Prise en charge de l'agilité de workflows collaboratifs par une approche dirigée par les événements

Anne-Marie Barthe-Delanoë

▶ **To cite this version:**

Anne-Marie Barthe-Delanoë. Prise en charge de l'agilité de workflows collaboratifs par une approche dirigée par les événements. Autre [cs.OH]. Ecole des Mines d'Albi-Carmaux, 2013. Français. NNT : 2013EMAC0003 . tel-00991683

HAL Id: tel-00991683

https://theses.hal.science/tel-00991683

Submitted on 15 May 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

École Nationale Supérieure des Mines d'Albi-Carmaux conjointement avec l'INSA de Toulouse

---

Présentée et soutenue par :
Anne-Marie BARTHE

le vendredi 22 novembre 2013

Titre :

Prise en charge
de l'agilité de workflows collaboratifs
par une approche dirigée par les événements

École doctorale et discipline ou spécialité :
EDSYS : Informatique 4200018

Unité de recherche :
Centre Génie Industriel, Ecole des Mines d'Albi-Carmaux

Directeur(s) de Thèse :

Hervé PINGAUD

Jury :
Jean-Pierre BOUREY, Professeur, Centrale Lille, Rapporteur
Chihab HANACHI, Professeur, Université de Toulouse 1, Rapporteur
Frédérick BÉNABEN, Maître Assistant HDR, Ecole des Mines d'Albi-Carmaux, Examinateur
Achim P. KARDUCK, Professeur, Hochschule Furtwangen University, Examinateur
Jean-Pierre LORRÉ, Directeur R&D Linagora, Examinateur
Hervé PINGAUD, Professeur, Centre universitaire J. F. Champollion - Albi, Directeur
Nicolas SALATGÉ, Ingénieur E&D EVS-OpenCube, Examinateur

Prise en charge de l'agilité de workflows collaboratifs par une approche dirigée par les événements

Anne-Marie Barthe

Dernière révision : 4 décembre 2013

Remerciements

Six chapitres rédigés sans angoisse de la page blanche et voici qu'au moment des remerciements, le doute s'installe. Comment rendre hommage à celles et ceux qui ont été à mes côtés durant la thèse, dans les moments joyeux et les phases délicates ?

Frédérick, merci de m'avoir donné envie d'embrasser une carrière d'Enseignant-Chercheur. Merci d'avoir répondu à mon appel en avril 2010. Merci pour ta patience, ton dynamisme et ton optimisme. Tu n'es pas un encadrant mais un grand frère, un coach avec lequel rien ne paraît impossible. Tu m'as poussée à aller toujours plus loin, plus haut, bref à me dépasser. Travailler avec toi a été un bonheur.

Hervé, merci d'avoir accepté d'être mon Directeur de thèse. Merci de la confiance que tu m'as accordée. Et promis, on se reverra autour d'un confit de canard cuit au couscoussier !

Je souhaiterais remercier Jean-Pierre Bourey et Chihab Hanachi, qui m'ont fait l'honneur de rapporter mon manuscrit de thèse avec un regard critique toujours constructif. Merci à Achim P. Karduck, qui a accepté d'être mon président de jury, et dont l'énergie et l'éternelle bonne humeur ont illuminé ma soutenance. Merci à Jean-Pierre Lorré et Nicolas Salatgé d'avoir accepté de participer à mon jury de thèse malgré leurs emplois du temps surchargés.

Tout naturellement, je tiens à remercier tous mes collègues du Centre Génie Industriel, passés et présents. Michel, Jacques, Lionel, Elise, Sabine, François M., Carine, Tiexin, Andres, Alexandre, Aurélie C., Saïna, Nicolas, Aurélien, Matthieu, Sébastien (l'autre grand frère), Franck, Didier, Elyès, Paul G., Safae, Diana, Xenia, Philippe, François G., Paul P., Jorge, Marc. Merci pour votre accueil, votre disponibilité mais aussi votre bienveillance et votre sourire. Merci également à Isabelle, qui est bien plus que la secrétaire du CGI : elle est un peu la maman des doctorants, toujours disponible et prête à aider. Merci également aux collègues des autres services des Mines d'Albi, et plus particulièrement à Claudine pour tout ce qu'elle a pu m'apprendre et la discipline acquise grâce aux R-Docs, et à Denis et Dolores pour les grandes conversations autour d'un café.

La thèse n'est pas qu'une aventure professionnelle. C'est aussi une aventure humaine,

où naissent de belles amitiés, souvent internationales, entre doctorants. Olina, j'ai apprécié chaque minute de ta présence au laboratoire : ton extrême gentillesse et ton humour me manquent déjà. Je suis heureuse d'avoir pu soutenir ma thèse avant ton retour en Chine. Aurélie, toujours partante, toujours là quand les soirées au laboratoire devenaient des nuits. Un MERCI en lettres capitales s'impose pour m'avoir écoutée, accueillie, hébergée. Olfa, merci pour ta bonne humeur et les formidables échanges que nous avons pu avoir ensemble, sur des sujets aussi variés que passionnants (incluant les catamarans). Guillaume, sans toi mon bureau aurait été bien calme : ton dynamisme et ton accent gersois ont donné vie au 0A01. Merci d'avoir supporté mes mois de rédaction sans prendre la fuite ! Sarah, merci pour ta douceur et ta générosité et l'attention que tu portes aux petits détails dans les relations humaines.

Merci GLaDOS, Chell, Diablo, Plants vs. Zombies, Deezer, Vin Diesel, Therion, Imagine Dragons, Radical Face, Anathema, Golem ! pour l'accompagnement musico-ludique de ces travaux de thèse. Et n'oublions pas « la » balle rose.

Ces remerciements ne seraient pas complets sans une pensée pour tous mes amis : Adeline, Anaïs, Cédric, Christelle, Christophe, Loris, Mélanie et tous les autres. Merci pour vos encouragements et votre soutien.

Merci à ma belle-famille pour m'avoir accueillie parmi eux et pour l'intérêt porté à mes travaux. Jacqueline, Dominique, Emilie : merci d'avoir fait le déplacement depuis la Bretagne pour assister à ma soutenance. Claire si Dorin, va multumesc pentru incurajarea dvs.

Merci à mes parents, Mireille et Bertrand, qui m'ont soutenue dans mes choix. Merci pour l'éducation que vous m'avez donnée, et pour le goût de l'instruction que vous m'avez insufflé. Merci aussi à Pauline, ma petite sœur et complice de nos (mes ?) expériences loufoques et improbables durant notre enfance.

Je ne pouvais pas terminer ces remerciements sans mentionner Pascal, mon mari. Avec ton amour, ta patience et ton soutien, j'ai pu mener à bien ces trois ans de travaux avec confiance et sérénité. Merci d'avoir accepté de marcher à mes côtés. Merci pour tout.

A celles et ceux qui ont croisé mon chemin,
et qui m'ont donné envie de toujours aller plus loin.

« Regarde attentivement car ce que tu vas voir n'est plus ce que tu viens de voir. »
Léonard de Vinci

Table des matières

Remerciements **iii**

Table des matières **xii**

Table des figures **xvi**

Introduction générale **1**

I Contexte et vision d'ensemble **7**
I.1 L'agilité des processus collaboratifs 7
I.1.1 La problématique de la collaboration entre acteurs hétérogènes . . 8
I.1.2 L'évolution de la collaboration et/ou de son environnement 10
I.2 Les principaux enjeux 11
I.2.1 Le recueil de l'information 11
I.2.2 Le suivi des activités 12
I.2.3 Le traitement de l'information 13
I.3 Les pistes de réponse à l'agilité des processus 13
I.3.1 Les événements 14
I.3.1.1 Définition de l'événement 14
I.3.1.2 Le recueil des événements 15
I.3.1.3 Le traitement des événements complexes 16
I.3.2 Les modèles de situation collaborative 17
I.3.2.1 Dépendance entre modèles de situation et processus collaboratifs 17
I.3.2.2 Détecter la divergence 17
I.3.2.3 Définir le besoin d'adaptation 18
I.4 La crise, une situation collaborative particulière 19
I.4.1 Présentation des projets PLAY et SocEDA 21
I.4.1.1 Le projet PLAY 21
I.4.1.2 Le projet SocEDA 22
I.4.1.3 Liens avec les travaux de thèse 23
I.4.2 Un cas d'étude d'actualité 24
I.4.2.1 Fukushima, le deuxième accident nucléaire le plus grave de l'Histoire 24

I.4.2.2 Le cas d'étude 26
I.4.2.3 La caractérisation de la crise 28
I.4.2.4 Le processus collaboratif 29
I.4.2.5 Les services disponibles 31
I.5 Positionnement dans l'écosystème de travail 32
I.5.1 Le projet MISE 1.0 32
I.5.2 Le projet MISE 2.0 33
I.5.3 Le projet MISE 3.0 34
I.6 Conclusion 35

II État de l'art autour de l'agilité des processus 37
II.1 Étude par rapport à l'objectif 37
II.1.1 Esquisse du paysage bibliographique 37
II.1.2 Définition de l'agilité 39
II.1.2.1 Agilité de l'entreprise 39
II.1.2.2 Agilité des chaînes logistiques 40
II.1.2.3 Synthèse 43
II.1.3 De l'agilité des processus 44
II.1.3.1 Concepts d'agilité des processus 45
II.1.3.2 Solutions existantes 50
II.1.4 Conclusion sur les moyens existants 53
II.2 Étude par rapport aux moyens 55
II.2.1 Le recueil automatisé des données 55
II.2.1.1 Les événements 55
II.2.1.2 Les différents types d'événements 56
II.2.1.3 Le traitement des événements 58
II.2.1.4 Les agents de traitement d'événements 59
II.2.1.5 L'architecture orientée événements 64
II.2.1.6 SOA et EDA 69
II.2.2 La transformation de données en information 71
II.2.2.1 Comparer les données 71
II.2.2.2 La comparaison d'arbres XML 72
II.2.3 La transformation d'information en connaissances 74
II.3 Conclusion 75

III Mise à jour des modèles : Détection (partie 1) 77
III.1 Le modèle de la situation collaborative 78
III.1.1 Le modèle de la crise 78
III.1.1.1 Le modèle à l'instant t_0 78
III.1.1.2 Le modèle à l'instant t 78
III.1.2 Le modeleur de la crise 79
III.1.3 Les liens entre les concepts 81
III.1.4 Conclusion sur les structures des modèles 83

III.2 Contexte technologique de la mise à jour 84
III.2.1 L'architecture orientée service 84
III.2.2 Le SOAP 85
III.2.3 Le WSDL 86
III.2.4 Les événements 87
III.2.5 La typologie d'événements 87
III.2.6 L'abonnement aux événements 89
III.2.6.1 La norme WS-Notification 89
III.2.6.2 Implémentation de WS-N 90
III.2.7 Conclusion du contexte technologique 94
III.3 Mise en œuvre de la mise à jour des modèles 95
III.3.1 Les événements de suivi de l'exécution des processus 95
III.3.2 Les événements complexes générés par le CEP 96
III.3.3 Préparation à la mise à jour des modèles 98
III.3.3.1 La structure du fichier contenant le modèle 98
III.3.3.2 La préparation des modèles 98
III.3.3.3 Choix du modèle à mettre à jour 98
III.3.4 La mise à jour du modèle attendu 100
III.3.5 La mise à jour du modèle terrain 101
III.3.6 La notation des règles métier 105
III.4 Conclusion 106

