no_opinion same_as near no_opinion	no_opinion same_as near no_opinion	no_opinion same_as near no_opinion	no_opinion same_as near no_opinion

Figure V- 2 : tableau de correspondance sémantique des instances de risque

Consequences	personnes_blessees	engorgement	interruption_trafic	incendie_reservoir
perturbation_de_l_evacuation	no_opinion	near	near	no_opinion
obstacle_circulation	no_opinion	near	near	no_opinion
fuite_produit_dangereux	no_opinion	no_opinion	no_opinion	no_opinion
incendie_faible_ampleur	no_opinion	no_opinion	no_opinion	same_as
victimes	near	no_opinion	no_opinion	no_opinion
aggravation_etat_blessees	near	no_opinion	no_opinion	no_opinion
massification_blessees	near	no_opinion	no_opinion	no_opinion
augmentation_personnes_en_danger	near	no_opinion	no_opinion	no_opinion
incendie_grande_ampleur	no_opinion	no_opinion	no_opinion	near

Figure V- 3 : tableau de correspondance sémantique des instances de conséquence

La cellule de gestion de crise a fait les choix présentés par les figures. On peut noter parmi ces choix le fait que la conséquence « incendie moteur » est équivalent à « un incendie de faible ampleur » et proche d'un « incendie de grande ampleur ». Au niveau des conditions, les conditions « zone accident organisée GIE » et « dispositif réalisé GIE » ont été définies comme équivalentes.

Conditions	zone_accident_organisee_GIE	dispositif_GIE_realise	Deviation_validee_DDE	besoin_validation_DDE	accompagnement_GIE
zone_accident_organisee_GIE	same_as	same_as	no_opinion	no_opinion	no_opinion
dispositif_GIE_realise	same_as	same_as	no_opinion	no_opinion	no_opinion
Deviation_validee_DDE	no_opinion	no_opinion	same_as	no_opinion	no_opinion
besoin_validation_DDE	no_opinion	no_opinion	no_opinion	same_as	no_opinion
accompagnement_GIE	no_opinion	no_opinion	no_opinion	no_opinion	same_as
itineraire_deplacement	no_opinion	no_opinion	no_opinion	no_opinion	no_opinion
Besoin_d_accompagnement	no_opinion	no_opinion	no_opinion	no_opinion	no_opinion
zone_de_stockage_disponible	no_opinion	no_opinion	no_opinion	no_opinion	no_opinion
itineraires_adaptes_GIE	no_opinion	no_opinion	no_opinion	no_opinion	no_opinion
itineraire_GIE_en_place	no_opinion	no_opinion	no_opinion	no_opinion	no_opinion
dispositif_SAMU_realise	no_opinion	no_opinion	no_opinion	no_opinion	no_opinion
moyens_acheminement_disponibles	no_opinion	no_opinion	no_opinion	no_opinion	no_opinion

Figure V- 4 : tableau de correspondance sémantique des instances de condition

Une fois les liens de correspondance sémantique définis, ils sont transmis au service de « *gestion de l'ontologie* » qui va injecter ces choix au sein de l'ontologie. Suite à cela, l'opération « *exécution de règle de déduction 1* » est exécutée. Cette opération consiste à exécuter, via le moteur d'inférence Jess, les règles de déduction portant d'une part sur la propagation de liens de correspondance sémantique et d'autre part, sur la déduction des services utilisables pour traiter chaque problème de la crise. Ces règles sont détaillées au chapitre II et correspondent aux règles II-1 à II-9.

Une fois ces règles exécutées, les listes de services utilisables pour traiter un problème de la crise sont extraites de l'ontologie et sont proposées à la cellule de gestion de crise.

### I.3 Priorité de traitement et choix des services

Une nouvelle interface est proposée à la cellule de gestion de crise. Cette interface lui permet de définir l'ordre de priorité de traitement. La figure V-5 présente cette interface avec le choix de priorité de traitement effectué à t0. L'ordre de priorité de traitement d'un problème correspond à la ligne dans le tableau de ce problème. Les boutons « + » et « - » permettent de modifier cet ordre.

La cellule de gestion de crise a décidé l'ordre suivant de priorité de traitement des problèmes :

1. le risque de « *contamination* »,
2. la conséquence « *incendie moteur* »,
3. la conséquence « *personnes blessées* »,
4. le risque « *incendie* » qui ici sous-entend l'expansion de *l'incendie moteur* à la *citerne du camion*,
5. la conséquence « *engorgement* ».

Les autres risques et conséquences, bien que présents dans le modèle de crise, ne font pour l'instant pas l'objet d'une attention particulière de la cellule de gestion de crise. Par conséquent, ils sont laissés dans un ordre aléatoire.

Le choix de la cellule de gestion de crise de laisser de côté des problèmes que l'on pourrait caractériser de « mineur » est permis grâce à la caractéristique d'agilité de la démarche. Le fait que la cellule puisse se focaliser sur les problèmes qu'elle estime « cruciaux » (parce que la possibilité de profiter de l'agilité du système et de revenir ultérieurement sur certains problèmes lui est autorisée) lui permet d'agir plus rapidement (et donc d'être plus réactive).

Probleme	Type	Commentaire	Etat	
risque_contamination	Risk		KO	+ -
personnes_blessees	Consequence		KO	+ -
incendie_reservoir	Consequence		KO	+ -
engorgement	Consequence		KO	+ -
interruption_trafic	Consequence		KO	+ -
risque_incendie	Risk		KO	+ -
risque_blessees	Risk		KO	+ -
risque_fuite_citerne	Risk		KO	+ -
risque_interruption_trafic	Risk		KO	+ -
risque_engorgement	Risk		KO	+ -

Figure V- 5 : définition de l'ordre de priorité de traitement à t0

Suite à la définition de cet ordre de priorité de traitement, la cellule de gestion de crise s'intéresse à trouver des solutions au premier problème, le risque de « contamination ». Pour cela, il suffit de cliquer sur le nom du problème afin d'avoir la liste des services pouvant être utilisés. La figure V-6 présente les services à disposition qui pourraient prévenir le risque de « contamination ».

Service des acteurs	confiance	accessibilité valeur	accessibilité commentaire	collatéraux valeur	collatéraux commentaire	acteur	choix
DDE_Determiner_nature_produit	1	1	???	1	???	DDE	<input checked="" type="radio"/> advisable <input type="radio"/> no_opinion <input type="radio"/> inadvisable
GIE_Adapter_le_perimetre_de_securite	1	0.33	???	1	???	GIE	<input type="radio"/> advisable <input checked="" type="radio"/> no_opinion <input type="radio"/> inadvisable

Service des acteurs	confiance	accessibilité valeur	accessibilité commentaire	collatéraux valeur	collatéraux commentaire	acteur	choix
DDE_Confiner_produit_dangereux	0,8	0.5	???	1	???	DDE	<input type="radio"/> advisable <input checked="" type="radio"/> no_opinion <input type="radio"/> inadvisable

Figure V- 6 : choix des services pour le risque de contamination

Les services sont répartis en deux tableaux, le premier listant les services répondant exactement à ce problème. Le deuxième liste les services répondant à un problème proche.



Concernant le risque de *contamination*, la cellule de gestion de crise a décidé de mettre en œuvre le service suivant : « *Déterminer la nature du produit* » proposé par la DDE. Ce choix se base sur le fait que le camion n'a pas de panneaux indiquant la nature du produit. Une fois ces choix validés la cellule de gestion de crise revient sur la première interface. La figure V-7 montre l'état de l'interface suite à la sélection du premier service.

On y remarque que le statut du risque « *contamination* » est passé à *OKcan* car au moins un des services sélectionnés prévient exactement ce risque. On y remarque aussi que le statut d'autre problème a changé. En effet, certains services sélectionnés permettant également de traiter d'autres problèmes, ces autres problèmes seront aussi traités lors de l'exécution dudit service (par exemple, le traitement du risque de contamination prévient également le risque de fuite de la citerne).

Probleme	Type	Commentaire	Etat	
risque_contamination	Risk		okcan	+ -
personnes_blessees	Consequence		ko	+ -
incendie_reservoir	Consequence		ko	+ -
engorgement	Consequence		ko	+ -
interruption_trafic	Consequence		ko	+ -
risque_incendie	Risk		ko	+ -
risque_blesses	Risk		ko	+ -
risque_fuite_citerne	Risk		okcould	+ -
risque_interruption_trafic	Risk		ko	+ -
risque_engorgement	Risk		ko	+ -

**Figure V- 7 : état de la première interface après la sélection des services pour le risque de contamination**

La cellule de gestion de crise doit ensuite sélectionner des services dans le but de résoudre la conséquence « *incendie réservoir* ». La figure V-8 présente ce choix.

Service des acteurs	confiance	accessibilité valeur	accessibilité commentaire	collatéraux valeur	collatéraux commentaire	acteur	choix
SAMU_Soigner_blesses	1	0.5	???	1	???	SAMU	<input type="radio"/> advisable <input checked="" type="radio"/> no_opinion <input type="radio"/> inadvisable

Service des acteurs	confiance	accessibilité valeur	accessibilité commentaire	collatéraux valeur	collatéraux commentaire	acteur	choix
GIE_Adapter_le_perimetre_de_securite	0,8	0.33	???	1	???	GIE	<input type="radio"/> advisable <input checked="" type="radio"/> no_opinion <input type="radio"/> inadvisable
GIE_Faire_evacuer_une_zone	0,8	1	???	1	???	GIE	<input type="radio"/> advisable <input checked="" type="radio"/> no_opinion <input type="radio"/> inadvisable
SDIS_Soigner_victimes	0,8	0.5	???	1	???	SDIS	<input checked="" type="radio"/> advisable <input type="radio"/> no_opinion <input type="radio"/> inadvisable
SAMU_Trier_les_blesses	0,8	1	???	1	???	SAMU	<input type="radio"/> advisable <input checked="" type="radio"/> no_opinion <input type="radio"/> inadvisable
SAMU_Acheminer_victimes	0,8	0.5	???	1	???	SAMU	<input type="radio"/> advisable <input checked="" type="radio"/> no_opinion <input type="radio"/> inadvisable

Figure V- 8 : choix des services à utiliser pour traiter la conséquence de personnes blessées

On remarque ici que la valeur de l'indice d'accessibilité du service « *Soigner victimes* » du SDIS est inférieure à 1. Par conséquent, ce service impliquera l'utilisation d'autres services soit en pré-requis soit a posteriori.

Au moment de ce choix, le SAMU a indiqué qu'il n'était pas en mesure d'intervenir pour le moment dans le cadre de la réponse à cette crise. La cellule de gestion de crise a donc demandé au SDIS de prendre en charge les blessés en sélectionnant le service « *Soigner victimes* ». La cellule de gestion de crise revient alors sur la première interface dont l'état correspond à la figure V-9.

Probleme	Type	Commentaire	Etat	
risque_contamination	Risk		okcan	+ -
personnes_blessees	Consequence		okcould	+ -
incendie_reservoir	Consequence		ko	+ -
engorgement	Consequence		ko	+ -
interruption_trafic	Consequence		ko	+ -
risque_incendie	Risk		ko	+ -
risque_blesses	Risk		okcould	+ -
risque_fuite_citerne	Risk		okcould	+ -
risque_interruption_trafic	Risk		ko	+ -
risque_engorgement	Risk		ko	+ -

Figure V- 9 : la première interface après la sélection des services pour les personnes blessées

La cellule de gestion de crise doit ensuite sélectionner des services dans le but de résoudre la conséquence « incendie réservoir ». La figure V-10 présente ce choix.

Service des acteurs	confiance	accessibilité valeur	accessibilité commentaire	collatéraux valeur	collatéraux commentaire	acteur	choix
SDIS_Eteindre_incendie_faible_ampleur	1	0.5	???	1	???	SDIS	<input checked="" type="radio"/> advisable <input type="radio"/> no_opinion <input type="radio"/> inadvisable
SDIS_Eteindre_incendie_grande_ampleur	0,8	0.5	???	1	???	SDIS	<input type="radio"/> advisable <input checked="" type="radio"/> no_opinion <input type="radio"/> inadvisable

Figure V- 10 : choix des services à utiliser pour traiter la conséquence d'incendie réservoir

A ce niveau de la démarche, deux services sont proposés. Le premier, « *Eteindre incendie de faible ampleur* », répond exactement à ce problème alors que le second, « *Eteindre incendie de grande ampleur* », pourrait répondre à ce problème. Cette double proposition est directement liée aux choix faits au moment de la définition des liens de correspondance sémantique (cf. figure V-3)

La cellule de gestion de crise a choisi d'utiliser le service du SDIS « *Eteindre incendie de faible ampleur* » pour traiter le problème. La cellule de gestion de crise revient alors sur la première interface dont l'état correspond à la figure V-11

Probleme	Type	Commentaire	Etat	
risque_contamination	Risk		okcan	+ -
personnes_blessees	Consequence		okcould	+ -
incendie_reservoir	Consequence		okcan	+ -
engagement	Consequence		ko	+ -
interruption_trafic	Consequence		ko	+ -
risque_incendie	Risk		okcan	+ -
risque_blesses	Risk		okcould	+ -
risque_fuite_citerne	Risk		okcould	+ -
risque_interruption_trafic	Risk		ko	+ -
risque_engorgement	Risk		ko	+ -

Figure V- 11: état de la première interface après la sélection des services pour l'incendie moteur

La cellule de gestion de crise choisi le service de la gendarmerie « Mettre en place une déviation routière » pour traiter le problème d'engorgement. Ce choix est représenté sur la figure V-12.

Service des acteurs	confiance	accessibilité valeur	accessibilité commentaire	collatéraux valeur	collatéraux commentaire	acteur	choix
DDE_Mettre_en_place_deviation_routiere	1	0.33	???	1	???	DDE	<input type="radio"/> advisable <input checked="" type="radio"/> no_opinion <input type="radio"/> inadvisable
GIE_Mettre_en_place_deviation_routiere	1	0.33	???	1	???	GIE	<input checked="" type="radio"/> advisable <input type="radio"/> no_opinion <input type="radio"/> inadvisable

Service des acteurs	confiance	accessibilité valeur	accessibilité commentaire	collatéraux valeur	collatéraux commentaire	acteur	choix
DDE_Deplacer_vehicule	0,8	0.33	???	1	???	DDE	<input type="radio"/> advisable <input checked="" type="radio"/> no_opinion <input type="radio"/> inadvisable
SDIS_Intervenir_sur_les_accidents_de_la_route	0,8	1	???	1	???	SDIS	<input type="radio"/> advisable <input checked="" type="radio"/> no_opinion <input type="radio"/> inadvisable

Figure V- 12 : des services à utiliser pour traiter la conséquence d'engorgement

Finalement, la cellule de gestion de crise décide de lancer la déduction du processus collaboratif de réponse à la crise à partir des choix fait.

## I.4 Dédution du processus collaboratif de réponse

Sur la base des choix de la cellule de gestion de crise le processus collaboratif est déduit. Cette déduction est réalisée en deux temps, tout d'abord les éléments du processus collaboratif sont créés grâce aux règles de déduction II-10 à II-14. Puis l'ordre d'exécution de ces éléments est défini grâce, d'une part aux règles de déduction II-15 à II-17 et, d'autre part à l'algorithme II-2 de construction des séquences du processus collaboratif.

A l'issue des mécanismes de déduction, les couples de services suivant peuvent être exécutés en parallèle :

- Le service « *Eteindre incendie de faible ampleur* » et le service « *Soigner victimes* » puisque les deux services ont besoin du même pré-requis, à savoir le service « *Adapter dispositif intervention* » du SDIS.
- Les services « *déterminer nature du produit* » et « *soigner victimes* » puisque ces deux services ne sont pas réalisés par le même acteur, ne traitent pas les mêmes problèmes qui impactent des composants système d'étude différents.
- Le service « *déterminer nature du produit* » et « *Mettre en place la déviation routière* » pour la même raison que le point précédent.

Le processus collaboratif déduit est proposé à la cellule de gestion de crise. Ce processus est représenté par la figure V-13. Ce processus est composé de quatre Pools, trois concernant les acteurs intervenant dans le cadre de la réponse (SDIS, GIE, DDE) et un représentant le médiateur. On y retrouve aussi l'ensemble des services sélectionnés ainsi que leurs pré-requis (NB : la figure suivante représente exactement le résultat graphique obtenu suite à la déduction du processus collaboratif).

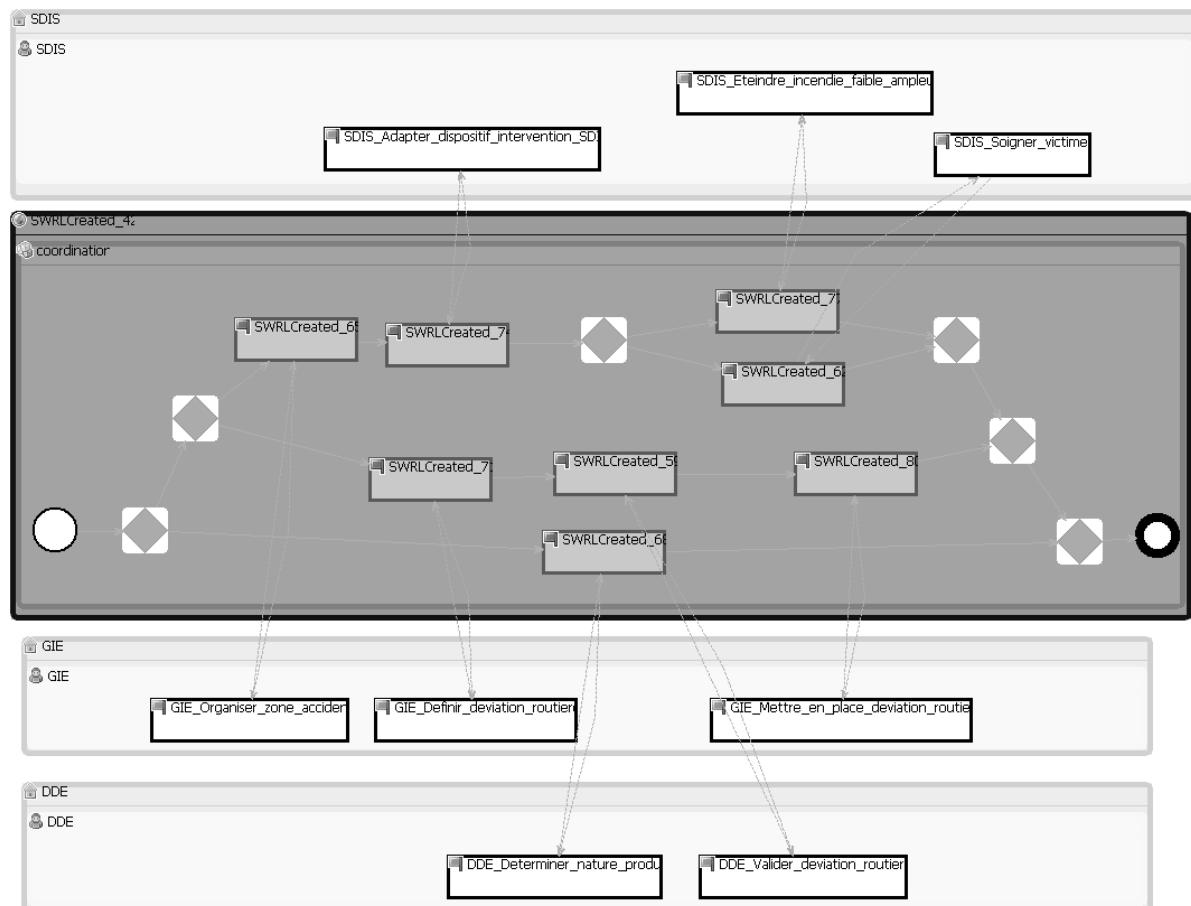


Figure V- 13 : processus collaboratif de réponse déduit à t0.

La cellule de gestion de crise décide de mettre en place ce processus de réponse. A partir de ce moment le reste de la démarche permettant le déploiement du système d'information de médiation est transparent pour la cellule de réponse à la crise.

## I.5 Déduction de l'architecture technique du SIM

Ce processus collaboratif est ensuite transformé en architecture logique puis technique. La figure V-14 représente une vue globale du modèle d'architecture technique dans sa version « brute de déduction ». Cependant, le nombre important d'éléments constituant ce modèle rend la figure illisible. Malgré ce manque de lisibilité, on retrouve au sein du modèle les trois grandes parties : une concernant les services, une autre les informations et une dernière le processus.

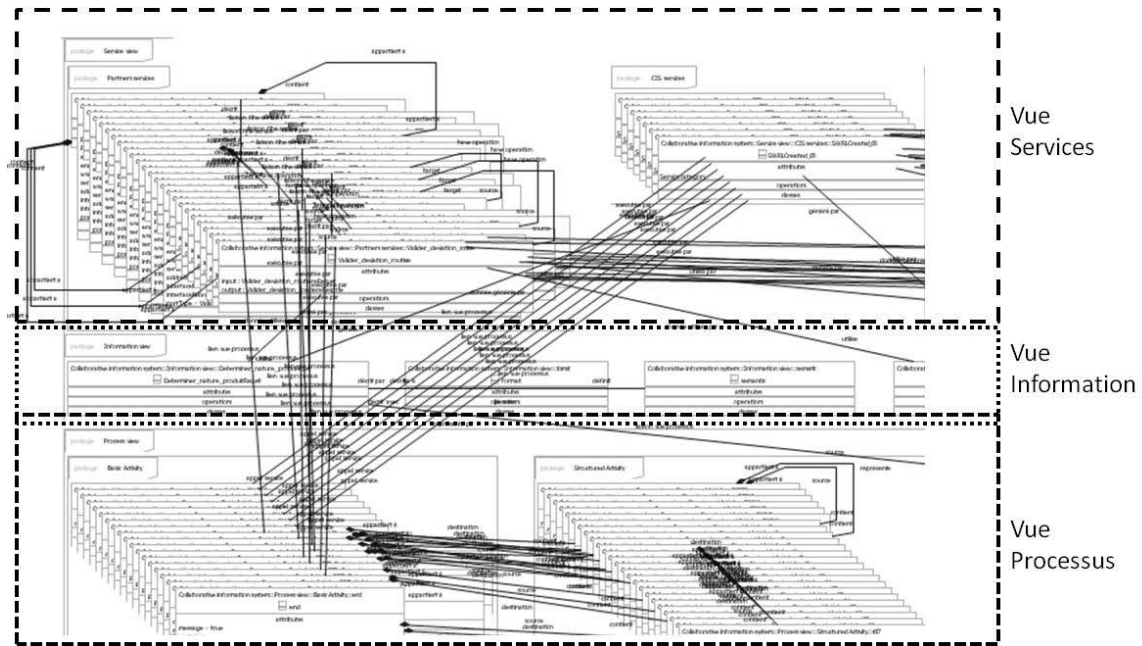


Figure V- 14 : Modèle d'architecture technique déduit

Ce modèle servira de base pour la phase de déploiement et notamment la déduction du BPEL.

### I.6 Déduction du processus au format BPEL

En respectant la transformation de modèle décrite au chapitre III, le BPEL déduit correspond à la figure V-15.

```

<?xml version="1.0" encoding="UTF-8" ?>
<bpmn:process xmlns:bpmn="http://www.omg.org/spec/BPEL/2004/04/process-instance" targetNamespace="http://exemple.org/" queryLanguage="urn:oasis:names:tc:wsbpel:2.0:nblang:xpath.0" expressionLanguage="XPath 1.0 in BPEL2.0" suppressOnFailure="yes">
  <bpmn:import namespace="http://www.isycri.com/Definir_deviation_routiere/" location="Definir_deviation_routiere.wsdl" importType="http://schemas.xmlsoap.org/wsdl/" />
  <bpmn:import namespace="http://www.isycri.com/Organiser_zone_accident/" location="Organiser_zone_accident.wsdl" importType="http://schemas.xmlsoap.org/wsdl/" />
  <bpmn:import namespace="http://www.isycri.com/Mettre_en_place_deviation_routiere/" location="Mettre_en_place_deviation_routiere.wsdl" importType="http://schemas.xmlsoap.org/wsdl/" />
  <bpmn:import namespace="http://www.isycri.com/Soigner_victimes/" location="Soigner_victimes.wsdl" importType="http://schemas.xmlsoap.org/wsdl/" />
  <bpmn:import namespace="http://www.isycri.com/Deployer_un_dispositif_d_intervention_SDIS/" location="Deployer_un_dispositif_d_intervention_SDIS.wsdl" importType="http://schemas.xmlsoap.org/wsdl/" />
  <bpmn:import namespace="http://www.isycri.com/Eteindre_incendie_faible_sapleur/" location="Eteindre_incendie_faible_sapleur.wsdl" importType="http://schemas.xmlsoap.org/wsdl/" />
  <bpmn:import namespace="http://www.isycri.com/Determiner_nature_produit/" location="Determiner_nature_produit.wsdl" importType="http://schemas.xmlsoap.org/wsdl/" />
  <bpmn:import namespace="http://www.isycri.com/Valider_deviation_routiere/" location="Valider_deviation_routiere.wsdl" importType="http://schemas.xmlsoap.org/wsdl/" />
  <bpmn:import namespace="http://www.vb.org/2005/05/addressing" location="vb-addr.xsd" importType="http://www.vb.org/2005/05/XMLSchema/" />
  <bpmn:import namespace="http://www.isycri.com/contractnet.xsd" location="contractnet.xsd" importType="http://www.vb.org/2001/XMLSchema/" />
  <bpmn:import namespace="http://www.isycri.com/IsyMediationInformation_Service/" location="IsyMediationInformation_Service.wsdl" importType="http://schemas.xmlsoap.org/wsdl/" />
  <bpmn:partnerLinks>
    <bpmn:partnerLink name="client" partnerRole="client" partnerType="tns:exemple" />
  </bpmn:partnerLinks>
  <bpmn:sequence name="S10">
    <bpmn:receive name="start" operation="process" portType="tns:exemple" createInstances="yes" variables="input" partnerLink="client"/>
  </bpmn:sequence>
  <bpmn:flow name="G46041">
    <bpmn:sequence name="S10Created_65">
      <bpmn:assign name="S10Created_65request_validate" validate="no"/>
      <bpmn:invoke name="S10Created_65request_validate" operation="validate" outputVariable="mediationResponse" inputVariables="sf10" partnerLink="mediation"/>
      <bpmn:assign name="S10Created_65response_validate" validate="no"/>
      <bpmn:invoke name="S10Created_65response_validate" operation="validate" outputVariable="mediationResponse" inputVariables="sf1" partnerLink="mediation"/>
      <bpmn:assign name="G1E_Organiser_zone_accidentRequest_validate" validate="no"/>
      <bpmn:invoke name="G1E_Organiser_zone_accidentRequest_validate" operation="validate" outputVariable="Organiser_zone_accidentResponse" inputVariables="sf10Response" partnerLink="mediation"/>
      <bpmn:assign name="G1E_Organiser_zone_accidentResponse_validate" validate="no"/>
      <bpmn:invoke name="G1E_Organiser_zone_accidentResponse_validate" operation="validate" outputVariable="endServiceResponse" inputVariable="sf10Response" partnerLink="mediation"/>
    </bpmn:sequence>
    <bpmn:sequence name="S10Created_77">
      <bpmn:assign name="S10Created_77request_validate" validate="no"/>
      <bpmn:invoke name="S10Created_77request_validate" operation="validate" outputVariable="mediationResponse" inputVariables="sf1" partnerLink="mediation"/>
      <bpmn:assign name="S10Created_77response_validate" validate="no"/>
      <bpmn:invoke name="S10Created_77response_validate" operation="validate" outputVariable="mediationResponse" inputVariables="sf1" partnerLink="mediation"/>
      <bpmn:assign name="S10Created_77request_validate" validate="no"/>
      <bpmn:invoke name="S10Created_77request_validate" operation="validate" outputVariable="mediationResponse" inputVariables="sf1" partnerLink="mediation"/>
      <bpmn:assign name="S10Created_77response_validate" validate="no"/>
      <bpmn:invoke name="S10Created_77response_validate" operation="validate" outputVariable="mediationResponse" inputVariables="sf1" partnerLink="mediation"/>
    </bpmn:sequence>
    <bpmn:sequence name="S10Created_62">
      <bpmn:assign name="S10Created_62request_validate" validate="no"/>
      <bpmn:invoke name="S10Created_62request_validate" operation="validate" outputVariable="mediationResponse" inputVariables="sf1" partnerLink="mediation"/>
      <bpmn:assign name="S10Created_62response_validate" validate="no"/>
      <bpmn:invoke name="S10Created_62response_validate" operation="validate" outputVariable="mediationResponse" inputVariables="sf1" partnerLink="mediation"/>
    </bpmn:sequence>
    <bpmn:sequence name="S10Created_88">
      <bpmn:assign name="S10Created_88request_validate" validate="no"/>
      <bpmn:invoke name="S10Created_88request_validate" operation="validate" outputVariable="mediationResponse" inputVariables="sf1" partnerLink="mediation"/>
      <bpmn:assign name="S10Created_88response_validate" validate="no"/>
      <bpmn:invoke name="S10Created_88response_validate" operation="validate" outputVariable="mediationResponse" inputVariables="sf1" partnerLink="mediation"/>
    </bpmn:sequence>
  </bpmn:flow>
</bpmn:process>

```

Figure V- 15 : fichier BPEL correspondant au processus de réponse à t0

Ce fichier est structuré de la façon suivante :

- les imports : renseigne les fichiers des services à récupérer,
- la définition des partnerLinks,
- la définition des variables,
- la définition de l'ordre d'invocation des services, on y retrouve les différentes séquences :
  - la séquence commençant par le service de la gendarmerie « organisation de la zone d'accident ». Cette séquence contenant deux sous séquences :
    - la séquence du service « Eteindre incendie de faible ampleur »,
    - la séquence du service « soigner les victimes ».
  - la séquence correspondant à la mise en place de la déviation routière,
  - la séquence correspondant à la détermination du produit contenu dans la citerne du camion.

Ce fichier BPEL, associé aux fichiers descriptifs des WebServices des partenaires (NB : ces WebServices ont été développés manuellement sous la forme d'interfaces correspondant à l'invocation des services métiers identifiés dans le workflow), peut ensuite subir les transformations finales permettant d'obtenir l'ESB configuré et déployé.

## II. Mise en évidence de l'agilité au travers d'évolutions

### II.1 Variation de la composition de la cellule à t1.

Nous sommes à t1, le système de systèmes de réponse basé sur le processus collaboratif déduit à t0 est en place et le processus de réponse est en cours d'exécution.

Au niveau des services, les services *finis* sont :

- organiser la zone d'accident,

Les services *en cours* sont :

- adapter le dispositif d'intervention,
- définir la déviation routière.
- déterminer la nature du produit.

Les autres services sont *en attente*.



C'est à ce moment t1 que le SAMU se déclare opérationnel pour participer à la réponse à la crise. La composition de la cellule de gestion de crise a donc évolué. Cependant, la cellule de gestion de crise décide d'attendre le retour du SDIS concernant le service « *soigner les victimes* » avant de modifier la réponse actuellement mise en place. Cette première évolution est principalement destinée à montrer qu'une évolution de la crise et/ou du système de réponse à la crise peut être ignorée (toujours en s'appuyant sur le fait que l'agilité du système permettra d'y revenir à loisirs).

## II.2 Evolution de la situation à t2.

### II.2.1 Etat de la réponse à t2

Nous sommes à t2, le processus de réponse se poursuit. Les figures V-16, V-17 et V-18 représentent les différentes parties de la fiche de suivi et par conséquent l'état de la réponse.

Service	Present_reponse	Ordre	Selectionne	utilise_pour	Etat	Acteur	Commentaire
GIE_Organiser_zone_accident	oui	1	deduce	SDIS_Soigner_victimes	Fini	GIE	
SDIS_Adapter_dispositif_intervention_SDIS	oui	3	deduce	SDIS_Soigner_victimes	EnCours	SDIS	
SDIS_Soigner_victimes	oui	5	advisable	personnes_blessees	EnAttente	SDIS	
SDIS_Eteindre_incendie_faible_ampleur	oui	5	advisable	risque_incendie	EnAttente	SDIS	
DDE_Determiner_nature_produit	oui	1	advisable	risque_contamination	EnCours	DDE	
GIE_Definir_deviation_routiere	oui	1	deduce	DDE_Mettre_en_place_...	Fini	GIE	
DDE_Valider_deviation_routiere	oui	3	deduce	DDE_Mettre_en_place_...	EnCours	DDE	
GIE_Mettre_en_place_deviation_routiere	oui	5	advisable	engorgement	EnAttente	GIE	

Figure V- 16 : état des services à t2

Nom	Etat_choisi	Priorite	Etat_deduit	Etat_reel	Commentaire
risque_ince...okcan		0	Nontraite	Present	
risque_bies...okcoud		6	Nontraite	Present	
risque_cont...okcoud		3	Nontraite	Present	
risque_inter...ko		7	Nontraite	Present	
risque_eng...okcan		8	Nontraite	Present	
risque_fute...ko		9	Nontraite	Present	

Nom	Etat_choisi	Priorite	Etat_deduit	Etat_reel	Commentaire
personnes_b...okcoud		1	Nontraite	Present	
engorgement...okcan		4	Nontraite	Present	
interruption_tr...okcoud		5	Nontraite	Present	
incendie_res...okcan		2	Nontraite	Present	

Figure V- 17 : état des problèmes à t2

Acteur	NbServices	Etat	Commentaire
GIE	3	EnAttente	
SDIS	3	EnCours	
DDE	2	EnCours	

Figure V- 18 : état des acteurs de la réponse à t2

Au niveau des services, les services *finis* sont :

- organiser la zone d'accident,
- définir la déviation routière.

Les services *en cours* sont :

- adapter le dispositif d'intervention,
- valider la déviation routière,

- déterminer la nature du produit.

Les autres services sont *en attente*.

## II.2.2 Evolution de la situation de crise

A t2, le moteur du camion explose provoquant une aggravation de l'incendie et les blessures par brûlures de personnes présentes sur les lieux (il est à noter que cette information remonte à la cellule de crise par le biais des moyens de communications internes aux partenaires ou du service *EarlyWarning* évoqué précédemment et localisé hors du périmètre de ces travaux). La réponse à la crise n'étant plus adaptée à la situation du fait de cette sur-crise, il est nécessaire de créer un nouveau processus de réponse.