IV Comparaison des modèles : Détection (partie 2) 107
IV.1 Détection des différences entre modèles 108
IV.1.1 La préparation des modèles 108
IV.1.1.1 La transformation XSL 109
IV.1.1.2 La suppression des espaces de nom 110
IV.1.1.3 Le formatage des données 111
IV.1.2 XMLUnit 113
IV.1.2.1 Principe de fonctionnement 113
IV.1.2.2 Le moteur de comparaison 114
IV.2 Coût de l'opération et importance de l'instance 116
IV.2.1 Le calcul de la divergence 116
IV.2.2 Le coût de l'opération 117
IV.2.3 L'importance de l'instance 119
IV.2.3.1 Sensibilité au temps 119
IV.2.3.2 Sensibilité au contexte 119
IV.2.3.3 Stockage 120
IV.2.4 Le calcul de la distance 120
IV.3 Algorithme de calcul de la distance entre les modèles 120
IV.3.1 Quelques précisions concernant la détection des différences 121
IV.3.2 L'objet DetailledDifference 122
IV.3.3 L'algorithme principal de XMLCompare 123

IV.3.4 L'implémentation de l'algorithme . . . 128
IV.4 Conclusion . . . 128

V Recommandations d'adaptation : Adaptation **129**
V.1 Contexte et limites . . . 130
V.1.1 Dépendance par rapport à la situation collaborative . . . 130
V.1.2 Pertinence des processus en cours d'exécution . . . 130
V.1.3 Sources d'adaptation . . . 131
V.2 Rappels sur la conception du SIM . . . 132
V.2.1 Modèle de collaboration . . . 133
V.2.2 Dynamique collaborative . . . 133
V.2.3 Modèle du SIM . . . 135
V.3 Différentes recommandations d'adaptation candidates . . . 137
V.3.1 Caractérisation de la situation collaborative . . . 138
V.3.2 Redéfinition du réseau collaboratif . . . 138
V.3.3 Réexécution d'un service . . . 139
V.4 Décision de la recommandation d'adaptation . . . 141
V.4.1 Facteurs de choix . . . 141
V.4.2 Nuance de l'adaptation . . . 141
V.4.3 Algorithme du choix de l'adaptation . . . 143
V.4.4 Algorithme de proposition d'adaptation à l'utilisateur . . . 153
V.4.5 Retour d'expérience . . . 155
V.5 Conclusion . . . 155

VI Illustration et implémentation **157**
VI.1 La réalisation du Service d'Agilité . . . 157
VI.1.1 Considérations préliminaires à la réalisation technique . . . 157
VI.1.2 Outils utilisés . . . 158
VI.1.3 Services utilisés . . . 159
VI.1.4 Architecture technique du Service d'Agilité . . . 160
VI.1.5 Licence logicielle . . . 164
VI.2 Présentation du cas d'étude . . . 165
VI.3 La mise en place de l'agilité . . . 167
VI.3.1 L'abonnement aux événements . . . 167
VI.3.2 Les règles métier de traitement des événements . . . 168
VI.3.3 Les matrices de coût et d'importance, le seuil . . . 170
VI.4 Déroulement du cas d'étude . . . 170
VI.4.1 À $t_0 + 20min$ . . . 171
VI.4.1.1 La mise à jour des modèles . . . 172
VI.4.1.2 Le calcul du delta . . . 173
VI.4.1.3 Le choix d'adaptation . . . 173
VI.4.2 À $t_0 + 40min$ . . . 175
VI.4.2.1 La mise à jour des modèles . . . 175

VI.4.2.2 Le calcul du delta . . . 176
VI.4.2.3 Le choix d'adaptation . . . 177
VI.4.3 À $t_0 + 70min$ . . . 178
VI.4.3.1 La mise à jour des modèles . . . 178
VI.4.3.2 Le calcul du delta . . . 179
VI.4.3.3 Le choix d'adaptation . . . 179
VI.5 Conclusion . . . 180

Conclusion et perspectives **183**

Bibliographie **200**

A Détails et chronologie du cas d'étude **201**
A.1 Première phase du scénario : t_0 à t_0+20 minutes . . . 201
A.2 Deuxième phase du scénario : t_0+20 minutes à t_0+30 minutes . . . 204
A.3 Troisième phase du scénario : t_0+30 minutes à t_0+105 minutes . . . 205

B Les processus collaboratifs BPMN du cas d'étude **209**

C Etude comparative des différents EPL **231**
C.1 Les solutions commerciales . . . 232
C.1.1 Progress Software - Progress Apama . . . 232
C.1.2 Tibco - TIBCO BusinessEvents . . . 232
C.1.3 Aleri - Aleri CEP 3.0 . . . 232
C.1.4 Coral8 Inc. - Coral8 . . . 233
C.1.5 Realtime Monitoring GMBH - RealTime Monitoring . . . 233
C.1.6 IBM - IBMWebSphere Business Events . . . 233
C.1.7 IBM - IBM Active Middleware Technology (Amit) . . . 233
C.1.8 Oracle - Oracle Complex Event Processing 10g . . . 234
C.1.9 BEA - BEAWebLogic . . . 234
C.1.10 Agent Logic - RulePoint . . . 234
C.1.11 StreamBase Systems - StreamBase . . . 235
C.1.12 Truviso Inc. - Truviso . . . 235
C.1.13 Rulecore - ruleCore CEP Server . . . 235
C.1.14 Senactive - Senactive InTime . . . 236
C.1.15 Event Zero - Event Zero . . . 236
C.2 Les moteurs issus de la recherche académique . . . 236
C.2.1 EsperTech Inc. - Esper/NEsper . . . 236
C.2.2 Middleware Systems Research Group (MSRG)/University of Toronto - PADRES . . . 236
C.2.3 CollabNet, Inc. - Intelligent Event Processor (IEP) . . . 237
C.2.4 UC Berkeley/University of Massachusetts Amherst - Stream-based And Shared Event Processing (SASE) . . . 237
C.2.5 Cornell University - Cayuga . . . 237

C.2.6 Brandeis University, Brown University, and MIT. - Aurora 238
C.2.7 Brandeis University, Brown University, and MIT - Borealis 238
C.2.8 Stanford University - STREAM 238
C.2.9 University of Marburg - PIPES 238

D Prioriser les risques en gestion de projet 239

E Format du fichier ServiceChoose.xml 241

F Exemple de fichier WSDL : contrat d'un Web Service permettant l'envoi d'email 245

G Schema XML des types d'événements en contexte de situation de crise CrisisEvents.xsd 249

H Schema XML du modèle de la situation allégé CrisisModel.xsd 253

I La norme WS-Notification 257
I.1 WS-BaseNotification 257
I.1.1 Abonnement 258
I.1.2 Suspendre/recommencer l'abonnement 258
I.1.3 Désabonnement 259
I.1.4 Notification 259
I.2 WS-BrokeredNotification 259
I.2.1 Notification 259
I.2.2 Abonnement 260
I.3 WS-Topic 260
I.3.1 Adressage des sujets 260
I.3.2 Topics ad-hoc 260

J Correspondance entre les événements et les éléments du modèle terrain 261

K Feuille de style pour la seconde transformations XSL 265

L Fichier de configuration d'Esper : esper.mise.cfg.xml 269

M Processus de réponse à la crise (cas d'étude simplifié) 277

N Règles métier implémentées dans Esper (cas d'étude simplifié) 282

Résumé 285

Table des figures

1 Organisation du manuscrit suivant la logique de conception et de réalisation de l'agilité des processus collaboratifs 4

I.1 Les alternatives d'architecture en réseau : de la centralisation à la médiation 9
I.2 Les quatre étapes du cycle de vie de la gestion de crise 19
I.3 Les principaux apports de la plateforme issue du projet européen PLAY [PLAY, 2010] 22
I.4 Vue aérienne du site de Fukushima Daiichi et représentation des parties submergées par le tsunami [Tokyo Electric Power Company, 2012] 25
I.5 Principes généraux du fonctionnement d'un réacteur à eau pressurisée, et les incidents conduisant à la situation de crise 27
I.6 Structure du méta-modèle de la situation collaborative [Bénaben, 2012] . 28
I.7 Structure du méta-modèle de la situation collaborative : le coeur [Bénaben, 2012] d'après [Macé-Ramète et al., 2012] 29
I.8 Structure du meta modèle de la situation collaborative, couche crise [Macé-Ramète et al., 2012] 30
I.9 Exemple d'un processus collaboratif du cas d'étude : implémentation du plan de circulation 31
I.10 Positionnement de la thèse par rapport aux travaux menés dans le projet MISE 33
I.11 Schéma d'architecture haut niveau et de positionnement du Service d'Agilité 36

II.1 Résultat des recherches sur ScienceDirect sur les 20 dernières années . . . 38
II.2 La Maison de l'Agilité, d'après [Charles, 2010] 41
II.3 Positionnement des actions de détection et d'adaptation dans le temps . . 45
II.4 Vue d'ensemble des approches de l'agilité appliquée au SI collaboratif , d'après [Hollingsworth, 1995] 46
II.5 Référentiel d'adaptation du comportement, [Bénaben, 2012] d'après [Schonenberg et al., 2008] et positionnement des concepts d'agilité 49
II.6 Positionnement des solutions étudiées dans le référentiel d'adaptation du comportement, [Bénaben, 2012] d'après [Schonenberg et al., 2008] 54
II.7 Les cinq catégories de traitement des événements [Etzion and Niblett, 2010] 58
II.8 Les quatre fonctions principales du traitement des événements [TIBCO, 2011] 60

II.9 L'EPA Filtre [Etzion and Niblett, 2010] 61
II.10 L'EPA Transformation [Etzion and Niblett, 2010] 62
II.11 Classification des différents types d'agent de transformation : les triangles représentent les instances d'événements, les flèches montrent le nombre de flux entrants et sortants par type d'événements [Etzion and Niblett, 2010] 62
II.12 L'EPA Détection de motif d'événement [Etzion and Niblett, 2010] 63
II.13 Principe du mécanisme Publication/Souscription [Maréchaux, 2006] . . . 64
II.14 Concepts de CEP d'après [Fülöp et al., 2012] 66
II.15 Similarité entre deux documents XML 72

III.1 Architecture simplifiée du Service d'Agilité et périmètre de l'étude dans le Chapitre III 77
III.2 Exemple très simple de l'utilisation du modeleur de la situation de crise . 80
III.3 Positionnement de l'étape d'obtention des liens entre *problèmes* et *solutions* dans MISE 81
III.4 Mise à jour des modèles : état des lieux 83
III.5 Principes du mécanisme d'une architecture SOA 85
III.6 Structure du WSDL 1.1 86
III.7 Les huit types d'événements couvrant une situation de crise 88
III.8 Le rôle du CEP dans le traitement des événements en vue de la mise à jour des modèles 97
III.9 Le fichier du modèle de la situation avant et après retrait des données concernant l'affichage graphique du modèle 99
III.10 Correspondance entre l'événement *Alert* généré par le CEP et les éléments du modèle *terrain* 104
III.11 Des évéments à la mise à jour des modèles : un traitement de type CEP et une analyse des événements émis par le CEP 105

IV.1 Architecture simplifiée du Service d'Agilité et périmètre de l'étude dans le Chapitre IV 107
IV.2 Résultat de la première transformation XSL : les espaces de nom ont été supprimés 111
IV.3 Principe de la nouvelle structure des données des fichiers XML des modèles terrain et attendu 112

V.1 Architecture simplifiée du Service d'Agilité et périmètre de l'étude dans le Chapitre V 129
V.2 Les trois niveaux de conception du SIM dans MISE : Design Time 132
V.3 La cartographie de processus dans MISE 2.0, d'après [Mu, 2012] 135
V.4 Principes de la réconciliation n-m des services 136
V.5 La réconciliation des formats de données des messages échangés entre les services : données liées et règles de transformation associées, d'après [Boissel-Dallier, 2012]. 137