Cette évolution de la situation impose la modification du modèle de crise. La nouvelle version du modèle de crise, figure V-19, est composé d'une part des éléments du précédent modèle de crise puisqu'aucun problème n'avait encore été traité et d'autre part, des nouveaux éléments comme la conséquence de « *personne brûlées* ».

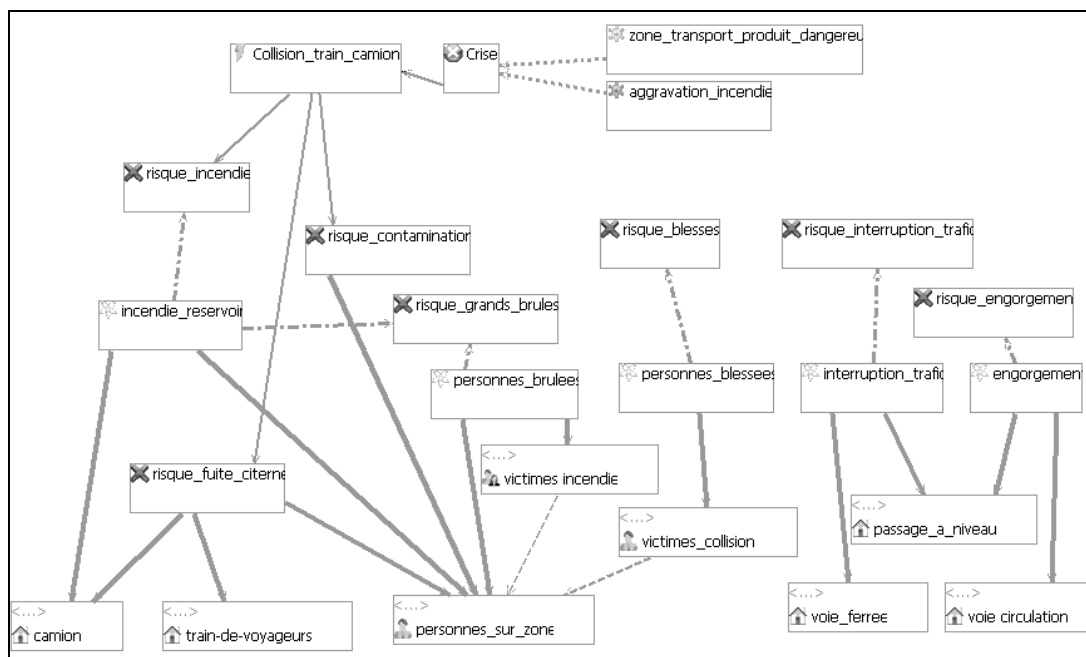


Figure V- 19 : modèle de crise à t2

A partir de ce nouveau modèle de crise, la chaîne de déduction du processus est relancée. Concernant la définition des liens de correspondance sémantique, une seule modification est apportée. La conséquence « *incendie moteur* » est maintenant équivalente à un « *incendie de grande ampleur* » et proche d'un « *incendie de faible ampleur* ».

L'ordre de priorité de traitement des problèmes est présenté ci-dessous avec pour chaque problème l'ensemble des services sélectionnés et en parenthèse l'acteur responsable du service:

1. incendie moteur,
  - éteindre incendie de grande ampleur (SDIS),
  - refroidir un objet (SDIS),
2. personnes brûlées,
  - acheminer les victimes (SAMU),

3. risque de contamination,
  - déterminer nature du produit (DDE),
  - adapter le périmètre de sécurité (GIE),
4. personne blessées,
  - soigner blessées (SAMU).

On peut remarquer que lors de ces choix, la gestion des personnes blessées ou brûlées a été confiée au SAMU plutôt qu'au SDIS. A partir de ces choix un nouveau processus collaboratif de réponse à la crise a été déduit, figure V-20.

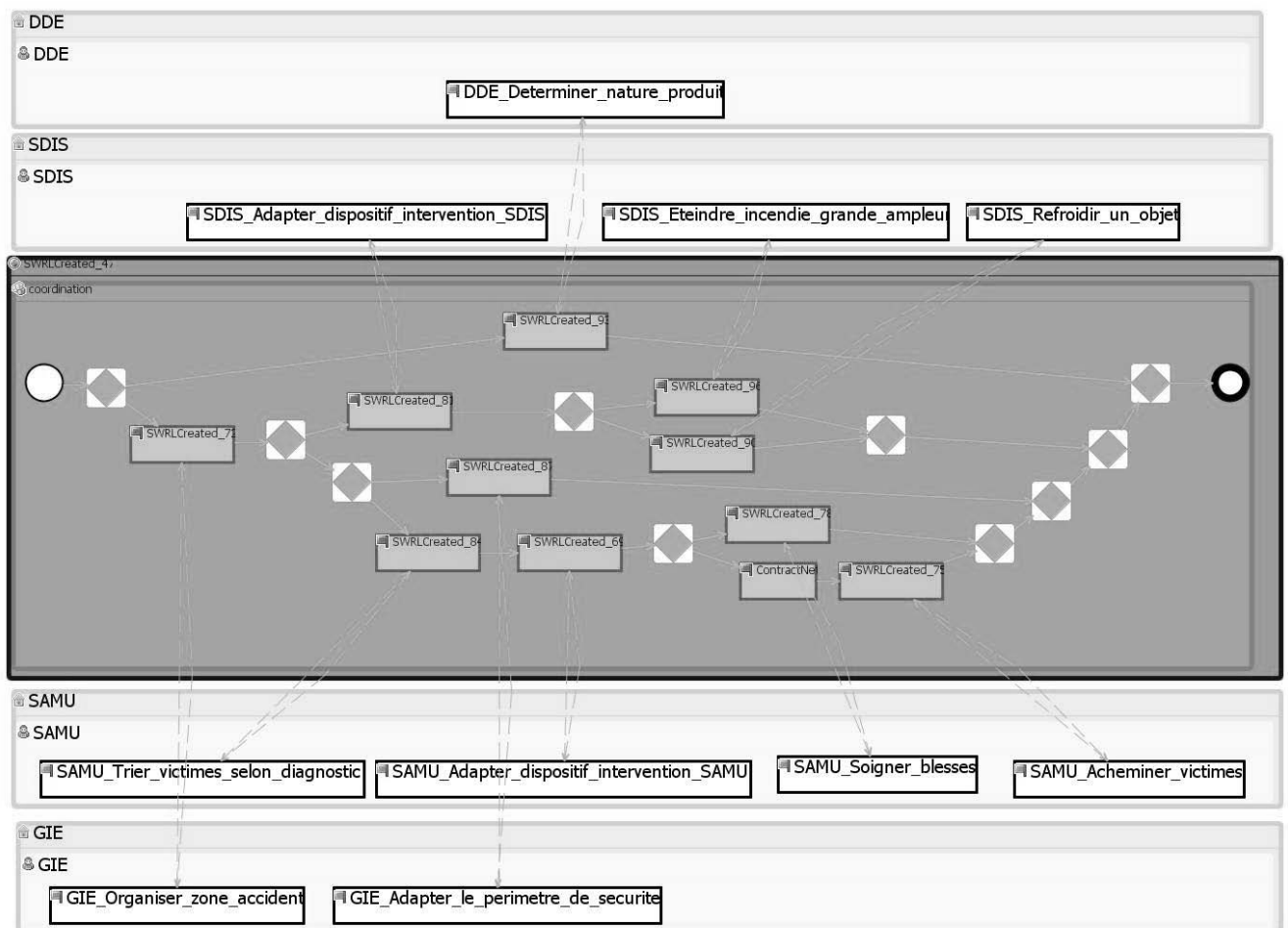


Figure V- 20 : processus de réponse à t2

Le système de systèmes de réponse à la crise est alors déployé et aucune autre évolution n'apparaîtra avant la fin de la réponse à la crise et un retour à une situation normale.

### III. Conclusion

Ce chapitre illustratif a pour objectif de montrer le point de vue utilisateur de la démarche de conception d'un système d'information de médiation. A travers la définition de la réponse de crise de cet

exemple, il a été montré que les interventions humaines, notamment les choix, ne se situent qu'au niveau métier de la démarche, les autres étapes étant totalement transparentes pour les utilisateurs.

Ce constat est un atout primordial de nos travaux. En effet, les interventions humaines ne servent qu'à définir les caractéristiques de la collaboration et ne concernent en rien la résolution des problèmes liés à la réalisation technique de l'outil. Ainsi la cellule de gestion de crise peut se concentrer sur son cœur de métier à savoir la définition, la mise en place et le suivi d'une réponse à une crise.



# Conclusion générale

De nos jours, les collaborations entre différents partenaires sont éphémères. Les partenaires collaborent essentiellement de manière opportuniste sur un laps de temps très court. Cette caractéristique de la collaboration moderne est due, pour les entreprises, à l'évolution des marchés mondiaux, qui fait de cette dynamique collaborative un facteur de compétitivité, mais elle relève aussi, dans certains cas tels que le domaine de la gestion de crise, d'un contexte incontournable au sein duquel réside nativement cette exigence de dynamisme et d'efficacité. Ce contexte correspond au cadre général de nos travaux, qui porte sur l'étude d'une approche d'interopérabilité par médiation dans le cadre d'une dynamique de collaboration appliquée à la gestion de crise.

Ce contexte nous a amené à nous intéresser à la notion d'interopérabilité, afin de répondre au questionnement suivant : quelle solution conceptuelle peut répondre au besoin d'entreprises de collaborer rapidement tout en gardant leur identité propre ? Et comment concevoir ce type de solutions ?

Notre réponse à la première question concerne la définition d'un système de systèmes à partir des informations relatives à la collaboration. Ce système de systèmes est bâti autour d'un médiateur et des différents partenaires, vus comme des entités indépendantes. Ce médiateur a pour objectif d'assurer la cohérence des actions des partenaires, en orchestrant leur comportement collaboratif, et de gérer l'hétérogénéité aussi bien au niveau des informations que des actions menées.

Afin de répondre à la deuxième question, nous nous sommes basés sur le fait qu'il existe une forte dépendance entre chaque partenaire et son système d'information (SI). La conception d'un système de systèmes peut alors être réalisé au travers d'un système de systèmes d'information composé de l'ensemble des SI des partenaires, ainsi que d'un système d'information de médiation (SIM), image du médiateur. De l'ensemble des éléments constituant le système de systèmes, seule la définition du SIM est dépendante des informations de la collaboration. Par conséquent, si à partir des informations de la collaboration il est possible de générer le SIM, alors le système de systèmes d'information sera généré.

La génération d'un SIM à partir des informations de la collaboration est l'objectif du projet MISE dans le domaine industriel, aussi bien celui du projet ISyCri dans un contexte de gestion de crise. Cette génération repose sur une approche dirigée par les modèles, consistant en une plongée en abstraction du niveau métier au niveau technique en passant par le niveau logique.

Dans cette thèse, plusieurs travaux ont été réalisés et notamment une démarche d'aide à la conception d'une réponse à la crise à partir des informations de la situation. Cette démarche repose sur une base de connaissance que nous faisons évoluer en fonction des partenaires disponibles et des informations de la crise. Afin de récolter ces informations, nous utilisons des modèles réalisés soit pendant la phase de préparation par un acteur, pour les modèles de services, soit au moment de la réponse à la crise par la cellule de gestion de crise. Deux éditeurs de modèles (DSL) ont été réalisés afin de permettre aux utilisateurs de créer graphiquement ces modèles, tout en étant certains de respecter le métamodèle de crise. Une fois la connaissance contenue dans les modèles injectée dans la base de connaissance, une première exécution de règles de déduction permet d'obtenir pour chaque problème de la crise une liste de l'ensemble des services permettant leur résolution. Ces listes sont alors proposées à la cellule de gestion de crise, afin de les aider à choisir le ou les bons services à utiliser. Une fois ces choix réalisés, une deuxième exécution de règle de déduction permet de construire un processus de réponse à la crise, respectant le métamodèle de processus collaboratif. Le modèle de processus collaboratif ainsi obtenu est alors proposé à la cellule de gestion de crise (au moyen d'un éditeur réalisé spécifiquement).

Une autre partie de ces travaux de thèse consistait à réaliser une transformation de modèle au niveau technique de la démarche afin de créer un fichier BPEL représentatif du workflow collaboratif à orchestrer. Ce fichier est nécessaire à la configuration de l'ESB PEtALS, cible technologique de ces

travaux. Pour être plus précis, ce fichier est utilisé par le moteur d'orchestration afin de connaître l'ordre d'invocation des différents services.

Enfin, la dernière partie de ces travaux de thèse s'intéressait à l'apport d'une faculté d'agilité au système de systèmes d'information et par conséquent au système de systèmes. Une étude de la notion d'agilité nous a menés à l'idée de concevoir une démarche de conception de SIM flexible. Afin d'atteindre cette flexibilité, la démarche de conception a été décomposée et chaque étape a donné naissance à un service qui peut être invoqué de façon indépendante des autres. Cette démarche de conception a donc été intégrée à l'ESB PEtALS, grâce à la transformation des étapes de la démarche de conception en WebServices. Un objectif sous-tendu par cette approche concerne l'abolition du clivage entre *Design-Time* et *Run-Time*.

Cet apport de flexibilité permet de réagir à l'apparition d'une évolution, mais ne permet pas de les détecter. Il a donc été également nécessaire de proposer une aide à la détection des variations. Cette aide est basée sur une fiche de suivi réalisée également au cours de cette thèse. Cette fiche de suivi repose sur la comparaison entre les informations venant de la situation et celles déduites automatiquement de l'avancement de la réponse à la crise.

La figure ci-dessous positionne l'ensemble des travaux réalisés dans le cadre de MISE 1.0.

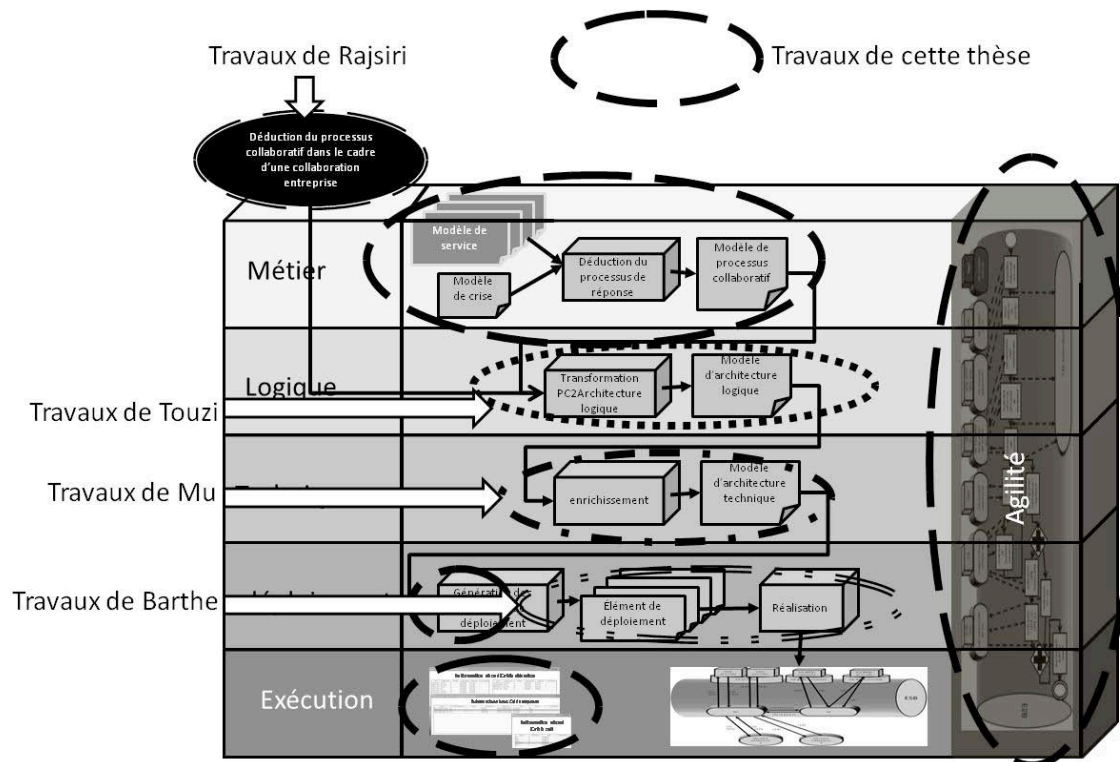


Figure 1 : travaux menant aux résultats de MISE 1.0

A la fin de cette thèse, une version 1.0 de la démarche de conception d'un système d'information de médiation est opérationnelle et disponible à travers le prototype réalisé.

Cependant, cette version comporte des limites, notamment sa dépendance envers le SOA. En effet, notre approche repose sur une vision orientée services et il est à ce titre indispensable que les partenaires de la collaboration soient en mesure de présenter leur système d'information selon cette même vision vis-à-vis de l'extérieur. Ce point constitue une limite de nos travaux, car il est contraire à notre point de vue d'approche fédérée (dans la mesure où il peut imposer une remise en question des systèmes des partenaires). Néanmoins, dans le cadre des niveaux de maturité collaborative qui ont été évoqués dans ce document, nous souhaitons rappeler le fait que les niveaux ouvert et fédéré exigent que les partenaires



puissent partager leurs fonctionnalités et services. On peut donc se demander si les types de technologies, tels que SOA, ne sont pas des pré-requis à ces niveaux de maturité collaborative.

Une autre limite concerne l'utilisation d'un seul processus pour caractériser la réponse à la crise. Ce processus ne décrit que les activités opérationnelles de la réponse à la crise (au milieu desquelles se trouvent « noyées » quelques considérations plus stratégiques et/ou logistiques), sans se soucier réellement de la question des processus décisionnels et des processus support au sens de la norme ISO (ISO, 2000). Il serait préférable de déduire des informations de la collaboration une cartographie de processus reprenant les niveaux décisionnel, opérationnel et support de la réponse à la crise.

Au sein de ces travaux, la gestion de l'information (acheminement et surtout traduction) est insuffisante, puisqu'elle ne permet que de faire des affectations de valeurs de type « un à un » (*i.e.* de copier-coller des valeurs sans pouvoir les transformer). De la même façon, la correspondance entre activités métier et services techniques, enjeu ô combien crucial, est solutionnée par la capitalisation *a priori* des compétences des acteurs et par les outils de correspondance sémantique (dédié au contexte de crise). La question de la réconciliation des données et des services ne trouve dans ces travaux qu'une solution spécifique et parcellaire. Néanmoins, l'étude de faisabilité de la gestion de l'information au cours de la réponse, en se basant sur le service de médiation ainsi que la création de règles au sein du BPEL, a été menée à bien. En effet, l'information est transférée au bon format, au bon moment, au bon partenaire. Afin d'obtenir une gestion de l'information plus satisfaisante, il serait intéressant de rajouter une sur couche permettant de définir de façon dynamique les transformations à exécuter.

Enfin, l'étude de l'agilité réalisée dans cette thèse constitue une première ébauche de solution, mais d'autres pistes restent encore à explorer, telle que l'intégration du choix et de la conception retardés au sein de la démarche de conception du SIM. Une autre consisterait à faire évoluer de façon dynamique l'ensemble des modèles utilisés pendant la démarche de conception en fonction de l'avancement de la réponse et des informations du terrain. Ceci permettrait de proposer à tout moment un SIM plus adapté à la situation.

La version 2.0 du projet MISE ambitionne de proposer une démarche de conception de SIM permettant de répondre à un certain nombre des limites précédentes. Cette version est articulée autour de quatre thèses en cours :

La première thèse, réalisée par W. Mu, vise à réaliser la partie abstraite du SIM, à savoir sa définition au niveau métier et logique. Un des buts de cette thèse est de pallier le problème de l'utilisation d'un unique processus collaboratif pour décrire la collaboration. Pour ce faire, à partir des informations de la collaboration, W. Mu projette de construire une cartographie des processus (basée sur les trois niveaux de processus, à savoir décisionnel, opérationnel et support). Cette cartographie vise également à être enrichie sémantiquement, afin de permettre une meilleure prise en charge de l'identification des données et des services (en particulier pour la gestion de l'information et l'affectation des services).

La deuxième thèse, réalisée par N. Boissel-Dallier, ambitionne quant à elle de réaliser concrètement le support de la collaboration en se basant sur les résultats de la partie abstraite (thèse de W. Mu). Pour ce faire, une étude est menée pour définir une approche de réconciliation sémantique dédiée à établir les correspondances entre informations (métiers) et données (techniques) d'une part, et entre activités (métiers) et services (techniques) d'autre part, sur la base des compléments sémantiques provenant des modèles abstraits (thèse de W. Mu) et des annotations sémantiques apportées aux couches technologiques (SA-WSDL, WSMO, etc.). En outre, la question de la réalisation automatisée de services techniques pour implémenter d'éventuelles activités métiers trop abstraites (cf. les compétences des pompiers qui sont peu

implémentables, mais que l'on peut invoquer au travers d'interfaces), pourra être étudiée au cours de ces travaux (BPEL4PEOPLE (OASIS, 2009)).

Les deux dernières thèses, quant à elles, ambitionnent, pour la première (A.-M. Barthe), d'appliquer les résultats des deux premières thèses au domaine de la gestion de crise et pour la seconde (S. Zribi), de se positionner transversalement en adressant des aspects non fonctionnels, tels que la gouvernance, la qualité de service, la sécurité, etc.

La figure suivante positionne sur la démarche MISE 1.0 en double « Y » les niveaux abstrait et concret correspondant à la vision MISE 2.0 (encouragée par l'émergence de BPMN 2.0).

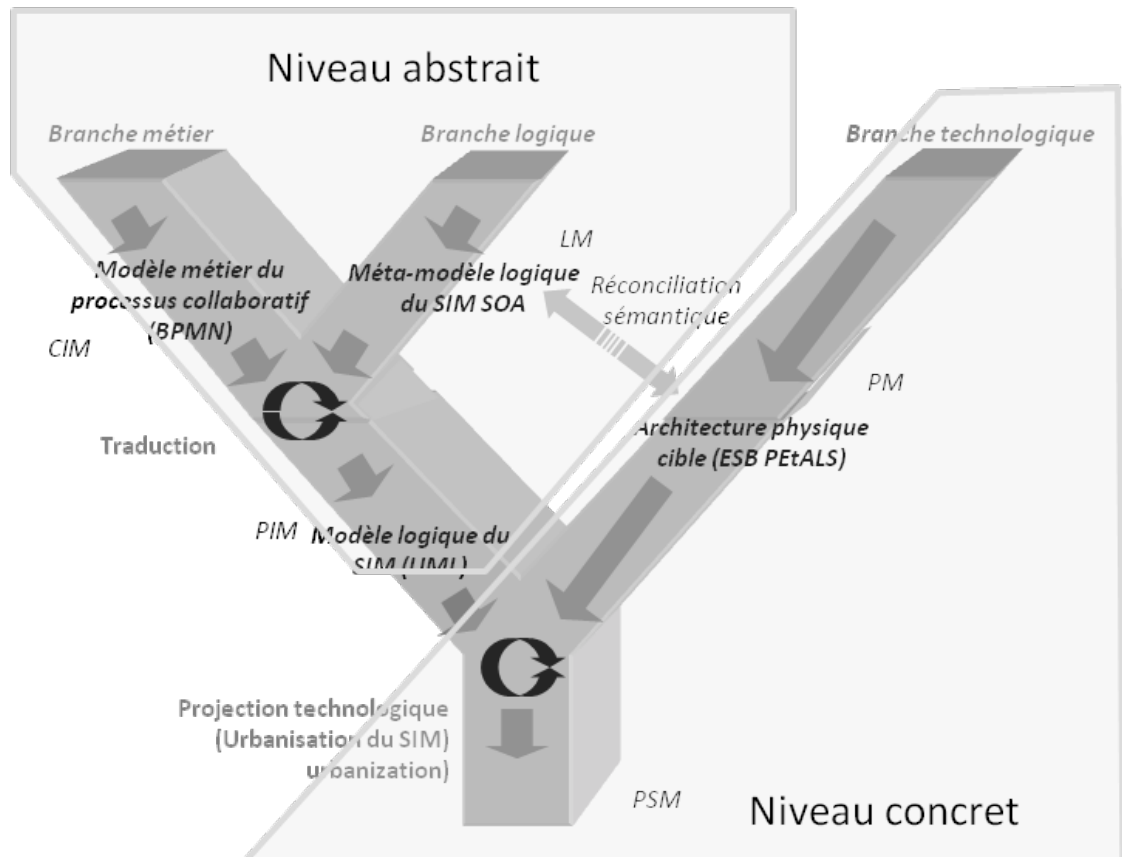


Figure 2: le cadre MISE 2.0 projeté sur le cadre MISE 1.0



# Glossaire :

ATL	<i>Atlas Transformation Language</i>
BC	<i>Binding Component</i>
BPEL	<i>Business Process Execution Language</i>
BPM	<i>Business Process Management</i>
BPMN	<i>Business Process Modelling Notation</i>
CIM	<i>Computer Independent Platform</i>
DDE	<i>Direction départementale de l'Équipement</i>
DSL	<i>Domain Specific Language</i>
EAI	<i>Enterprise Application Integration</i>
EDI	<i>Electronic Data Integration</i>
EIF	<i>Enterprise Interoperability Framework</i>
ESB	<i>Enterprise Service Bus</i>
GIE	<i>Gendarmerie</i>
GMF	<i>Graphical Modelling Framework</i>
IDM	<i>Ingénierie dirigée par les modèles</i>
ISyCri	<i>Interopérabilité des Systèmes en situation de Crise</i>
JB1	<i>Java Business Integration</i>
JSR	<i>Java Specification Request</i>
MDA	<i>Model Driven Approach</i>
MDE	<i>Model Driven Engineering</i>
MDI	<i>Model Driven Interoperability</i>
MISE	<i>Mediation Information System Engineering</i>
NRBC	<i>Nucléaire, Radiologique, Bactériologique, Chimique</i>

---

OMG	<i>Object Management Group</i>
OWL	<i>Web Ontology Language</i>
PIM	<i>Platform Independent Model</i>
PIM4SOA	<i>Platform Independent Model For Service-Oriented Architecture</i>
PSM	<i>Platform Specific Model</i>
SA	<i>Service Assembly</i>
SA-BPEL	<i>Service Assembly en lien avec le moteur d'orchestration BPEL</i>
SE	<i>Service Engine</i>
SDIS	<i>Services Départementaux: d'Incendie et de Secours</i>
SE-BPEL	<i>Service Engine correspondant au moteur d'orchestration BPEL</i>
SI	<i>Système d'information</i>
SIM	<i>Système d'Information de Médiation</i>
SOA	<i>Service Oriented Architecture</i>
SOAP	<i>Simple Object Access Protocol</i>
SoS	<i>Système de Systèmes</i>
SoSI	<i>Système de Systèmes d'Information</i>
SU	<i>Service Unit</i>
SU-BPEL	<i>Service Unit utilisé par le moteur d'orchestration BPEL</i>
SWRL	<i>Semantic Web Rule Language</i>
UML	<i>Unified Modelling Language</i>
UMLT	<i>Métamodèle d'architecture technique du SIM</i>
WSDL	<i>Web Service Description Language</i>
XML	<i>Extensible Markup Language</i>
XSLT	<i>XML Transformation</i>
WS	<i>WebService</i>

# Références Bibliographiques

- (Akehust *et al.*, 2002) Akehust, D.H., Kent, S. : *A relational approach to defining transformations in a metamodel*, Proceedings of the 5th International Conference on the UML 2002, Vol. 2460, pp.243-258, 2002.
- (Aligne *et al.*, 2009) Aligne, F.: *Which Information and Decision Support System for the Crisis Management?*, Information Systems and Technology Panel (IST) Symposium. Bucharest, Romania., 2009.
- (Aligne *et al.*, 2010) Aligne, F. et Savéant, P.: *Gestion de crise : optimisation de la mise en oeuvre des plans de secours*, WISG, 2010.
- (Altay *et al.*, 2005) Altay, N. et Green III, W. G.: *OR/MS research in disaster operations management*, European Journal of Operational Research, 175(1), p.475–493, 2005.
- (Andonoff *et al.*, 2007) Andonoff, E., Bouaziz, B. et Hanachi, C.: *Protocol Management Systems as a Middleware for Inter-Organizational Workflow Coordination*, International Journal of Computer Science & Application, IV, p.23-41, 2007.
- (Andrews *et al.*, 2003) Andrews, T., Curbera, F., Dholakia, H., Golland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S. et others: *Business process execution language for web services*, Citeseer., 2003.
- (apfa, 2004) Association pour Promouvoir le Français des Affaires, [www.pressefrancophone.org](http://www.pressefrancophone.org), 2004.
- (Archimate, 2007) Archimate, *Deliverables of Archimate Project*, <https://doc.telin.nl/dscgi/ds.py/Index/Collection-4483>, 2007.
- (Aubert *et al.*, 2002) Aubert, B., Dussart, A. : *Systèmes d'Information Inter-Organisationnels*, Rapport Bourgogne, CIRANO,2002.
- (Badot, 1998) Badot, O.: *Théorie de l'entreprise agile*, Harmattan., 1998.
- (Bakir, 2003) Bakir, S.: *Contribution à une démarche d'intégration des processus de gestion des risques et des projets : étude de la fonction planification*, Thèse de doctorat, Institut National Polytechnique de Toulouse., 2003.
- (Bataille *et al.*, 2001) Bataille, V., Castellani, X. : *Métamodélisation et ingénierie des systèmes d'information*, Ingénierie des systèmes d'information, Hermes, pp.149 – 174, 2001.
- (Bauer *et al.*, 2006) Bauer, B., Müller, J.P., Roser, S. : *A Decentralized Broker Architecture for Collaborative Business Process Modelling and Enactement*, Enterprise Interoperability: New Challenges and Approaches- Springer Verlag, 2006.

- (Beamon *et al.*, 2004) Beamon, B. M. et Kotleba, S. A.: *Inventory modelling for complex emergencies in humanitarian relief operations*, International Journal of Logistics Research and Applications, 9(1), p. 1–18, 2004.
- (Bénaben *et al.*, 2007) Bénaben, F., Touzi, J., Rajsiri, V. et Pingaud, H.: *L'Interopérabilité des systèmes d'information comme moyen vers l'intégration de l'écosystème industriel*, 7e Congrès international de génie industriel, Trois-Rivières, Québec., 2007.
- (Bénaben *et al.*, 2008) Bénaben, F., Touzi, J., Rajsiri, V., Truptil, S., Lorré, J. P. et Pingaud, H.: *Mediation Information System Design in a Collaborative SOA Context through a MDD Approach*, First International Workshop on Model Driven Interoperability for Sustainable Information Systems (MDISIS'08) held in conjunction with the CAiSE'08 Conference, 2008.
- (Bénaben *et al.*, 2010) Bénaben, F., Mu, W., Truptil, S., Lorré, J. P. et Pingaud, H.: *Information Systems design for emerging ecosystems : A model-driven approach diving into abstraction layers*, IEEE-DEST'10, Springer-Verlag, Dubai, EAU., 2010.
- (Bénaben *et al.*, 2010) Bénaben, F., Boissel-Dallier, N., Lorré, J. P. et Pingaud, H.: *Semantic Reconciliation in Interoperability Management through Model-Driven Approach*, PRO-VE'10, 2010.
- (Benguria *et al.*, 2006) Benguria, G., Larrucea, X., Elvesaeter, B., Neple, T., Beardsmore, A. et Friess, M.: *A platform independent model for service oriented architectures*, I-ESA'06, Bordeaux., 2006.
- (Berners-Lee *et al.*, 2001) Berners-Lee T., Hendler J., Lassila O. *The Semantic Web*. Scientific American, May, 2001.
- (Bernus *et al.*, 1996) Bernus, P., Nemes, L. et Williams, T. J.: *Architectures for enterprise integration*, Springer., 1996.
- (Berre *et al.*, 2007) Berre, A.-J., Elveaster, B., Figay, N., Guglielmina, C., Johnsen, S. G., Karlsen, D., Knothe, T., Lippe, S. : *The ATHENA Interoperability Framework*, Enterprise interoperability: New challenges and approaches II, Springer, 2007.
- (Bézivin *et al.*, 2008) Bézivin, J., Blay, M., Bouzhegoub, M., Estublier, J., Favre, J.M., Gérard, S., Jézéquel, J.M. : *Rapport de synthèse de l'AS CNRS sur le MDA ( Model Driven Architecture)*, 16 p., 2004.
- (Bézivin *et al.*, 2009) Bézivin, J. et Jouault, F.: *Acquis et Défis de l'Ingénierie Dirigée par les Modèles*, Génie logiciel, (90), p.55–60, 2009.
- (Bézivin, 2004) Bézivin, J. : *Sur les principes de base de l'ingénierie des modèles*, RSTI –L'objet – 10/2004, *Où en sont les objets ?*, p. 124 - 156, 2004.

- (Boissel-Dallier *et al.*, 2009) Boissel-Dallier, N., Lorré, J. P. et Benaben, F.: *Management Tool for Semantic Annotations in WSDL*, On the Move to Meaningful Internet Systems: OTM 2009 Workshops, p. 898–906., 2009.
- (Booch *et al.*, 2004) Booch, G., Jacobson, I., Rumbaugh, J. : *UML 2.0 Guide de référence*, Paris, CampusPress, 2004.
- (Bourey *et al.*, 2007) Bourey, J. P., Grangel, R., Doumeingts, G. et Berre, A. J.: *Report on model driven interoperability*, Deliverable DTG2, 3, 2007.
- (Bouzguenda, 2006) Bouzguenda, L.: *Coordination Multi-Agents pour le Workflow Inter-Organisationnel Lâche*, Thèse de doctorat, Université des Sciences Sociales, Toulouse., 2006.
- (BPML, 2007) BPML, *BPML - Business Process Modeling Language*, [www.bpmi.org](http://www.bpmi.org).
- (Caplat, 2003) Caplat, G., Sourrouille, J.L. : *Considerations about model mapping*, Wisme., 2003.
- (Chapron, 2006) Chapron, J. : *L'urbanisation organisationnelle : méthodes et aides à la décision pour piloter l'évolution du système d'information de l'entreprise*, Thèse de doctorat, Ecole Nationale Supérieure des Mines de Saint Etienne & Université Jean Monnet de Saint Etienne, 2006.
- (Chen *et al.*, 2003) Chen, D. et Doumeingts, G.: *European initiatives to develop interoperability of enterprise applications—basic concepts, framework and roadmap*, Annual Reviews in Control, 27(2), p. 153–162, 2003.
- (Chen *et al.*, 2006) Chen, D., Dassisti, M., Elvaester, B.: *Interoperability Knowledge Corpus, Intermediate Report*. Deliverable DI.1b, Network of Excellence InterOp, Contract No.IST-508011, 2006.
- (Chen *et al.*, 2008) Chen, D., Doumeingts, G. et Vernadat, F.: *Architectures for enterprise integration and interoperability: Past, present and future*, Computers in industry, 59(7), p.647–659, 2008.
- (Combemale, 2008) Combemale, B.: *Ingénierie Dirigée par les Modèles (IDM) -- État de l'art*, <http://hal.archives-ouvertes.fr/hal-00371565/>, 2008.
- (D'Antonio, 2005) D'Antonio, F. : *Deliverable MoMo.2 TG MoMo Roadmap*, Rapport INTEROP (Interoperability Research for Network Enterprises Applications and Software), 2005.
- (Daclin *et al.*, 2009) Daclin, N. et Chapurlat, V.: *An Anticipative Effects-Driven Approach for Analyzing Interoperability in Collaborative Processes*, Leveraging Knowledge for Innovation in Collaborative Networks, p.441–448, 2009.
- (Dautun, 2004) Dautun, C.: *Organisation de la gestion de crise «Les secours face à une crise de grande ampleur»*, Mémoire de Mastère Sécurité industrielle et environnement. Alès: École des Mines d'Alès., 2004.



- (Deguil, 2008) Deguil, R.: *Mapping entre un référentiel d'exigences et un modèle de maturité: application à l'industrie pharmaceutique*, Thèse de doctorat, Institut National Polytechnique, Toulouse, 2008.
- (Devlin, 2006) Devlin, E.: *Crisis management planning and execution*, Auerbach Publications, Boca Raton FL., 2006.
- (DoD, 2001) Department of Defense. *Department of defense Dictionary of Military and Associated Terms*, United States of America Department of Defense, 2001.
- (EBM, 2005) EBM Websourcing, *Nouvelles technologies pour l'intégration : les ESB*, White Paper, 2005
- (Eclipse, 2007) Eclipse Foundation, *Eclipse - an open development platform*, [www.eclipse.org](http://www.eclipse.org).
- (Ecole militaire, 2007) Ecole militaire: *Faire Face aux situations d'exception ; Scénario D : Attentat terroriste dans le métro*, Commission armées-jeunesse, 2007.
- (Faure et al., 2010) Faure, C., Andonoff, E., Hanachi, C., Sibertin-Blanc, C. et Salatge, N.: *Flexibilité de processus de gestion de crise par intégration de protocoles d'interaction*, Ingénierie des systèmes d'information, 15(3), 2010.
- (Fowler, 2004) Fowler, M. : *UML 2.0 : initiation aux aspects essentiels de la notation*, CampusPress, 2004.
- (Geram, 1997) GERAM, *Generalised Enterprise Reference Architecture and Methodology*, Task Force on Enterprise Integration, 1997.
- (GMF, 2005) GMF: Eclipse Graphical Modelling Framework, <http://www.eclipse.org/gmf/>, 2005.
- (Gourc, 2006) Gourc, D.: *Vers un modèle général du risque pour le pilotage et la conduite des activités de biens et de services : Propositions pour une conduite des projets et une gestion des risques intégrées*, Thèse de HDR, Institut national polytechnique de Toulouse., 2006.
- (Grimm et al., 2007) Grimm, S., Hitzler, P. et Abecker, A.: *Knowledge Representation and Ontologies : Logic, Ontologies and Semantic Web Languages, Semantic web services*, Semantic web services, 2007.
- (Gruber, 1993) Gruber TR. *A translation approach to portable ontologies*, Knowledge Acquisition, 5(2), p.199-220, 1993
- (Gruber, 1995) Gruber TR. *Toward principles for the design of ontologies used for knowledge sharing*, 1995.
- (Guilhous, 2009) Guilhou, X.: *Cyclone Katrina : Enseignements du retour d'expérience d'une crise "hors cadre"*, Doctrine, n°2009/01, 2009.
- (Hansen-Glize, 2008) Hansen-Glize, C.: *De la «maîtrise» des crises*, Techniques de l'ingénieur, 2008.

- (IEEE, 1990) IEEE: *IEEE Standard Computer Dictionary: Compilation of IEEE Standard Computer Glossaries*, 1990.
- (ISO 16100, 2002) ISO 16100. *Industrial automation systems and integration – Manufacturing software capability profiling for interoperability – Part 1: Framework*, ISO/TC 184/SC 5/WG 4, 2002.
- (ISO, 2000) ISO: *Norme européenne NF EN ISO 9001 version 2000, Systèmes de management de la qualité – Exigences*, AFNOR, 2000.
- (ISyCri, 2008) livrable interne du projet ISyCri portant sur le lot 2 sous la responsabilité de EBM-Websourcing, 2008
- (Jouault *et al.*, 2005) Jouault, F. et Kurtev, I.: *Transforming models with ATL*, Satellite Events at the MoDELS 2005 Conference, p. 128–138., 2006.
- (Jouault, 2006) Jouault, F. : Contribution à l'étude des langages de transformation de modèles, thèse de doctorat, Université de Nantes, 2006.
- (JSR, 2005) JSR 208: The Java Community Process(SM) Program - JSRs: *Java Specification Requests - detail JSR# 208*, 2005.
- (Kidd, 1994) Kidd, P. T.: *Agile manufacturing: forging new frontiers*, Addison-Wesley., London., 1994.
- (Kleppe, 2003) Kleppe, A., Warmer, J., Bast, W. : *MDA explained : the model driven architecture : practice and promise*, Reading, Addison-Weasley, 2003.
- (Konstantas *et al.*, 2005) Konstantas, D., Bourrières, J.-P., Léonard, M., Boudjlida, N. : Preface de : *Interoperability of Enterprise Software and Applications*, I-ESA'05, Genève, 2005.
- (Lagadec, 1992) Lagadec, P.: *La Gestion Des Crises - Outils De Réflexion À L'usage Des Décideurs*, Ediscience International., Paris., 1992.
- (Lagadec, 1995) Lagadec, P.: *Cellules de crise—Les conditions d'une conduite efficace*, Les Editions d'Organisation, Paris., 1995.
- (Lassila *et al.*, 1999) Lassila O., Swick R. *Resource description framework (RDF) model and syntax specification*, W3C Recommendation, <http://www.w3.org/TR/RECrdf-syntax/>, 1999.
- (Le Moigne, 1977) Le Moigne, J.-L. : *La théorie du système général, théorie de la modélisation*, Presses Universitaires de France, (3ème édition, 1990), 1977
- (Le Moigne, 1990) Le Moigne, J.-L. : *La modélisation des systèmes complexes*, Afcet-systèmes, Dunod, 1990.

- (Levendovskin *et al.*, 2002) Levendovskin, T., Karsai, G., Maroti, M., Ledeczi, A., Charaf, H. : *Model reuse with meta-model based transformations*, the 7th International Conference on Software Reuse : Methods, Techniques and Tools, 2002.
- (Lindberg, 1990) Lindberg, P.: *Strategic Manufacturing Management: A Proactive Approach*, International Journal of Operations & Production Management, 10(2), p.94-106, 1990.
- (Lonpégé, 2002) Lonpégé, C. : *Le projet d'urbanisation du système d'information*, Dunod, 2002.
- (Luzeaux *et al.*, 2008) Luzeaux, D. et Ruault, J. R.: *Systèmes de systèmes: concepts et illustrations pratiques*, Hermes science, 2008.
- (Maier, 1998) Maier, M. W.: *Architecting principles for systems-of-systems*, *Systems Engineering*, Systems Engineering, 1(4), p.267–284, 1998.
- (Malone *et al.*, 1999) Malone, T. W., Crowston, K., Lee, J. et Pentland, B.: *Tools for inventing organizations: Toward a handbook of organizational processes*, Management Science, 3(45), p.425-443, 1999.
- (Mc Cullen *et al.*, 2006) Mc Cullen, P., Saw, R., Christopher, M. et Towill, D.: *The F1 Supply Chain: Adapting the Car to the Circuit-The Supply Chain to the Market*, Supply Chain Forum, vol. 7, p. 14-23., 2006.
- (McGuinness *et al.*, 2000) McGuinness D., Fikes R., Rice J., Wilder S. *The Chimaera Ontology Environment*. In: *Rosenbloom*, 17th National Conference on Artificial Intelligence (AAAI'00), Austin, Texas, p. 1123–1124, 2000.
- (Mega, 2007) Mega, *Mega Suite*, www.mega.fr.
- (Miller *et al.*, 2003) Miller, J. et Mukerji, J.: *MDA Guide Version 1.1*, 2003.
- (Ministère de l'intérieur, 2008) Ministère de l'intérieur de l'outre-mer et des collectivités territoriales *Exercice de simulation d'attentats simultanés à Paris*, 2008.
- (Monateri *et al.*, 1999) Monateri, J.C., Sapina, M. : *Dynamique des relations entre entreprises, stratégies manufacturières et arrangements contractuels durables* , 3<sup>o</sup> Congrès international de génie industriel, L'intégration des ressources humaines et des technologies : le défi, 1999.
- (Monfort *et al.*, 2004) Monfort, V., Goudeau, S. : *Web services et interopérabilité des SI*, Dunod/01 Informatique, Collection InfoPro, 2004.
- (Morley *et al.*, 2002) Morley, C.: *La modélisation des processus: typologie et proposition utilisant UML*, Processus et Systèmes d'information—Journées ADELI, Paris, France, 2002.
- (Morley *et al.*, 2005) Morley, C., Hugues, J., Leblanc, B. et Hugues, O.: *Processus Métiers et SI: Evaluation, modélisation, mise en oeuvre*, Dunod, 2005.

- (Neches *et al.*, 1991) Neches R., Fikes RE., Finin T., Gruber TR., Senator T., Swartout WR. *Enabling technology for knowledge sharing*. AI Magazine 12(3), p36–56, 1991.
- (OASIS, 2006) OASIS: *Emergency Data Exchange Language*, OASIS CoverPages Report, 2006.
- (OASIS, 2007) OASIS: *Web Services Business Process Execution Language*, <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>, 2007.
- (OMG, 2002) Object Management Group, *Meta Object Facility (MOF) Specification*, Version 1.4, 2002.
- (OMG, 2003) Object Management Group, *MDA Guide*, Version 1.0.1, 2003.
- (OMG, 2003) OMG, *MDA guide version 1.0.1*, document number : omg/2003-06-01 edition. Object Management Group, 2003.
- (OpenGroup, 2005) The Open Group, *TOGAF - The Open Group Architecture Framework*, Version 8, The Open Group, 2005.
- (Ouskel, 1999) Ouskel, A. M. et Sheth, A.: *Semantic interoperability in global information systems*, ACM Sigmod Record, 28(1), p.5–12, 1999.
- (Ouyang *et al.*, 2008) Ouyang, C., Dumas, M., Hofstede, A. H. et Van Der Aalst, W. M.: *Pattern-based translation of BPMN process models to BPEL web services*, International Journal of Web Services Research, 5(1), p.42–62, 2008.
- (Panetto, 2006) Panetto, H.: *Meta-Modèles et Modèles pour l'Intégration et l'Interopérabilité des Applications d'Entreprises de Production*, Thèse de HDR, Université Nancy 1, 2006.
- (Peaucelle, 1999) Peaucelle, J. L.: *Systèmes d'information: le point de vue des gestionnaires*, Economica., 1999.
- (Pera, 2007) PERA, *The Purdue Enterprise Reference Architecture*, <http://www.pera.net>.
- (Pingaud, 2003) Pingaud, H.: *Logistiques et technologies de l'information et de la communication : les guides experts*, WEKA., 2003.
- (Pingaud, 2009) Pingaud, H.: *Rationalité du développement de l'interopérabilité dans les organisations, Management des technologies organisationnelles - Journées d'étude 2009*, p. 19-30, 2009.
- (Protégé, 2000) Protégé: <http://protege.stanford.edu/>, 2000.
- (Putman, 2001) Putman, J. R. : *Architecting with RM-ODP*, Prentice Hall, 2001.
- (Rajsiri, 2009) Rajsiri, V., Lorré, J. P., Bénaben, F. et Pingaud, H.: *Knowledge-based system for collaborative process specification*, Computers in Industry, 61(2), p161–175, 2010.

- (Rajsiri, 2009) Rajsiri, V.: *Knowledge-based system for collaborative process specification*, Thèse de doctorat, Institut National Polytechnique, Toulouse., 2009.
- (Raymond ; 2007) Raymond, G.: *SOA : architecture logique, principes, structures et bonnes pratiques*, www.softteam.fr, 2007.
- (Reichert *et al.*, 1998) Reichert, M. et Dadam, P.: *ADEPT<sub>flex</sub>—supporting dynamic changes of workflows without losing control*, Journal of Intelligent Information Systems, 10(2), p.93–129, 1998.
- (Reix *et al.*, 1995) Reix, R. et Helfer, J. P.: *Systèmes d'information et management des organisations*, Vuibert Paris., 1995.
- (Reix *et al.*, 1995) Reix, R. et Helfer, J. P.: *Systèmes d'information et management des organisations*, Vuibert Paris., 1995.
- (Roques, 2004) Roques, P.: *UML 2 par la pratique*, Paris, Eyrolles, 2004.
- (Rueppel *et al.*, 2007) Rueppel, U. et Wagenknecht, A.: *Improving emergency management by formal dynamic process-modelling*, 4th Conf. on Information Technology in Construction (W78), p. 559–564., 2007.
- (Ryfman, 1999) Ryfman, P.: *La question humanitaire: histoire, problématiques, acteurs et enjeux de l'aide humanitaire internationale*, Ellipses., 1999.
- (Saadoun, 2000) Saadoun, M.: *Technologies de l'information et management*, Hermès science, 2000.
- (Salem *et al.*, 2008) Salem, R. B., Grangel, R. et Bourey, J.: *A comparison of model transformation tools: Application for Transforming GRAI Extended Actigrams into UML Activity Diagrams*, Computers in Industry, 59(7), p. 682-693, 2008.
- (Scheer, 2007) Scheer, IDS : *ARIS Tool Set*, <http://www.ids-scheer.fr>.
- (Schonenberg, 2008) Schonenberg, H., Mans, R., Russell, N., Mulyar, N. et Aalst, W.: *Process flexibility: A survey of contemporary approaches*, Advances in Enterprise Engineering I, p.16–30, 2008.
- (Sharifi *et al.*, 1999) Sharifi, H. et Zhang, Z.: *A methodology for achieving agility in manufacturing organisations: An introduction*, International Journal of Production Economics, 62(1-2), p.7-22, 1999.
- (Sheffi, 2004) Sheffi, Y.: *Demand Variability and Supply Chain Flexibility*, Entwicklungspfade und Meilensteine moderner Logistik, Glaber, p.87-113, 2004.
- (Sienou, 2009) Sienou, A.: *Proposition d'un cadre méthodologique pour le management intégré des risques et des processus d'entreprise*, Thèse de doctorat, Institut National Polytechnique, Toulouse, 2009

- (Terrasse *et al.*, 2005) Terrasse, M.N., Savonnet, M., Leclercq, E., Grison, T., Becker, G. : *Points de vue croisés sur les notions de modèle et métamodèle*, Premières journées sur l'Ingénierie Dirigée par les Modèles, Paris, 2005.
- (Tomasini *et al.*, 2004) Tomasini, R. M. et Van Wassenhove, L. N.: *Genetically modified food donations and the cost of neutrality: logistics response to the 2002 food crisis in Southern Africa*, INSEAD Case, 03/2004-5169, 2004.
- (Tomasini *et al.*, 2009) Tomasini, R. et Van Wassenhove, L.: *Humanitarian logistics*, Palgrave Macmillan., 2009.
- (Touzi *et al.*, 2009) Touzi, J., Benaben, F., Pingaud, H. et Lorré, J. P.: *A model-driven approach for collaborative service-oriented architecture design*, International Journal of Production Economics, 121(1), p.5–20, 2009.
- (Touzi, 2007) Touzi, J.: *Aide à la conception de Système d'Information Collaboratif , support de l'interopérabilité des entreprises*, Thèse de doctorat, Institut National Polytechnique, Toulouse, 2007.
- (Truptil *et al.*, 2008) Truptil, S., Bénaben, F., Couget, P., Lauras, M., Chapurlat, V. et Pingaud, H.: *Interoperability of Information Systems in Crisis Management: Crisis Modeling and Metamodeling*, I-ESA'08, Berlin, Allemagne., 2008.
- (Truptil *et al.*, 2009) Truptil, S., Bénaben, F., Hanachi, C., and Pingaud, H., *Collaborative process design for Mediation Information System Engineering*. Proceedings of ISCRAM'09. Göteborg, Sweden.
- (Truptil *et al.*, 2009a) Truptil, S., Bénaben, F., Pingaud, H., and Hanachi, C., *Une architecture de système d'information collaboratif pour la gestion de crise*. INFORSID'09. Toulouse, France.
- (Truptil *et al.*, 2009b) Truptil, S., Bénaben, F., and Pingaud, H., *Démarche de création d'un processus collaboratif de réponse à une crise*, CIGP'09. Bagnères-de-Bigorres, France.
- (Truptil *et al.*, 2010) Truptil, S., Benaben, F. et Pingaud, H.: *Une architecture de système d'information collaboratif pour la gestion de crise: Approche basée sur la médiation des systèmes*, Numéro Spéciale de la revue Ingénierie des systèmes d'information, Hermès, vol. 15, n°3/2010, pp.11-36, 2010.
- (Truptil *et al.*, 2010a) Truptil, S., Bénaben, F., Couget, P., Lauras, M., Chapurlat, V., et Pingaud, H.: *Mediation Information System Engineering for Interoperability Support in Crisis Management*, I-ESA'10, Coventry, 2010.
- (Truptil *et al.*, 2010b) Truptil, S., Bénaben, F., and Pingaud, H., *Collaborative process deduction to help the crisis cell emerging ecosystem to coordinate the crisis response*, IEEE-DEST'10, Springer-Verlag, Dubai, UAE.

- (Truptil *et al.*, 2010c) Truptil, S., Bénaben, F., and Pingaud, H : *A Mediation Information System to help to coordinate the response of a crisis*, Pro-VE'10, Springer, IFIP, Saint-Etienne, France.
- (van der Aalst *et al.*, 2004) van der Aalst, W. M., Aldred, L., Dumas, M. et ter Hofstede, A. H.: *Design and implementation of the YAWL system*, Advanced Information Systems Engineering, p. 281–305., 2004.
- (van der Aalst *et al.*, 2005) van der Aalst, W. M., Weske, M. et Grünbauer, D.: *Case handling: a new paradigm for business process support*, Data & Knowledge Engineering, 53(2), p.129–162, 2005.
- (van der Aalst, 1998) van der Aalst, W. M.: *Changing Workflows: On the interplay between flexibility and support*, p.5-6, Dagstuhl Seminar Report, 217, Dagstuhl, 1998.
- (Van Wassenhove, 2005) Van Wassenhove, L. N.: *Humanitarian aid logistics: supply chain management in high gear*, Journal of the Operational Research Society, 57(5), p.475–489, 2005.
- (Vanderhaeghen *et al.*, 2004) Vanderhaeghen, D., Zang, S., Hofer, A., Adam, O. : *XML-based Transformation of Business Process Models – Enabler for Collaborative Business Process Management -*, Proc. of the 2nd GI Workshop XML4BPM , XML Interchange Formats for Business Process Management, 2004.
- (Vernadat, 1996) Vernadat, F. B.: *Enterprise Modelling and Integration, principles and applications*, Chapman & All., 1996.
- (Vernadat, 1999) Vernadat, F. : *techniques de modélisation en entreprise : Applications aux processus Opérationnels*, economica 1999.
- (Vernadat, 2006) Vernadat, F. : *Interoperable enterprise systems : architecture and methods*, plenary lecture, IFAC/INCOM conference, Saint-Etienne, 2006.
- (White *et al.*, 2004) White, S. A.: *Business Process Modeling Notation (BPMN) Version 1.0*, Business Process Management Initiative, BPMI. org, 2004.
- (Wiederhold, 1992) Wiederhold, G.: *Mediators in the architecture of future information systems*, Computer, 25(3), p.38–49, 1992
- (Wohed *et al.*, 2003) Wohed, P., van der Aalst, W., Dumas, M. et Hofstede, A.: *Analysis of web services composition languages: The case of BPEL4WS*, Conceptual Modeling-ER 2003, p. 200–215, 2003.
- (Zachman, 1987) Zachman, J. : *A Framework for Information Systems Architecture*, IBM Systems Journal, Vol. 26, No 3, 1987.

# Annexes :

Annexe A : Mémo LM-RP-NG Après le séisme en Haïti. ....	154
Annexe B : Modèles de descriptions des services des acteurs .....	156
I. Modèle de la Croix Rouge .....	156
II. Modèle de la gendarmerie .....	157
III. Modèle du SAMU.....	158
IV. Modèle du SDIS .....	159
Annexe C : Interopérabilité.....	160
I. Interopérabilité.....	160
I.1 Définitions de la notion d'interopérabilité.....	160
I.2 Cadre de référence de l'interopérabilité.....	161
I.3 Synthèse.....	163
II. Les systèmes d'information .....	165
II.1 Définition de la notion de système d'information .....	165
II.2 Architecture de système d'information.....	166
II.3 SI et un système de systèmes d'information. ....	167
Annexe D : Du modèle à la transformation de modèles .....	169
I. Du modèle à la transformation de modèles.....	170
I.1 Modèle.....	170
I.2 Modèle de référence : Métamodèle.....	172
I.3 Transformation de modèles.....	173
I.4 Les ontologies.....	176
Annexe E : Solutions Technologiques .....	178
I. De la Collaboration au Système d'information.....	178
I.1 L'architecture orientée service (SOA) .....	178
I.2 Caractéristiques d'un service de système d'information .....	179
I.3 Entreprise Service Bus : Une technologie de médiation.....	181
II. Technologie pour la transformation de modèles :.....	182
II.1 XML .....	182
II.2 Langage de description de métamodèle .....	183
II.3 . Le choix d'un outil de transformation .....	184
Annexe F : Règles de mappings et fichiers ATL de la transformation UMLT à BPEL..	187
I. Présentation des Règles de transformation. ....	187
II. La transformation de modèles du modèle d'architecture technique au BPELArtifact	190
III. La transformation de modèles du modèle d'architecture technique au BPEL.....	193
Annexe G : UML2JBI .....	223



## Annexe A : Mémo LM-RP-NG Après le séisme en Haïti.

Après le séisme en Haïti,

il est urgent que l'ONU crée des "casques rouges"

LM RP/NG<sup>1</sup>19 fev10

Haïti est plongé dans le chaos et la désolation depuis plus d'un mois. Les premiers bilans font état de plus de 200 000 morts et 300 000 blessés. Des centaines de milliers de rescapés sont installés dans des camps de fortune qui risquent d'occuper durablement le paysage. Et pourtant, saluons l'élan de solidarité de la communauté internationale qui s'est mobilisée, dès les premières heures de la catastrophe, pour aider Haïti. Des équipes de secours ont été dépêchées par centaines. Pas moins de 74 avions sont arrivés en vingt-quatre heures. Résultat, l'aéroport a été engorgé et des ONG sont restées bloquées des jours durant sur le tarmac, pendant que les survivants périssaient encore sous les décombres.

L'explication est simple: Haïti n'était pas prêt à accueillir cette affluence de bonnes volontés! Américains, Européens, Chinois,... ont fait preuve d'une générosité sans précédent. Porte-avions, hôpitaux de campagne, rations alimentaires, tentes, hélicoptères... ont été acheminés des quatre coins de la planète. Mais par manque d'organisation et de coordination, nous avons perdu du temps et de trop nombreuses vies humaines.

Ce qu'il aurait fallu dans l'urgence, c'est un état-major humanitaire qui aurait,

1. identifié les besoins et
2. réalisé un état des lieux des ressources disponibles.

Ce qui aurait pu changer la donne, c'est une force de réaction rapide qui aurait,

- préconisé un schéma directeur d'intervention et
- coordonné les actions des équipes opérationnelles.

Haïti aura été la catastrophe de trop et il n'aura pas fallu beaucoup de temps pour comprendre que nous n'avons pas retenu les leçons du tsunami. La communauté humanitaire n'est plus assez forte pour relever seule les défis imposés par la nature déchaînée. Nous n'avons pas besoin d'un acteur supplémentaire: les ONG et les agences intergouvernementales accomplissent déjà un travail considérable. Encore moins d'une

---

<sup>1</sup>René Prével et Nicole Guedj.

nouvelle philosophie. Il n'y en a qu'une qui vaille : sauver des vies. Le monde humanitaire a besoin de préparation, d'encadrement, de coordination, de régulation, de structuration, de veille... Il lui faut une "intelligence humanitaire" pour anticiper et unifier son action.

Nous sommes capables d'inventer un humanitaire efficace et durable, un humanitaire qui sache optimiser les fonds alloués et surmonter l'écueil du gaspillage.

De Port-au-Prince, sur le théâtre de la plus grande désorganisation humanitaire de ces dernières années, nous avons proposé une solution. Celle des "casques rouges", les frères humanitaires des casques bleus, qui, sous l'égide de l'ONU, auraient la légitimité de remplir cette mission. Ban Ki-moon, le secrétaire général des Nations unies, a ouvert la voie en nommant Bill Clinton coordonnateur de l'aide internationale en Haïti. Hélas, cette décision est intervenue trois semaines après le séisme. Mais gageons qu'une véritable volonté de coordination présidera au bon ordre des opérations de reconstruction.

L'ONU doit assumer sa "responsabilité de protéger" et garantir le droit au secours à toutes les populations. Ne nous étonnons pas que les casques bleus ne parviennent pas à organiser la distribution alimentaire. Ce sont des forces de maintien de la paix et non des secouristes. Des militaires et non des humanitaires!

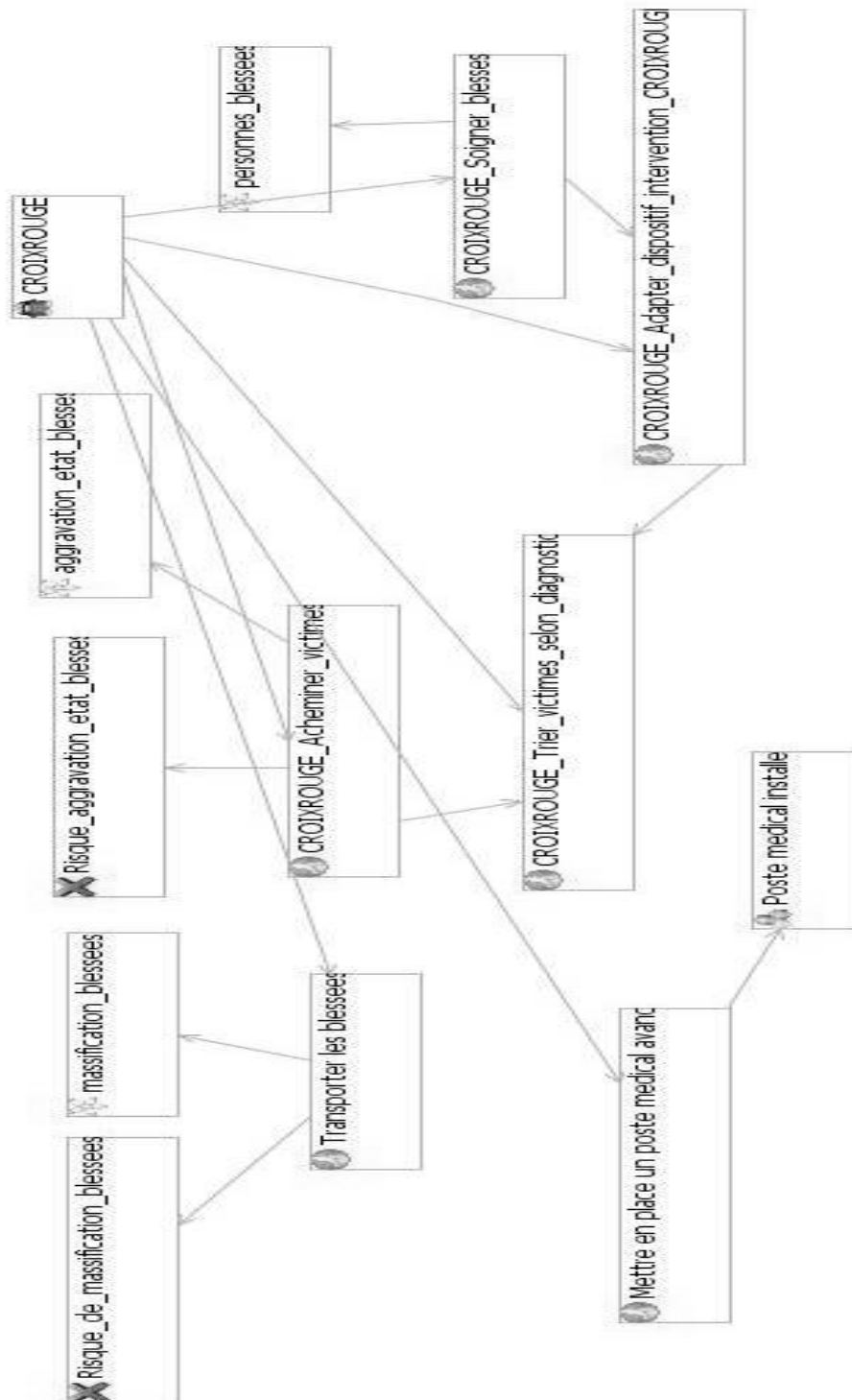
L'ONU doit se doter d'une force 100 % humanitaire pour organiser et coordonner les secours. Il ne s'agit ni de lever une armée ni de remplacer les ONG. Mais plutôt de réunir un groupe d'experts composé de logisticiens, médecins, ingénieurs, pompiers... à même de définir une stratégie globale pour gérer une crise dans l'urgence et réguler l'aide proposée par la communauté internationale. Le premier jour et pas plus tard!

C'est en responsabilité que nous lançons, à l'ensemble des chefs d'Etat, un appel à consensus. C'est par solidarité qu'ils doivent veiller à ce que l'histoire ne se répète plus. Pas un point de la planète n'est à l'abri d'une prochaine catastrophe. La question de la coordination des secours se reposera tant que nous n'aurons pas pris les décisions qui s'imposent. Mais la prochaine fois, on ne nous le pardonnera pas. A la veille de la conférence internationale pour la reconstruction d'Haïti, nous formons le vœu que des "casques rouges" soient créés à l'ONU.

## Annexe B : Modèles de descriptions des services des acteurs

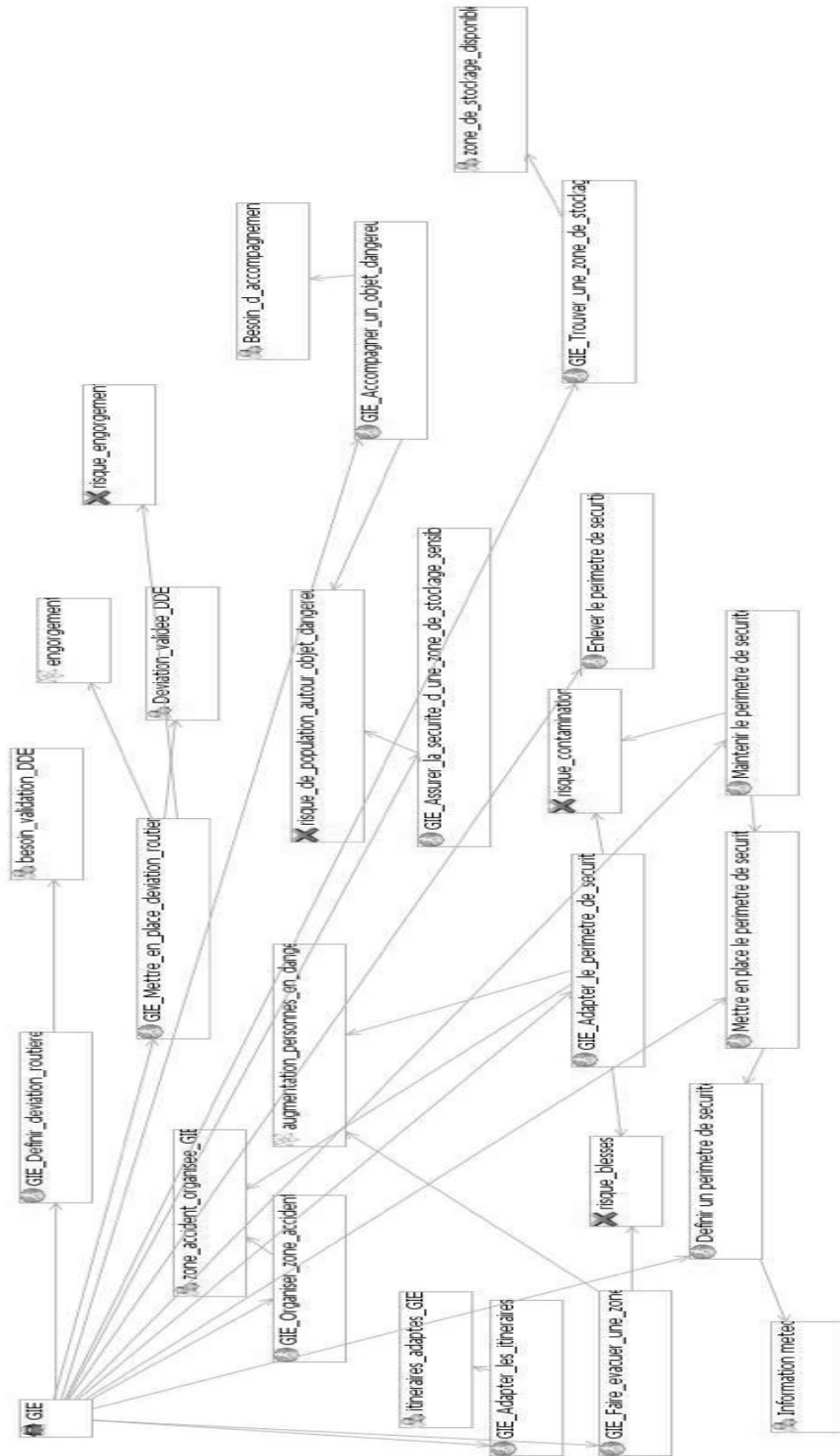
Dans cette annexe, nous présentons l'ensemble des modèles de description de services utilisé dans le cadre des illustrations du chapitre II et V.