V.6 Problématique actuelle : effectuer la correspondance entre la divergence et les recommandations d'adaptation . . . 143
V.7 Le méta-modèle (réduit) de la crise . . . 145
V.8 Le diagramme de classes simplifié du modèle de la situation de crise . . . 145
V.9 Le cas de la mise à jour d'un *Sous Composant* et de la suppression son lien avec une *Perturbation* (voire la suppression de la *Perturbation*) . . . 146
V.10 Le cas de la suppression d'une *Perturbation* et d'un *Sous Composant* qui lui est lié . . . 147
V.11 Le cas de l'ajout d'un *Sous Composant* sans lien avec une *Perturbation* . 148
V.12 Le cas de la suppression d'une *Perturbation* et de son lien avec un *Service* 148
V.13 Le cas de la suppression d'un *Service* et de toutes ses *Perturbations* liées . 149
V.14 Le cas de la suppression d'un *Service* mais pas de toutes ses *Perturbations* liées . . . 149
V.15 Le cas de la suppression d'un *Partenaire*, de tous ses *Service* et de toutes les *Perturbations* liées à ses *Service* . . . 150
V.16 Le cas de la suppression d'un *Partenaire*, de tous ses *Service* mais pas de toutes les *Perturbations* liées à ses *Service* . . . 150
V.17 Le cas de la panne de service . . . 151

VI.1 Exemple d'utilisation de l'outil de génération des Web Services . . . 158
VI.2 Web Services de Design Time et de Runtime . . . 160
VI.3 Architecture technique de l'implémentation du Service d'Agilité . . . 161
VI.4 Modèle de la situation de crise à t_0 . . . 166
VI.5 Mise en place de la déviation par la DIRSO . . . 167
VI.6 Règle de détection d'un incendie écrite en EQL . . . 169
VI.7 Evénements *Mesure* émis par les capteurs . . . 171
VI.8 Réception d'événements *Mesure* émis et déclenchant la règle d'émission d'une *Alerte* « incendie » par MISECep . . . 171
VI.9 Réception par ModelsUpdate de l'événément *Alerte* émis par MISECep . 172
VI.10 Modèle de la situation de crise à $t_0 + 20min$ . . . 172
VI.11 Résultat de la détection à $t_0 + 20min$ . . . 173
VI.12 Proposition d'adaptation à $t_0 + 20min$ . . . 173
VI.13 Choix de retour à la caractérisation de la situation à $t_0 + 20min$ . . . 174
VI.14 Résultat de l'adaptation à $t_0 + 20min$ . . . 175
VI.15 Réception d'événements *Activité* émis par la supervision des workflows et déclenchant la règle de filtrage des *Activité*par MISECep . . . 176
VI.16 Modèle terrain (à gauche) et modèle attendu (à droite) de la situation de crise à $t_0 + 40min$ . . . 177
VI.17 Résultat de la détection à $t_0 + 40min$ . . . 177
VI.18 Proposition d'adaptation à $t_0 + 40min$ . . . 178
VI.19 Réception d'événements *OfferResource* émis et déclenchant la règle de filtrage des *OffreRessource*par MISECep . . . 178
VI.20 Modèle terrain de la situation de crise à $t_0 + 70min$ . . . 179

VI.21Résultat de la détection à $t_0 + 70min$ 179
VI.22Proposition d'adaptation à $t_0 + 70min$ 180
VI.23Positionnement de la thèse par rapport aux travaux menés dans les trois itérations du projet MISE 188

A.1 Le workflow de gestion de la situation 202
A.2 Le workflow de gestion de la situation 203
A.3 Le workflow de protection de la population 205
A.4 Le workflow de d'implémentation du plan de circulation 206
A.5 Le workflow de gestion des ressources de la DIR 207

D.1 Exemple de matrice de criticité 240

I.1 Principes de fonctionnement du NotificationBroker 259

Introduction générale

De nos jours les organisations (qu'il s'agisse d'entreprises, d'institutions ou d'administrations) doivent œuvrer ensemble et prendre part à des collaborations pour être capables de partager leurs compétences afin de faire face à un environnement instable (mondialisation, crise).

Ce besoin d'interconnexion, et plus précisément de *collaboration*, est révélé par des contextes aussi nombreux que variés tels que les réseaux sociaux, la domotique, les partenariats industriels, la sous-traitance, ou même les situations de crises, qu'elles soient économiques, civiles, militaires, humanitaires, sanitaires, sociales, environnementales.

De plus, notre environnement tend à devenir un immense réseau de données où les personnes, les appareils et les organisations sont impliqués dans des processus collaboratifs qui les interconnectent.

Considéré comme la partie émergente des organisations et leur support « fonctionnel », le Système d'Information (SI) devient le support de la collaboration en permettant aux organisations de partager leurs compétences au sein de processus collaboratifs, en vue d'atteindre le but commun.

Cependant participer à un processus collaboratif et devenir un partenaire d'une collaboration n'est pas nécessairement une chose aisée, et ce d'autant plus que la collaboration est éphémère. En outre, les collaborations sont instables de par l'environnement fluctuant dans lequel elles évoluent, mais aussi une mauvaise perception de ce dernier. La nature même des collaborations est elle aussi cause d'instabilité : les organisations membres peuvent évoluer, disparaître, leurs services être indisponibles, ou a contrario d'autres organisations peuvent rejoindre la collaboration, de nouveaux services peuvent être proposés. Dans ce contexte, la poursuite des objectifs peut être menacée par les évolutions de la collaboration et de son environnement.

La courte durée de vie et l'instabilité de la collaboration sont particulièrement visibles dans le domaine de la réponse à la crise [1]. Le besoin de flexibilité et de réactivité des plans de réponse à la crise n'est plus à démontrer : si des plans existent et sont établis, l'expérience montre que ces derniers sont bien souvent dépassés par l'ampleur de

1. Le domaine de la gestion de crise servira de cadre à notre étude de cas « fil rouge » dans ces travaux de thèse, afin de représenter la classe d'application que sont les Workflows Inter-Organisationnels en environnement instable.

la crise. L'imprévu reste par essence imprévisible et la cellule de crise doit pouvoir être informée de la réalité du théâtre des opérations mais aussi des effets réels des workflows collaboratifs de réponse sur la crise. Ces derniers ont-ils l'effet attendu (i.e. répondent-ils aux objectifs) ? Si ces workflows n'ont pas l'effet escompté, comment réagir ?

De ce constat, deux problématiques majeures émergent :

1. *Comment détecter le moment et les causes de l'inadéquation des processus collaboratifs exécutés par rapport aux objectifs poursuivis à l'instant t ?*
2. *Comment redéfinir la meilleure réponse possible à l'instant t —i.e. à un niveau concret les meilleurs workflows, ainsi que leur orchestration et leur chorégraphie—, dans un laps de temps proche du temps réel, en fonction de la situation de collaboration, de son contexte, de l'avancée de l'exécution des processus, et l'état des acteurs et ressources.*

Ces travaux de thèse cherchent à répondre à ces deux problématiques en abordant la question de la définition de *l'agilité des workflows collaboratifs*. La *définition* que nous en avons retenue nous a amené à nous pencher sur la manière de *recueillir les informations pertinentes* pour *caractériser* la situation collaborative en temps réel, puis ensuite à traiter le problème de la *détection* d'un manque de pertinence des processus collaboratifs exécutés à un instant t.

Cette thèse a été menée et encadrée au sein de l'axe *Interopérabilité des Organisations*, dans le laboratoire Centre de Génie Industriel de Mines Albi. L'héritage reçu des travaux de thèse réalisés précédemment au sein de cet axe sur la thématique de l'interopérabilité et de la médiation est particulièrement présent dans le cadre du projet interne MISE (Mediation Information System Engineering). MISE vise à définir et concevoir, de façon automatique et via une approche dirigée par les modèles, le support de l'interopérabilité des organisations engagées dans des collaborations, c'est-à-dire un Système d'Information de Médiation. Les présents travaux de thèse s'inscrivent dans la seconde itération du projet. MISE 2.0 vise notamment à apporter l'agilité nécessaire au SIM (objet de ces travaux), afin de répondre aux problématiques présentées plus avant dans cette introduction.

Les présents travaux de thèse s'inscrivent dans le cadre de deux projets de recherche. Le premier est un projet de recherche financé par l'ANR (Agence Nationale de la Recherche) et intitulé SocEDA (Social Event Driven Architecture). SocEDA propose une architecture orientée services et dirigée par les événements. Le but du projet est de fournir une plateforme permettant de collecter les événements émis par l'environnement d'exécution des services des utilisateurs de la plateforme. Une fois analysés (du point de vue de la confiance dans les données mais aussi du point de vue métier), ces événements peuvent permettre le pilotage de l'orchestration de processus. Un des enjeux du projet est de créer un outil permettant de rendre les processus collaboratifs agiles et d'automatiser leur adaptation à la situation collaborative courante. Les travaux présentés dans ce

manuscrit ont été logiquement utilisés pour répondre à cet enjeu.

Le projet PLAY (Pushing dynamic and ubiquitous interaction between services Leveraged in the Future Internet by ApplYing complex event processing) est un projet européen (FP7 ICT) visant à réaliser une plateforme fournissant à ses utilisateurs la possibilité d'acquérir une sensibilité à l'environnement d'exécution de leurs systèmes et le moyen de réagir en conséquence, y compris en affectant l'exécution de leurs processus métier. Cette plateforme est alimentée par plusieurs cas d'études, dont un cas réaliste de réponse à une crise nucléaire civile qui met en évidence la nécessité d'être à l'écoute des évolutions de l'environnement de la collaboration mais aussi le besoin d'adapter le comportement collaboratif à ces évolutions.

Nous avons étudié de quelle façon nous pouvions conduire les changements nécessaires dans les processus collaboratifs pour les *adapter* à leurs nouvelles conditions d'exécution : de quelle façon identifier la source majeure du manque de pertinence, comment déterminer et proposer des solutions réalistes compte tenu des contraintes de délai. Enfin, nous avons implémenté ces mécanismes dans un *prototype* alimenté par un cas d'étude inspiré d'une crise nucléaire civile. Ce cheminement, illustré par la Figure 1, nous amène tout naturellement au plan suivant.

- Le Chapitre I expose la problématique et le cadre de ces travaux de thèse. Dans un premier temps nous détaillerons le contexte de ces travaux, à savoir le besoin d'agilité des processus collaboratifs, compte tenu de l'évolution de l'environnement de la collaboration. Les différents enjeux de la problématique soulevée consistent à recueillir l'information, suivre l'état des activités des processus et le traitement de l'information collectée. Ensuite, différentes pistes de réponse seront présentées afin de répondre aux enjeux identifiés. Enfin, nous positionnerons cette thèse par rapport aux différents projets de recherche (ANR SocEDA, européen FP7 PLAY et interne au laboratoire MISE) dans lesquels elle s'inscrit.
- Le Chapitre II présente l'état de l'art concernant l'agilité des processus collaboratifs, tant dans le domaine académique que dans le domaine commercial. Nous nous attacherons en premier lieu à étudier la notion d'agilité des processus collaboratifs. Nous verrons que si l'agilité des processus en général et l'agilité des collaborations sont largement discutées dans la littérature, peu d'auteurs se sont penchés sur la notion d'agilité des processus collaboratifs. Sur la base de notre étude, nous en *proposerons une définition* sur laquelle nous baserons la suite de notre exposé. Puis nous positionnerons notre approche de l'agilité (adaptation *ad-hoc*, au sens du cadre défini par [Schonenberg et al., 2008]) par rapport aux travaux existants. Nous creuserons également les pistes de réponse évoquées dans le chapitre précédent afin de mettre en œuvre notre définition de l'agilité.
- Le Chapitre III s'intéresse à développer les mécanismes de recueil d'information à propos de la situation collaborative. Dans un premier temps, les données émises par le théâtre de la collaboration et la supervision des workflows collaboratifs sont collectées. Ces données sont filtrées voire agrégées afin de déduire de l'informa-