### I. Modèle de la Croix Rouge



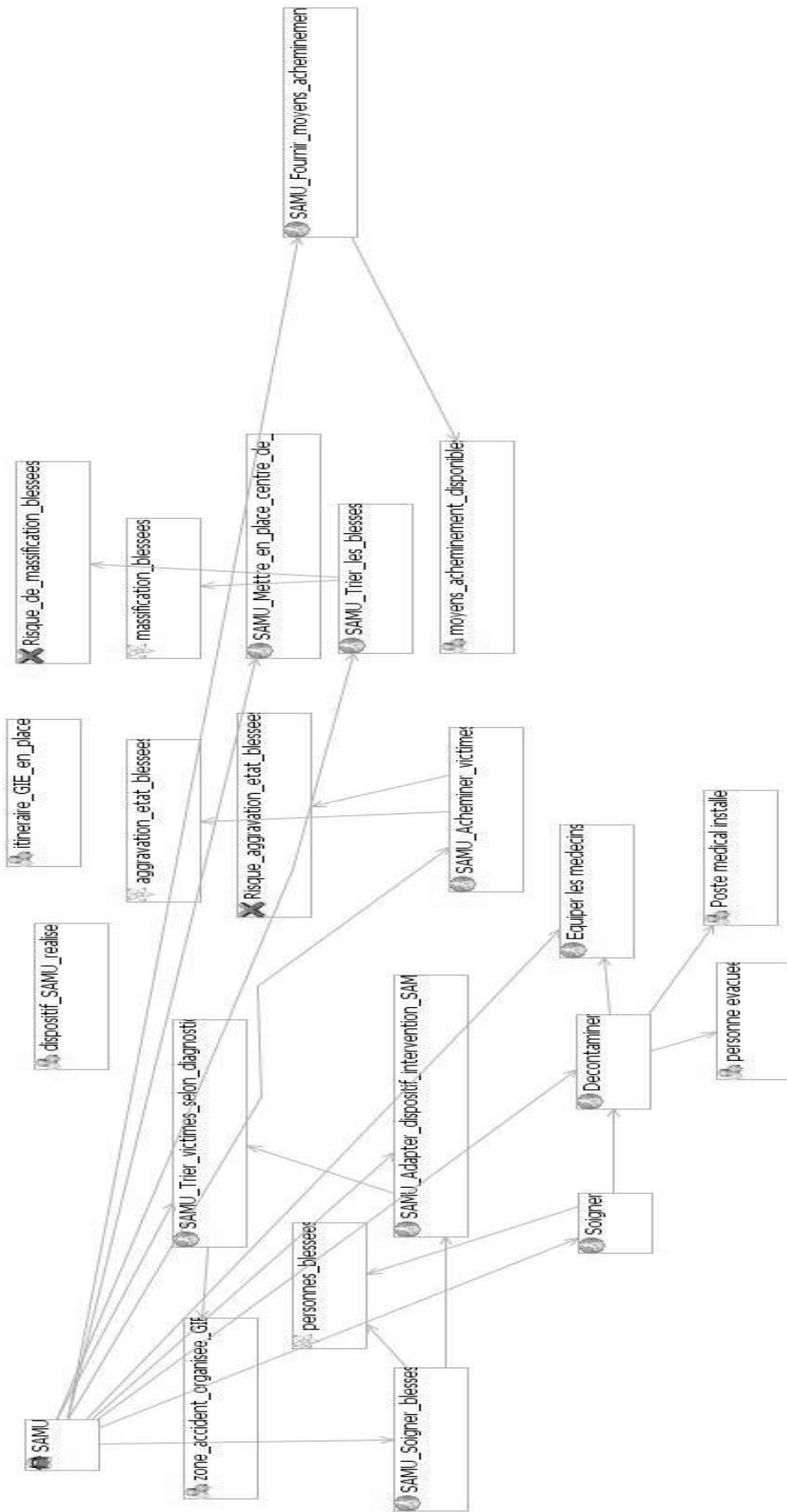
## II. Modèle de la gendarmerie

□

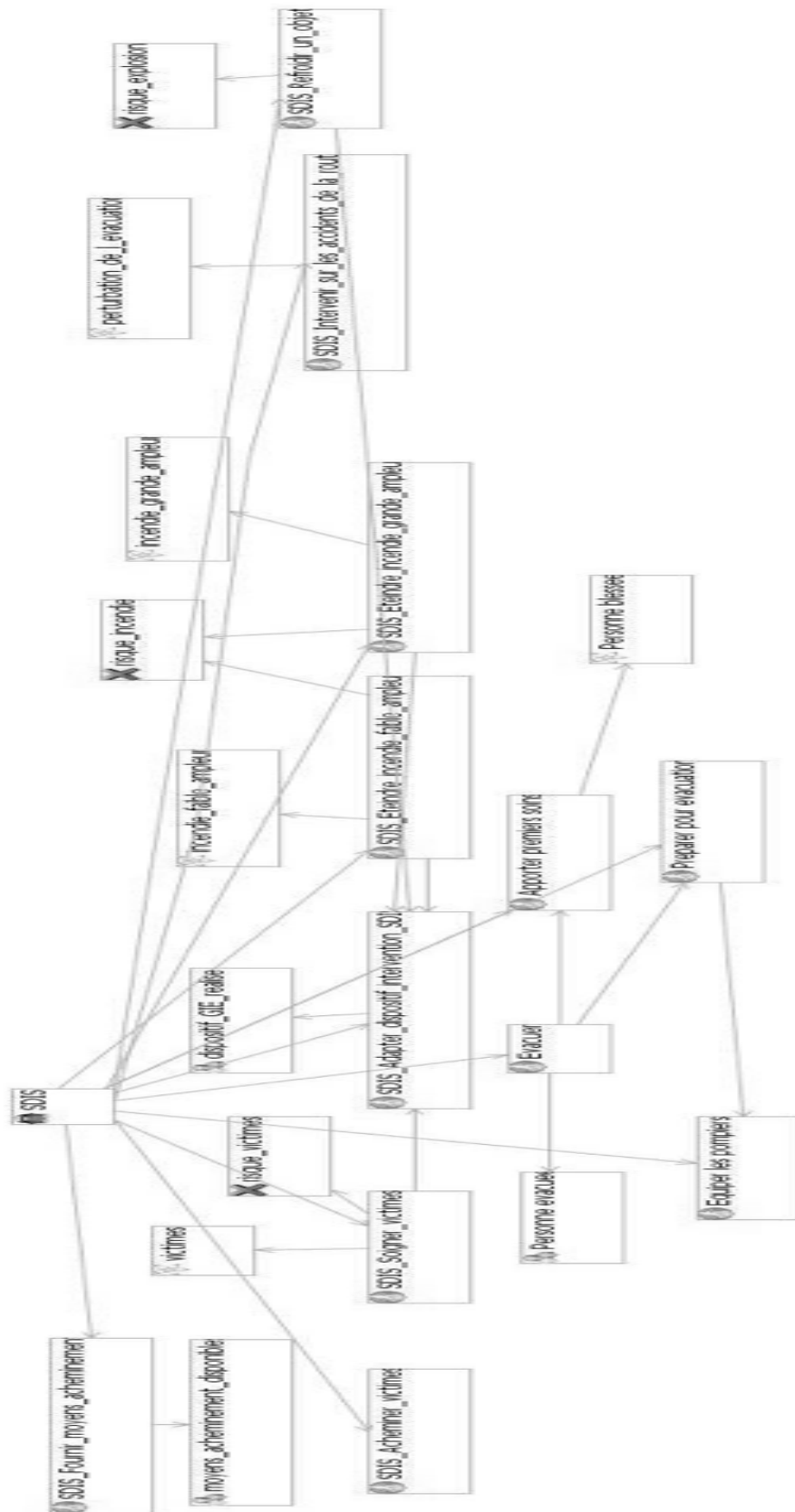


### III. Modèle du SAMU

□



## IV. Modèle du SDIS



## Annexe C : Interopérabilité

L'évolution du tissu industriel est caractérisée comme « la transformation d'une structure cristallisée en un environnement fluide » (Bénaben *et al.*, 2007) ou encore comme « la transformation d'une construction statique de lego® vers un organisme vivant » (Luzeaux *et al.*, 2008). Ces métaphores insistent sur l'évolution des collaborations inter-entreprises. Elles ne se fondent plus sur une optique à long terme avec des collaborations stables, exigeantes pour leur mise en place et longues à établir. Elles se fondent sur des collaborations ponctuelles rapides à mettre en place et à dissoudre.

L'observation de cette évolution est le point de départ des recherches effectuées dans les domaines de l'interopérabilité et de l'agilité appliquées au thème de la collaboration d'entreprises. Ces deux axes de recherche s'intéressent aux conséquences de cette évolution sur les traitements de l'information ainsi que sur les systèmes d'information. Même s'il ne faut pas limiter l'interopérabilité à la seule solution physique, comme le précise (Berre *et al.*, 2007)<sup>2</sup>, l'aspect matériel du système d'information est incontournable (Monateri *et al.*, 1999). En effet, le souhait des entreprises de travailler rapidement ensemble tout en préservant leur caractère confidentiel implique que l'accès à leur SI soit contrôlé. Cette limitation aux autres partenaires contraint fortement la collaboration en termes de formats de données et de coordination des échanges et pose ainsi de réels problèmes d'interopérabilité.

L'objectif de cette première partie de chapitre est de mener une exploration des notions liées à cette problématique. Cet axe est décomposé en deux parties : la première partie s'intéresse à la notion d'interopérabilité entre des organisations hétérogènes. La deuxième partie s'intéresse aux systèmes d'information en tant qu'éléments majeurs des solutions d'interopérabilité.

### I. Interopérabilité

#### I.1 Définitions de la notion d'interopérabilité

Le concept d'interopérabilité est apparu dans les années 1990 dans le domaine des systèmes d'information et a été étendu par la suite à plusieurs domaines tel que les domaines militaire, médical, des transports, etc. (DoD, 2001), (ISO 16100, 2002), (apfa, 2004). Depuis lors, de nombreuses définitions de ce concept ont été proposées. Parmi la masse de définitions disponibles, nous avons décidé de retenir les suivantes :

« L'interopérabilité des applications d'entreprise est définie comme la capacité pour un système d'échanger de l'information et des services dans un environnement technologique et organisationnel hétérogène » (IEEE, 1990).

Selon le réseau d'excellence européen InterOp NoE (Kontansas *et al.*, 2005), la notion d'interopérabilité fait référence à « la capacité de systèmes à travailler ensemble, sans nécessiter d'effort particulier de la part des utilisateurs de ces systèmes ».

Ces définitions soulignent que l'interopérabilité est une solution aux nécessités de partage d'information et de services entre des organisations hétérogènes. Cependant, ces définitions soulignent

---

<sup>2</sup> « Interoperability should not only be considered a property of ICT systems, but also concerns the business process and the business context of an enterprise »

aussi que ce partage doit pouvoir être mis en place sans effort particulier des partenaires, c'est-à-dire sans modification spécifique pour chaque collaboration. Nous retenons aussi la définition suivante :

« l'interopérabilité désigne une capacité de systèmes, nativement étrangers les uns par rapport aux autres, à interagir afin d'établir des comportements collectifs harmonieux et finalisés, sans avoir à modifier en profondeur leur structure ou leur comportement » (Pingaud, 2009). Cette définition insiste sur le fait que chaque système reste indépendant vis-à-vis des autres systèmes.

Cette notion d'indépendance constitue pour certains auteurs la principale différence entre l'interopérabilité et l'intégration. Dans (Chen et al., 2008) les auteurs différencient l'intégration, représentant l'uniformisation, de l'interopérabilité, représentant un environnement autonome et fédérateur. Le tableau ci-dessous essaye de synthétiser la différence entre l'intégration et l'interopérabilité en s'inspirant des travaux de (Chen *et al.*, 2006) et (Chen et al., 2008)

**Tableau II- 1 : différence entre interopérabilité et intégration**

Interopérabilité	Intégration
Identité et diversité préservées	Assimilation
Couplage lâche (composants indépendants)	Couplage fort (composants interdépendants)
Vocabulaire non normalisé	Vocabulaire normalisé
Cohérence des objectifs globaux	Cohérence des objectifs locaux et globaux

## I.2 Cadre de référence de l'interopérabilité

Le but de nos travaux étant d'essayer de supporter l'interopérabilité entre différentes organisations dans un contexte de gestion de crise, nous présentons dans cette partie, un cadre de référence de l'interopérabilité. La criticité de la notion d'interopérabilité a induit le développement de nombreux cadres de référence liés à différents domaines tel que : IDEAS (Interoperability Development for Enterprise Application and Software), AIF (ATHENA Interoperability Framework), e-GIF (e-Government Interoperability Framework), etc. De l'ensemble de ces cadres de références, nous avons préféré présenter celui qui à nos yeux nous semble le plus couvrant vis-à-vis des différentes dimensions de la notion d'interopérabilité : le cadre EIF (Enterprise Interoperability Framework),

Le cadre EIF (Enterprise Interoperability Framework), développé dans le cadre du projet InterOp Noe (Interoperability Research for Networked Enterprise Applications and software, FP6 508011) (Chen et al., 2006), a pour objectif de définir un cadre de référence des recherches dans le domaine de l'interopérabilité. Généralement, la recherche de nouvelles solutions support d'interopérabilité est pilotée par les problèmes rencontrés. Ce cadre de références est composé de trois dimensions devant couvrir l'ensemble des problèmes pouvant être rencontrés ainsi que les compétences et solutions pouvant les traiter. Ces trois dimensions sont : les approches adoptées, les niveaux d'application et les barrières (Chen et al., 2008).

Ce cadre de travail distingue tout d'abord trois **approches de l'interopérabilité** entre systèmes :



- L'approche **intégrée** correspond à l'utilisation par tous les tiers (partenaires, outils...) du même format de données. Ce format n'est pas forcément un standard, mais il doit être connu, accepté et utilisé par toutes les parties afin de créer et d'adapter les différents systèmes en jeu.
- L'approche **unifiée** correspond à la définition d'un format commun servant de modèle aux différents formats de données des partenaires. Ce format de données n'est pas directement utilisé, mais permet de faire les correspondances entre les formats de données (pivot).
- L'approche **fédérée** signifie qu'aucun format de donnée n'est imposé. Chaque partenaire utilise alors ses propres concepts et l'interopérabilité est réalisée « à la volée ». Seule une base de connaissance définissant un vocabulaire de référence est partagée entre les tiers.

Le cadre de référence EIF définit quatre **niveaux d'interopérabilité** à des niveaux d'abstraction différents pour l'entreprise :

- L'interopérabilité des **données** a pour but de mettre en relation différents modèles de données venant d'applications diverses. Cela permet de partager des données entre des systèmes hétérogènes.
- L'interopérabilité des **services** concerne l'identification, la composition, et la réalisation de fonction d'application afin de mettre en relation divers services et applications (développés indépendamment les uns des autres). La notion de service ne se limite pas au composant informatique mais elle concerne aussi les savoir faire des entreprises
- L'interopérabilité des **processus** ambitionne de mettre en relation différents processus de travail : un processus y est défini comme la séquence de services à exécuter dans un objectif rechercher par l'entreprise. Lors d'une collaboration, il est aussi nécessaire d'étudier la connection entre des processus de partenaires différents afin de créer un processus commun.
- L'interopérabilité **métier** fait référence aux caractéristiques de l'entreprise et de la collaboration. Cette caractérisation de la collaboration permet de déduire les objectifs communs ainsi que la définition des interactions possibles avec les autres entreprises en fonction des variétés de culture, de mode de décision et de méthodes de travail de chaque entreprise.

Enfin, le cadre EIF décrit les **barrières** ou incompatibilités empêchant la mise en place d'une solution d'interopérabilité :

- La barrière **conceptuelle** concerne les différences syntaxiques et sémantiques présentes entre les informations échangées. Elle est due aux différentes façons de nommer, structurer et représenter les concepts. La barrière conceptuelle peut être due à des différences syntaxiques (différents formats), à des différences schématiques (différents schémas, modèles) ou encore des différences sémantiques (différents noms pour la même entité et même nom pour différentes entités).
- La barrière **technologique** concerne les différences entre les systèmes informatiques (architecture, infrastructure, standards utilisés...) entravant la collaboration entre les systèmes.
- La barrière **organisationnelle** concerne les incompatibilités au niveau de l'organisation, des responsabilités ou des droits accordés à chacun. Elle peut être due aux différentes façons de travail, aux différentes structures des organisations.

En prenant en compte ces trois dimensions, le cadre EIF est représenté sous la forme d'un cube (ou plutôt d'un parallélépipède rectangle) figure 2.

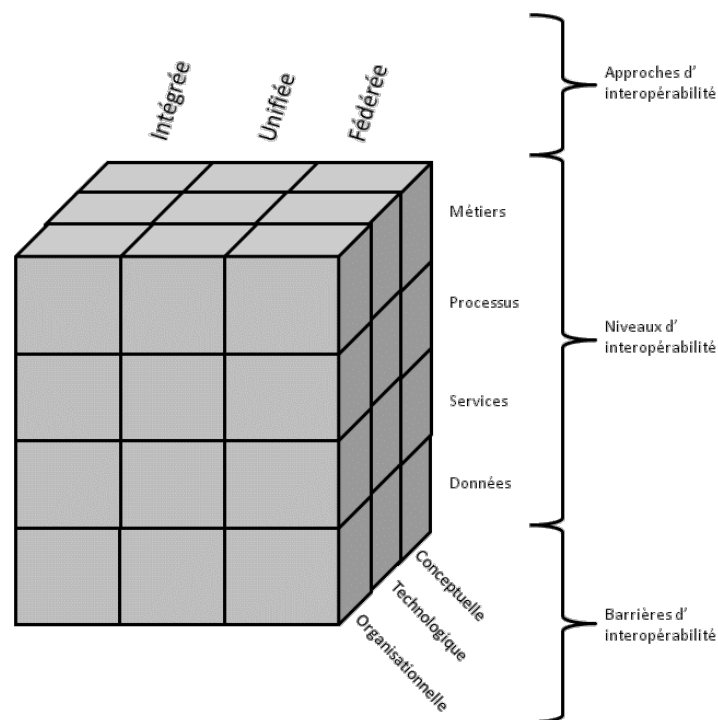


Figure 1 : Représentation du cadre EIF (Chen et al., 2008),

### I.3 Synthèse

Nous venons de voir que l'interopérabilité est définie comme permettant à des organisations hétérogènes de travailler ensemble sans le moindre effort tout en gardant leurs particularités et leur indépendance les unes par rapport aux autres. Quant aux cadres de références de l'interopérabilité, ils s'intéressent à mettre en avant les différents niveaux de mise en place de l'interopérabilité en pointant les exigences des solutions techniques, les propriétés des systèmes de communication et les spécificités « métier » de la collaboration. Cependant, l'hétérogénéité des différents partenaires, aussi bien au niveau de leur plateforme technologique que de leurs méthodes de travail, leur pose des problèmes de compréhension mutuelle. Cette hétérogénéité peut être de différentes natures : syntaxique, sémantique et technique (Ouksel et al., 1999). L'hétérogénéité technique est liée aux plateformes logicielles et matérielles ainsi qu'aux protocoles de communication. L'hétérogénéité syntaxique concerne les conflits des formats et des modèles de représentation de données, quant à l'hétérogénéité sémantique elle résulte des discordances d'interprétation d'une valeur, d'un terme, d'un concept ou d'une conceptualisation du monde réel.

Les solutions aux problèmes liés à cette hétérogénéité peuvent être classées selon le cadre de référence EIF. En effet, afin de résoudre ces problèmes d'hétérogénéité, différentes approches peuvent être adoptées selon le degré de convergence de l'ensemble des participants vers des choix de technologie et de langage commun. Plus la convergence est importante plus les problèmes organisationnel, technologique et conceptuel sont faibles.

Ces solutions peuvent correspondre à différents niveaux de collaboration. En effet, il existe une similitude entre les niveaux de collaboration (communication, coordination, coopération, intégration) et les niveaux d'interopérabilités (métiers, processus, services, données) proposés par l'EIF. La *communication* traduisant l'échange d'information entre différents partenaires, elle concerne le niveau d'interopérabilité des données. La *coordination* induisant, en plus de la circulation des informations, la synchronisation des

activités (sans réelle nécessité de construire de processus collaboratif), les niveaux d'interopérabilité associés correspondent à ceux des données et des services. La *coopération*, quant à elle, implique la nécessité de pouvoir définir, d'une part, des objectifs communs partagés et d'autre part, des processus collaboratifs permettant leur réalisation (en plus de l'échange d'informations et du partage des services). Il est donc nécessaire de définir l'interopérabilité aux niveaux données, services, processus et métier. Enfin, l'*intégration* constitue une « fluidification » du niveau de coopération, et à ce titre, n'a pas à proprement parler de niveau d'interopérabilité associé autres que ceux concernés par la coopération.

Dans le cadre de nos travaux, nous cherchons à mettre en place un système de système de réponse à la crise (cf chapitre I). Ce système est présenté comme une solution d'interopérabilité entre les différents partenaires de la réponse à la crise. Par conséquent, ce système doit pouvoir être utilisé dans le cas de crise civile, militaire ou humanitaire, *i.e.* dans des contextes nécessitant tous les niveaux de collaboration : communication, coordination, coopération et intégration. Par conséquent et compte tenu des considérations précédentes, il doit couvrir les niveaux d'interopérabilité : données, services, processus et métiers. De plus, le caractère exceptionnel et unique des situations de crise nécessite une réponse adaptée et vierge, autant que possible, de tout standard. Au sens de l'EIF, le système de systèmes doit donc être obtenu via une approche fédérée. La mise en œuvre d'une telle approche entraîne alors des problèmes de définitions de concepts et des rôles de chacun, d'hétérogénéité de formats de données, et de supports technologiques. Ces problèmes relèvent des barrières organisationnelle, technologique et conceptuelle du cadre EIF.

La figure 3 résume le positionnement de la solution proposée par ces travaux de thèse vis-à-vis du cadre de référence EIF.

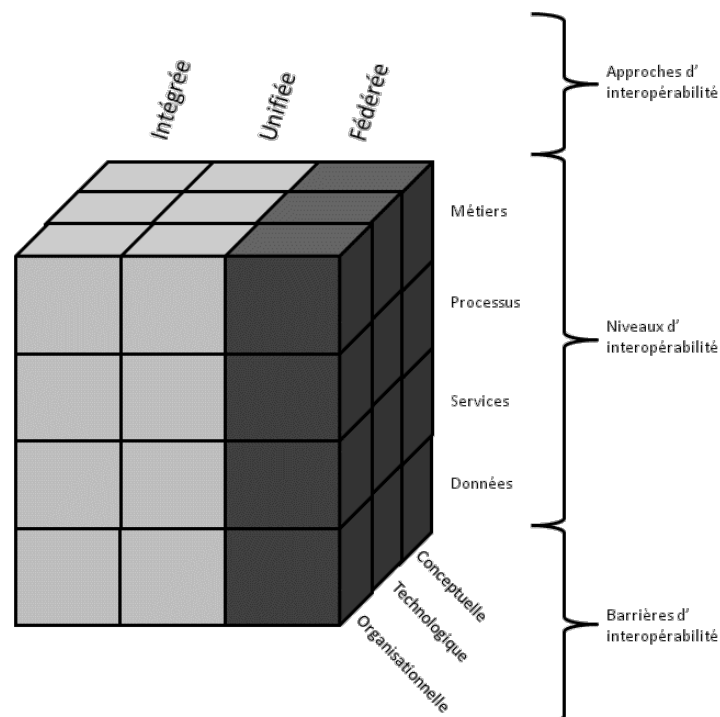


Figure 2 : positionnement de nos travaux vis à vis du cadre de référence EIF

## II. Les systèmes d'information

Bien qu'il ne faille pas la limiter à cela, la question de la prise en charge de l'interopérabilité ne peut pas être absolument dissociée des systèmes d'informations. Dans cette section, nous allons donc revenir sur le concept de système d'information d'entreprise.

### II.1 Définition de la notion de système d'information

Dans les années soixante, la montée en puissance de l'informatique a permis d'outiller et de matérialiser le concept de système d'information. Il était alors assimilé à un système informatique applicatif, c'est-à-dire un ensemble d'applications de gestion au service de l'entreprise. Les premiers systèmes d'information ont permis d'automatiser des traitements répétitifs comme la comptabilité, la gestion de la paie du personnel ou la tenue des comptes par clients (Le Moigne, 1977) (Peaucelle, 1999).

Suite aux progrès dans le domaine de l'informatique, le système d'information a pris une nouvelle dimension au sein de l'entreprise, comme l'explique la définition du système d'information donnée par (Reix et al., 1995) : « un système composé d'entités diverses (employés, ordinateurs, réseaux, applications, base de données, règles, ...) chargées de stocker et de traiter les informations relatives au système opérant afin de les mettre à disposition du système de pilotage. Il peut recevoir les informations du système de pilotage. Il peut émettre vers le système opérant des informations ». Cette définition relève d'une approche systémique appliquée à la notion de système d'information. Le système d'information y est vu comme un sous-système interagissant à la fois avec le système opérant et le système de pilotage (le système opérant devant transformer des entrées en sorties selon une finalité et le système de pilotage, devant élaborer des instructions en fonction des informations (Le Moigne, 1977), (Le Moigne, 1990)).

La notion d'information est bien entendu primordiale dans la définition du système d'information. Dans sa définition Robert Reix attribut au système d'information la capacité de « ... stocker et traiter les informations ... ». Nous pouvons cependant aller plus loin au niveau des fonctionnalités assurées par le système d'information. (Saadoun, 00) explique que « la raison d'être d'un système d'information est l'accès au bon moment à la bonne information pour prendre la bonne décision ». Afin d'ajouter une dimension « inter-entreprises », nous proposons de compléter cette proposition en y adjoignant « le bon utilisateur ». Cette approche fonctionnelle du système d'information impose à celui-ci de pouvoir acquérir, traiter, stocker et communiquer des informations. Pour cela, (Reix et Helfer, 1995) caractérise le système d'information comme étant un ensemble organisé de ressources matérielles, de personnel, de connaissances et de logiciels. Cette qualification du système d'information nous permet d'introduire une vision structurelle et architecturale.

Depuis très longtemps, l'Homme sait consigner l'information sous forme de dessin, de chiffres ou d'écrits. Il a aussi compris l'intérêt des archives et sait transformer l'information en fonction de règles et de procédures définies. Ce constat apporte la réponse à la question portant sur l'existence possible d'un système d'information sans support informatique. Bien qu'un système d'information puisse exister sans support informatique, une vision moderne du système d'information ne peut plus se concevoir sans l'informatique. La définition d'un système d'information de C. Morley (Morley, 2002) (Morley et al., 2005) propose cette séparation entre le système d'information et le support informatique puisqu'elle propose de voir un SI comme la composition de deux sous-systèmes : **le système de traitement de l'information** (comprenant les acteurs, les données et les processus) **et le système informatique** (comprenant les ressources matérielles et logicielles, les bases de données et les fonctions). Le système informatique y est vu comme l'infrastructure du système d'information.

## II.2 Architecture de système d'information

L'architecture des systèmes d'information vise à décrire la structuration d'un système d'information en termes d'organisation de ces constituants. Dans (Morley et al., 2005), on identifie une architecture de système d'information comme « une représentation abstraite des différentes parties du système d'information qui permet des décisions globales et de s'assurer de la pertinence de l'assemblage, notamment la cohérence et l'efficacité technique... ».

L'architecture représente alors un enjeu fondamental en vue de permettre une certaine maîtrise de la conception et de l'évolution des systèmes d'information. La maîtrise de la conception passe par une description détaillée des composants et de leur organisation généralement représentée par différents modèles de conception. La maîtrise de l'évolution, quant à elle, passe par une gestion efficace des changements qui interviennent durant le cycle de vie du système (Vernadat 2006). Tout l'enjeu de cette maîtrise de la conception et de l'évolution est de permettre à l'organisation, généralement une entreprise, d'avoir un système d'information à son image en termes d'architecture, tout en permettant aux deux systèmes (entreprise et SI) d'évoluer en fonction du contexte.

Concernant les architectures d'entreprises, il existe bon nombre de références permettant de structurer ce domaine scientifique :

- des frameworks d'architectures d'entreprises (ODP (Putman, 2001), IEEE 1471 (IEEE, 2000), Zachman (Zachman, 1987), TOGAF (OpenGroup, 2005), ARCHIMATE (Archimate, 2007), etc.),
- des méthodologies ou approches d'ingénierie pour de telles architectures (PERA (Pera, 2007), CIMOSA (Vernadat 1996), GERAM (Geram, 1997), Urbanisme (Lonpégé, 2002), etc.),
- des langages de modélisation de telles organisations (UML (Fowler et al., 2004), UEMML (Panetto, 2006), BPML (BPML, 2007), BPMN (White et al., 2004), etc.),
- et des outils dédiés (Eclipse (Eclipse, 2007), ARIS (Sheer, 2007), Mega (Mega, 2007), etc).

Concernant les architectures de systèmes d'information, il en existe une typologie riche et variée. Ces architectures représentent en particulier les différents niveaux d'abstraction d'un système d'information. (Touzi, 2007) propose de classer les architectures de la façon suivante :

- **architecture fonctionnelle** : permet de spécifier les besoins fonctionnels des utilisateurs de système d'information. Ainsi, un système d'information est présenté en termes de fonctions, liées entre elles par des flux d'entrée/sortie (*SADI*) ou liées aux acteurs qui les manipulent (*diagramme de cas d'utilisation UML*). Nous parlons aussi d'urbanisme de système d'information, qui consiste à découper le SI en modules autonomes faisant référence aux fonctions de SI. Cette architecture intervient directement pour faire évoluer ou refondre un système d'information.
- **Architecture logique** : permet d'identifier la structuration d'un système d'information en adoptant une logique indépendante des considérations techniques. Les architectures logiques les plus connues sont : l'architecture orientée composants et l'architecture orientée services (SOA). Les composants ou les services sont caractérisés par le fait qu'ils sont des entités autonomes qui réagissent par échange de messages et sont définis sur différents niveaux de granularité.
- **Architecture informatique (ou physique)** : décrit la structuration d'un système informatique en termes d'organisation de fonctions et des constituants qui le composent. Cette architecture définit d'autres « sous-architectures » :

1. **Architecture logicielle** : l'agencement et l'interaction des composants logiciels. Ces derniers peuvent être définis sur plusieurs couches (couche présentation, couche applicative, couche données).
2. **Architecture matérielle** : l'agencement et l'interaction des composants physiques (disque dur, unité centrale, etc.).
3. **Architecture intergicielle** (Middleware) : l'agencement et l'interaction des composants servant à faire communiquer plusieurs applications entre elles.
4. **Architecture réseau** : l'architecture permettant la communication au sein d'un système d'information.

## II.3 SI et un système de systèmes d'information.

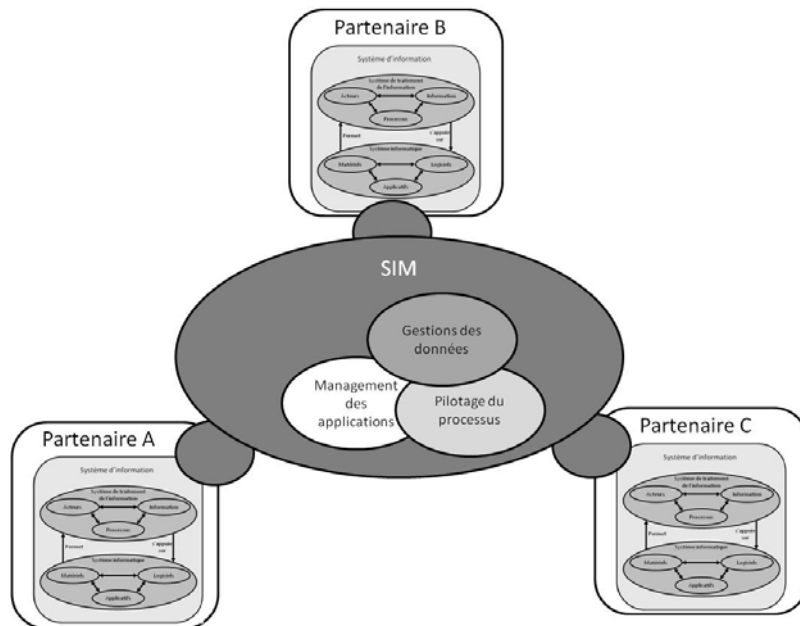
Nous venons de voir qu'un système d'information a pour vocation de gérer l'information afin que la bonne information soit disponible, au bon moment, au bon format. Pour cela il repose généralement, même si ce n'est pas obligatoire, sur un support informatique. L'information y est ensuite traitée selon des règles « métiers » et véhiculée selon un cheminement prédéfini.

Cependant qu'en est-il pour un système de systèmes d'information ? En se référant aux cinq critères de Maier (Maier, 1998), un système de systèmes se met en place entre différents partenaires, indépendants sur le plan managérial et sur le plan opérationnel, et reposant par conséquent sur des systèmes d'information hétérogènes. Cette indépendance des partenaires et cette hétérogénéité de leurs SI posent des problèmes de traitement de l'information du fait, d'une part de la nécessité de transformer l'information (l'hétérogénéité des SI se traduisant aussi par une hétérogénéité de formalismes et de langages), et d'autre part de la difficulté de définir les séquences et ordonnancements des actions d'échange et de traitement de l'information (l'indépendance des partenaires se traduisant aussi par une absence totale ou partielle de définition de leur collaboration, coordination, synchronisation). Il serait donc profitable d'envisager qu'un système tiers puisse se connecter à l'ensemble des systèmes d'information des partenaires afin de prendre en charge non seulement la traduction des données mais également l'orchestration des échanges et des actions selon la prescription fournie par un modèle de processus. Il s'agit bien là d'un système d'information spécifique puisqu'il ferait intervenir des acteurs (les systèmes contributeurs) qui interagissent selon le processus collaboratif (*cf.* chapitre I) qui véhicule des flux d'information entre les services.

Un tel système peut être assimilé à un système d'information de médiation (SIM) inspiré des travaux de (Wielderhold, 1992). Ce SIM est défini comme un composant intermédiaire entre les systèmes contributeurs. Ce médiateur peut être perçu comme un média capable de supporter et de faciliter la collaboration entre les partenaires (Touzi, 2007). Ce système d'information est en charge de trois missions (Bénaben *et al.*, 2008) et (Aubert *et al.*, 2002) :

- piloter le processus collaboratif,
- gérer les données (transformer et transférer l'information),
- manager les applications.

Afin de réaliser ces missions, le SIM doit pouvoir être connecté à l'ensemble des systèmes d'information des partenaires afin de traduire les données, d'assurer leur échange et d'exécuter les services selon la prescription fournie par le modèle de processus (et ce malgré la non-standardisation des formats de données et des protocoles d'accès aux applications ou d'orchestration des processus). La figure 3 résume les différentes fonctionnalités que doit proposer le SIM.



**Figure 3 : un système d'information de médiation pour supporter l'interopérabilité**

Ainsi, l'adjonction d'un SIM, au sein d'une communauté de systèmes d'information hétérogènes peut s'avérer une solution raisonnable pour prendre en charge les fonctions inhérentes au support de l'interopérabilité des partenaires. Ce SIM, s'il devient incontestablement la composante cruciale du bon comportement du réseau d'organisation, n'en est pas pour autant critique vis-à-vis des partenaires eux-mêmes. Une hypothétique disparition n'entraînerait qu'un retour du système de systèmes à un ensemble de systèmes, indépendants mais intacts dans leurs connaissances, leurs compétences et leurs savoir-faire.

Nous n'oublions pas les difficultés liées aux différences sémantiques dans l'identification des services et dans les interprétations de données. Ces considérations ne font pas l'objet d'une étude particulière au sein de ces travaux (les hypothèses permettant de s'en affranchir seront présentées ultérieurement), néanmoins le problème est formulé dans le cadre de travaux menés en parallèle aux nôtres, et fait l'objet de recherches particulières pour étoffer le médiateur (Rajsiri, 2009), (Boissel-Dallier et al., 2009).

## Annexe D : Du modèle à la transformation de modèles

Dans la partie précédente, nous avons évoqué l'évolution du tissu industriel et ses conséquences sur les collaborations inter-entreprises. En effet, suite à cette évolution, le laps de temps disponible pour la mise en place d'un système d'information de médiation implique la nécessité de pouvoir le développer le plus rapidement possible à partir des informations disponibles, relatives à la collaboration. L'automatisation de cette démarche est une source conséquente d'optimisation et c'est justement tout l'intérêt de l'ingénierie dirigée par les modèles (IDM).

L'IDM est née dans les années 2000 avec la proposition d'une approche dirigée par les modèles (MDA) faite par l'OMG (OMG, 2003) dont l'objectif était de permettre de spécifier et faire évoluer des modèles d'applications au rythme requis par l'organisation et non par l'évolution des plateformes technologiques. L'IDM peut être définie comme le domaine de l'informatique mettant à disposition des outils, concepts et langages pour créer et transformer des modèles. Cette discipline propose de nouvelles approches qui se décomposent généralement en deux phases distinctes :

- la construction (élaboration) d'un (ou plusieurs) modèle(s),
- la manipulation de ce(s) dernier(s).

Pour cela, l'IDM s'articule autour de quatre natures de modèles (Miller *et al.*, 2003) (Bézivin, 2004) :

- **CIM (*Computation Independent Model*)** : appelé aussi modèle de domaine ou modèle métier, il montre le système dans l'environnement organisationnel dans lequel il va être exécuté. Son but est d'aider à la compréhension mutuelle du problème. Les exigences exprimées dans le CIM doivent être traçables dans les modèles suivants (PIM et PSM).
- **PIM (*Platform Independent Model*)** : modèle indépendant des plateformes techniques, le PIM représente par exemple les différentes entités fonctionnelles d'un système et la description de leurs interactions, vis-à-vis de la logique d'entreprise. Dans le domaine des systèmes d'information un modèle PIM représente généralement un modèle d'architecture logique.
- **PDM (*Platform Description Model*)** ou **PM (*Platform Model*)** : décrivant la plateforme sur, ou grâce à laquelle le système va être exécuté, ce modèle peut décrire des langages de programmation ou dans notre cas les caractéristiques de configuration d'un ESB (Entreprise Service Bus).
- **PSM (*Platform Specific Model*)** : le modèle PSM représente la projection du modèle PIM sur la plateforme choisie, décrite par le modèle PM.

Tout l'intérêt de l'IDM est alors de proposer la réalisation d'une plongée en abstraction au fur et à mesure des modèles créés. Cette démarche permet de ramener les différentes exigences à leur juste niveau de considération et d'éviter ainsi de se retrouver avec une activité de configuration de solution technique qui tente de prendre en compte l'ensemble des problématiques inhérentes au déploiement d'une application (exigences métier, objectifs logiques et contraintes techniques). La suite de cette partie présente les concepts de base permettant de réaliser une démarche dirigée par les modèles. Pour cela, nous commencerons par introduire les notions de modèles et métamodèles, avant de nous intéresser aux principes de transformation de modèles.



## I. Du modèle à la transformation de modèles

Comme nous avons pu le souligner en introduction les modèles tiennent évidemment une place de choix dans une démarche d'ingénierie dirigée par les modèles. En plus des quatre types de modèles listés précédemment, (Bézivin *et al.*, 2009) propose une taxonomie partielle des modèles utilisés en IDM. Cette taxonomie n'est pas exhaustive car certains types de modèles, comme les modèles de mesure et de vérification, ne sont pas cités.

La figure 4 illustre cette taxonomie, on y retrouve la notion de modèle qui représente une partie du système considéré. Ces modèles utilisés pour décrire le système doivent respecter des modèles de référence, *i.e.* un métamodèle pour un modèle et un métamétamodèle pour un métamodèle. Ces modèles de référence servent ensuite à définir des modèles terminaux servant de base soit à une transformation de modèles soit à la définition de correspondances entre les modèles.

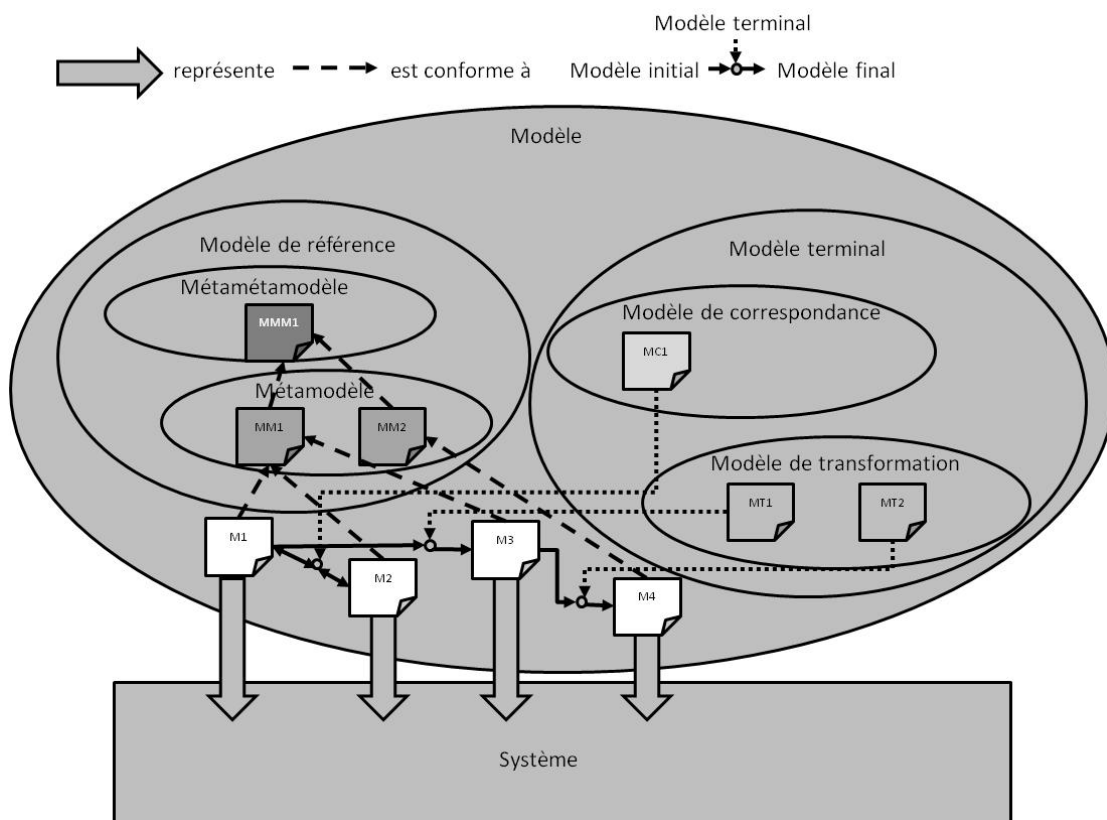


Figure 4 : Taxonomie partielle des modèles en IDM inspirée de (Bézivin et al, neptunes)

Dans cette partie, nous allons donc nous intéresser à ces différentes notions, nous commencerons par faire un tour d'horizon des définitions de la notion de modèle, appliquée au domaine de l'ingénierie, avant de nous intéresser au modèle de référence, puis au mécanisme de transformation de modèles.

### I.1 Modèle

Dans le domaine de l'ingénierie, un modèle est généralement défini comme « une représentation d'une abstraction d'une partie du monde réel, exprimée dans un langage de représentation » (Vernadat, 1999). D'après (Terrasse *et al.*, 2005) « le modèle d'un système est la spécification formelle des fonctions, de la structure et/ou du comportement de ce système dans son environnement, dans un certain but »

Un modèle est donc indissociable de la syntaxe utilisée pour le représenter. Celle-ci dépend de l'intention de la personne qui modélise. Modéliser un système, dans un langage donné, ne permet pas (et ce n'est pas le but d'une telle démarche) de représenter toutes les facettes de ce système (statique / dynamique, fonctionnelle / structurelle...). Puisque modéliser est une activité d'abstraction, un modèle donné ne concernera simultanément qu'une ou plusieurs facette(s) du système, en fonction de l'objectif recherché, c'est-à-dire en fonction du point de vue adopté. Un système pourra donc être décrit par un ensemble de modèles. En fonction des points de vue choisis, les modèles représentant un même système pourront être totalement différents. L'exemple suivant, tiré de (Bézivin *et al.*, 2004) et représenté par la figure 5, illustre nos propos. Le système « France métropolitaine » est modélisé selon trois de points de vue différents : « administratif », « fluvial » et « infrastructures routières ». Les trois modèles obtenus, chacun correspondant à un point de vue, ne représentent pas la même information et par conséquent ne représente pas le système de la même façon.

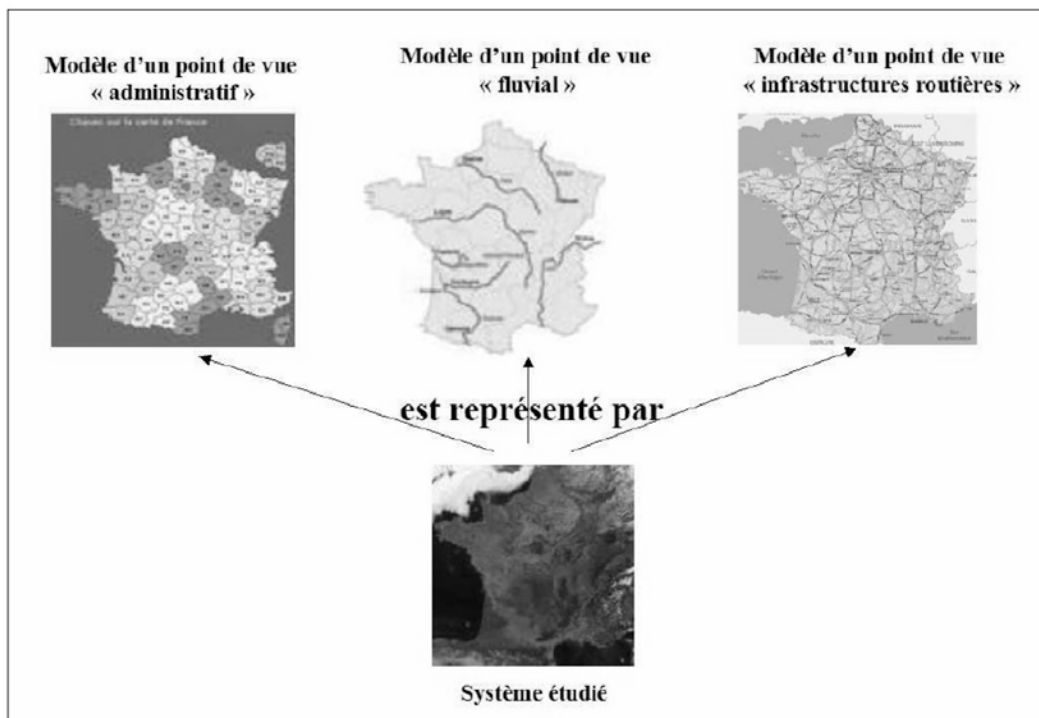


Figure II- 5 : illustration de la notion de modèle (Bézivin *et al.*, 2004)

Dans le cadre d'une démarche IDM, le « langage de représentation » du modèle doit être sans équivoque de façon à pouvoir être exploitable par différentes catégories d'utilisateurs, aussi bien humaine que logicielle. Reprenons l'exemple précédent et plus précisément le modèle d'infrastructures routières. Au niveau de ce modèle, les infrastructures routières sont représentées par des lignes différentes selon le type de route (autoroute, nationale, départementale, etc.). De plus, ces lignes ne sont pas placées n'importe où, une ligne ne pourra pas se situer au milieu de l'océan et plusieurs lignes peuvent se croiser au niveau de la représentation de certaines villes. Nous pouvons donc dire que le « langage de représentation » du modèle a besoin de définir à la fois une composante sémantique (correspondance entre le symbole et l'objet) et syntaxique (relation entre les composants). C'est tout l'intérêt du métamodèle.

## I.2 Modèle de référence : Métamodèle

Un métamodèle est défini par l'OMG comme étant: « un modèle qui définit le langage utilisé pour exprimer un modèle » (OMG, 2002). Il s'agit donc d'un modèle de langage de modélisation. Plus pragmatiquement, un métamodèle est « une description de tous les concepts d'un langage. La métamodélisation permet de définir directement les concepts manipulés, leur sémantique et la syntaxe du langage liée à l'utilisation de ces concepts » (Chapron, 2006). Cette définition souligne l'intérêt de l'utilisation de métamodèles afin de définir un langage de représentation compréhensible par différents utilisateurs. Si ces définitions placent le modèle et le métamodèle au même niveau, ce n'est pas le cas d'autres auteurs qui proposent la définition suivante : « un métamodèle est un modèle dont les instances sont des modèles » (Bataille *et al.*, 2001). Cette montée en abstraction au niveau du métamodèle est particulièrement intéressante par rapport à l'utilisation d'une « technologie objet ».

Nous faisons donc le choix d'adopter la vision proposée par (Bézivin *et al.*, 2004) qui définit la relation qui unit le système étudié à son modèle par « **est représenté par** », et la relation qui unit un modèle à son métamodèle par « **est conforme à** ».

Nous venons de voir qu'un modèle et un métamodèle ne se situe pas au même niveau d'abstraction. Cependant, ce métamodèle peut correspondre à un modèle conforme à autre métamodèle. On parle alors de métamétamodèle défini par l'OMG (OMG, 2002) comme un « modèle qui définit le langage utilisé pour exprimer un métamodèle ». Suite à cette définition nous voyons qu'il est possible d'avoir une infinité de (méta)<sup>n</sup>métamodèle. La figure ci-dessous inspirée de (OMG, 2002) représente les différents niveaux de modélisation dont la boucle définit l'infinité de niveaux possibles.

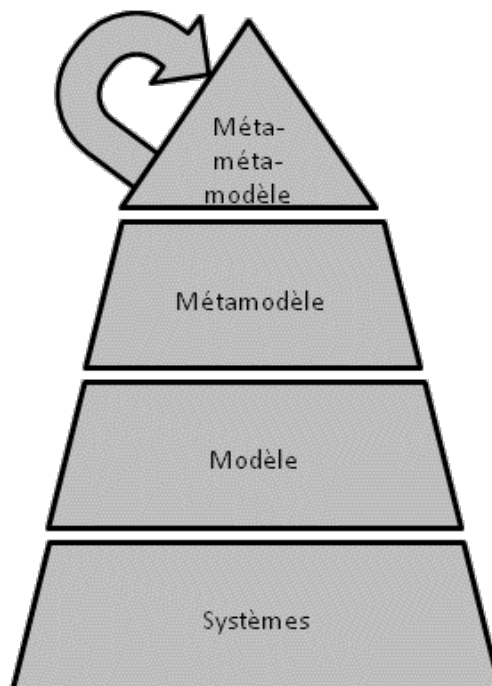


Figure 6 : Relation entre les modèles et métamodèles (OMG, 2002)

## I.3 Transformation de modèles

### I.3.1 Mécanisme de transformation de modèles

La « transformation de modèles » a déjà fait l'objet d'une littérature conséquente (OMG, 2003), (D'Antonio, 2005), (Akehus *et al.*, 2002), (Kleppe *et al.*, 2003), (Touzi, 2007), (Combemale, 2008). Il s'agit d'un concept clé d'une démarche IDM.

D'un point de vue formel, la transformation a été définie de la manière suivante (D'Antonio, 2005) : soit  $A$  un modèle et  $t : \text{MOD} \rightarrow \text{MOD}$  une fonction ( $\text{MOD}$  désigne l'ensemble des modèles),  $t$  est alors une fonction de transformation de modèle.  $B = t(A)$  est alors le modèle  $A$  transformé par  $t$  (mathématiquement,  $B$  est l'image de  $A$  par la fonction  $t$ ). Plus pragmatiquement, (Jouault, 2006) définit une transformation de modèles comme « une opération qui crée automatiquement un ensemble de modèles cible à partir d'un ensemble de modèles source ».

Une transformation s'appuie sur des « Règles de transformation ». La mise en œuvre de ces règles est réalisée par le « processus de transformation ». Cette opération n'est en rien automatique : tout le processus est à spécifier et à outiller avant d'être exécuté. La figure 7 (Touzi, 2007), illustre le mécanisme de transformation. Les règles de transformation sont établies entre le métamodèle source et le métamodèle cible. Le processus de transformation prend en entrée un (ou des) modèle(s) conforme(s) au métamodèle source et génère, en fonction des règles de transformation, un (ou des) modèle(s) conforme(s) au(x) métamodèle(s) cible(s).

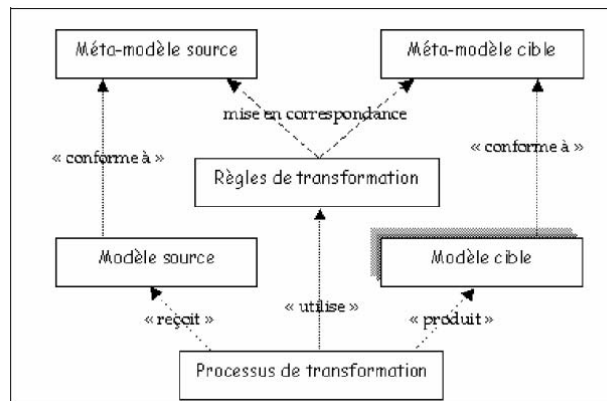


Figure 7 : Processus de transformation de modèles (Touzi, 2007)

### I.3.2 Les règles de transformation

Les règles de transformation ont par conséquent une place prépondérante dans le mécanisme de transformation de modèle. Dans (Levendovski *et al.*, 2002), ces règles de transformation sont assimilées à la notion de « mapping ». Dans (Deguil, 2008), l'auteur propose quatre types de mappings de modèles :

les mappings de type « 1 : 1 » (type 1) utilisent des relations qui associent un élément d'un modèle source à un et un seul élément d'un modèle cible.

Les mappings de type « 1 : n » (type 2) qui peuvent associer un élément du modèle source à plusieurs éléments du modèle cible.

Les mappings de type « m : 1 » (type 3) peuvent associer plusieurs éléments du modèle source à un seul élément du modèle cible.

Les mappings de type « m : n » (type 4) associent plusieurs éléments du modèle source à plusieurs éléments du modèle cible.

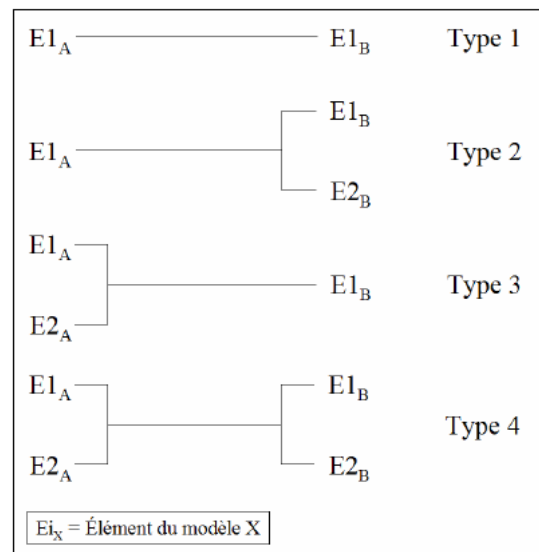


Figure 8 : différents types de mappings (Deguil, 2008)

Un autre type de mapping est proposé dans (Combemale, 2008) : les « mappings de mise à jour » ou encore appelée « transformation sur place » qui consistent à modifier un modèle par ajout, modification ou suppression d'une partie de ses éléments. Dans ce type de transformation, les métamodèles source et cible sont confondus. Une telle transformation agit directement sur le modèle source sans créer de modèle cible.

### I.3.3 Taxonomie des transformations de modèle

Partant de la nature des métamodèles source et cible, on distingue les transformations dites endogènes et exogènes combinées à des transformations dites verticales et horizontales.

Une transformation est dite endogène si les modèles cible et source sont conformes au même métamodèle, exogène dans le cas contraire (Caplat *et al.*, 2003). Une transformation est dite *verticale* si elle met en jeu différents niveaux d'abstraction (par exemple le passage de PIM vers PSM), alors qu'une transformation est dite *horizontale* si elle demeure à un même niveau d'abstraction (par exemple la transformation d'un modèle de processus exprimé sous la forme d'un diagramme d'activité en un autre modèle de processus représenté sous la forme d'un diagramme EPC).

Dans (Combemale, 2008), l'auteur propose une taxonomie de transformation de modèle, représentée par la figure 9, illustrant chacune de combinaisons possibles de transformation de modèle : la restructuration, la normalisation et l'intégration de patrons sont des exemples de transformation endogène et horizontale alors que le raffinement est une transformation endogène et verticale. La migration de plateformes et la fusion de modèles sont des exemples de transformation exogène et horizontale, tandis que la rétro-conception est un exemple de transformation exogène et verticale.

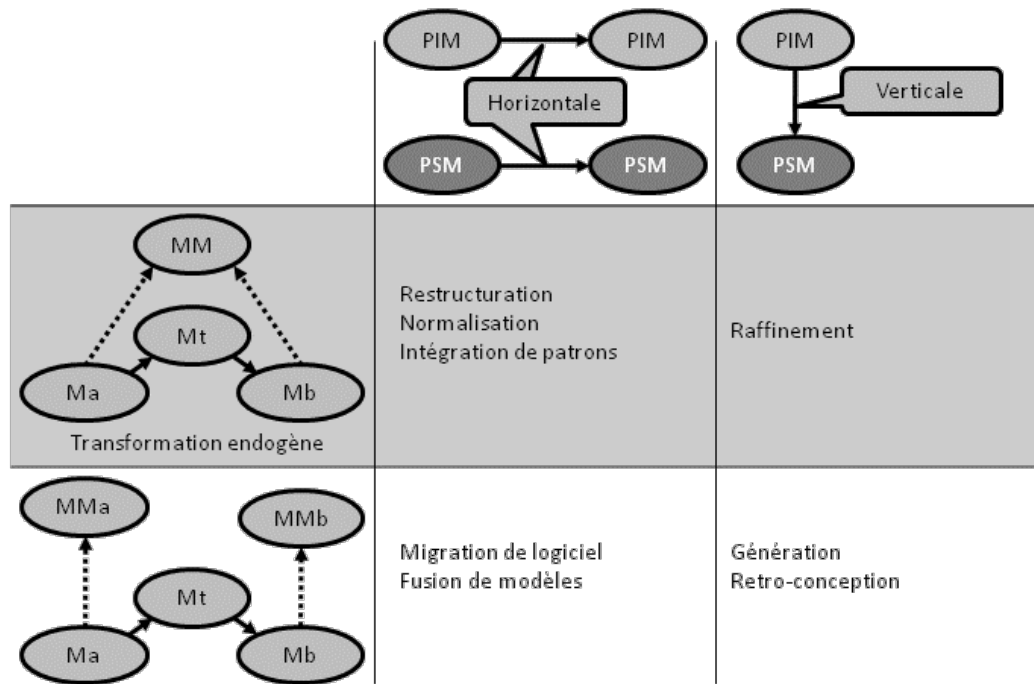


Figure 9 : Taxonomie des transformations de modèles (Combemale, 2008)

### I.3.4 Synthèse sur la transformation de modèle

Nous venons de voir qu'une transformation de modèle partait d'un modèle source, conforme à un métamodèle source, afin d'obtenir un modèle cible, conforme à un métamodèle cible, en fonction des règles de mapping portant sur les éléments des deux métamodèles. Cependant, les règles de mapping ne peuvent être réalisées qu'à la condition que le métamodèle source et le métamodèle cible aient des concepts partagés. Par concepts partagés, nous ne sous-entendons pas que les mêmes concepts doivent apparaître selon la même syntaxe et sémantique au niveau des deux métamodèles, nous parlons de concepts pouvant au minimum être mis en relation. Cet ensemble de concepts partagés devient alors l'espace de définition des règles de mappings.

Les métamodèles source et cible n'étant pas limités à l'espace des concepts partagés, lors d'une transformation une partie du modèle source est potentiellement perdue. En effet, un modèle étant conforme à un métamodèle, le modèle source contient des éléments définis par la partie spécifique (*i.e.* non partagé) du métamodèle. Ainsi une partie des connaissances ne se retrouveront pas dans le modèle cible. Ces éléments devront être sauvegardés afin de ne pas perdre de connaissance. La réciproque de ce raisonnement implique que le modèle cible peut potentiellement nécessiter une étape d'enrichissement.

La figure ci-dessous (Truptil *et al.*, 2010a) synthétise, la définition d'une transformation de modèle que nous conserverons tout au long de ce manuscrit.

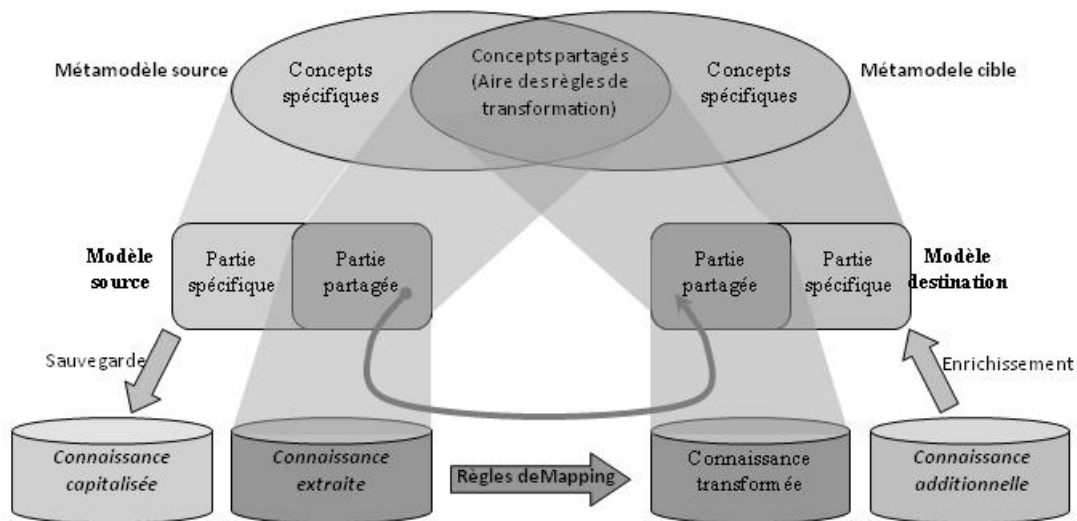


Figure 10 : Principes de transformation de modèles (Truptil *et al.*, 2010a)

Dans le cadre de ces travaux de thèse, nous avons choisi deux approches pour la réalisation de transformation de modèles en fonction de la nature endogène ou exogène de la transformation. Dans le cas d'une transformation exogène, nous avons choisi le langage ATL, présenté plus en détail par la suite. Pour le cas des transformations endogènes, notre choix s'est porté sur l'utilisation d'ontologies et de moteur d'inférence.

## I.4 Les ontologies

L'étymologie du terme ontologie renvoie à la « science de l'être », c'est-à-dire, selon le dictionnaire Robert à « la partie de la métaphysique qui s'applique à l'être, indépendamment de ses déterminations particulières » permettant « une explication systémique de l'existant ». Cette notion, éloignée du domaine des systèmes d'information, a été reprise par la communauté travaillant sur l'ingénierie de la connaissance. Dans le cadre de ces travaux de thèse, nous nous intéresserons uniquement à la notion d'ontologie appliquée au domaine de l'ingénierie de connaissance.

De nombreuses définitions du terme ontologie sont disponibles au sein de la littérature. Nous pouvons néanmoins citer l'une des premières : (Neches *et al.*, 1991) décrit une ontologie comme « la définition des termes et relations de bases comprenant le vocabulaire d'un domaine d'étude ainsi que les règles pour définir les extensions au vocabulaire ».

(Gruber, 1993) définit une ontologie comme : « une spécification explicite d'une conceptualisation » Cette définition englobe certaines caractéristiques intéressantes d'une ontologie (Gruber, 1995), (Grimm *et al.*, 2007) :

- Un aspect **formel** : une ontologie est exprimée dans un langage de représentation des connaissances qui fournit une sémantique formelle. L'ontologie doit être compréhensible par une machine.
- Un aspect **explicite** : les types de concepts utilisés et les contraintes sur leur utilisation sont explicitement définis. Une ontologie exprime explicitement les connaissances pour les rendre accessibles par les machines. Les notions qui ne sont pas explicitement exprimées dans l'ontologie

ne pourront pas être interprétées par un machine même si, au nom du bon sens, les Humains les prennent pour acquises.

- Un aspect de **partage** : Puisqu'une ontologie capture une connaissance consensuelle et reflète un accord sur une conceptualisation de domaine, elle peut difficilement être créée par un seul acteur.
- Un aspect de **conceptualisation** : une ontologie spécifie la connaissance de manière conceptuelle en termes de symboles représentant des concepts et leurs relations. D'autre part, elle décrit une conceptualisation en termes généraux afin de couvrir l'ensemble des situations pouvant potentiellement se produire.
- Une **spécificité de domaine** : une ontologie se limite à la connaissance d'un domaine particulier. Plus le domaine particulier est restreint, plus l'ontologie est définie en détail par rapport à ce domaine plutôt que de couvrir un large éventail de sujets connexes.

(Rajsiri, 2008) résume les raisons générales de la construction des ontologies comme suit :

- Partager une compréhension commune entre des personnes ou des agents logiciels.
- Permettre la réutilisation des connaissances d'un domaine.
- Proposer une définition explicite du domaine : l'explicite spécification de la connaissance du domaine pouvant être utile pour les nouveaux utilisateurs qui doivent apprendre les concepts.
- Séparer les connaissances du domaine de la connaissance opérationnelle.
- Analyser les connaissances du domaine : l'analyse formelle des termes est extrêmement précieuse à la fois pour réutiliser ou étendre des ontologies existantes (McGuinness et al. 2000).

Une autre utilisation des ontologies, proposée par (Grimm et al, 2007), est l'intégration d'information. Il explique que les ontologies doivent souvent manipuler des sources d'information hétérogènes à condition que chacune respecte la définition explicite du domaine proposée par l'ontologie.

Nous pouvons donc noter des similitudes entre les définitions d'un métamodèle et d'une ontologie. En effet, dans les deux cas, les concepts d'un domaine doivent être définis explicitement et les informations manipulées par l'ontologie doivent respecter leur définition explicite (on retrouve alors des relations similaires à celles unissant un modèle à son métamodèle). L'intérêt de l'ontologie vis-à-vis de la manipulation de données issues de sources hétérogènes, explique son utilisation dans le cas d'une transformation endogène. Cependant, l'exécution des règles de transformation ne peut pas être directement réalisée par l'ontologie. Il est donc nécessaire d'utiliser à la fois une ontologie (pour la description de la connaissance) et un moteur d'inférence (pour la dynamique de transformation) afin de réaliser une transformation de modèle.

L'inférence est l'opération qui consiste à admettre une proposition en raison de son lien avec une proposition préalable tenue pour vraie. C'est un terme général dont les mots raisonnement, déduction, induction, etc., sont des cas spéciaux. Le moteur d'inférence est le programme qui réalise les déductions logiques d'un système expert à partir d'une base de connaissances et d'une base de règles.



## Annexe E : Solutions Technologiques

### I. De la Collaboration au Système d'information

Lors de la définition de la notion d'interopérabilité, nous avons évoqué le fait que l'interopérabilité portait à la fois sur le niveau métier de la collaboration ainsi que sur les caractéristiques des systèmes d'information. Le lien entre les niveaux métier et technique est loin d'être trivial. Cependant, une architecture orientée service (SOA) permet de réaliser ce lien de façon plus intuitive qu'un autre type architecture de système d'information. Dans cette partie, nous allons donc nous intéresser à cette notion de SOA ainsi qu'aux notions connexes permettant la mise en place de l'interopérabilité.

#### I.1 L'architecture orientée service (SOA)

L'architecture orientée services (SOA) est le terme utilisé pour désigner un « style d'architecture organisé à partir de services métiers communs mutualisés pour un ensemble de lignes métiers ou d'applications » (Raymond, 2007). Toutes ces applications logicielles sont communément appelées « services ». Ces services sont ensuite exécutés selon un ordre préétabli lors de l'apparition d'un événement. Ces éléments de base du SOA identifiés par (Vernadat, 2006), sous les termes « événements », « services » et « processus », doivent être compris et maîtrisés aux niveaux « métier » et « applicatif ». C'est cette correspondance entre le niveau « métier » (dans notre cas, la description d'une collaboration entre organisations), et le niveau « applicatif » (dans notre cas, le système d'information), qui est intéressante dans le but de mettre en place un système de systèmes d'information support du système de systèmes de réponse à la crise. Il est à noter que ce type de correspondance n'est pas forcément de type 1-1, car plusieurs services dit « applicatif » peuvent être exécutés selon une certaine composition afin de réaliser un seul service dit « métier ». Cette vision de SOA montre le lien fort entre le système d'information et les processus métier d'une entreprise qui a adopté cette architecture. Dans ce cas, un processus métier peut, pour s'exécuter, faire appel à l'image sur le système d'information d'un ensemble de services « métiers ». Afin d'alléger la lecture, dans la suite de ce chapitre nous entendrons par « service », un service applicatif.