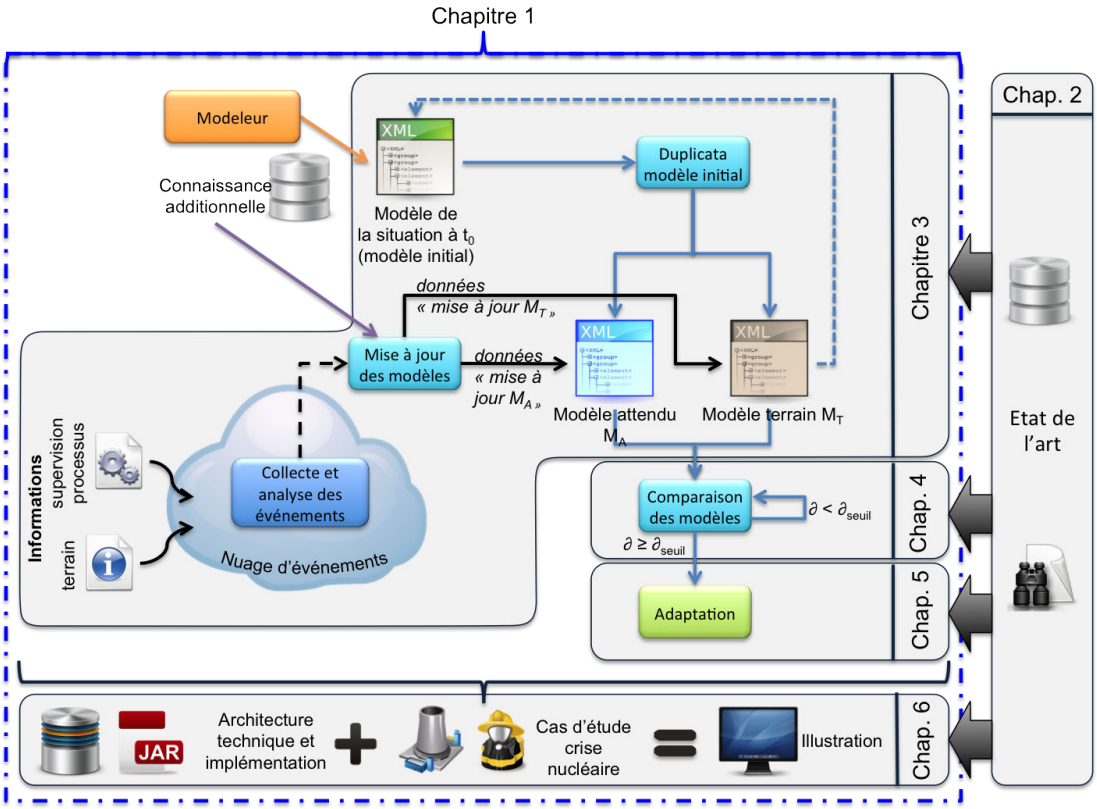


FIGURE 1 – Organisation du manuscrit suivant la logique de conception et de réalisation de l'agilité des processus collaboratifs

tion concernant la situation collaborative et de mettre en lumière ses évolutions. Les modèles *terrain* et *attendu* sont des représentations de la situation collaborative réalisées suivant deux points de vue : la photographie, réalisée à partir des données émises par les équipes, appareils et capteurs présents sur le théâtre de la collaboration (modèle *terrain*), et le portrait-robot, esquissé à partir des événements émis par les activités des workflows collaboratifs en cours d'exécution (modèle *attendu*). L'intégration d'une EDA (Event-Driven Architecture, pour Architecture Dirigée par les Evénements) à une SOA (Service Oriented Architecture, pour Architecture Orientée Services) permet de faciliter ce recueil des données émises tant par le terrain de la collaboration que par le suivi de l'exécution des workflows, ainsi que leur interprétation.

- Le Chapitre IV se focalise sur la détection des causes d'inadéquation des processus collaboratifs en cours d'exécution avec les objectifs communs de la collaboration. Si les processus exécutés n'ont pas les effets escomptés, la photographie de la situation collaborative va présenter de moins en moins de similitudes avec son portait-robot. Il s'agit ici de détecter et d'identifier les différences δ_i présentes entre les modèles *terrain* et *attendu* et de les interpréter pour calculer l'ampleur de

la différence globale δ entre les modèles. La détection est l'étape qui va *analyser les différences trouvées* et autoriser une modification (ou adaptation) des processus collaboratifs.

- Le Chapitre V s'attache à proposer un mécanisme d'aide à la décision concernant l'adaptation des processus collaboratifs. Cette adaptation consiste en trois options, cohérentes avec les travaux menés par [Mu, 2012] et [Boissel-Dallier, 2012] sur la conception et le déploiement automatisés de processus collaboratifs. Les solutions d'adaptation issues de leurs travaux consistent en une refonte (i) partielle ou (ii) totale de la logique collaborative ainsi qu'en (iii) une re-exécution d'une sous-partie des processus collaboratifs déployés. Notre travail consiste à *définir les profils de* δ (par analyse de ses composantes δ_i) conduisant à telle ou telle solution d'adaptation. Le résultat de cette étape propose à l'utilisateur d'une liste de choix d'adaptation.
- Le Chapitre VI expose l'architecture technique du Service d'Agilité défini dans ces travaux de thèse ainsi que son implémentation. Ce Service d'Agilité couvre tous les aspects étudiés dans les chapitres précédents : collecte, analyse et déduction des événements par un moteur de traitement des événements complexes, mise à jour des modèles dirigée par les événements, détection des divergences entre les modèles et proposition d'adaptation des processus collaboratifs. Ces quatre composantes majeures du Service d'Agilité ont été développées sous forme de librairies Java open-source qui sont appelées par un serveur d'applications (via une application web) et un bus de services. Une démonstration de l'utilisation du Service d'Agilité est faite à travers le déroulement d'un cas d'étude de type crise nucléaire. Cet exemple est inspiré des cas d'études développés dans les projets de recherche PLAY et SocEDA menés dans le cadre de cette thèse.

Chapitre I

Contexte et vision d'ensemble

L'objectif de ce premier chapitre est d'expliquer les objectifs de ces travaux de thèse. Nous allons d'abord nous intéresser à la problématique et au contexte dans lequel elle se positionne pour ensuite nous pencher sur les différentes méthodes envisageables pour la résoudre. Puis nous proposerons une méthodologie complète avant de présenter l'exemple fil rouge issu des projets de recherche français SocEDA et européen PLAY. Enfin, nous positionnerons ces travaux de thèse par rapport aux autres travaux menés dans le laboratoire, notamment vis-à-vis du projet MISE 2.0 (Mediation Information System Engineering) et des thèses liées à ce projet.

I.1 L'agilité des processus collaboratifs

De nos jours, les organisations (telles que les entreprises, les institutions, les administrations, etc.), les personnes et même les systèmes connectables (appareils, capteurs, etc.) doivent travailler ensemble et prendre part à des collaborations de façon à faire face à un environnement instable. Cette nécessité de connexion, et plus précisément de collaboration, transparait dans des contextes aussi nombreux que les réseaux sociaux, la domotique, les collaborations professionnelles, la sous-traitance, les situations de crise.

Ainsi notre environnement tend à devenir un vaste réseau de personnes, d'appareils et d'organisations (à savoir les partenaires de collaboration), qui exécutent des processus collaboratifs entre eux. D'autre part, non seulement les organisations se dirigent vers la mise en réseau global, mais on assiste à un lent changement de leurs réseaux organisationnels qui tendent à être remplacés par des structures plus fluides, amorphes [1] et souvent transitoires, basées sur des alliances, partenariats et collaborations.

Cette mutation entraîne alors le problème de l'agilité des processus collaboratifs : qu'est-ce que l'agilité des processus collaboratifs ? Comment se traduit-elle ? S'agit-il d'introduire les notions de l'agilité, à savoir le besoin de flexibilité, réactivité ou encore d'adaptabilité, comme définit par [Kidd, 1994], [Badot, 1998] et [Sharifi and Zhang, 1999] dans la définition des processus ? Qu'en est-il au niveau de l'exécution des processus ?

1. Au sens physique du terme, c'est-à-dire non cristallisées.

Comment rendre ces processus agiles ? Toutes ces questions sur l'agilité sont discutées en détails dans le Chapitre II.

I.1.1 La problématique de la collaboration entre acteurs hétérogènes

Être capable de collaborer avec les clients, les fournisseurs ou même des concurrents a toujours été une exigence critique dans notre écosystème moderne basé sur des organisations multiples. Toutefois, si la collaboration concerne habituellement des organisations proches (d'un point de vue géographique), requérant du temps pour définir une relation stable et durable, ceci n'est plus le cas : de nos jours, les organisations doivent établir leurs collaborations, potentiellement éphémères, avec des partenaires venant du monde entier. Ceci doit être réalisé de façon réactive afin de pouvoir saisir des opportunités d'affaires souvent éphémères.

De plus, les attentes des partenaires vis-à-vis de la collaboration deviennent de plus en plus élevées [Bénaben, 2012] : d'abord cantonnée à un partage des (ou plutôt de certaines) données entre les participants (faible intensité collaborative), la collaboration a vu son rôle évoluer vers un partage et une synchronisation des tâches pour former une sorte de lointain ancêtre du processus collaboratif. Elle gagne alors en profondeur. Mais ce n'est qu'avec la réelle poursuite d'un objectif commun à tous les partenaires que le processus collaboratif voit le jour au sein de la collaboration et donne par la même sa raison d'être à la collaboration. Enfin, ultime strate dans l'intensité de la collaboration, l'appartenance à une même structure globale ([Bénaben, 2012] parle de « fusion ») qui peut être virtuelle ou réelle. Cette fusion complète la strate précédente en devenant le support de l'échange de données, du partage des tâches et de la poursuite de l'objectif commun.

On peut affirmer que l'écosystème de l'entreprise voire de l'organisation au sens large a évolué d'une structure fortement cristallisée vers un environnement très fluide. Dans ce contexte de libre circulation, la collaboration est plus une manière de saisir une opportunité et de rester dynamiquement sur le haut de la vague qu'un élément structurant, en charge de définir l'intensité de l'intégration des organisations dans leur environnement géographique et métier.

En outre, les systèmes d'information (SI) peuvent être considérés d'une part comme l'épine dorsale fonctionnelle des organisations —dans la mesure où ils assument la gestion de leur information, de leurs fonctions et de leur comportement— et d'autre part comme l'interface principale avec tout partenaire potentiel. Par conséquent, la gestion de la collaboration des organisations doit nécessairement envisager la réalisation de l'interopérabilité des systèmes d'information.

La mise en place d'une situation collaborative implique de travailler avec des organisations hétérogènes. Considérant le fait que le SI est la partie visible d'une organisation, le but est de résoudre la question de la collaboration des organisations grâce à l'interopérabilité des SI en satisfaisant les exigences métier. D'après le Laboratoire

Virtuel International InterOp VLab, l'intéropérabilité est « la capacité d'un système ou d'un produit à travailler avec d'autres systèmes ou produits sans effort particulier du client ou de l'utilisateur » [Konstantas et al., 2006]. L'interopérabilité est également définie par Pingaud dans [Pingaud, 2009a] comme « la 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 individuel ». Comme il apparaît que les SI des partenaires ne peuvent nativement prendre en charge les fonctions de transfert et de traduction (syntaxique et sémantique) des données, de la gestion des services et de l'orchestration des workflows [2] collaboratifs (excepté avec de hautes contraintes de normalisation technique, ce qui ne correspond pas aux définitions données de l'interopérabilité), un moyen de réaliser ces fonctions collaboratives doit être trouvé.