La SOA peut être vue comme, ou tout du moins contribuer à développer, une architecture agile de système d'information. En effet, la SOA consiste à substituer la découpe strictement applicative d'un système d'information par une structuration en services plus réduits et potentiellement plus simple à faire évoluer (Raymond, 2007). Par ailleurs, d'après (Monfort *et al.*, 2004), cette découpe facilite les échanges inter-entreprises. Nous pouvons donc dire que SOA est une réponse conceptuelle adaptée à la problématique d'interopérabilité, d'une part par l'agilité et l'accessibilité qu'elle apporte au système d'information, et d'autre part par la possibilité qu'elle offre de projeter, sur les systèmes d'information, une collaboration définie au niveau métier. Cependant SOA ne répond pas en elle-même aux problématiques d'hétérogénéité sémantique. Quelques travaux de recherche essayent de l'enrichir afin de palier cette lacune, tel que Semantic Markup for Web Services (OWL-S), Semantic Annotations for WSDL and XML Schema (SA-WSDL) ou encore WSMO-Lite.

Cependant l'accessibilité des services du système d'information d'une entreprise, rendue possible par la SOA, pose des problèmes, d'une part, de concurrence, comme l'explique (Vanderhaeghen *et al.*, 2004) : « pour des raisons stratégiques liées à la compétitivité de l'entreprise, la plupart des entreprises partenaires d'un réseau collaboratif ne dévoilent pas directement leurs savoir-faire et connaissances liés à leurs processus métiers internes », et d'autre part, de confidentialité des données (Panetto, 2006) puisque « le

partage d'information entre les applications de l'entreprise pose encore des problèmes liés au secret professionnel et à la propriété intellectuelle. Le processus d'interopérabilité dépend aussi des applications mises en place, de leur ouverture et de la disponibilité de leurs modèles internes ». La SOA a donc besoin de prendre en compte les aspects de confidentialité des procédés et des informations.

Dans le cadre d'une collaboration inter-entreprises, ce besoin de confidentialité est particulièrement important. Afin de répondre à ce besoin, un système d'information doit être composé de deux parties (Bauer et al, 2006) :

- **Une partie privée** : correspondant à la partie totalement propriétaire du système d'information. C'est dans cette partie que les données, applications, organisation et même stratégie de l'entreprise sont cachées aux partenaires externes (Vanderhaeghen et al, 2004).
- **Une partie publique ou abstraite** : cette partie du système d'information, visible par les partenaires, définit les fonctionnalités mises à disposition, ainsi que pour chacune les données d'entrées et de sorties sans en détailler les spécificités.

Un système d'information basé sur une architecture SOA et pouvant donc exposer chacun de ces services doit donc intégrer cette notion de confidentialité. Le découpage en parties publique/privée peut se projeter sur chaque service du système d'information. La partie privée étant par définition propre à chaque service, nous allons maintenant nous intéresser à la description de la partie publique : le contrat de service.

## I.2 Caractéristiques d'un service de système d'information

Un service d'un système d'information basé sur une architecture SOA, est caractérisé par un point d'accès. Ce point d'accès correspond à la partie publique du service et à chacun de ces point d'accès est associé un contrat de service. Ce contrat de service précise à la fois la structure des messages d'entrée que peut recevoir le service (via ce point d'accès), ainsi que la structure des messages de sortie qu'il produira.

Ce contrat de service s'inspire directement du modèle des contrats professionnels de service. Il s'agit d'un document qui développe l'ensemble des éléments permettant de décrire le service. Dans le monde des relations professionnelles, un contrat de service est un document comportant plusieurs parties et dont les éléments terminaux sont généralement appelés articles et clauses. Dans le monde des SOA, les éléments du contrat de service décrivent les fonctions du service, les messages échangés, le protocole de communication employé ainsi que l'adresse physique du service.

Dans un but de collaboration, il est important que ce contrat de service soit, en même temps, lisible par l'homme et exploitable par des logiciels. C'est dans cette optique que des langages de description de services ont été proposés, nous pouvons citer le plus communément utiliser : le Web Services Definition Language (WSDL) (W3C, 2003). Pour chaque service un document WSDL est défini afin de décrire le contrat de service. Ce document WSDL est composé de 6 éléments principaux et d'un certain nombre d'autres éléments optionnels:

- **Definitions** : cet élément contient l'ensemble des informations liées au service. Notamment le nom du service, les espaces de noms utilisés ainsi que les autres éléments décrits ci-dessous.
- **Types** : cet élément décrit tous les types de données utilisés par le service. Un type peut être simple (type chaîne de caractère, entier, etc.) ou complexe. Un type complexe est généralement composé d'un ensemble d'éléments de type simple.

- **Message** : lors d'une invocation d'une opération d'un service, un message d'entrée peut être demandé et un message de sortie peut être renvoyé. Dans cette partie du WSDL, les messages utilisés par les opérations du service sont décrits. Cette description consiste d'une part à nommer le message et d'autre part à définir l'ensemble des éléments qui le constitue. Ces éléments ont chacun un type (simple ou complexe) défini dans la partie précédente du WSDL.
- **PortType** : cette partie définit pour chaque opération le type de message d'entrée et de sortie liés à l'exécution de l'opération du service. Si pour une même opération, il est possible d'avoir différents messages d'entrée et de sortie alors chaque combinaison de messages sera sauvegardée sans un portType différents.
- **Binding** : décrit les spécifications concrètes concernant la manière dont le service sera contacté : le protocole de communication et, via le portType, les messages des opérations du service.
- **Service** : cette partie permet d'une part de définir les adresses permettant d'invoquer le service donné (*i.e.* l'adresse du point d'accès au service) et d'autre part les informations liées au protocole de communication via le Binding, qui permet via le PortType de connaître les messages des opérations. Messages dont la structure est définie selon des types simples ou complexes de données.

La figure 11 présente un exemple de WSDL pour un service nommer « *Eteindre incendie de faible ampleur* ». Ce service ne propose qu'une opération portant le même nom que le service. Cette opération prend en entrée un message nommé « *Eteindre incendie de faible ampleur Request* » et renvoie en sortie un message nommé « *Eteindre incendie de faible ampleur Response* ». La dernière partie de ce WSDL, correspondant à la partie Service, est composée, d'une part, de la description de l'adresse où contacter le service à savoir [http://192.168.0.4:9205/Eteindre\\_incendie\\_faible\\_ampleurEndpoint](http://192.168.0.4:9205/Eteindre_incendie_faible_ampleurEndpoint) et, d'autre part, du binding utilisé. Ce binding est décrit dans la partie juste au dessus de la partie Service. On y retrouve le protocole de communication SOAP, le nom de l'opération et le renvoi vers le PortType. Ce dernier étant décrit sur la partie juste au dessus du binding, renseigne les informations portant sur les messages utilisés pour cette opération (messages décrits juste avant). La description des éléments des messages est réalisée grâce aux types d'éléments définis.

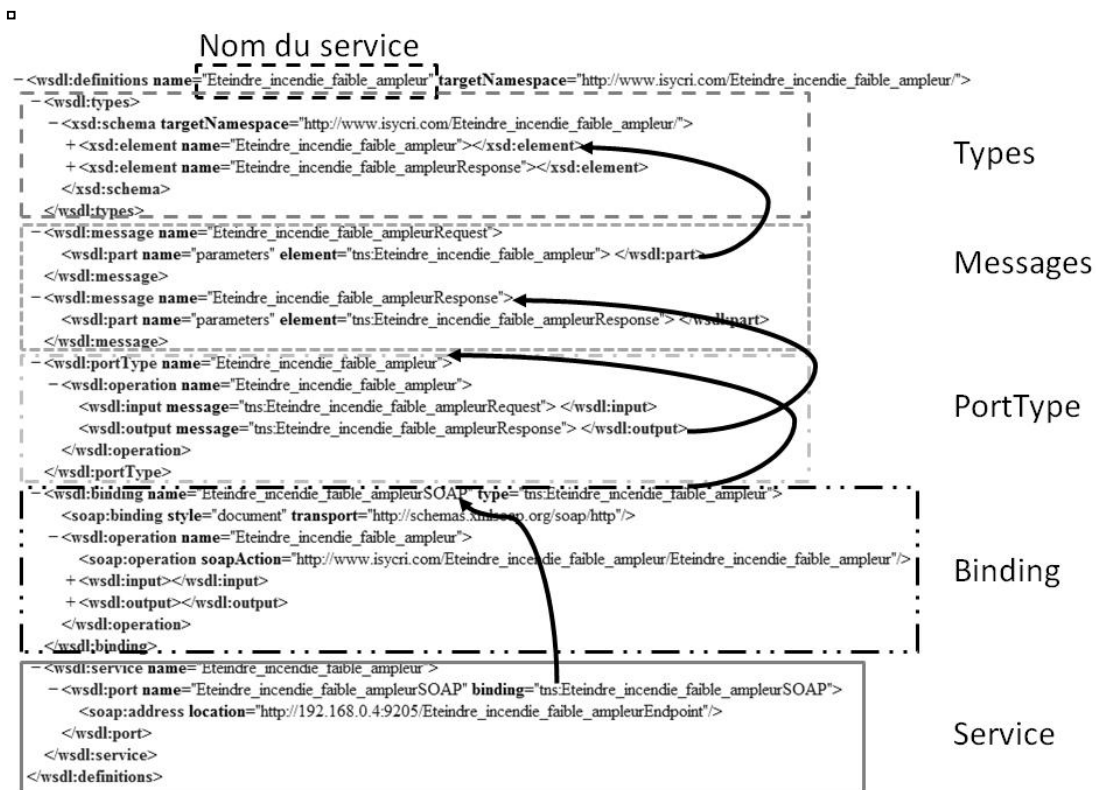


Figure II- 11 : Exemple de WSDL

### I.3 Entreprise Service Bus : Une technologie de médiation

Nous venons d'expliquer que le type d'architecture SOA nous permettait de faire le lien entre les services dits métiers et les services d'un système d'information. Cette correspondance est l'élément clé qui permet de faire correspondre un processus collaboratif, défini au niveau « métiers », avec un processus « technique » devant être exécuté par le système d'information de médiation. Cependant une architecture SOA ne suffit pas à créer un médiateur agile. Ce médiateur a pour objectif de (i) piloter un processus collaboratif pouvant évoluer dans le temps, (ii) gérer les données, (iii) manager les applications. Pour cela, le médiateur a besoin :

- De se connecter de façon dynamique aux services de différents systèmes d'information.
- De pouvoir transformer de l'information.
- De pouvoir exécuter un processus.

Nous avons décidé d'utiliser comme support technologique répondant à ces fonctionnalités un bus de service d'entreprise<sup>3</sup> (ESB). Un ESB, comme son nom l'indique, est centrée sur la notion de bus. La distribution des services constitue le principal avantage d'une telle infrastructure (EBM, 2005). L'ESB permet une connexion indirecte entre différents services de systèmes d'information : à partir du moment où un service d'un SI est connecté au bus, il est en relation potentielle avec l'ensemble des services déjà connectés, même si ces services sont issus de technologies différentes. Cependant le fait que les services des systèmes d'information puissent être issus de technologies différentes peut engendrer un problème de connectivité. Afin de résoudre ce problème, l'ESB propose un ensemble de connecteurs basés sur la

<sup>3</sup> Entreprise Service Bus

norme J2CA (J2EE Connector Architecture) (EBM, 2005) autorisant les connexions de services largement hétérogènes. Finalement, comme le montre la figure 12, l'ESB permet d'éviter la création de connexion point à point entre les services, et autorise une connexion rapide et dynamique des services.

Enfin un ESB doit offrir des services techniques, comme la transformation des messages, le routage basé sur le contenu et éventuellement l'orchestration des services (EBM, 2005). Dans (ISyCri, 2008), l'orchestration est définie comme « une expression des conditions et des enchaînements des invocations aux services. Dans le cadre des architectures orientées services (SOA), ces orchestrations sont vues comme des compositions de services définissant de nouveaux services. Une orchestration définit un workflow entre les services. Dans le cadre d'ISyCri, ce workflow doit être observable et contrôlable. ». L'ESB doit donc être capable d'exécuter un processus collaboratif défini dans un langage que son moteur d'orchestration puisse dérouler. Nous présenterons dans le chapitre IV de cette thèse ce langage : BPEL.

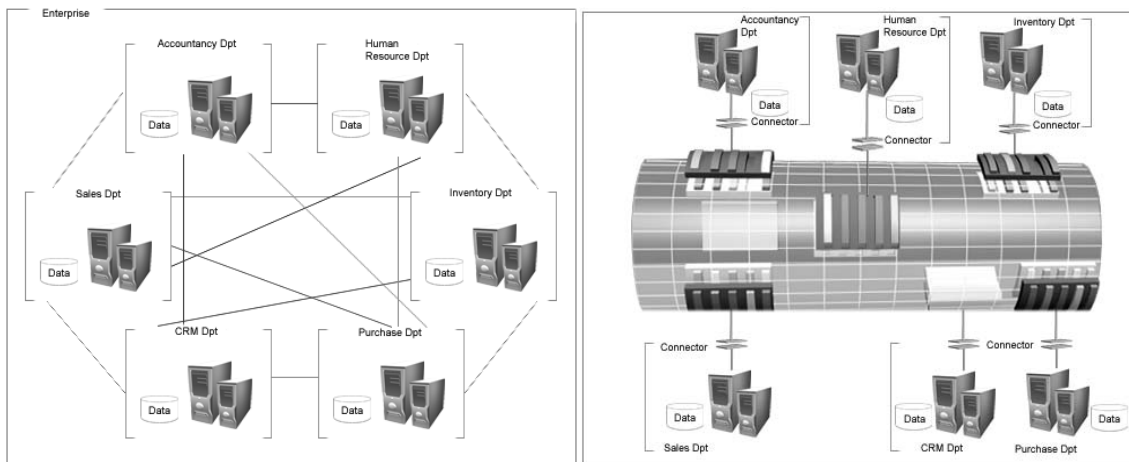


Figure 12: connexion point à point entre les services VS connexion entre services via l'ESB (EBM, 2005)

## II. Technologie pour la transformation de modèles :

Dans cette partie, nous allons nous intéresser aux différentes technologies que nous avons choisies d'utiliser lors de ces travaux de thèse. Cette partie est décomposée en trois sous-parties : Les langages utilisés pour décrire les métamodèles : UML et OWL puis les différents langages de description des règles de transformation : ATL et SWRL. Mais avant de les présenter, nous commencerons par une présentation succincte du langage de base de l'ensemble des modèles que nous utiliserons : le langage XML (Extensible Markup Language).

### II.1 XML

Dans le domaine de l'IDM, et plus précisément de la transformation de modèle, la structuration de données prend une place prépondérante. En effet, le lien entre métamodèle et modèle, ainsi qu'entre éléments au sein d'un même modèle nécessite de pouvoir organiser et structurer les informations dans le but de les réutiliser. Dans cette optique, XML (Extensible Markup Language) constitue un langage de structuration de données, adapté aux transformations de modèles.

XML est un langage de balisage extensible mis au point par le XML Working Group sous l'égide du World Wide Web Consortium (W3C). XML est un standard qui sert de base pour créer des langages balisés spécialisés. A ce titre, on peut le qualifier de « méta langage ». Il est suffisamment général pour que

les langages basés sur XML puissent être utilisés pour décrire toutes sortes de données et de textes. Son objectif est, dans un échange entre systèmes informatiques, de transférer des données et leurs structures.

Cependant, chaque utilisateur pouvant créer ses propres balises, chargées d'un certain sens sémantique, les risques de « collision de noms de balise » deviennent plus importants. Afin d'éviter ces collisions une extension a été ajoutée au langage XML : XML Namespaces permet ainsi de créer des espaces de nommage. Les espaces de noms d'XML permettent de qualifier de manière unique des éléments et des attributs. On sait alors à quel domaine de définition se rapporte un objet et comment il doit être interprété, selon sa spécification. Différencier des espaces de nommage permet de faire coopérer, dans un même document, des objets ayant le même nom, mais une signification différente.

## II.2 Langage de description de métamodèle

### II.2.1 UML

Le langage UML (Unified Modeling Language) a été développé dans le but de répondre à quatre objectifs (Booch *et al.*, 2004) et (Roques, 2004) :

- représenter complètement un système en utilisant plusieurs diagrammes permettant chacun de modéliser un point de vue du système.
- Etablir un couplage explicite entre les concepts et un code exécutable.
- Prendre en compte les facteurs d'échelle qui sont inhérents aux systèmes complexes et critiques.
- Créer un langage de modélisation utilisable à la fois par les humains et les machines.

Afin de réaliser ces objectifs, le langage UML permet de décrire un système selon différentes vues complémentaires, supportées par des diagrammes. Chaque type de diagramme UML possède une structure (les types des éléments de modélisation qui le composent sont prédéfinis) et véhicule une sémantique précise. Les vues du système que supportent ces diagrammes peuvent être classées de bien des façons (statique / dynamique, logique / physique, architectural / structurel, comportemental / fonctionnel, etc.).

Par ailleurs, un avantage notable d'UML (celui qui nous intéresse plus particulièrement dans le cadre de nos travaux) concerne sa capacité à servir de langage de métamodélisation. En effet, UML permet de créer des langages de modélisation, ce qui revient à réaliser un métamodèle. Les métamodèles sont alors représentés par des diagrammes de classes. Nous donnerons un exemple de métamodèle décrit en UML dans le chapitre II.

### II.2.2 OWL

Le langage OWL est une recommandation du W3C. Le langage OWL est un langage basé sur XML avec des espaces de nommages spécifiques traduisant l'élaboration en plusieurs phases du langage. Nous ne ferons qu'un bref retour sur l'historique du langage OWL, pour plus de détail, voir (Rajsiri, 2009).

Les ontologies sont apparues dans le domaine de l'informatique, et plus particulièrement d'Internet, afin de résoudre des problèmes sémantiques. Pour cela, le W3C a proposé le langage RDF (Lassila et al, 1999) comme un langage de description des ressources disponibles sur Internet. Une première extension a été proposée au langage RDF : RDF Schéma ou RDFS (Berners-lee et al, 2001). Cette extension ajoute des relations entre concepts, représentées par des propriétés. Cependant le couplage RDF et RDFS, ne

suffisait pas pour décrire une ontologie puisqu'il manquait la possibilité d'ajouter des équivalences, des relations transitives, etc.

Afin de répondre à ces lacunes, le langage OWL a été mis au point. Le langage OWL se compose de trois sous-langages qui proposent une expressivité croissante : OWL Lite, OWL DL, OWL Full. Chacun est une extension de son prédécesseur.

- **OWL Lite** répond à des besoins de hiérarchie de classification et de fonctionnalités de contraintes simples de cardinalité 0 ou 1. Ces cardinalités 0 ou 1 correspondent à des relations fonctionnelles, par exemple, « une personne a une adresse ». Toutefois, cette personne peut avoir un ou plusieurs prénoms, OWL Lite ne suffit donc pas pour cette situation.
- **OWL DL** concerne les utilisateurs qui souhaitent une expressivité maximum couplée à la complétude du calcul (cela signifie que toutes les inférences seront assurées d'être prises en compte) et la décidabilité du système de raisonnement (c'est-à-dire que tous les calculs seront terminés dans un intervalle de temps fini). Ce langage inclut toutes les structures OWL avec certaines restrictions, comme la séparation des types : une classe ne peut pas être simultanément un individu et une propriété. Il est nommé DL car il correspond à la logique descriptive.
- **OWL Full** se destine aux personnes souhaitant une expressivité maximale. Il a l'avantage de la compatibilité complète avec RDF/RDFS, mais son haut niveau de capacité de description s'accompagne de l'impossibilité de garantir la complétude et la décidabilité des calculs liés aux inférences sur l'ontologie.

## II.3 . Le choix d'un outil de transformation

Dans cette partie, nous allons expliquer nos choix concernant les outils permettant de faire des transformations de modèles. Dans le cadre de nos travaux, deux types de transformation de modèles ont été réalisés : une transformation endogène horizontale au niveau du CIM de la démarche et des transformations verticales exogènes, chacune utilisée lors d'un passage d'un niveau à un autre. (CIM  $\square$  PIM, PIM  $\square$  PSM, PSM  $\square$  Déploiement).

Ainsi l'outil dédié aux transformations horizontales endogènes doit permettre de :

- déduire des modèles des informations ou de la connaissance en se basant sur un fichier de correspondance.
- construire à partir des informations à disposition un nouveau modèle respectant le métamodèle source.

Quant à l'outil dédié aux transformations verticales exogènes, il doit permettre de :

- définir des métamodèles qu'ils soient « faits maison » (par exemple, le métamodèle de processus collaboratif), standard (par exemple, le formalisme UML), ou encore issus de profils UML (par exemple, le métamodèle d'architecture SOA défini par (Touzi, 2007)),
- définir des règles de correspondances directes entre les éléments, *i.e.* lorsque l'instance du concept A du métamodèle source est présent il est transformé en instance du concept B du métamodèle cible,

- définir des règles correspondant à des fonctions, *i.e.* contrairement aux précédentes elles ne sont utilisées que dans certains cas (par exemple, si un élément A vérifie certaines conditions pré-établies, alors la fonction sera appelée pour l'utiliser dans une transformation vers le concept B)

Suite à l'identification de ces besoins, nous avons décidé d'utiliser les ontologies avec des règles de déduction écrites en SWRL et exécutées par le moteur d'inférence Jess pour les transformations horizontales endogènes. Ce choix se base sur :

- la possibilité de regrouper la connaissance contenue au sein de différents modèles au niveau de l'ontologie.
- La rédaction de règles de correspondance en SWRL. Ces règles permettent d'une part d'enrichir la connaissance de l'ontologie, notamment au niveau des liens de correspondances sémantiques entre les instances d'un concept et d'autre part, de créer une nouvelle connaissance à partir de la connaissance disponible,
- la possibilité d'extraire de l'ontologie certaines connaissances grâce à l'utilisation de feuilles de styles.

Le langage ATL (ATLAS Transformation Language) a été choisi comme outil de transformation verticale exogène. Ce choix se base sur une étude de comparaison des outils de transformation faite par (Salem et al., 2008). Il ressort de cette étude que seul ATL répond aux critères précédemment cités. Il est à noter que nous nous sommes limités à des d'outils proposés comme plug-in eclipse open source, dans l'optique de pouvoir les intégrer à un prototype.

### II.3.1 Transformation exogène : ATL

ATL est un langage à vocation déclarative, mais qui est en réalité hybride et permet de faire des transformations de modèles aussi bien endogènes qu'exogènes. Les outils de transformation liés à ATL sont intégrés sous forme de plug-in ADT (ATL Development Tool) pour l'environnement de développement Eclipse.

Différents types de règles existe en ATL, nous pouvons notamment citer les règles de type déclarative, impérative et enfin les fonctions supports.

Une règle déclarative d'ATL, aussi appelée Matched Rule, est désignée par un nom, est utilisée pour chaque instance d'un concept du métamodèle source et crée pour chacune de ces instances une ou plusieurs instances d'un ou plusieurs concepts du métamodèle cible. Depuis la version 2006 d'ATL, de nouvelles fonctionnalités ont été ajoutées telles que l'héritage entre les règles et la possibilité d'avoir plusieurs modèles en entrée.

Le style impératif d'ATL est supporté par deux constructions différentes. En effet, on peut utiliser soit des règles impératives appelées Called Rule, soit un bloc d'instructions impératives utilisé avec différents types de règles. Contrairement à une Matched Rule, une Called Rule peut être vue comme une fonction, elle est donc appelée explicitement en utilisant son nom et en initialisant ses paramètres.

Enfin, ATL fournit le mécanisme de Helpers. Ce mécanisme peut être vu comme des fonctions élémentaires à partir desquelles il est plus facile de définir les autres types de règles. Il sert donc à éviter la redondance de code et la création de règle de grandes expressions. Ils font référence à des procédures stockées (qui peuvent ainsi être appelées dans une règle lorsqu'il y en a besoin). Les Helpers induisent donc une meilleure lisibilité des programmes ATL. Un helper en ATL peut recevoir des paramètres et



retourner une valeur. La figure 13 ci-dessous présente la composition d'une règle de transformation ATL. Le chapitre IV de ce manuscrit présentant en détail une transformation réalisée avec ATL.

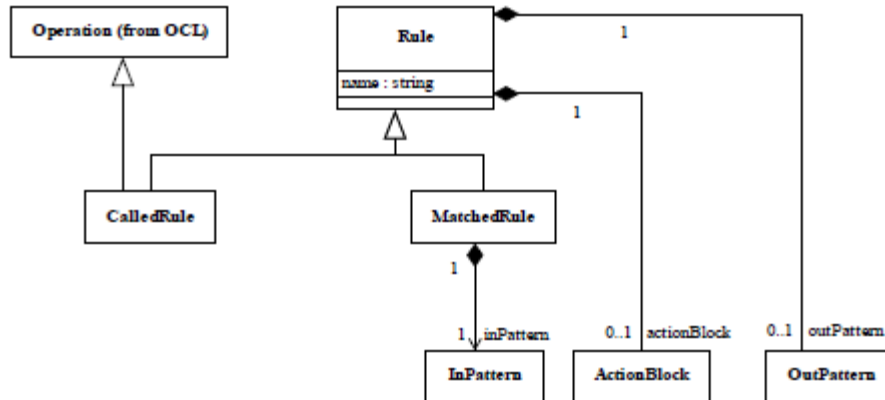


Figure 13 : Extrait du métamodèle ATL

### II.3.2 Transformation endogène : SWRL

Le langage de règles SWRL (Semantic Web Rule Language), proposé en 2004 par le W3C, est un langage qui a été créé à l'origine pour enrichir la sémantique d'une ontologie définie en OWL. SWRL permet, contrairement à OWL, de manipuler des instances au travers de variables ( $?x, ?y, ?z$ ). Il est à noter que SWRL ne permet pas nativement de créer des concepts ni des relations. Il permet simplement d'ajouter des relations déjà définies ou une valeur à un attribut suivant la satisfaction des règles. Dernièrement, certaines extensions, comme SWRL Built-in Specification, ont été proposées afin d'enrichir le langage SWRL. Elles permettent la création de nouvelles instances, l'utilisation d'opérateur logique (égale, supérieur,...), l'utilisation de fonction arithmétique (division, module,...), la manipulation de chaîne de caractère (contient, ...) tout en respectant la structure des règles en SWRL.

Les règles SWRL sont construites suivant ce schéma: antécédent  $\square$  conséquent. L'antécédent et le conséquent sont des conjonctions d'atomes. Un atome est une instance de concept, une relation OWL. Le fonctionnement d'une règle est basée sur le principe de satisfiabilité de l'antécédent ou du conséquent. Le langage SWRL est donc un langage déclaratif.

(O'Connor et al, 2005) donne un exemple de règle SWRL. Cette règle explique que si une personne a un enfant dont le sexe est masculin alors il a un fils. Cette règle nécessite la définition des concepts personne, sexe, enfant et fils dans l'ontologie. Le concept personne peut être représenté dans l'ontologie par la classe *Personne* alors que les autres peuvent être représenté par des relations *àSexe*, *àEnfant*, *àFils*. La règle en SWRL se traduirait donc par :

$$\text{Personne}(?x) \wedge \text{àEnfant}(?x, ?y) \wedge \text{àSexe}(?y, \text{« masculin »}) \rightarrow \text{àFils}(?x, ?y)$$

Le chapitre II de ce manuscrit présente précisément une transformation réalisée avec SWRL, nous illustrerons donc en détail ce langage dans ce chapitre.

## Annexe F : Règles de mappings et fichiers ATL de la transformation UMLT à BPEL

Cette annexe présente les règles de transformation du modèle d'architecture technique du SIM aux modèles de BPEL et BPELArtifact. Ces règles de mapping entre des éléments des différents métamodèles ont été réalisées. Ces règles de mapping seront appelées soit pour l'ensemble des éléments d'un même concept du métamodèle source (équivalent de Matched Rule en ATL), soit par un des algorithmes présentés au chapitre IV.

Pour présenter ces règles de transformations, nous allons utiliser un tableau dont la première colonne correspond au nom de la règle, la deuxième colonne indique le ou les concept(s) de(s) l'élément(s) source(s) de la règle (concepts appartenant au métamodèle d'architecture technique). La troisième colonne renseigne le ou les concept(s) de(s) éléments créé(s) (concepts appartenant au métamodèle BPEL). Enfin la quatrième colonne présente les liens de correspondance entre les attributs de ces éléments. Pour faciliter la lecture des alias seront utilisés, ainsi dans la quatrième colonne le nom des concepts des métamodèles seront remplacés par des alias définis entre parenthèse suite au concept.

Suite à la présentation de ces tableaux, les fichiers de transformations ATL utilisés seront présentés. Ces fichiers reprennent l'ensemble des règles décrites dans cette annexe ainsi que les algorithmes et la gestion de l'information présentés au chapitre IV.

### I. Présentation des Règles de transformation.

Le premier tableau présente les règles exécutées dès qu'un élément du modèle source correspond aux concepts du métamodèle d'architecture technique sur lesquels reposent la règle.

Nom	Concept du métamodèle d'architecture technique	Concept du métamodèle BPEL	Lien de correspondance
GenerateImport	Partner Service Description (PSD)	Import (I)	I.location ← PSD.wsdluri, I.namespace ← PSD.interfaceNsUrl I.importType ← 'http://schemas.xmlsoap.org/wsdl/'
GenerateVariable	Message Variable (MV) Business Object (BO)	Variable (V)	V.name ← MV.name; V.messageType ← concat('{'; BO.namespace; '}'; BO.type)
GeneratePartner Link	partner_service (PS)	PartnerLink (PL)	PL.name ← concat('plt_'; PS.name) PL.partnerLinkType ← concat( 'http://exemple.org/Artifacts/plt_'; PS.name) PL.partnerRole ← concat('role_'; PS.name)

GeneratePartner LinkType	partner_service (PS)	PartnerLinkType (PLT)	PLT.name ← concat('plt_'; PS.name)
GenerateRole	partner_service description (PSD)	Role (R)	R.name ← concat('role_'; PSD.name) R.portType ← PSD.portType;

Les règles présentées dans le tableau ci-dessous sont appelées et par conséquent ont des paramètres. Ces paramètres correspondent à un des concepts du métamodèle d'architecture technique.