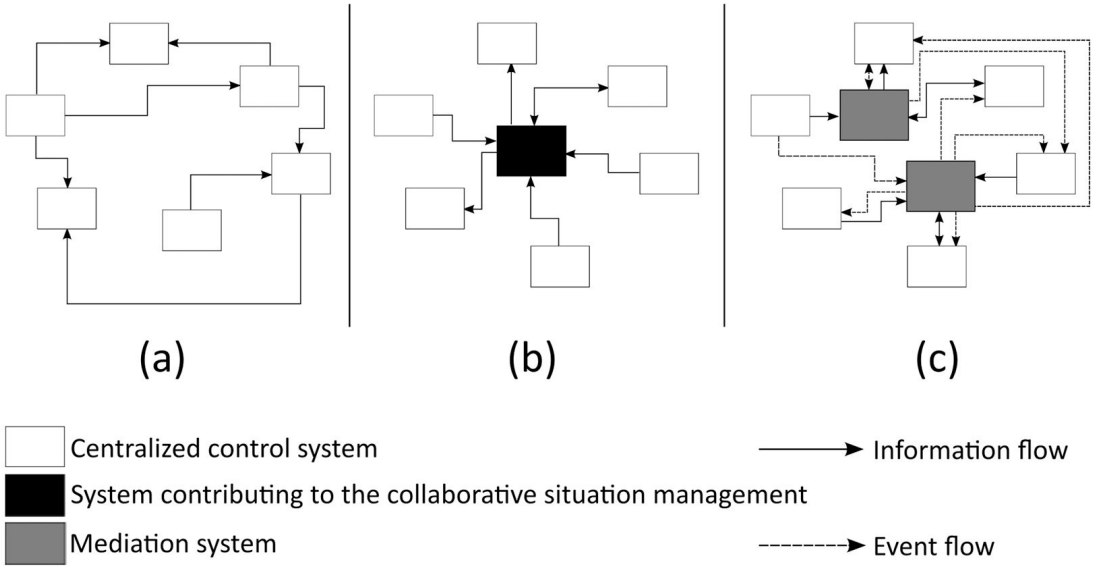


FIGURE I.1 – Les alternatives d'architecture en réseau : de la centralisation à la médiation

La Figure I.1 présente trois types d'alternatives architecturales. Le cas (a) est basé sur une approche point-à-point : si tous les partenaires partagent les mêmes standards, il est clair que la réactivité et la flexibilité sont favorisées par une telle architecture. Mais il s'agit d'une architecture fortement couplée, ce qui signifie que dès que les standards ne sont plus partagés les contraintes techniques d'implémentation se révèlent coûteuses au regard du contexte dynamique de la collaboration et du temps nécessaire pour réaliser une telle architecture de collaboration.

A l'inverse, le cas (b) propose une architecture centralisée : relativement simple, elle garantit que tout partenaire capable de connecter son SI au point de centralisation est alors connecté aux autres SI du réseau (le point de centralisation assurant le rôle de traducteur de données entres les SI des partenaires). Le défaut d'une telle architecture

2. Un workflow est la vision automatisée de la totalité ou d'une partie d'un processus.

est de concentrer le pilotage sur l'unique point de centralisation.

Enfin, le cas (c) expose une architecture de médiation (telle que décrite par Wiederhold [Wiederhold, 1992]) : ce type d'architecture est à couplage lâche (ce qui correspond aux contraintes de l'évolution de la collaboration), où un SI orchestre tous les échanges. La différence avec l'architecture centralisée réside dans le fait que les centralisations peuvent être multipliées (suivant des critères thématiques ou de sécurité), en les connectant de pair-à-pair. Un SIM (Système d'Information de Médiation), basé sur les principes de la SOA (Service Oriented Architecture pour Architecture Orientée Services), à savoir un système tiers en charge de la coordination des activités des acteurs suivant des processus collaboratifs, est un moyen crédible et pertinent pour devenir le support de l'interopérabilité des SI [Bénaben et al., 2008].

Dans la suite de ce chapitre, nous verrons qu'un prototype de SIM a été réalisé dans le cadre du projet de recherche MISE (cf. Section I.5).

I.1.2 L'évolution de la collaboration et/ou de son environnement

Comme souligné précédemment, les situations collaboratives tendent à devenir de plus en plus plastiques de par leur nature même ou de par l'évolution de la cinétique de leur environnement. Il convient donc de prendre en compte les aléas auxquels la dynamique opérationnelle de la collaboration peut être exposée, afin de faire évoluer le SIM si besoin est. Les sources d'adaptation peuvent être :

- Internes à la collaboration : arrivée ou départ de partenaires, mise à disposition d'une nouvelle ressource, modification des objectifs de la collaboration, etc.,
- Externes à la collaboration : changement des caractéristiques de l'environnement dans lequel la collaboration se déroule, par exemple.

Dans l'écosystème actuel, la dynamique opérationnelle de la collaboration devient une composante essentielle de la collaboration entre organisation. Il est donc crucial de tenir compte des évolutions possibles dans une situation collaborative, afin d'y mener les actions nécessaires à la garantie de sa pertinence vis-à-vis des objectifs collectifs. D'après [Pingaud, 2009b], les types d'évolutions des situations collaboratives peuvent être classées ainsi :

- Evolution du contexte. Ce premier type d'évolution concerne l'environnement de la collaboration qui diffère de celui pris en compte pour définir le comportement collectif du réseau collaboratif (i.e. les processus collaboratifs). Il peut ainsi s'agir d'un réel changement (par exemple la mutation d'une catastrophe naturelle en catastrophe technologique, l'évolution du marché économique) ou de la simple prise en compte d'informations ignorées jusqu'alors (par exemple, la découverte d'une panne dans un système de refroidissement, ou la communication de la mauvaise santé économique d'un donneur d'ordres),
- Evolution du réseau. Ce second type relève des partenaires, de leurs compétences et/ou de leurs ressources. Il peut s'agir, par exemple, du départ d'un partenaire (et de ses compétences et ressources associées), de l'arrivée d'un nouvel acteur, de l'indisponibilité temporaire ou définitive d'une compétence, l'épuisement ou la

soudaine disponibilité d'une ressource,

- Occurence du dysfonctionnement. Ce dernier type d'évolution souligne la possibilité qu'une ou plusieurs activités n'aient pas eu l'effet escompté, de par une mauvaise évaluation des activités à mener ou de par une mauvaise exécution de ces dernières (panne, par exemple). L'un des exemples les plus marquants de ces dernières années en la matière est le largage d'eau sur les réacteurs et les piscines de combustible usagé de la centrale nucléaire de Fukushima Daiichi. D'une part le largage n'a pu être réalisé correctement (mauvaise exécution), d'autre part l'apport d'eau n'a permis de refroidir ni réacteurs ni piscines (mauvaise évaluation). La dynamique comportementale ne peut être poursuivie en restant sourde et aveugle aux informations remontant du théâtre de la collaboration : elle doit être interrompue et adaptée.

Compte tenu de ces éléments, il devient essentiel de suivre d'une part l'évolution de l'environnement, théâtre de la collaboration, et de la collaboration elle-même d'autre part, afin de détecter une éventuelle inadéquation entre les objectifs poursuivis et les moyens (au sens large : ressources, activités et leurs états associés) à un instant t pour y parvenir.

L'agilité du SIM est un véritable enjeu de la situation collaborative, qui va avoir des effets sur l'architecture même du SIM. La notion d'agilité sera plus largement discutée dans la suite du manuscrit (Chapitre II).

I.2 Les principaux enjeux

« Savoir, c'est pouvoir » affirmait Sir Francis Bacon. Si le savoir (i.e. la connaissance) semble être un pré-requis naturel au pouvoir (i.e. à la capacité d'agir) il ne confère pas pour autant beaucoup de pouvoir. Le pouvoir réside dans la capacité à utiliser le savoir et dans le savoir-faire nécessaire pour utiliser le savoir. Autrement dit, il ne s'agit pas seulement de recueillir de la connaissance à propos de la collaboration et de son environnement, il faut aussi pouvoir exploiter cette connaissance et en extraire la substantifique moelle qui nous permettra de prendre une décision, c'est-à-dire d'agir. Recueillir, observer et analyser l'information constituent les trois grands enjeux de l'agilité des processus en vue d'agir, i.e. d'adapter nos processus collaboratifs en fonction du besoin identifié.

I.2.1 Le recueil de l'information

Le recueil de l'information et de la connaissance relative à la situation considérée sont essentiels afin de pouvoir définir (i) le contexte de la collaboration (i.e. l'environnement dans lequel se positionne la collaboration), (ii) les objectifs de la collaboration (i.e. sa raison d'être ainsi que les motivations collectives des partenaires de la collaboration), et enfin (iii) les moyens de la collaboration (i.e. les compétences et ressources que les partenaires sont prêts à partager).

Ce recueil de l'information doit pouvoir suivre les enjeux de la collaboration : si au cours de la vie de la collaboration, il devient crucial d'obtenir un certain type d'infor-

mation, le recueil doit en être facilité. De même, si à un moment donné, il n'est plus pertinent de collecter un certain type d'information, il doit être possible d'abandonner cette source d'information de la manière la plus souple possible, sans avoir à réaliser de modifications majeures dans l'infrastructure du SIM de la collaboration.

Un autre enjeu du recueil de l'information est la confiance que l'on peut accorder aux données. Bien qu'hors sujet dans le cadre de ces travaux de thèse, nous tenons à en signaler l'existence. Cet enjeu porte sur deux niveaux : d'une part sur la source de données, d'autre part sur les données elle-même. En effet, la confiance dans la source de données doit pouvoir être établie avant de recueillir ses données, mais doit être également vérifiée au cours de la vie de la collaboration. Ensuite, même si la source de données est fiable, rien n'interdit que certaines de ses données soient aberrantes (à cause d'une panne, d'une défaillance, d'un mauvais paramètrage de capteur, etc.). Il doit être possible de vérifier la fiabilité de la source de données et de ses données à n'importe quel moment et, si nécessaire, de rompre l'échange de données avec la source incriminée.

I.2.2 Le suivi des activités

Nous avons pu voir précédemment que l'agilité des processus collaboratifs était un enjeu crucial pour assurer la survie et la pérennité de la collaboration, du fait du caractère évolutif de la collaboration (causes internes) ou de son environnement (causes externes).

Il faut donc suivre la progression de l'exécution de la logique collaborative pour déterminer l'efficacité des processus collaboratifs, compte tenu de l'état du réseau collaboratif et de son environnement à un instant donné. La supervision des processus passe par l'observation de l'exécution des workflows, et plus précisément de leurs activités. Suivre l'exécution des activités (granularité fine) permet de vérifier la bonne marche du processus (granularité haute) et de déduire une sorte de bilan de santé de la situation collaborative. Par exemple, mettons que l'on fait appel à l'activité « Evacuer victimes » effectuée par une équipe du SAMU pour évacuer des victimes. Si l'activité « Evacuer victimes » indique que son état passe de « en cours » à « terminé », on peut en déduire qu'il n'y a plus de victimes sur place.

Cependant, le suivi des activités ne peut en effet s'effectuer sur la seule foi du suivi de l'exécution des processus (monitoring) en terme de statut d'activité (i.e. en attente, commencé, en cours, terminé). Si l'on se fie uniquement aux statuts des activités pour en déduire les effets sur la collaboration et son environnement, le risque est grand de rester sourd et aveugle face à des dysfonctionnements ayant lieu sur le théâtre de la collaboration et qui n'ont pu, par définition, être traduits par les statuts des activités. Si nous reprenons l'exemple de l'évacuation des victimes, le SAMU a bien évacué les victimes recensées. Malheureusement, après le départ du dernier véhicule d'évacuation (le statut de l'activité passe à « terminé »), le SAMU découvre deux nouvelles victimes sur les lieux de la crise. Il faut également les évacuer. L'activité « Evacuer victimes » n'est donc pas terminée. Il est donc nécessaire de pouvoir obtenir une vision fidèle de ce

qui se produit sur le terrain.