Nom	Concept du métamodèle d'architecture technique	Concept du métamodèle BPEL	Lien de correspondance
GenerateInvoke(X)	<p>X ∈ Invoke (IV)</p> <p>MV1 et MV4 ∈ MessageVariable<sup>2</sup> correspondent aux séquence flows avant et après X</p> <p>MV2 et MV3 ∈ MessageVariable<sup>2</sup> correspondent aux message flows entrant et sortant pour X</p> <p>PS ∈ Partner Service lié à l'invoke</p> <p>O ∈ Operation lié à l'invoke</p>	<p>IV1 ∈ Invoke correspond au service de médiation</p> <p>IV2 ∈ Invoke correspond au service d'acteur</p> <p>A1 ∈ Assign</p> <p>A2 ∈ Assign</p>	<p>IV1.name ← « Service de médiation »</p> <p>IV1.partnerLink ← “Mediation”</p> <p>IV1.operation ← “match”</p> <p>IV1.inputVariable ← MV1</p> <p>IV1.outputVariable ← res</p> <p>A1 récupère les informations de res pour faire MV2 (voir chapitre 4)</p> <p>IV2.name ← X.name</p> <p>IV2.partnerLink ← PS.name</p> <p>IV2.operation ← O.name</p> <p>IV2.inputVariable ← MV2</p> <p>IV2.outputVariable ← MV3</p> <p>A2 récupère le message de sortie MV3 et créer MV4 à partir de MV1 et MV3 (cf chapitre IV)</p>
GenerateReceive(X)	X ∈ Receive	Receive(RC)	<p>RC.name ← X.name</p> <p>Si X=événement initial alors</p> <p>RC.createInstance ← “true”</p> <p>RC.partnerLink ← “client”</p> <p>RC.portType ← “tns:exemple”</p> <p>RC.operation ← “process”;</p> <p>RC.variable ← “input”;</p> <p>Fin Si</p>

GenerateReply(X)	X ∈ Reply	Reply (RP)	<p>RP.name ← X.name</p> <p>Si X=événement final alors</p> <p>RP.partnerLink ← “client”</p> <p>RP.portType ← “tns:exemple”</p> <p>RP.operation ← “process”;</p> <p>RP.variable ← “output”;</p> <p>Fin Si</p>
GenerateFlow(X)	X ∈ Flow	Flow (F)	F.name ← X.name
GenerateWhile(X)	X ∈ While	While (W)	W.name ← X.name
GenerateSwitch(X)	X ∈ Switch	Switch (S)	S.name ← X.name

## II. La transformation de modèles du modèle d'architecture technique au BPELArtifact

```

module UMLT2BPELArtifact; -- Module Template
create OUT : BPELARTIFACT from IN : UMLT;

--
--
-- Fonctions
--
--

helper context UMLT!Element def: hasStereotype(stereotype :
String) : Boolean =
self.getAppliedStereotypes() -> collect(st | st.name) ->
includes(stereotype);

--pour trouver la valeur d'un attribut d'une classe
helper context UMLT!"uml::Class" def:FindAttribute (attribute:String):
Set(UMLT!"uml::Element")=

        let maVariable :String =
        self.ownedMember
        ->select (a|a.name = attribute)
        in
        if maVariable->asSet()->isEmpty() -- pas de noeud qui va bien

            then 'faux'
            else maVariable.first().default
        endif;

helper context UMLT!Element def: isSameName (nst : String) : Boolean =
if self.name=nst
    then true
    else false
endif;

--
--
-- generation du bpelartifact
--
--

rule GenerateBPELartifact {
    from
    model: UMLT!Model
    to
    p: BPELARTIFACT!definitions
    (
        name<-'processArtifacts',
        targetNamespace<-'http://exemple.org/Artifacts',

-- traitement des imports
import <- model.packagedElement-> select(a | a.isSameName('Service
view'))->collect(a|a.packagedElement)->flatten()->
        select(c | c.isSameName('Partners services'))-
>collect(c|c.packagedElement)->flatten()
-
>select(b|b.hasStereotype('partner_service_description'))->flatten(),

```

```

--ajout des partnerLinks
    partnerLinkType <- model.packagedElement->select(a |
a.isSameName('Service view'))->collect(a|a.packagedElement)->flatten()->
    select(c | c.isSameName('Partners services'))-
>collect(c|c.packagedElement)->flatten()
    ->select(b|b.hasStereotype('partner_service'))-
>flatten()
    )
    do{

        thisModule.InitialisationImport(p,'3');
        thisModule.InitialisationPartnerLinks(p,'2');

    }
}

rule GenerateImportArtifact {
    from m:UMLT!Class (m.hasStereotype('partner_service_description'))
    to
i: BPELARTIFACT!Import
do{
    i.namespace <-m.FindAttribute('interfaceNsUrl');
    i.location <-m.FindAttribute('wsdlURI');
}}

rule GeneratePartnerLinkArtifact{
    from m:UMLT!Class(m.hasStereotype('partner_service'))
to
i: BPELARTIFACT!PartnerLinkType
(
    name<- 'plt_'+m.name
    )do{
    thisModule.GenerateRole(i,m);
    }}

rule GenerateRole(i:BPELARTIFACT!PartnerLinkType,m:UMLT!Class){
    using{
        VarGlo_information : UMLT!Package
            = UMLT!Package.allInstances()-> select(a | a.name =
'Information view')-> first().packagedElement;
        VarGlo_service : UMLT!Package
            = UMLT!Package.allInstances()-> select(a | a.name =
'Service view')-> first().packagedElement
            -> select(e | e.name = 'Partners services')-
>first().packagedElement;    }
to
r: BPELARTIFACT!Role
(
    name<- 'role_'+m.name
    )do{
    for(fiser in VarGlo_service->flatten()->select(b|b.name='liaison fihe
service')->flatten()){
        for(fiserelement in fiser.endType->flatten()){
            if(fiserelement=m){
                for(fiche in fiser.endType->flatten()){

                    if(fiche.hasStereotype('partner_service_description')){
                        --
                        r.portType<-
fiche.FindAttribute('interfaceNsUrl')+ '$'+fiche.FindAttribute('portType');
                        }}}}
                    i.role<-r;

```

```

    }
}

--
rule InitialisationImport(process : BPELARTIFACT!definitions,int :String) {
to
p: BPELARTIFACT!Import
do{
if(int='1'){
    p.namespace<-'http://exemple.org/Artifacts';
    p.location<-'processArtifacts.wsdl';
    p.importType<-'http://schemas.xmlsoap.org/wsdl/';
}
if(int='2'){
    p.namespace<-'http://exemple.org/';
    p.location<-'process.wsdl';
    p.importType<-'http://schemas.xmlsoap.org/wsdl/';
}
if(int='3'){
    p.namespace<-'http://www.isycr.com/IsyMediationInformation_Service/';
    p.location<-'IsyMediationInformation_Service.wsdl';
}
process.import<-p;}}
--

rule InitialisationPartnerLinks(process :
BPELARTIFACT!definitions,num:String) {
to
p: BPELARTIFACT!PartnerLinkType
do{

    if(num='2'){

        p.name<-'Mediation';
        thisModule.InitialisationRole(p,num);
        process.partnerLinkType<-p;
    }
}}
--

rule InitialisationRole(process : BPELARTIFACT!PartnerLinkType,num:String)
{
to
p: BPELARTIFACT!Role
do{

    if(num='2'){

        p.name<-'MediationRole';
        p.portType<-'http://www.isycr.com/IsyMediationInformation_Service/$IsyMediationInforma
tion_Service';
        process.role<-p;
    }
}}
--

```

### III. La transformation de modèles du modèle d'architecture technique au BPEL

```

module UMLT2BPEL_V100709; -- Module Template
create OUT : BPEL from IN : UMLT;

-----
--                               Fonctions
-----

helper context UMLT!Element def: hasStereotype(stereotype :
String) : Boolean =
self.getAppliedStereotypes() -> collect(st | st.name) ->
includes(stereotype);

helper context UMLT!Element def: hasStereotypeEmpty() : Boolean =
  let maVariable : OrderedSet(String) =
    self.getAppliedStereotypes()
  in
  if maVariable->asSet()->isEmpty()
  then true
  else false endif;

helper context UMLT!Element def: isSameName (nst : String) : Boolean =
if self.name=nst
  then true
  else false
  endif;

helper context UMLT!Element def: whichStereotype () : String =
self.getAppliedStereotypes();

-----
-----

--pour trouver la valeur d'un attribut d'une classe
helper context UMLT!"uml::Class" def:FindAttribute (attribute:String):
Set(UMLT!"uml::Element")=

  let maVariable :String =
    self.ownedMember
    ->select (a|a.name = attribute)
  in
  if maVariable->asSet()->isEmpty() -- pas de noeud qui va bien

  then 'faux'
  else maVariable.first().default
endif;

helper context UMLT!Element def:hasAttribute (attribute:String):
Boolean=

  let maVariable :String =
    self.ownedMember
    ->select (a|a.name = attribute)
  in
  if maVariable->asSet()->isEmpty() -- pas de noeud qui va bien

```



```

    then false
  else true
endif;

helper def:isInside(name : String,liste: OrderedSet(String)): Boolean =
  let maVariable : OrderedSet(String) =
    liste->select(e|e=name)
  in
  if maVariable->asSet()->isEmpty()
  then false
  else true endif;

helper def: ClassCorrespondante(elementName : String, nomStereotype :
String):UMLT!Class=
  let maVariable : OrderedSet(UMLT!Class)=
    UMLT!Class.allInstances()->select(e|e.name=elementName)->flatten()
  in
  if maVariable.size()>1
  then
    let maVariable2 : UMLT!Class =
      maVariable->select(e|e.name=elementName)->first()
    in
    if maVariable2.hasStereotype(nomStereotype)
    then maVariable2
    else
      maVariable->select(e|e.name=elementName)->last() endif
  else
    let maVariable3 : UMLT!Class =
      maVariable->first()
    in
    maVariable3
  endif;

helper def: isOperation(elementName : String): Boolean=
  let maVariable : UMLT!Class=
    UMLT!Package.allInstances()-> select(a | a.name = 'Service view')-
>first().packagedElement
  -> select(e | e.name = 'Partners services')-
>first().packagedElement
  ->flatten()->select(u|u.name=elementName)-
>select(u|u.hasStereotype ('partner_service'))->flatten()
  in
  if maVariable->isEmpty()
  then false
  else
    let maVariable2 : UMLT!Class= maVariable->first() in
      if maVariable2.hasStereotype ('partner_service')
      then true
      else false endif
    endif;
  --*****
helper def: returnOperation(elementName : String): UMLT!Class=
  let maVariable : UMLT!Class=
    UMLT!Package.allInstances()-> select(a | a.name = 'Service view')-
>first().packagedElement
  -> select(e | e.name = 'Partners services')-
>first().packagedElement
  ->flatten()->select(u|u.name=elementName)-
>select(u|u.hasStereotype ('partner_service'))->flatten()

```



```

helper def: PSMsgFlowIn:String='';
helper def: PSMsgFlowOut:String='';
helper def: PSPortType:String='';
helper def: PSPartnerLink:String='';

--
--                                     Debut des règles
--
rule InitialisationVariable(process : BPEL!Variables,int :Integer) {
to
p: BPEL!Variable
do{
if(int=0){
    p.name<-'input';
    p.type<-'{http://exemple.org/}exempleRequestMessage';
}
if(int=1){
    p.name<-'output';
    p.type<-'{http://exemple.org/}exempleResponseMessage';
}
if(int=2){
    p.name<-'mediationResponse';
    p.messageType<-
' {http://www.isycricom/IsyMediationInformation_Service/}IsyMediationInform
ation_ServiceResponse';
}
if(int=3){
    p.name<-'endServiceResponse';
    p.messageType<-
' {http://www.isycricom/IsyMediationInformation_Service/}IsyMediationInform
ation_ServiceResponse';
}
    process.variable<-p;}}
--

rule InitialisationPartnerLinks(process : BPEL!PartnerLinks,num:String) {
to
p: BPEL!PartnerLink
do{
    if(num='1'){
        p.name<-'client';
        p.partnerLinkType<-'http://exemple.org/exemple';
        p.myRole<-'exempleProvider';

        process.partnerLink<-p;
    }
    if(num='2'){
        p.name<-'mediation';
        p.partnerLinkType<-'http://exemple.org/Artifacts/Mediation';
        p.partnerRole<-'MediationRole';
        process.partnerLink<-p;
    }
}}
--

rule InitialisationImport(process : BPEL!Variables,int :String) {
to
p: BPEL!Import
do{
if(int='1'){
    p.namespace<-'http://exemple.org/Artifacts';
    p.location<-'processArtifacts.wsdl';
}
}
}

```

```

        p.importType<-'http://schemas.xmlsoap.org/wsdl/';
    }
if(int='2'){
    p.namespace<-'http://exemple.org/';
    p.location<-'process.wsdl';
    p.importType<-'http://schemas.xmlsoap.org/wsdl/';
}
if(int='3'){
    p.namespace<-
'http://www.isycr.com/IsyMediationInformation_Service/';
    p.location<-'IsyMediationInformation_Service.wsdl';
    p.importType<-'http://schemas.xmlsoap.org/wsdl/';
}
    process.import<-p;}}

--
--

rule GenerateBPEL {
from
model: UMLT!Model
to
process: BPEL!Process
(
    --attribut du process
    name<- 'Mediation_information_system',
    exitOnStandardFault <- true,
    expressionLanguage <- 'XPath 1.0 in BPEL2.0',
    queryLanguage <- 'urn:oasis:names:tc:wsbpel:2.0:sublang:xpath1.0',
    suppressJoinFailure <- true,
    targetNamespace <- 'http://exemple.org/',
-- traitement des imports
    import <- model.packagedElement-> select(a | a.isSameName('Service
view'))->collect(a|a.packagedElement)->flatten()->
        select(c | c.isSameName('Partners services'))-
>collect(c|c.packagedElement)->flatten()
-> select(b|b.hasStereotype('partner_service_description'))->flatten(),

--ajout des partnerLinks
    partnerLinks <- model.packagedElement-> select(a |
a.isSameName('Service view'))->flatten()
        ->first(),
--ajout du process
    activity <-model.packagedElement-> select(a | a.isSameName('Process
view'))-> first().packagedElement
        -> select(e | e.isSameName('Basic Activity'))-
>first().packagedElement
        -> select(b|b.hasStereotype('receive'))-
>select(b | b.isSameName('start'))-> flatten()
        ->first()
)
do{
--ajout des variables
    thisModule.GenerateVariables(process);
    thisModule.InitialisationImport(process,'1');
    thisModule.InitialisationImport(process,'2');
    thisModule.InitialisationImport(process,'3');
}}
--
*****
*****

```

```

rule GenerateVariables(process : BPEL!Process) {
using{
    messageType : Sequence(UMLT!Class)=UMLT!Package.allInstances()->
select(a | a.isSameName('Process view'))->first().packagedElement
        ->select(b|b.hasStereotype('message_type'))-
>select(b|b.hasAttribute('message_type'))->flatten();
    messageComponent: Sequence(UMLT!Class)=UMLT!Package.allInstances()->
select(a | a.isSameName('Process view'))->first().packagedElement
        ->select(b|b.hasStereotypeEmpty())-
>select(b|b.hasAttribute('message_component'))->flatten();
}
to
p: BPEL!Variables
do{
--Initialis tion des variables pour le BPEL
    thisModule.InitialisationVariable(p,0);
    thisModule.InitialisationVariable(p,1);
    thisModule.InitialisationVariable(p,2);
    thisModule.InitialisationVariable(p,3);

    UMLT!Package.allInstances()-> select(a | a.isSameName('Process
view'))->first().packagedElement
        ->select(b|b.hasStereotype('message_variable'))-
>flatten().debug('liste des variables');

    UMLT!Package.allInstances()-> select(a | a.isSameName('Process
view'))->first().packagedElement
        ->select(b|b.hasStereotypeEmpty())-
>flatten().debug('liste des sans stereotypes');

    messageType.debug('messageType');
    messageComponent.debug('messageComponent');

    for(var in UMLT!Package.allInstances()-> select(a |
a.isSameName('Process view'))->first().packagedElement
        ->select(b|b.hasStereotype('message_variable'))-
>flatten()){
        for(vartype in UMLT!Package.allInstances()->select(a |
a.isSameName('Process view'))->first().packagedElement
            ->select(e | e.name = 'Structured Activity')-
>first().packagedElement->select(b|b.isSameName(var.name))->flatten()){
            if (vartype.hasStereotype('sequence')){

                thisModule.GenerateVariable(var,p,var.name,'IsyMediationInformation_S
erviceRequest','http://www.isycricri.com/IsyMediationInformation_Service/');

                thisModule.GenerateVariable(var,p,var.name+'Response','IsyMediationIn
formation_ServiceRequest','http://www.isycricri.com/IsyMediationInformation_Se
rvice/');
            }
        }
--on génère les messages flows
        for(vartype in UMLT!Package.allInstances()->select(a |
a.isSameName('Process view'))->first().packagedElement
            ->select(c|c.name='have')->flatten()){
            for(stype in vartype.endType->flatten()){
                if (stype = var ){
                    for(dtype in vartype.endType->flatten()){

```

```

        if (dtype.hasAttribute('message_type')){
            --on va chercher le type de l'element
            thisModule.namespace<-'';
            thisModule.findNamespace(var);
            thisModule.GenerateVariable(var,p,var.name,dtype.name,thisModule.name
space);
        }}}}
    process.variables<-p;}}
--
*****
*****
rule GenerateVariable(m : UMLT!Class,process :
BPEL!Variables,name:String,type:String,namespace:String) {

to
p: BPEL!Variable
(
)do{
    if(name='') {
        p.name<- m.name;
    }else{
        p.name<-name;
        p.messageType<- '{'+namespace+'}' +type;
    }
    process.variable<-p;
}}
--
*****
*****

rule GeneratePartnerLinks {
from
m: UMLT!Package(m.name = 'Service view')
to
process: BPEL!PartnerLinks
(
    partnerLink<- m.packagedElement ->
        select(c | c.isSameName('Partners services'))-
>first().packagedElement->flatten()
        ->select (e|e.hasStereotype('partner_service'))-
>flatten()
    )
do{
    thisModule.InitialisationPartnerLinks(process,'1');
    thisModule.InitialisationPartnerLinks(process,'2');
    }
}
--
*****
*****

rule GeneratePartnerLink {
from
m: UMLT!Class (m.hasStereotype('partner_service'))
to
process: BPEL!PartnerLink
(
    --myRole          :
    --partnerRole     :
    --PartnerLinkType :
    name<- 'plt_'+m.name,

```

```

        partnerLinkType<-'http://exemple.org/Artifacts/plt_'+m.name,
        partnerRole<-'role_'+m.name
    )
}
--
*****
*****

rule GenerateImport {
from
m: UMLT!Class (m.hasStereotype('partner_service_description'))
to
process: BPEL!Import
(
    namespace <-m.FindAttribute('interfaceNsUrl'),
    location <-m.FindAttribute('wsdlURI'),
    importType <-'http://schemas.xmlsoap.org/wsdl/')
--
*****
*****

rule GenerateSequence {
from
m: UMLT!Class (m.isSameName('start'))
to
p: BPEL!"model::Sequence"

do{
--    m.debug('debutsequence');
thisModule.SequenceEnCours<- thisModule.SequenceEnCours->union(m.name->asSet()->asSet());
thisModule.trouverActiviteSuiivante(m,'sequenceInitiale');
thisModule.SequenceInitiale.debug('SequenceInterne(premiere du BPEL)');
if ( m.name = 'start'){
    p.name<-'SIM';
}else{
    p.name <-m.name;
}
--__on a la séquence à implenter, pour chaque on va générer un composant
thisModule.count<-0;
for (activiteName in thisModule.SequenceInitiale->flatten()){
    thisModule.activite <-
thisModule.ClassCorrespondante(activiteName,'invoke');

    if(thisModule.activite.hasStereotype('receive')){
        thisModule.GenerateReceive(thisModule.activite,p);
    }else if(thisModule.activite.hasStereotype('reply')){
--dans le cas des reply on ne sait pas si c'est un partner service ou un
element structurant du process (endevent par exemple)
        if(thisModule.isOperation(activiteName)=true){
--si il y a une class dont le stereotype est operation qui a le meme nom
que la class reply alors on est dans le cas d'un service de partenaire

--si il y a une class dont le stereotype est operation qui a le meme nom
que la class reply alors on est dans le cas d'un service de partenaire

activiteName.debug('activiteName');
thisModule.GenerateInvoke(thisModule.returnOperation(activiteName),p);
-- après l'invocation du service des partenaires on enregistre la reponse
dans le sequenceflow de reponse

```

```

    thisModule.GenerateAssign(thisModule.returnOperation(activiteName),p,
'responseService');
-- après l'invocation du service de partenaire on appel un deuxième service
de mediation afin de determiner que le service est fini

    thisModule.GenerateEndServiceIsyMediationService(thisModule.returnOpe
ration(activiteName),p,thisModule.VariableEntrante,thisModule.VariableSorta
nte);

        }else{
thisModule.GenerateReply(thisModule.activite,p);
        }
    }else if(thisModule.activite.hasStereotype('invoke ')){

thisModule.GenerateIsyMediationInformation(thisModule.activite,p);

    }else if(thisModule.activite.hasStereotype('flow')){
        if(thisModule.count=1){
            --on est dans le cas d'un flow fermant
            thisModule.count<-0;
        }else{
            thisModule.debutSequences<-OrderedSet{};
        }

        thisModule.ClassCorrespondante(thisModule.SequenceInitiale.at(thisMod
ule.SequenceInitiale->indexOf(activiteName)+1),'invoke');

        thisModule.trouverDebutDesSequences(thisModule.activite,thisModule.Cl
assCorrespondante(thisModule.SequenceInitiale.at(thisModule.SequenceInitial
e->indexOf(activiteName)+1),'invoke'));
            thisModule.debutSequences.debug('sequence A generer');

        thisModule.GenerateFlow(thisModule.activite,thisModule.ClassCorrespon
dante(thisModule.SequenceInitiale.at(thisModule.SequenceInitiale-
>indexOf(activiteName)+1),'invoke'),p);
            thisModule.count<-1;
        }
        }else if(thisModule.activite.hasStereotype('partner_service')){
            thisModule.GenerateInvoke(thisModule.activite,p);
        }else if(thisModule.activite.hasStereotype('specific_service')){
        }
    }
}
}
}
--
*****
*****

rule GenerateReceive (m : UMLT!Class,process : BPEL!"model::Sequence"){
to
p: BPEL!Receive
(
    name <-m.name
)
do{
    if(m.name='start'){
        p.createInstance<-true;
        p.partnerLink<- thisModule.returnPartnerLink('client');
        p.portType<- 'tns:exemple';
        p.operation<- 'process';
        p.variable<-
            thisModule.returnVariable('input');
    }
}
}
}

```



```

    }
    process.activities<-p;}}
--
*****
*****
rule GenerateReply(m : UMLT!Class,process : BPEL!"model::Sequence"){
to
p: BPEL!Reply
(
    name <-m.name
)do{
    if(m.name='end'){
        p.partnerLink<- thisModule.returnPartnerLink('client');
        p.portType<-'tns:exemple';
        p.operation<-'process';
        p.variable<- thisModule.returnVariable('output');
    }
    process.activities<-p;}}
--
*****
*****
rule GenerateIsyMediationInformation (m : UMLT!Class,p :
BPEL!"model::Sequence"){
do{
-- pour générer le service de mediation on doit :
--créer un assign qui copie les valeurs dans le sequence flow avant
--créer un invoke du service (ecrit en dur pour l'instant)
--créer un assign entre la sortie du service de mediation et le
service de partenaire suivant

--on initialise les valeurs des variables
thisModule.VariableEntrante='';
thisModule.VariableSortante='';
--on recupere le nom du sequenceflow entrant qui nous donnera le nom
de la variable entrante
thisModule.trouverVariablesIMS(m);
thisModule.VariableEntrante.debug('VariableEntrante');
thisModule.VariableSortante<-thisModule.VariableEntrante+'Response';
thisModule.VariableSortante.debug('VariableSortante');

thisModule.GenerateAssign(m,p,'request');
thisModule.GenerateInvokeIsyMediationService(m,p,thisModule.VariableE
ntrante,thisModule.VariableSortante);
thisModule.GenerateAssign(m,p,'response');

}}
--
*****
*****
--
*****
*****
rule GenerateAssign (m : UMLT!Class,p :
BPEL!"model::Sequence",type:String){
to
assign: BPEL!Assign
(
    name <- m.name+type,
    validate<-false
)do{

```

```

    if (type='request'){
        thisModule.GenerateCopy(m,assign,type,'','');
    }else if (type='responseService'){
-- on doit copier les valeurs du sequence flow et de la sortie du service
dans le sequence flow reponse
        thisModule.MsgFlowSuivant<-'';
        thisModule.SeqFlow<-'';
        thisModule.MsgFlowparameters<-'';
        thisModule.MsgFlowElement<-OrderedSet{};
        thisModule.ExpCopieResponse<-'';

        thisModule.trouverAssignServiceResponseElement(m);

--        thisModule.MsgFlowSuivant.debug('thisModule.MsgFlowSuivant');
        thisModule.trouverMsgFlowElement(thisModule.MsgFlowSuivant);
--        thisModule.MsgFlowElement.debug('seb');
        thisModule.ExpCopieResponse<-
thisModule.ExpCopieResponse+'["'+thisModule.MsgFlowSuivant+'"]';
        for(element in thisModule.MsgFlowElement->flatten()){
            --on créer l'element de copie

            thisModule.ExpCopieResponse<-
thisModule.ExpCopieResponse+'{"'+element+'"}'+'$'+thisModule.MsgFlowSuiva
nt+'.'+thisModule.MsgFlowparameters+'/' +element;
        }
        thisModule.ExpCopieResponse.debug('thisModule.ExpCopieResponse');
--on ecrit la copie
        thisModule.GenerateCopy(m,assign,'responseService',thisModule.ExpCopi
eResponse,'',thisModule.SeqFlow);

    }else{
--on doit générer une copie par element de la variable d'entrée du service
suivant + copier l'entrée +sortie du service de médiation dans la variable
seqflow
        thisModule.MsgFlowSuivant<-'';
        thisModule.MsgFlowparameters<-'';
        thisModule.MsgFlowElement<-OrderedSet{};
        thisModule.trouverMsgFlowSuivant(m);
        thisModule.trouverMsgFlowElement(thisModule.MsgFlowSuivant);
        for(element in thisModule.MsgFlowElement->flatten()){
            --on génère pour chaque element sortant

            thisModule.GenerateCopy(m,assign,'response',thisModule.MsgFlowSuivant
,thisModule.MsgFlowparameters,element);
        }
    }
    p.activities<-assign;}}
--
*****
*****
--
*****
*****
rule GenerateCopy (m : UMLT!Class,assign :
BPEL!Assign,type:String,msgflow:String,parameters:String,element:String){
to
cop: BPEL!Copy
( name<-m.name+type+msgflow+parameters+element
)
do{
        thisModule.GenerateTo(m,cop,type,msgflow,parameters,element);

```

```

        thisModule.GenerateFrom(m,cop,type,msgflow,parameters,element);

        assign.copy<-cop;}}
--
*****
*****
rule GenerateTo (m : UMLT!Class,cop :
BPEL!Copy,type:String,msgflow:String,parameters:String,element:String){
using{
}to
too: BPEL!To
do{
    thisModule.GenerateExpression(m,too,'',type,msgflow,parameters,elemen
t);

    cop.to<-too;}}
--
*****
*****
rule GenerateFrom (m : UMLT!Class,cop :
BPEL!Copy,type:String,msgflow:String,parameters:String,element:String){
to
fromm: BPEL!From
do{
    thisModule.GenerateExpression(m,'',fromm,type,msgflow,parameters,elem
ent);
    cop.from<-fromm;}}
--
*****
*****
rule GenerateExpression (m : UMLT!Class,too : BPEL!To,fromm :
BPEL!From,type:String,msgflow:String,parameters:String,element:String){
using{
    bpeVariable : BPEL!Variable =BPEL!Variable.allInstances()-
>flatten();
}to
exp: BPEL!Expression
do{
    exp.expressionLanguage<-'XPath 1.0 in BPEL2.0';

    if (too=''){
        if(type='request'){
            --on est dans le cas de avant le service de mediation, on liste
les seqflow d'avant
                --on commence par sauvegarder le seqflow en question
(necessaire pour le service de mediation)
                thisModule.expressionEnCours<-
'concat("[Process]", "{Seqflow}", "";
                thisModule.expressionEnCours<-
thisModule.expressionEnCours+thisModule.VariableEntrante+'';
                --on recupere le sequence flow avant et on l'ajoute à
l'expression

                thisModule.addSeqFlowBefore(m,thisModule.VariableEntrante);

                --on enregistre l'expression
                exp.body<-thisModule.expressionEnCours+'';
                fromm.expression<-exp;
            }else if (type='response'){
                --on doit decomposer la variable de sortie

```

```

        exp.body<-'substring-
after($mediationResponse.IMIResponse,"{'+element+'}");
        --"in"');

        fromm.expression<-exp;
        --on sauvegarde la valeur de l'entrée et du retour du
partenaire dans le seqflowresponse
        --    thisModule.expressionEnCours<-'![CDATA[concat('$'+
+'$mediationResponse.IMIResponse';
        }else if (type='responseService'){
        exp.body<-'concat('$'+element+'.IMIRequest,'+msgflow+'));
        fromm.expression<-exp;

        }
        }
    if(fromm=''){
        if(type='request'){
            --on stock les valeurs des seq flows précédent dans la
variable du seqflow
            exp.body<- '$'+thisModule.VariableEntrante+'.IMIRequest';
            too.expression<-exp;
        }else if (type='response'){
            exp.body<- '$'+msgflow+'.'+parameters+'/' +element;
            too.expression<-exp;
        }else if (type='responseService'){
            exp.body<- '$'+element+'Response.IMIRequest';
            too.expression<-exp;

        }
    }
}--fin do
}--fin rule
--
*****
*****
rule GenerateInvokeIsyMediationService (m : UMLT!Class,process :
BPEL!"model::Sequence", variableEntrante:String,variableSortante:String){
using{
}
to
p: BPEL!Invoke
(
    name <-m.name
)do{

    p.outputVariable<-thisModule.returnVariable('mediationResponse');
    p.inputVariable<-thisModule.returnVariable(variableEntrante);
-- partnerLink
    p.partnerLink<- BPEL!PartnerLink.allInstances()->flatten()-
>select(a|a.name='mediation')->first();
--operation
    p.operation<-'match';
--on l'ajoute au process
    process.activities<-p;}}
--
*****
*****
--
*****
*****

```

```

rule GenerateEndServiceIsyMediationService (m : UMLT!Class,process :
BPEL!"model::Sequence", variableEntrante:String,variableSortante:String){
using{
}
to
p: BPEL!Invoke
(
)do{
p.name<-m.name+'Response';
    p.outputVariable<-thisModule.returnVariable('endServiceResponse');
    p.inputVariable<-
thisModule.returnVariable(variableEntrante+'Response');
-- partnerLink
    p.partnerLink<- BPEL!PartnerLink.allInstances()->flatten()-
>select(a|a.name='mediation')->first();
--operation
    p.operation<-'endService';
--on l'ajoute au process
    process.activities<-p;}}
--
*****
*****_
*****
*****
rule GenerateInvoke (m : UMLT!Class,process : BPEL!"model::Sequence"){
using{
}
to
p: BPEL!Invoke
(
    name <-m.name
)do{
    thisModule.PSOperation='';
    thisModule.PSMsgFlowIn='';
    thisModule.PSMsgFlowOut='';
    thisModule.PSPortType='';
    thisModule.PSPartnerLink='';

    thisModule.trouverInfoPartnerService(m);

-- on va chercher les class du service et du service description pour
l'operation
    p.outputVariable<-thisModule.returnVariable(thisModule.PSMsgFlowOut);
    p.inputVariable<-thisModule.returnVariable(thisModule.PSMsgFlowIn);
-- partnerLink
    p.partnerLink<- thisModule.returnPartnerLink(thisModule.PSPartnerLink);
--operation
    p.operation<-thisModule.PSOperation;
--on l'ajoute au process
    process.activities<-p;}}
--
*****
*****
rule GenerateFlow (flowEntree : UMLT!Class,flowSortie : UMLT!Class,p :
BPEL!"model::Sequence") {
using{
    debutSequencesInterne : OrderedSet(String)=
    thisModule.debutSequences;
}
to
process: BPEL!Flow

```

```

(
    name <-flowEntree.name
)
do{
    for (sequenceADebuter in debutSequencesInterne->flatten()){
        sequenceADebuter.debug('sequenceADebuter');
        thisModule.SequenceEnCours<-OrderedSet{};
        thisModule.SequenceEnCours<-thisModule.SequenceEnCours-
>union(sequenceADebuter->asSet());
--on ajoute le partnerService associé à la premiere activité si il y en a
un

        thisModule.trouverPartnerService(thisModule.ClassCorrespondante(seque
nceADebuter,'invoke'));
--on initialise le nombre de flux
        thisModule.NbFluxEntrant<-0;
        thisModule.NbFluxSortant<-0;

        thisModule.trouverSubSequence(thisModule.ClassCorrespondante(sequence
ADebuter,'invoke'),flowSortie);

        thisModule.GeneratesubSequence(thisModule.ClassCorrespondante(sequenc
eADebuter,'invoke'),flowSortie,process);
    }
    p.activities<-process;}}
--
*****
*****
rule GeneratesubSequence (m : UMLT!Class,flowSortie : UMLT!Class,p :
BPEL!Element) {
using{
    sequenceAFaire:OrderedSet(String) = thisModule.SequenceEnCours;
    VarGlo_structuredActivity : UMLT!Package
        = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packagedElement
        -> select(e | e.name = 'Structured Activity')->first();
    VarGlo_basicActivity : UMLT!Package
        = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packagedElement
        -> select(e | e.name = 'Basic Activity')->first(); }
to
process: BPEL!"model::Sequence"
(
    name <-m.name)
do{
    thisModule.count<-0;
for (activiteName in sequenceAFaire->flatten()){
    thisModule.activite <-
thisModule.ClassCorrespondante(activiteName,'invoke');

    if(thisModule.activite.hasStereotype('receive')){
    thisModule.GenerateReceive(thisModule.activite,process);
    }else if(thisModule.activite.hasStereotype('reply')){
    if(thisModule.isOperation(activiteName)=true){
--si il y a une class dont le stereotype est operation qui a le meme nom
que la class reply alors on est dans le cas d'un service de partenaire

```

```

    thisModule.GenerateInvoke(thisModule.returnOperation(activiteName),process);
    -- après l'invocation du service des partenaires on enregistre la reponse
    dans le sequenceflow de reponse

    thisModule.GenerateAssign(thisModule.returnOperation(activiteName),process,
'responseService');
    -- après l'invocation du service de partenaire on appel un deuxième service
    de mediation afin de determiner que le service est fini

    thisModule.GenerateEndServiceIsyMediationService(thisModule.returnOperation(activiteName),process,
thisModule.VariableEntrante,thisModule.VariableSortante);
    }else{
        thisModule.GenerateReply(thisModule.activite,process);
    }
    }else if(thisModule.activite.hasStereotype('invoke ')){
        --on est dans le cas d'un service de mediation
        --pour l'instant on a qu'un service de transfert d'info et
matching semantique
        thisModule.GenerateIsyMediationInformation(thisModule.activite,process);
    }else if(thisModule.activite.hasStereotype('flow')){
        if(thisModule.count=1){
            --on est dans le cas d'un flow fermant
            thisModule.count<-0;
        }else{
            thisModule.debutSequences.debug('debutSequences');
            thisModule.debutSequences<-OrderedSet{};
            thisModule.activite.debug('activite');
            sequenceAFaire.debug('sequenceAFaire');
            sequenceAFaire.at(sequenceAFaire->indexOf(activiteName)+1).debug('flow de sortie du subsequence');

            VarGlo_structuredActivity.packagedElement -
>select(a|a.hasStereotype('flow'))-
>select(a|a.isSameName(sequenceAFaire.at(sequenceAFaire->indexOf(activiteName)+1)))-
>first().debug('flow_sortie_de_la_subsequence');
            thisModule.trouverDebutDesSequences(thisModule.activite,VarGlo_structuredActivity.packagedElement ->select(a|a.hasStereotype('flow'))-
>select(a|a.isSameName(sequenceAFaire.at(sequenceAFaire->indexOf(activiteName)+1)))->first());
            thisModule.debutSequences.debug('premiere activité des Sequences a lancer');

            thisModule.GenerateFlow(thisModule.activite,VarGlo_structuredActivity.packagedElement ->select(a|a.hasStereotype('flow'))-
>select(a|a.isSameName(sequenceAFaire.at(sequenceAFaire->indexOf(activiteName)+1)))->first(),process);
            thisModule.count<-1;
        }
    }
    }else if(thisModule.activite.hasStereotype('partner_service')){
        thisModule.GenerateInvoke(thisModule.activite,process);
    }else if(thisModule.activite.hasStereotype('specific_service')){
    }else{
        thisModule.activite.debug('stereotype pas pris en
compte');

```

```

    }

    p.activities<-process;}}
}
--
--      fonctions
--
rule trouverActiviteSuivante(acti :UMLT!Class,fluxSortantSeqSup: String){
    using{
        VarGlo_structuredActivity : UMLT!Package
            = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packagedElement
            -> select(e | e.name = 'Structured Activity')->first();
        VarGlo_basicActivity : UMLT!Package
            = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packagedElement
            -> select(e | e.name = 'Basic Activity')->first();    }
do{
acti.debug ('activite d entree');
if (thisModule.SequenceEnCours.isEmpty()==true){
thisModule.SequenceEnCours<-acti.name->asSet().debug('String');
}

if(acti.hasStereotype('flow')){
    thisModule.NbFlux(acti);
}
if(acti.hasStereotype('flow')and
thisModule.NbFluxEntrant<thisModule.NbFluxSortant){
    thisModule.traitementFlow(acti,fluxSortantSeqSup);
}
else{

--if(thisModule.FlagFinSeqInitiale=0){
for (s in VarGlo_structuredActivity.packagedElement-
>select(e|e.name='source')){
    for(stype in s.endType->flatten() ){
        if (stype = acti ) {
-- c'est une association qui a pour source l'activité
--on recupere le nom du ou des (gateway) sequence flow
        for(sfgood in s.endType-
>select(e|e.hasStereotype('sequence'))->flatten()){
            sfgood.debug('sequenceflow');
            for(d in
VarGlo_structuredActivity.packagedElement->select(e|e.name='destination')){
                for(dtype in d.endType-
>select(e|e.hasStereotype('sequence'))->flatten() ){
                    if (dtype = sfgood){
--
                        for(dgood in d.endType->flatten()){
                            if
(dgood.hasStereotype('sequence')){
                                }else{
--dans le cas ou l'on a un gateway on doit regarder si il est ouvert,
--si c'est le cas on doit verifier que c'est une gateway merge, car ca
entraîne la création de sequence
--si c'est un merge on regarde la valeur de notre pointeur qui permet de
trouver si c'est la bonne fermeture
                                if(dgood.name=fluxSortantSeqSup){

```



```

--on est dans le cas ou on est
revenu au flow fermant la sequence supérieur
    }else{
    thisModule.SequenceEnCours<-
thisModule.SequenceEnCours->union(dgood.name->asSet());
-- dans le cas ou l'activite
suivante est un service de coordination on va chercher le service de
partner
-- ainsi que l'information
vehicule
    if(dgood.name='end'){
thisModule.SequenceInitiale <-thisModule.SequenceEnCours;
    }else{

    thisModule.trouverPartnerService(dgood);

    thisModule.trouverActiviteSuivante(dgood,fluxSortantSeqSup);
    }}}}
}}}
}}
-----
-----
-----
rule trouverPartnerService(acti :UMLT!Class){
using{
    VarGlo_structuredActivity : UMLT!Package
        = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packagedElement
        -> select(e | e.name = 'Structured Activity')->first();
    VarGlo_basicActivity : UMLT!Package
        = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packagedElement
        -> select(e | e.name = 'Basic Activity')->first();
    VarGlo_InformationView : UMLT!Package =
        UMLT!Package.allInstances()-> select(a | a.name =
'Information view')->first();
    VarGlo_servicesView : UMLT!Package =
        UMLT!Package.allInstances()-> select(a | a.name =
'Service view')->first();
    VarGlo_partnerServices : UMLT!Package =
        UMLT!Package.allInstances()-> select(a | a.name =
'Service view')->first().packagedElement
        -> select(e | e.name = 'Partners services')->first();}
do{
    for (m in VarGlo_InformationView.packagedElement-
>select(e|e.name='donnée générée par')){
        for(mtype1 in m.ownedEnd->flatten() ){
            if (mtype1.type.name = acti.name ){
-- c'est une association qui va du service de coordination au service du
partner
--on recupere le nom de la donnée transmise avec le message flow
                for(mtype in m.ownedEnd->flatten() ){

                    if (mtype.type.name = acti.name ){}else{
                        mtype.type.debug('_____info
transferer');

                        for(message in
VarGlo_InformationView.packagedElement->select(e|e.name='donnée utilisée
par')){

                            for(messagetype1 in
message.ownedEnd->flatten() ){

```

```

                                if (messagetype.type =
mtype.type ){
                                for(messagetype in
message.ownedEnd->flatten() ){
                                if
(messagetype.type = mtype.type ){}else{

                                messagetype.type.debug('messagetype');
-- on recupere le nom de l'operation qui utilise le message flow
for (s in VarGlo_partnerServices.packagedElement->select(e|e.name='have
operation')){
                                for(stype in s.endType->flatten() ){
                                if (stype = messagetype.type ){
-- c'est une association qui a pour source l'activité
--on recupere le nom du service qui correspond
                                for (sgood in s.endType->flatten() ){
                                if(sgood= stype){
                                }else{
-- on va recuperer le nom du partnerservice
sgood.debug('-----partnerService');

thisModule.SequenceEnCours<-thisModule.SequenceEnCours->union(sgood.name-
>asSet());
                                }}}}}

                                }}}}}}}}}}}}}

--
-----
--
-----
rule NbFlux(acti :UMLT!Class){
    using{
        VarGlo_structuredActivity : UMLT!Package
            = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packagedElement
            -> select(e | e.name = 'Structured Activity')->first();
        VarGlo_basicActivity : UMLT!Package
            = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packagedElement
            -> select(e | e.name = 'Basic Activity')->first();
    }
do{
for (s in VarGlo_structuredActivity.packagedElement-
>select(e|e.name='source')){
                                for(stype in s.endType->flatten() ){
                                if (stype = acti ){
-- c'est une association qui a pour source le flow
thisModule.NbFluxSortant<-thisModule.NbFluxSortant+1;
                                }}}
for(d in VarGlo_structuredActivity.packagedElement-
>select(e|e.name='destination')){
                                for(dtype in d.endType->flatten() ){
                                if (dtype = acti ){
-- c'est une association qui a pour destination le
flow
                                thisModule.NbFluxEntrant<-thisModule.NbFluxEntrant+1;
                                }}}}}

```

---

```

rule traitementFlow(acti :UMLT!Class,fluxSortantSeqSup : String){
do{
--on commence par ajouter le flow a la sequence
thisModule.SequenceEnCours<- thisModule.SequenceEnCours->union(acti.name->asSet());

    thisModule.trouverFlowSuivant(acti,fluxSortantSeqSup);

}}
--

```

---

```

rule trouverFlowSuivant(acti :UMLT!Class, fluxSortantSeqSup: String){
    using{
        VarGlo_structuredActivity : UMLT!Package
            = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packagedElement
            -> select(e | e.name = 'Structured Activity')->first();
        VarGlo_basicActivity : UMLT!Package
            = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packagedElement
            -> select(e | e.name = 'Basic Activity')->first();
        VarGlo_InformationView : UMLT!Package =
            UMLT!Package.allInstances()-> select(a | a.name =
'Information view')->first();
        VarGlo_servicesView : UMLT!Package =
            UMLT!Package.allInstances()-> select(a | a.name =
'Service view')->first();
        VarGlo_partnerServices : UMLT!Package =
            UMLT!Package.allInstances()-> select(a | a.name =
'Service view')->first().packagedElement
            -> select(e | e.name = 'Partners services')->first();
    }
do{
for (s in VarGlo_structuredActivity.packagedElement->select(e|e.name='source')){
    for(stype in s.endType->flatten() ){
        if(stype = acti ){
-- c'est une association qui a pour source l'activité
--on recupere le nom du ou des (gateway) sequence flow
            for(sfgood in s.endType->select(e|e.hasStereotype('sequence'))->flatten()){
                for(d in
VarGlo_structuredActivity.packagedElement->select(e|e.name='destination')){
                    for(dtype in d.endType->select(e|e.hasStereotype('sequence'))->flatten() ){
                        if (dtype = sfgood){
-- on a le meme numero de sequenceflow pour une association destination
-- on recupere le ou les activites suivantes
                            for(dgood in d.endType->flatten()){
                                if
(dgood.hasStereotype('sequence')){
                                    }else{
--dans le cas ou l'on a un gateway on doit regarder si il est ouvert,
--si c'est le cas on doit verifier que c'est une gateway merge, car ca
entraîne la création de sequence
--si c'est un merge on regarde la valeur de notre pointeur qui permet de
trouver si c'est la bonne fermeture
                                        if (dgood.hasStereotype('flow'))

```

---

```

        {
            thisModule.NbFlux(dgood);

            if (fluxSortantSeqSup=dgood.name){
--c'est le gateway superieur on arrete donc le traitement
            }else{
                if (thisModule.NbFluxEntrant=
thisModule.NbFluxSortant){
                    thisModule.NbFluxEntrant<-0;
                    thisModule.NbFluxSortant<-0;
                    thisModule.NbFlux(dgood);
                    thisModule.NbFluxEntrant<-0-
thisModule.NbFluxEntrant;
                    thisModule.NbFluxSortant<-0-
thisModule.NbFluxSortant;

                    dgood.debug('ajout a la sequence
en cours');
                    thisModule.SequenceEnCours<-
thisModule.SequenceEnCours->union(dgood.name->asSet());

-- il faut verifier qu'il n'y a pas d'activité autre
--on doit faire un test sur le flow finisant la sequence avant
fluxSortantSeqSup.debug('fluxSortantSeqSup');

                thisModule.trouverActiviteSuiivante(dgood,fluxSortantSeqSup);
                --
                    }else{

                    thisModule.trouverFlowSuiivant(dgood,fluxSortantSeqSup);
                    }}
                    }else{

                    thisModule.trouverFlowSuiivant(dgood,fluxSortantSeqSup);}
                --
                    }
                }))))))}}
--

```

```

rule trouverDebutDesSequences(actidebut :UMLT!Class, actifin :UMLT!Class){
    using{
        VarGlo_structuredActivity : UMLT!Package
            = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packagedElement
            -> select(e | e.name = 'Structured Activity')->first();
        VarGlo_basicActivity : UMLT!Package
            = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packagedElement
            -> select(e | e.name = 'Basic Activity')->first();
    }
do{

for (s in VarGlo_structuredActivity.packagedElement-
>select(e|e.name='source')){
    for(stype in s.endType->flatten() ){
        if (stype = actidebut ){
-- c'est une association qui a pour source l'activité
--on recupere le nom du ou des (gateway) sequence flow
        for(sfgood in s.endType-
>select(e|e.hasStereotype('sequence'))->flatten()){

```

```

        for(d in
VarGlo_structuredActivity.packageElement->select(e|e.name='destination')){
            for(dtype in d.endType-
>select(e|e.hasStereotype('sequence'))->flatten() ){
                if (dtype = sfgood){
-- on a le meme numero de sequenceflow pour une association destination
-- on recupere le ou les activites suivantes
                    for(dgood in d.endType->flatten()){
                        if
( dgood.hasStereotype('sequence')){
                            }else{
                                if(dgood=actifin){
                                    }else{
                                        thisModule.debutSequences<-
thisModule.debutSequences->union(dgood.name->asSet());
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

--
}

rule trouverSubSequence(actidebut :UMLT!Class, actifin :UMLT!Class){
    using{
        VarGlo_structuredActivity : UMLT!Package
            = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packageElement
            -> select(e | e.name = 'Structured Activity')->first();
        VarGlo_basicActivity : UMLT!Package
            = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packageElement
            -> select(e | e.name = 'Basic Activity')->first();
    }

do{
if(actidebut.hasStereotype('flow')){
    actidebut.debug('flow depart');
    thisModule.NbFlux(actidebut).debug('flow de depart');
    actifin.debug('flow arrivee');
}

if(actidebut.hasStereotype('flow')and
thisModule.NbFluxEntrant<thisModule.NbFluxSortant){
    thisModule.traitementFlow(actidebut,actifin.name);
}else{
for (s in VarGlo_structuredActivity.packageElement-
>select(e|e.name='source')){
    for(stype in s.endType->flatten() ){
        if (stype = actidebut ){
-- c'est une association qui a pour source l'activité
--on recupere le nom du ou des (gateway) sequence flow
            for(sfgood in s.endType-
>select(e|e.hasStereotype('sequence'))->flatten()){
                for(d in
VarGlo_structuredActivity.packageElement->select(e|e.name='destination')){
                    for(dtype in d.endType-
>select(e|e.hasStereotype('sequence'))->flatten() ){
                        if (dtype = sfgood){
-- on a le meme numero de sequenceflow pour une association destination
-- on recupere le ou les activites suivantes
                            for(dgood in d.endType->flatten()){
                                if
( dgood.hasStereotype('sequence')){

```



```

>select(e|e.name=sgood.name)-
>first().debug('partnerServiceDescriptionEnCours');
        }
    }
}]]]]
--
rule trouverVariablesIMS(acti :UMLT!Class){
    using{
        VarGlo_structuredActivity : UMLT!Package
            = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packagedElement
            -> select(e | e.name = 'Structured Activity')->first();
        VarGlo_basicActivity : UMLT!Package
            = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packagedElement
            -> select(e | e.name = 'Basic Activity')->first();    }
do{

--on recupere le sequenceflow avant
for (s in VarGlo_structuredActivity.packagedElement-
>select(e|e.name='destination')){
    for(stype in s.endType->flatten() ){
        if (stype = acti ){
-- c'est une association qui a pour destination l'activité
--on recupere le nom du ou des (gateway) sequence flow
            for(sfgood in s.endType-
>select(e|e.hasStereotype('sequence'))->flatten()){
                thisModule.VariableEntrante<-sfgood.name;
            }}}}}
-----
rule addSeqFlowBefore(acti : UMLT!Class,seqflow:String){
using{
    VarGlo_structuredActivity : UMLT!Package
        = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packagedElement
        -> select(e | e.name = 'Structured Activity')->first();
    VarGlo_basicActivity : UMLT!Package
        = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packagedElement
        -> select(e | e.name = 'Basic Activity')->first();    }
do{

--on recupere le sequenceflow avant
for (s in VarGlo_structuredActivity.packagedElement-
>select(e|e.name='source')){
    for(stype in s.endType->flatten() ){
        if (stype.name = seqflow ){
-- c'est une association qui a pour source le sequenceflow
--on recupere le nom du ou des (gateway) sequence flow
            for(actigood in s.endType->flatten()){
if(actigood.hasStereotype('sequence')){
}else{
                actigood.debug('sequenceflow');
                --on doit trouver les sequence flows qui
arrive a cette activite
                    for(sfgood in s.endType-
>select(e|e.hasStereotype('sequence'))->flatten()){
                        for(d in
VarGlo_structuredActivity.packagedElement->select(e|e.name='destination')){

```

```

        for(dtype in d.endType ){
            if (dtype = actigood){
-- on a le meme numero de sequenceflow avant
                for(seqflowbefore in d.endType){
if(seqflowbefore.hasStereotype('sequence')){
                    thisModule.expressionEnCours<-
thisModule.expressionEnCours+', $'+seqflowbefore.name+'Response.IMIRequest';
                }
                    }}}}
                }}}}
            }
}

-----
-----
-----

rule findNamespace(acti : UMLT!Class){
using{
    VarGlo_information : UMLT!Package
        = UMLT!Package.allInstances()-> select(a | a.name =
'Information view')-> first().packageElement;
    VarGlo_service : UMLT!Package
        = UMLT!Package.allInstances()-> select(a | a.name =
'Service view')-> first().packageElement
        -> select(e | e.name = 'Partners services')-
>first().packageElement;    }
do{

--on recupere le business object correspondant au message_variable
for (s in VarGlo_information->flatten()->select(b|b.name='liaison vue
processus')->flatten()){
    for(stype in s.endType->flatten() ){
        if (stype = acti ){
-- c'est une association qui a le messagevariable
--on recupere le business object correspondant
            for(bobject in s.endType->flatten()){
                if(bobject.hasStereotype('business_object')){
                    --on doit trouver l'operation qui utilise ce
business_object
                        for(sfgood in VarGlo_information->flatten()-
>select(b|b.name='donnée utilisée par')->flatten()){
                            for(d in sfgood.endType->flatten()){
                                if (d =    bobject){
-- c'est une association qui a le businnesobject
--on recupere l operation correspondante
                                    for(ope in sfgood.endType-
>flatten()){

                                        if(ope.hasStereotype('operation')){
                                            --
                                                for(haveop in VarGlo_service-
>flatten()->select(b|b.name='have operation')->flatten()){

                                                    for(haveopelement in haveop.endType->flatten()){

                                                        if(haveopelement=ope){

                                                            for(pser in haveop.endType->flatten()){

                                                                if(pser.hasStereotype('partner_service')){

```



```

        for(fiser in VarGlo_service->flatten()-
>select(b|b.name='liaison fihe service')->flatten()){
            for(fiserelement in fiser.endType->flatten()){
                if(fiserelement=pser){
                    for(fiche in fiser.endType->flatten()){
                        if(fiche.hasStereotype('partner_service_description')){
                            thisModule.namespace<-
fiche.FindAttribute('interfaceNsUrl');
                        }
                    }
                }
            }
            for(sfgood in VarGlo_information->flatten()-
>select(b|b.name='donnée générée par')->flatten()){
                for(d in sfgood.endType->flatten()){
                    if (d = bobject){
                        -- c'est une association qui a le businnesobject
                        --on recupere l operation correspondante
                        for(ope in sfgood.endType-
>flatten()){
                            if(ope.hasStereotype('operation')){
                                for(haveop in VarGlo_service->flatten()-
>select(b|b.name='have operation')->flatten()){
                                    for(haveopelement in haveop.endType->flatten()){
                                        if(haveopelement=ope){
                                            for(pser in haveop.endType->flatten()){
                                                if(pser.hasStereotype('partner_service')){
                                                    for(fiser in VarGlo_service->flatten()-
>select(b|b.name='liaison fihe service')->flatten()){
                                                        for(fiserelement in fiser.endType->flatten()){
                                                            if(fiserelement=pser){
                                                                for(fiche in fiser.endType->flatten()){
                                                                    if(fiche.hasStereotype('partner_service_description')){
                                                                        thisModule.namespace<-
fiche.FindAttribute('interfaceNsUrl');
                                                                    }
                                                                }
                                                            }
                                                        }
                                                    }
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
}
-----
-----
-----
-----
rule trouverMsgFlowSuivant(acti : UMLT!Class){
using{
    VarGlo_structuredActivity : UMLT!Package
}
}

```

```

        = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packagedElement
        -> select(e | e.name = 'Structured Activity')->first();
        VarGlo_basicActivity : UMLT!Package
        = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packagedElement
        -> select(e | e.name = 'Basic Activity')->first();
        VarGlo_InformationView : UMLT!Package =
        UMLT!Package.allInstances()-> select(a | a.name =
'Information view')->first();
        VarGlo_servicesView : UMLT!Package =
        UMLT!Package.allInstances()-> select(a | a.name =
'Service view')->first();
        VarGlo_partnerServices : UMLT!Package =
        UMLT!Package.allInstances()-> select(a | a.name =
'Service view')->first().packagedElement
        -> select(e | e.name = 'Partners services')->first();}
do{
    for (m in VarGlo_InformationView.packagedElement-
>select(e|e.name='donnée générée par')){
        for(mtype1 in m.ownedEnd->flatten() ){
            if (mtype1.type.name = acti.name ){
-- c'est une association qui va du service de coordination au service du
partner
--on recupere le nom de la donnée transmise avec le message flow
                for(mtype in m.ownedEnd->flatten() ){

                    if (mtype.type.name = acti.name ){}else{
                        thisModule.MsgFlowSuivant<-
mtype.type.name;

                    }}}}
}
-----
-----
-----
rule trouverMsgFlowElement(mf : String){
using{
    VarGlo_process : UMLT!Package
    = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packagedElement;
}do{
    for (m in VarGlo_process->flatten()->select(e|e.name='have')){
        for(mt in m.endType->flatten() ){
            if (mt.name = mf and mt.hasStereotype('message_variable')
){
-- c'est une association qui a le messagevariable
--on recupere le messagetype correspondant
                for(mtype in m.endType->flatten() ){

                    if (mtype.name = mf and
mtype.hasStereotype('message_variable')){}else{
                        thisModule.MsgFlowparameters<-
mtype.FindAttribute('parameter');

                    for (me in VarGlo_process->flatten()-
>select(e|e.name='have element')){

                            for(mte in me.endType->flatten()

                                if (mte = mtype ){
                                    -- c'est une
association qui a le messagetype

```

```

--on recupere le
messagecomponent correspondant
me.endType->flatten() ){
    for(mtypee in
        if (mtypee =
mtype){}else{
    thisModule.MsgFlowElement<-
thisModule.MsgFlowElement.union(mtypee.FindAttribute('value')->asSet());
        }}}}
-----
rule trouverInfoPartnerService(acti : UMLT!Class){
using{
    VarGlo_information : UMLT!Package
        = UMLT!Package.allInstances()-> select(a | a.name =
'Information view')-> first().packagedElement;
    VarGlo_service : UMLT!Package
        = UMLT!Package.allInstances()-> select(a | a.name =
'Service view')-> first().packagedElement
        -> select(e | e.name = 'Partners services')-
>first().packagedElement;
do{
    thisModule.PSPartnerLink<-'plt_'+acti.name;
--on recupere l'operation correspondant au message_variable
for (s in VarGlo_service->flatten()->select(b|b.name='have operation')-
>flatten()){
    for(stype in s.endType->flatten() ){
        if (stype = acti ){
-- c'est une association qui a le partnerservice
--on recupere l operation correspondante
            stype.debug('stype');
            for(bobject in s.endType->flatten()){
                if(bobject.hasStereotype('operation')){
                    thisModule.PSOperation<-bobject.name;
                }
            }
            thisModule.PSOperation.debug('thisModule.PSOperation');
--on doit trouver l'operation qui utilise ce
business_object
            for(sfgood in VarGlo_information->flatten()-
>select(b|b.name='donnée utilisée par')->flatten()){
                for(d in sfgood.endType->flatten()){
                    if (d = bobject){
-- c'est une association qui a le businnesobject
--on recupere l operation correspondante
                    for(ope in sfgood.endType-
>flatten()){
                        if(ope.hasStereotype('business_object')){
                            for(haveop in VarGlo_information->flatten()-
>select(b|b.name='liaison vue processus')->flatten()){
                                for(haveopelement in haveop.endType->flatten()){
                                    if(haveopelement=ope){
                                        for(pser in haveop.endType->flatten()){
                                            if(pser.hasStereotype('message_variable')){

```

```

    thisModule.PSMsgFlowIn<-pser.name;

    }}}}}}}}}
    for(sfgood in VarGlo_information->flatten()-
>select(b|b.name='donnée générée par')->flatten()){
        for(d in sfgood.endType->flatten()){
            if (d =      bobject){
-- c'est une association qui a le businessobject
--on recupere l operation correspondante
            for(ope in sfgood.endType-
>flatten()){

                if(ope.hasStereotype('business_object')){
                    for(haveop in
VarGlo_information->flatten()->select(b|b.name='liaison vue processus')-
>flatten()){

                        for(haveopelement in haveop.endType->flatten()){

                            if(haveopelement=ope){

                                for(pser in haveop.endType->flatten()){

                                    if(pser.hasStereotype('message_variable')){

                                        thisModule.PSMsgFlowOut<-pser.name;

                                        }}}}}}}}}}}}}}}}}}}

```

```

-----
rule trouverAssignServiceResponseElement(acti : UMLT!Class){
using{
    VarGlo_structuredActivity : UMLT!Package
        = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packagedElement
        -> select(e | e.name = 'Structured Activity')->first();
    VarGlo_basicActivity : UMLT!Package
        = UMLT!Package.allInstances()-> select(a | a.name =
'Process view')-> first().packagedElement
        -> select(e | e.name = 'Basic Activity')->first();
    VarGlo_informationView : UMLT!Package =
        UMLT!Package.allInstances()-> select(a | a.name =
'Information view')->first();
    VarGlo_servicesView : UMLT!Package =
        UMLT!Package.allInstances()-> select(a | a.name =
'Service view')->first();
    VarGlo_partnerServices : UMLT!Package =
        UMLT!Package.allInstances()-> select(a | a.name =
'Service view')->first().packagedElement
        -> select(e | e.name = 'Partners services')->first();}
do{
acti.debug ('*****$$$$$*****activite d entree');
    for (d in VarGlo_partnerServices.packagedElement-
>select(e|e.name='have operation')){
        for(dtype1 in d.ownedEnd->flatten() ){
            if (dtype1.type = acti ){
                for(dtype in d.ownedEnd->flatten() ){

                    if (dtype.type = acti){}else{

```

```

                --on recupere l'operation
                dtype.type.debug('operation');
            for (m in VarGlo_InformationView.packagedElement-
>select(e|e.name='donnée générée par')){
                for(mtype1 in m.ownedEnd->flatten() ){
                    if (mtype1.type = dtype.type ){
-- c'est une association qui va du service de partner au service de
coordination
--on recupere le nom de la donnée transmise avec le message flow
                for(mtype in m.ownedEnd->flatten() ){

                    if (mtype.type = dtype.type ){}else{

                        thisModule.MsgFlowSuivant<-
mtype.type.name;
                    }}}}
--on recupere le service de coordination qui appelle le service d'acteur

            for (m in VarGlo_InformationView.packagedElement-
>select(e|e.name='donnée utilisée par')){
--      m.debug('m');
                for(mtype1 in m.ownedEnd->flatten() ){
                    if (mtype1.type = dtype.type ){
-- c'est une association qui va du service de coordination au service
d'acteur
--on recupere le nom de la donnée transmise avec le message flow
                for(mtype in m.ownedEnd->flatten() ){
                    if (mtype.type.name = acti.name ){}else{
                        for (m2 in
VarGlo_InformationView.packagedElement->select(e|e.name='donnée générée
par')){

                            for(mtype2 in m2.ownedEnd-
>flatten() ){

                                if (mtype2.type.name =

mtype.type.name ){

                                    for(mtype3 in

m2.ownedEnd->flatten() ){

                                        if

(mtype3.type.name = mtype.type.name ){}else{
                                            ----on recupere le sequence flow

                                                for (s in
VarGlo_structuredActivity.packagedElement->

select(e|e.name='destination')->flatten()) {

for(s2 in s.ownedEnd->flatten() ){

    if (s2.type.name = mtype3.type.name ){

        for(s3 in s.ownedEnd->flatten() ){

            if (s3.type.name = mtype3.type.name ){}else{
                thisModule.SeqFlow<-s3.type.name;
                }}}}}}}}}}}}}}}}}}}
            }}}}}
        }}
    }
}
-----

```

## Annexe G : UML2JBI

Cette annexe présente une vue globale des travaux de A.M. Barthe, nommé UML2JBI, sur la création des différents SU et SA nécessaires à la configuration de l'ESB.

Comme le montre la figure 14, ces travaux sont décomposés en trois grandes étapes :

- La première étape, représentée par la partie haute de la figure, consiste à extraire de l'architecture technique l'ensemble des éléments nécessaires à la création des SU et de récupérer d'une part, la liste des composants disponibles (SE, BC) et d'autre part le fichier BPEL et BPELArtifact.
- La deuxième étape, représenté en bas à droite de la figure, consiste à créer les SU puis les SA.
- La dernière partie, représenté en bas à gauche de la figure, consiste à créer un fichier Ant qui permettra le déploiement automatique de l'ensemble de ces SU et SA sur un ESB.

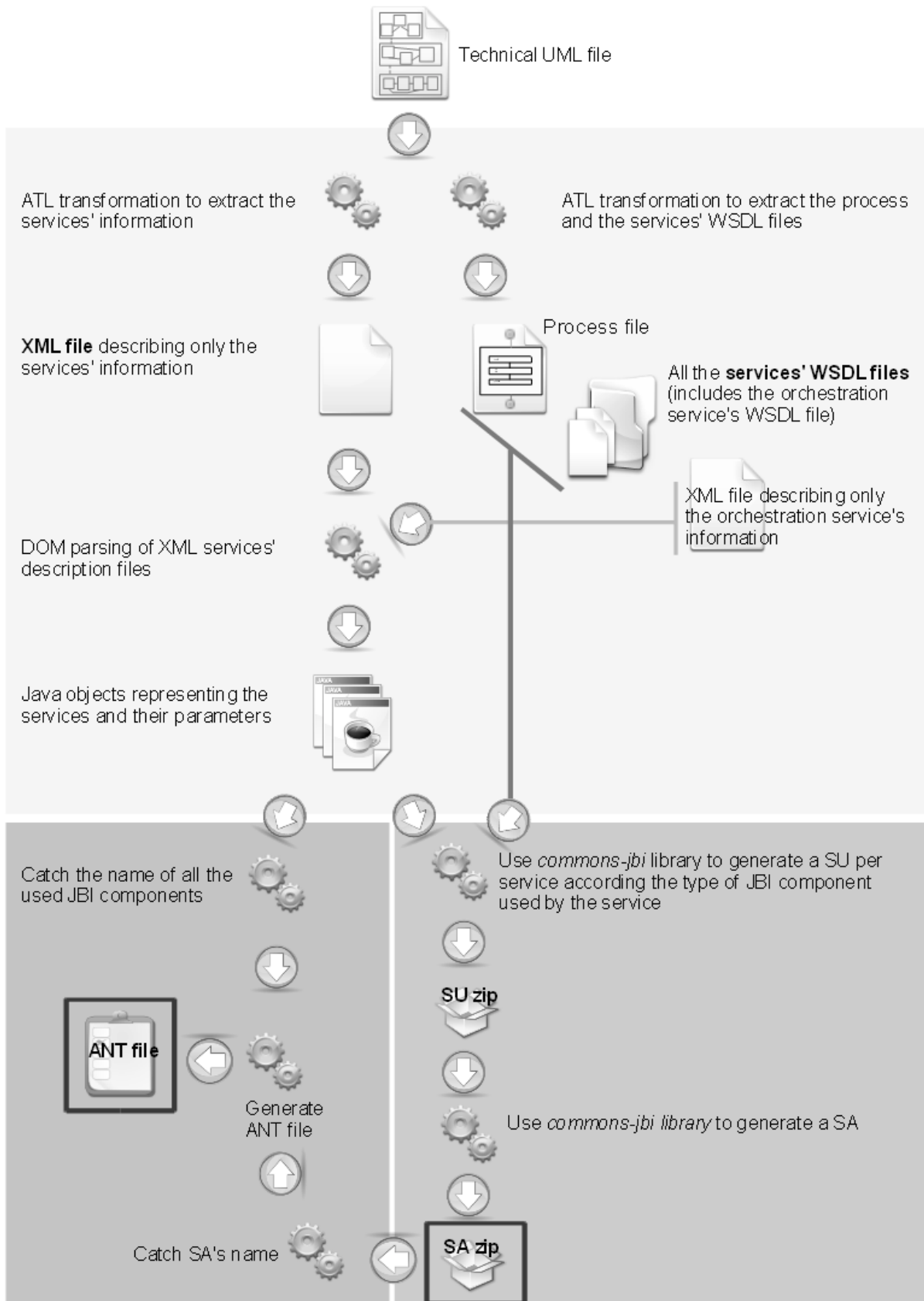


Figure 14 : vue d'ensemble de UML2JBI





**Résumé :** les collaborations inter-organisationnelles relèvent généralement de circonstances opportunistes et s'avèrent par conséquent éphémères. Les organisations doivent alors être disposées à s'intégrer dans ce type de collaboration tout en gardant leur identité propre. Ce constat est le point de départ du projet MISE (*Mediation Information System Engineering*), qui aborde cette notion de collaboration d'organisations selon l'angle du système d'information, en proposant une démarche de conception d'un SIM (système d'information de médiation). Ce SIM constitue un système tiers, médiateur des SI des diverses organisations, destiné à prendre en charge, d'une part la coordination des actions des partenaires (orchestration de la dynamique collective) et d'autre part, de gérer la circulation de l'information au sein de la collaboration (acheminement et traduction des données). La conception du SIM repose sur une démarche d'ingénierie dirigée par les modèles (IDM). Par ailleurs, la notion de crise, reposant par définition sur la sollicitation d'acteurs hétérogènes concernés par une collaboration opportuniste (qui plus est dans le cadre d'un phénomène évolutif d'une durée indéterminée), fait du domaine de la gestion de crise un parfait cas d'étude pour le projet MISE. Ces travaux de thèse, liés au projet ANR-CSOSG ISyCri, présentent cette démarche de conception du SIM appliqué au domaine de la gestion de crise. Le manuscrit parcourt la démarche MISE appliquée au domaine de la gestion de crise depuis la définition conceptuelle jusqu'à la réalisation technique selon les trois étapes de cette démarche IDM : (i) au niveau « métier » : l'utilisation d'une base de connaissance, représentée par une ontologie, permet, à partir des caractéristiques de la situation de crise et du savoir-faire des partenaires de la collaboration, de définir le processus collaboratif représentatif de la succession des activités à exécuter dans le cadre de la réponse à la crise. (ii) au niveau « logique » : une transformation de modèle permet de construire, à partir du modèle de processus collaboratif obtenu au niveau « métier », une architecture logique du SIM (orientée service, selon les préceptes SOA). (iii) au niveau « technique » : une deuxième transformation de modèles permet de générer les éléments nécessaires à la configuration du SIM, notamment le fichier BPEL. L'agilité du SIM ainsi déployé constitue une exigence incontournable. Les travaux présentés dans ce manuscrit proposent donc d'intégrer ces différentes étapes de conception du SIM sous la forme de composants logiciels indépendants, sollicités à loisir au sein d'une architecture orientée service. Cette solution apporte une grande flexibilité structurelle à la démarche, en autorisant la reconfiguration partielle du SIM à partir du niveau adapté à la situation.

**Mots Clés :** Système d'information, Interopérabilité, Transformation de modèles, IDM, Système d'Information de Médiation, Processus collaboratif, ESB, Ontologie.

---

**Abstract:** organizations should be able to take part into opportunistic and brief collaborative networks. However, they should also control their identity. The MISE project (*Mediation Information System Engineering*) aims at dealing with that issue from the information system point of view. The main principle is to design a specific third part mediation information system (MIS) in charge of, first, orchestrating the collaborative workflow of the collaborative network and, second, managing information (carrying and translating data). Designing such a MIS is based on a model-driven engineering approach (MDE). Considering crisis management field, it is obvious that such a domain requires opportunistic collaboration of heterogeneous partners involved in the crisis response (furthermore, crisis management is a very dynamic process where agility is a crucial point). Directly linked to the French funded ISyCri project, this PhD research work presents the overall approach for MIS design in a crisis management context. That MDE approach is based on three steps: (i) "Business" level: a collaborative process model is deduced from a knowledge base represented through an ontology. (ii) "Logical" level: an abstract service-oriented architecture of MIS is built, based on a model transformation from the previously obtained collaborative process model. (iii) "Technical" level: all the required deployment files are generated (including BPEL file), based on another model transformation, from the logical architecture. Besides, agility is a strong requirement for such a MIS. Therefore, these three steps are integrated, as independent software components, in a service-oriented architecture of a MIS-design tool. This solution brings structural flexibility to the overall approach by allowing partial redesign of the MIS (at the expected step).

**KeyWords :** Information System, Interoperability, Model transformation, MDE, Médiation Information System, Collaborative process, ESB, Ontology.