Il faut donc pouvoir réaliser un autre suivi des activités en parallèle du monitoring, en adoptant cette fois-ci un point de vue terrain : les effets de la collaboration sont directement constatés et non déduits. La corrélation activité-effet devra permettre de confirmer ou d'infirmer les déductions réalisées à partir des événements provenant de la surveillance de l'exécution des activités au sens du workflow.

I.2.3 Le traitement de l'information

L'enjeu premier du traitement de l'information va être de filtrer les données pertinentes et de leur donner un sens, transformant ainsi le flux de données entrantes en information exploitable par le SIM. Cet aspect là est essentiel dans l'établissement de la situation collaborative (définition de la logique collaborative) puis dans sa gestion en phase d'exécution. Il est à noter que dans un contexte où la réactivité est cruciale, les responsables de l'établissement de la collaboration sont soit submergés par les données (saturation des opérateurs), soit en manque de données (données sans importance prises séparément, signaux faibles ou absents) [Lagadec, 1992] . Dans ces conditions, le SIM doit pouvoir décharger l'humain d'une partie de son travail de recherche des données et de corrélation entre les données reçues (pour pallier au manque de données), et avoir un rôle de support d'aide à la prise de décision (pour faciliter l'analyse des nombreuses données).

En outre, à l'instar du cas du recueil de l'information, le traitement de celle-ci se doit de pouvoir évoluer au gré des besoins de la collaboration. Si l'on reprend l'exemple des réseaux de capteurs dans les structures industrielles, ceux-ci sont physiquement reliés au centre de contrôle. Les règles d'interprétation des données entrantes émises par les capteurs sont littéralement figées par les connexions physiques [Barthe-Delanoë et al., 2013a]. Or il est essentiel de pouvoir faire évoluer les règles d'interprétation, de filtrage et d'agrégation des données dans le but de pouvoir d'une part suivre les exigences de la collaboration au cours de sa vie (il n'est plus pertinent de surveiller certaines données à un instant t), et d'autre part éduquer le système en favorisant les règles les plus adaptées à la situation collaborative courante (aspect capitalisation et retour d'expérience du traitement de l'information).

I.3 Les pistes de réponse à l'agilité des processus

L'événement permet de représenter une donnée émise par un système connecté (capteur, par exemple), un être humain ou une organisation. Cette donnée peut traduire un état, une observation, une décision, etc. L'intérêt des événements réside dans le fait qu'ils sont émis dans un réseau et reçus par tout membre de ce réseau qui s'intéresse à ces événements (sans que la source émettrice de l'événement sache exactement qui reçoit ses événements). Cela permet une grande flexibilité dans le recueil des données. D'autre part, l'utilisation des événements est facilité par une architecture de type dirigée par les événements. Elle se charge de les collecter mais aussi de les traiter par filtrage,

agrégation, dérivation, etc. permettant ainsi de déduire de nouvelles données. Ce type d'architecture semble répondre à notre besoin d'analyser les données entrantes en vue d'aider à la décision. Les raisons de ces choix sont expliquées en détails dans le Chapitre II.

I.3.1 Les événements

I.3.1.1 Définition de l'événement

Le terme événement provient du latin *evenire*, qui signifie advenir. Dans son acception étymologique, le mot événement recouvre tout ce qui se produit, arrive ou apparaît (sens général). Au pluriel, les événements désignent l'ensemble des faits qui créent telle ou telle situation, telle conjoncture [Larousse (dictionnaire), 2013].

Dans la littérature scientifique, deux courants de pensée existent : le premier affirme que l'événement est forcément lié à l'activité, le second défend le fait que l'événement n'est pas nécessairement lié à l'activité et précise que l'activité peut ou non utiliser l'événement (cette deuxième tendance est plus proche de ce que nous allons utiliser.) [Morley et al., 2007], pour qui un événement est « quelque chose qui arrive et qui provoque le déclenchement d'une activité », s'inscrit clairement dans le premier cas. L'acteur responsable de cette activité doit être informé que l'événement s'est produit. C'est pourquoi l'événement est souvent matérialisé par de l'information. L'événement est aussi vu dans [Morley et al., 2007] du point de vue processus, comme « un stimulus qui provoque une réaction dans une activité ». La notion de stimulus traduit le fait qu'un événement est quelque chose qui arrive, sans impliquer aucun acteur de l'activité et sans consommer aucune de ses ressources. Toujours d'après [Morley et al., 2007], un événement est toujours associé à une ou plusieurs activités sur lesquelles il agit. Un même événement peut agir sur plusieurs activités, permettant alors d'indiquer les activités pouvant se dérouler en parallèle. Inversement, plusieurs activités peuvent être associées à la même activité. Dans ce cas, l'activité est dotée d'une règle de synchronisation indiquant si les événements doivent être ou non simultanés.

Cependant [Chandy and Schulte, 2009] donnent une définition plus large de l'événement, sans le lier nécessairement à l'activié. L'événement est « quelque chose qui arrive ». Ils ajoutent que la caractéristique fondamentale des événements est qu'ils ne peuvent pas être entièrement prévus. Compte tenu de ce fait, nous ne pouvons pas prédire quand un événement critique va arriver.

On peut trouver une définition proche de celle de Chandy dans le Glossaire du Management des Événements, créé par le symposium EPTS (Event Processing Technical Society) [EPTS, 2008] dans le but de poser les fondations d'une standardisation des concepts de traitement d'événements. Dans ce glossaire dont l'approche consiste à définir chaque terme indépendamment d'un produit, d'une implémentation ou d'un domaine d'application particulier, (Luckham et al., 2008) définissent un événement suivant deux dimensions :

- L'activité qui a lieu et engendre l'événement. L'événement est ainsi vu comme tout ce qui advient ou dont l'avènement est envisagé, par exemple une opération financière, un atterrissage d'avion, un capteur qui fournit des données, un changement d'état dans une base de données ou d'une machine à états finis, le séisme au large des côtes de Honshu, l'abolition de l'esclavage, la bataille de Marathon.
- La formalisation, c'est-à-dire la représentation de cette activité dans un système informatique (objet, message, tuple). L'événement est alors un objet qui représente, encode, enregistre un fait, généralement à des fins de traitement informatique. Par exemple : un ordre d'achat (enregistré comme une activité d'achat), un email de confirmation d'une réservation de vol, une fiche de réclamation client, le résultat d'une lecture d'un capteur RFID (Radio Frequency IDentification).

Les événements reçus sont manipulés par les systèmes informatiques par traitement de leurs représentations en tant qu'objets événement. La même activité peut être représentée par plus d'un objet événement, et chaque objet événement peut enregistrer différents attributs de l'activité.

I.3.1.2 Le recueil des événements

Tels que décrits dans le précédent paragraphe, les événements sont porteurs de données concernant l'état d'une partie du système ou du système de traitement. Ils représentent les données informant du statut d'une activité, les données émises par un capteur, l'émission d'un rapport par un pompier sur le terrain, la réception d'un ordre par un sous-traitant, etc. Ainsi que définis par [Luckham and Schulte, 2008], les événements peuvent être considérés comme les instances d'un type d'événement : par exemple, tous les événements émis par un certain type de capteur auront le même type, c'est-à-dire la même structure et les mêmes propriétés.

Le mécanisme d'émission et de réception des événements est basé sur le paradigme publication/abonnement (publish/subscribe). Dans un système publication/abonnement, les abonnés, aussi appelés consommateurs, peuvent exprimer leur intérêt pour un type d'événement ou un motif de types d'événements, et être notifiés de la génération de n'importe quel événement émis par les producteurs qui correspond à cet intérêt. Les événements sont propagés de façon asynchrone[3]. En effet, un service prévient tous ses abonnés par émission d'un événement qu'il vient de réaliser une opération donnée. Ensuite, c'est aux consommateurs de traiter cet événement.

Par conséquent, l'ensemble du système est chargé de faire la correspondance entre les événements émis et les abonnements, et de ne transmettre des producteurs aux consom-

3. En informatique, l'asynchronisme signifie que le programme A qui envoie des données à un programme B via la fonction a n'attend pas la réponse de B à A pour continuer à s'exécuter. Typiquement, une communication par email est asynchrone (A envoie un email à B, et n'attend pas la réponse de B pour vaquer à ses occupations) alors qu'une communication téléphonique est une communication synchrone (A appelle B et doit attendre que B ait fini de lui parler pour raccrocher et passer à une autre activité).

mateurs que les événements pertinents. L'intérêt d'un tel système est la possibilité de disséminer une grande quantité d'information à un large ensemble de consommateurs. Chaque consommateur reçoit alors uniquement l'information qui correspond à son abonnement : ceci évite de surcharger les consommateurs en information inutile (« flood ») et de rendre cette dernière inexploitable (au mieux).

Une de nos hypothèses de travail est que l'architecture des SI des partenaires de la collaboration est orientée services (SOA). L'architecture SOA est une forme d'architecture de médiation qui repose sur les interactions entre des composants logiciels appelés services. Un service est une fonctionnalité définie d'un SI qui est divisée en opérations. Les Services Web présentent l'avantage supplémentaire d'être accessible à travers un réseau (Internet, intranet, etc.) via des interfaces permettant l'appel des opérations sans accéder à leur mécanique proprement dite. Les services communiquent entre eux via des messages.

Notre nouvelle hypothèse est que les web services des partenaires de la collaboration sont capable d'émettre et de recevoir des événements. A cette fin, ils proposent des fonctionnalités d'émission et d'abonnement aux événements. Pour cela, ils implémentent les recommandations de la norme WS-Notification pour gérer les mécanismes publication/abonnement [OASIS, 2006b, OASIS, 2006c, OASIS, 2006a]. Ainsi, un web service peut envoyer des événements à propos de son état (invoqué, en cours d'exécution, terminé, interrompu) en sus des habituels échanges de messages (par exemple, un relevé de mesure en réponse à une requête émise par le système de surveillance de la centrale) avec les autres web services. Les capteurs et autres appareils situés sur le terrain envoient également des événements concernant les mesures qu'ils observent : ces événements permettent de constituer une photographie de la situation collaborative et de suivre son évolution. De façon à collecter tous les événements concernant la situation collaborative, l'abonnement portera sur tous les types d'événements provenant tant de l'exécution des workflows collaboratifs que du terrain de la collaboration.

I.3.1.3 Le traitement des événements complexes

Une fois les événements collectés, il faut les traiter pour en extraire l'information pertinente (au regard des objectifs de la collaboration). Pour cela, un traitement informatique va réaliser des opérations sur les événements, incluant la lecture, la création, la transformation, la suppression et l'abstraction des données portées par ces événements. C'est ce que l'on appelle le CEP (Complex Event Processing). Le moteur de CEP va « écouter » les événements entrants et les passer au crible d'un ensemble de règles métier (contenant des motifs d'événements [4]) définies par les acteurs de la situation collabo-

4. Un motif d'événement permet la détection de séquences d'événements liés par des relations temporelles, de causalité, etc.. Lorsqu'une séquence d'événements correspond à un motif, elle constitue une instance de ce motif. Exemple (très simple) de motif : l'événement B suit l'événement A dans un laps de temps de 20 secondes.

rative. Si des événements entrants réalisent une instance de motif d'événements, alors les règles liées à ce motif sont exécutées et permettent au moteur de CEP de générer de nouveaux événements. Ces nouveaux événements constituent donc de nouvelles données, qui peuvent enrichir la connaissance concernant la situation collaborative et son environnement.

Un exemple de règle métier très simple peut être :

```
SI (vitesse du vent > 50 km/h) et (taux de radiation augmente de 5%)
durant les 5 dernières minutes)
ALORS GénérerEvénement(Alerte())
```

Un moteur de CEP peut, à travers les règles métier, réaliser plusieurs types de filtrage et d'agrégation des données : ceci sera détaillé dans les chapitres suivants.

De façon à collecter tous les événements concernant la situation collaborative et à pouvoir en déduire de nouveaux événements (i.e. de nouvelles données), le moteur de CEP s'abonne à tous les types d'événements provenant tant de l'exécution des workflows collaboratifs que du terrain.

I.3.2 Les modèles de situation collaborative

I.3.2.1 Dépendance entre modèles de situation et processus collaboratifs

La situation collaborative est représentée par un modèle, qui est construit à partir de la connaissance recueillie sur la situation collaborative du méta-modèle [Macé-Ramète et al., 2012]. C'est sur la base de ce modèle qui représente les partenaires, leurs compétences, le terrain de la situation collaboration (environnement, biens, risques, etc.) que les processus collaboratifs sont conçus. Ces processus sont ensuite transformés en workflows collaboratifs pour être exécutés (cf. section I.5.2). Il y a donc un lien de cause à effet du modèle de la crise vers les workflows collaboratifs. Si la caractérisation de la situation collaborative à l'instant t ne correspond plus à la caractérisation de la situation collaborative à t_0 alors les processus risquent de se baser sur des éléments (partenaires, activités, ressources) ne correspondant plus à la réalité et/ou de ne plus répondre aux objectifs de la collaboration.

Il est donc nécessaire de pouvoir d'une part suivre l'exécution des processus collaboratifs définis à t_0 tout en ayant d'autre part une photographie du théâtre de la collaboration, et ce à chaque instant.

I.3.2.2 Détecter la divergence

Notre but est de superviser la situation : il s'agit là de vérifier si l'exécution des processus collaboratifs nous fournit les résultats escomptés. Il faut donc comparer ce que l'on attend avec le réel. L'analogie suivante permet de mieux comprendre le concept de divergence (qui peut conduire à une adaptation) : l'élève A doit rédiger un rapport de 20 pages en cinq jours. Il prévoit donc d'écrire 4 pages par jour pour équilibrer sa charge de travail. A J+2 jours, il prévoit donc d'avoir écrit 8 pages (situation attendue). Or, il s'avère qu'il n'en a écrit que 2 (situation réelle) au lieu de 8 (détection de la divergence).

S'il veut respecter le délai imposé, il va devoir modifier son comportement et accélérer son rythme d'écriture (adaptation).

Nous pouvons utiliser les événements générés par le CEP pour détecter les changements qui se produisent dans le modèle de la situation collaborative, de la façon suivante :

- Le modèle de la situation collaborative à t_0 (appelé modèle initial) est dupliqué,
- Les deux copies du modèle initial sont mises à jour de manière continue grâce aux événements, générant ainsi deux modèles de la situation collaborative :
  1. **Le modèle attendu** : ce modèle correspond au modèle de la situation collaborative tel qu'on l'attend à un instant t, i.e. le modèle de la situation que l'on s'attend à obtenir après exécution de tout ou partie des workflows collaboratifs (suivant l'instant t considéré). Ce modèle est obtenu en mettant à jour le modèle initial en utilisant les événements témoignant du suivi de l'exécution des workflows (état des activités, rapport d'exécution des services),
  2. **Le modèle terrain** : ce modèle représente la situation collaborative réelle à un instant t, quels que soient les workflows collaboratifs exécutés, i.e. la situation telle qu'elle est réellement sur le terrain à cet instant t. Ce modèle est obtenu en mettant à jour le modèle initial via les événements dont l'origine est le théâtre de la collaboratif (mesures, comptes-rendus, etc.).
- A un instant t, la mesure de la divergence δ entre le modèle attendu et le modèle terrain est effectuée.

La détection consiste en cette mesure du δ entre le modèle terrain et le modèle attendu afin de diagnostiquer une divergence significative.

I.3.2.3 Définir le besoin d'adaptation

L'adaptation des workflows collaboratifs se décide en étudiant le δ calculé dans l'étape de détection. La mesure du δ étant réalisée sur l'ensemble des points des modèles précédemment présentés, l'amplitude, la nature et l'origine de la divergence peuvent être déterminés. Dans un premier temps, l'adaptation n'est proposée que si la divergence δ est au moins égale à un seuil δ_{seuil}(déterminé en amont). Si tel est le cas, l'analyse fine de la nature de la divergence détectée va permettre de suggérer le meilleur niveau d'adaptation possible.

Il est à noter que nos travaux s'appuient sur les résultats de thèses et projets de recherche antérieurs. Ces résultats nous fournissent les outils nécessaires à l'obtention du modèle de la situation collaborative ainsi que le SIM (outils de conception et déploiement). Soulignons également que l'approche retenue dans ces travaux s'inspire largement du principe de la conception retardée [Dellen et al., 1997]. Dans notre contexte, il s'agit principalement de pouvoir interrompre à tout moment l'orchestration du ou des workflows collaboratifs afin de pouvoir invoquer (toujours dans la phase d'exécution des processus) les outils de conception des dits-processus. Ceci afin de redéfinir le ou les processus concernés de la manière la plus adéquate. Cette option permet ainsi de revenir au niveau d'abstraction le plus pertinent par rapport à l'adaptation souhaitée, à savoir :

1. Caractériser la situation collaborative : les évolutions de la collaboration et/ou de son environnement sont importants, il faut redéfinir les processus collaboratifs en partant de la caractérisation actuelle à l'instant t. Puis, compte tenu de la caractérisation de la situation collaborative, quels sont les processus permettant d'atteindre les objectifs collectifs ?
2. Concevoir le SIM : les partenaires et/ou leurs ressources et activités associées ont subi des modifications (arrivée, départ, (in)disponibilité, etc.), quel est l'impact sur les processus et le passage des processus aux workflows ?
3. Déployer le SIM : exécution ou re-exécution de tout ou partie des workflows.

I.4 La crise, une situation collaborative particulière

Comme défini par [Devlin, 2006], une situation de crise peut être politique, militaire, économique, humanitaire, sociale, technologique, environnementale ou sanitaire. Quelque soit sa nature, une crise est une situation anormale, qui est le résultat d'une instabilité qui impacte une sous-partie du monde (appelé écosystème ou système) avec des conséquences inacceptables [Devlin, 2006] [Lagadec, 1992].

D'après [Alexander, 2002, Altay and Green III, 2006, Beamon and Kotleba, 2006], le cycle de vie de la gestion de crise est composé de quatre phases principales (cf. Figure I.2) :

FIGURE I.2 – Les quatre étapes du cycle de vie de la gestion de crise

1. La phase de prévention : antérieure à la survenue de la crise, elle a pour objectif la diminution de la probabilité d'apparition des risques liés à la crise et leurs conséquences s'ils surviennent. C'est cette phase qui peut conduire, par exemple, à la construction de bâtiments pouvant supporter des séismes sans s'effondrer comme c'est le cas au Japon. Pour prendre un exemple encore plus simple, cela correspond au passage du permis de conduire : on forme les gens à bien conduire, à respecter le code de la route pour éviter les accidents et on valide leurs capacités et acquis avant de les laisser conduire un véhicule seuls.

2. La phase de préparation : également antérieure à la survenue de la crise, elle vise à établir de nouveaux processus de réponse adaptés aux crises à venir (planification, plans d'intervention, etc.). Les exercices d'entraînement comme une évacuation en cas d'alerte incendie en sont un exemple. Pour rester dans le domaine de la conduite, la phase de préparation rassemble tous les équipements de sécurité qui permettent de protéger les occupants du véhicule en cas d'accident : les ceintures de sécurité, les airbags, les carrosseries qui absorbent les chocs, la géolocalisation automatique du véhicule accidenté et l'appel aux secours, etc.
3. La phase de réponse : la crise a lieu, l'objectif de cette phase est de mettre en place un ensemble d'actions qui agira sur le système impacté par la crise pour qu'il revienne le plus tôt possible dans son état nominal. C'est dans cette phase que les plans de secours et autres opérations définis dans la phase de préparation vont être déclenchés. Dans notre exemple, quand l'accident de voiture se produit, tous les éléments définis pendant la phase de préparation se déclenchent : la cartouche libère le coussin de l'airbag, la carrosserie se déforme de façon à absorber au mieux le choc, le véhicule envoie un signal géolocalisé aux services de secours, etc. Dans la suite de ce manuscrit, nous nous intéresserons essentiellement à la phase de réponse.
4. La phase de rétablissement : il s'agit là de faire en sorte que le système impacté revienne à un état stable à long terme. Un exemple d'action menée pendant cette phase est la réhabilitation du réseau routier et la reconstruction d'habitations. Toujours dans le cadre de notre exemple, il s'agit d'apporter des soins aux blessés, de réparer le véhicule, de remettre en état les rails de sécurité, etc.

Si nous nous attardons sur la phase de réponse, nous pouvons logiquement constater que l'efficience de cette phase est déterminée par la vitesse et la pertinence avec lesquelles chaque information peut être gérée et échangée entre les partenaires de la cellule de crise, à savoir les organisations, les personnes physiques, mais également les appareils (physiques ou logiciels). Il est donc nécessaire de pouvoir avoir accès à la bonne information, au bon moment.

Un autre point à prendre en compte lors de la phase de réponse à la crise est le caractère évolutif d'une situation de crise : la crise évolue d'elle-même, de par sa nature, son environnement et/ou de par les actions menées en vue de résoudre la crise. A cause de cela, le système de réponse à la crise doit suivre les évolutions de la crise afin de rester conforme aux attentes des acteurs de la cellule de crise (à savoir répondre à la crise). Ceci implique de mesurer l'efficience de la réponse à la crise, de façon à prendre en compte les changements affectant la crise elle-même ou la cellule de crise.

Ces différents éléments nous montrent à quel point la situation de crise est un excellent exemple du besoin d'agilité des processus au sein d'une situation collaborative, en exacerbant les aspects généraux d'une collaboration : il faut pouvoir détecter les changements au sein de la situation collaborative (tant du côté du théâtre de la crise, que des

objectifs et des acteurs de la cellule de crise), vérifier la bonne adéquation des processus collaboratifs vis-à-vis des objectifs compte tenu des évolutions détectées et si nécessaire, y remédier en proposant d'adapter les processus collaboratifs, le tout dans un laps de temps le plus court possible, compte tenu de l'urgence induite par la crise.

I.4.1 Présentation des projets PLAY et SocEDA

Les présents travaux de thèse ont été menés dans le cadre de deux projets de recherche : un projet européen (PLAY) et un projet français (SocEDA).

I.4.1.1 Le projet PLAY

Le projet européen PLAY (Pushing dynamic and ubiquitous interaction between services Leveraged in the Future Internet by ApplYing complex event processing) est réalisé en partenariat avec FZI (Allemagne), ICCS (Grèce), CIM Group (Serbie), Orange (France), Linagora (France), et INRIA (France). L'objectif principal du projet PLAY est de définir et fournir une plateforme ouverte et hautement distribuée permettant les interactions dirigées par les événements entre les services. Le projet se déroule d'octobre 2010 à octobre 2013 (ICT Call 5, topic ICT- 2009.1.2, EU/FP7).

Cette plateforme est basée sur une architecture fédérée qui pourra répondre à la problématique de la multiplicité et de l'hétérogénéité des réseaux de service, ainsi qu'aux exigences de qualité de service (telles que la fiabilité, la sécurité et la montée en charge) à l'échelle de très grands réseaux basés sur Internet. En quelques mots, la plateforme sera capable de combiner les événements provenant de n'importe quel service pour déduire de nouvelles informations et de les transmettre (sous forme d'événements) à n'importe quel service à très grande échelle (Figure I.3). Les principaux résultats de ce projet de recherche sont :

- Une couche intergicielle fédérée : un réseau pair-à-pair en surcouche combiné à un mécanisme publication/abonnement, qui a pour mission de collecter les événements provenant de services hétérogènes et distribués,
- Un moteur de gestion d'événements complexes distribué : ce moteur est élastique, basé sur l'informatique en nuage (cloud computing) distribuée pour transformer/-combiner des événements provenant de différents services, de manière à détecter des situations intéressantes face auxquelles le système devrait réagir,
- Un ensemble de mécanismes permettant de modéliser et de reconnaitre des événements en fonction de connaissances concernant les liens sociaux dans un but de filtrage des événements ainsi qu'un moteur d'orchestration piloté par les événements et susceptible d'adapter les processus en cours d'exécution en fonction du contexte événementiel,
- Un composant d'adaptation métier sensible à la situation courante : un moteur de recommandation d'adaptation et de modification de la gestion des services et des processus métier en cours d'exécution, de manière non pré-définie, en veillant à maintenir la cohérence du système,

FIGURE I.3 – Les principaux apports de la plateforme issue du projet européen PLAY [PLAY, 2010]

- Deux cas d'études générant et utilisant un grand nombre de données, permettant de piloter et valider les résultats.

Le premier cas d'étude (Nuclear Crisis) [Barthe et al., 2011b] concerne le domaine de la gestion de crise (le scenario développé est celui d'un accident nucléaire), qui souligne l'intérêt du recours aux événements comme source d'information sur la situation courante dans le cas d'une situation de crise complexe et la nécessité de rendre les processus collaboratifs agiles afin d'atteindre les objectifs de la collaboration quelques soient les évolutions du système impacté et de la cellule de traitement. Le second cas d'étude (LiveBoxPro) est lié au développement d'un nouveau service permettant à la fois de diagnostiquer les pannes de connexion Internet établies via une LiveBoxPro (exemples : déterminer l'origine de la panne —DSLAM [5] ou BAS (Broadband Access Server)—) et de gérer les connexions établies à partie de la LiveBoxPro. Ceci doit permettre l'amélioration des services LiveBoxPro et d'augmenter la satisfaction client.

I.4.1.2 Le projet SocEDA

Le projet ANR SocEDA (Social-Event Driven Architecture) fait appel aux partenaires français suivants : ActiveEon, I3S, LIG, Mines Albi, INRIA Adam, LIRIS, Orange, Linagora et Thalès-Communications. Le projet SocEDA concerne l'exploitation d'outils

5. Digital Subscriber Line Access Multiplexer

de gestion des événements complexes afin de définir un environnement capable de réagir à la réception d'événements au sein des systèmes membres de cet environnement. Il se déroule de novembre 2010 à novembre 2013 (ANR/2010/ARPEGE).

Le but du projet SocEDA est de développer et de valider une plate-forme flexible et fiable basée sur une architecture orientée service fédérée destinée aux contextes événementiels largement distribués. Les événements traités proviennent d'environnements où les services sont largement distribuées et hétérogènes. Une telle architecture permet l'échange d'information contextuelle entre services hétérogènes, rendant possible l'optimisation et la personnalisation de leur exécution en se basant sur les liens sociaux connus. Dans la plate-forme de services, résultat du projet, les échanges sont pilotés par les événements. Cette plate-forme est composée des éléments suivants :

- Une couche intergicielle fédéré : un réseau en surcouche type pair-à-pair combiné avec un mécanisme d'abonnement de type publication/abonnement, permettant de recevoir les événements issus des services hétérogènes distribués,
- Un moteur distribué de traitement d'événements complexes : un moteur flexible et distribué pour le traitement d'événements complexes provenant de services et permettant de détecter des situations sur lesquelles d'autres services doivent réagir,
- Un composant d'agilité métier sensible à la situation courante : un moteur de recommandation d'adaptation et de modification de la gestion des services et des processus métier en cours d'exécution, de manière non pré-définie, en veillant à maintenir la cohérence du système,
- Un composant de vérification de la confiance accordée aux données recueillies par la plateforme,
- Un ensemble d'outils de supervision et de gouvernance ainsi qu'un environnement de mashup afin de décrire et de gérer les services ainsi que les événements métier,
- Le système est validé grâce à deux cas d'utilisation, l'un dans le domaine de la gestion de crises et l'autre dans le domaine de la gestion des flux aériens.

I.4.1.3 Liens avec les travaux de thèse

Ces travaux de thèse ont été fortement liés aux projets PLAY et SocEDA :

- D'une part, dans la fourniture des cas d'études liés au domaine de la gestion de crise, et plus particulièrement le cas d'un accident nucléaire. Un cas d'étude général inspiré des cas d'études des projets et de la récente actualité servira de fil rouge dans la suite de ce manuscrit.
- D'autre part, dans la conception et la réalisation d'un service d'agilité des processus collaboratifs (projet SocEDA), qui répond à la problématique exposée dans ce manuscrit. Il est à noter que le projet PLAY propose une autre variante du service d'agilité, qui s'appuie sur des hypothèses différentes des nôtres. Cette autre vision du service d'agilité sera abordée dans le chapitre II.

I.4.2 Un cas d'étude d'actualité

La situation de crise est, comme nous avons pu le voir précédemment, une situation collaborative particulière faisant appel à de nombreux acteurs hétérogènes, qui doivent oeuvrer ensemble dans un environnement en perpétuel mouvement, dans des laps de temps très courts. Un cas d'étude extrait d'une situation de crise permet d'éprouver les prototypes issus des projets PLAY et SocEDA, tant au niveau du filtrage et de la scalabilité (quantité et qualité des informations à échanger, rapidité des transmissions), que de l'agilité du système pour faire face à cet environnement mouvant.

I.4.2.1 Fukushima, le deuxième accident nucléaire le plus grave de l'Histoire

Plus que jamais, les situations de crises sont à la une de l'actualité : crise économique, guerres, pandémies, catastrophes naturelles, catastrophes technologiques sont devenues des titres habituels. L'un des exemples les plus frappants de ces dernières années est la situation de crise déclenchée par le séisme de la côte Pacifique du Tōhoku et qui a rapidement évolué en situation d'accident nucléaire, évalué à 7 sur 7 (i.e. accident majeur) sur l'échelle INES (International Nuclear Event Scale) [AFP, 2011].

Le vendredi 11 mars 2011, à 14h46 heure locale, un séisme de magnitude 8,9 sur l'échelle de Richter survient au large des côtes du nord-est de l'île de Honshu [Baumont, 2011]. Il s'agit du séisme le plus violent jamais enregistré au Japon, dont les secousses ont été ressenties jusqu'à Beijing. Cinquante et une minutes après la première secousse, la première vague du tsunami provoqué par le séisme, haute de 15 mètres, atteint la centrale nucléaire de Fukushima Daiichi, exploitée par TEPCO (Tokyo Electric Power Company) [Alberganti, 2011].

Si les systèmes de sécurité de la centrale se sont déclenchés lors de la secousse et ont entraîné un arrêt automatique des réacteurs en service, ils se sont retrouvés littéralement submergés par le tsunami (Figure I.4), provoquant ainsi la perte des groupes électrogènes alimentant les systèmes de refroidissement des réacteurs 1 et 2 de la centrale nucléaire de Fukushima Daiichi, ainsi que les piscines de désactivation des réacteurs 1 à 4 dans lequel le combustible usé était entreposé. Les événements ne s'arrêtent malheureusement pas là : s'ensuivront sept explosions dans les bâtiments 1, 2, 3 et 4. Puis la fusion des réacteurs 1, 2 et 3 sera avérée et peu de temps après la perte du confinement des bâtiments 1, 2 et 3 est devenue une certitude [Kokkai, 2012].

Le 17 mars 2011, le monde entier découvre les images de deux hélicoptères s'approchant et larguant de l'eau sur les réacteurs 3 et 4. Le but de la manoeuvre, à savoir refroidir le réacteur 3 et remplir les piscines contenant les combustibles usés du réacteur 4, ne sera pas atteint. Seul un largage atteint correctement le réacteur 3. Le taux de radiation à proximité des réacteurs, ainsi que le bouleversement de la topographie des lieux empêchent les pilotes de viser correctement leur cible. Plus tard dans la journée, des camions-citernes sont dépêchés pour arroser les réacteurs. Mais l'opération reste vouée à l'échec, le vent dispersant la majeure partie de l'eau déversée.

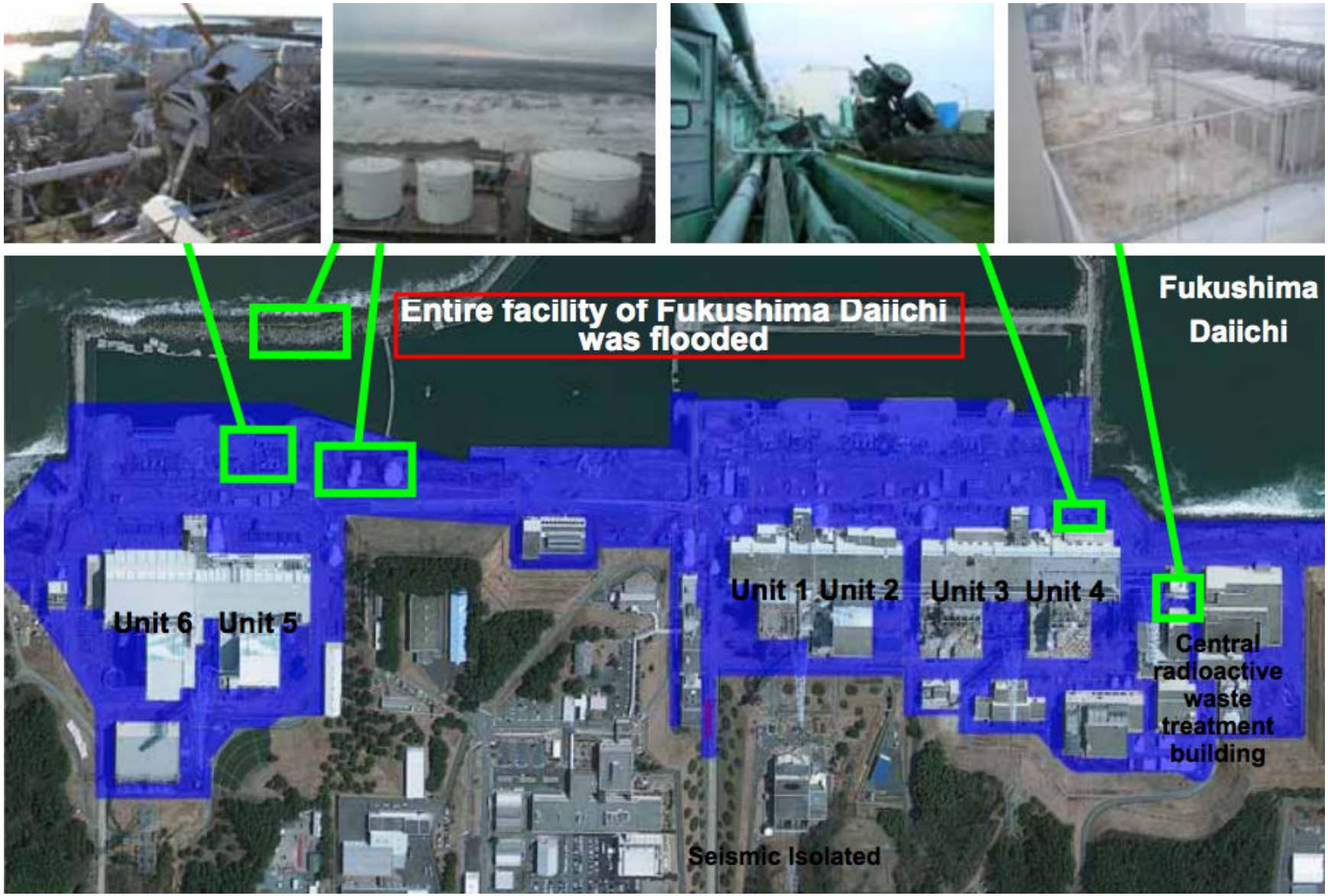


FIGURE I.4 – Vue aérienne du site de Fukushima Daiichi et représentation des parties submergées par le tsunami [Tokyo Electric Power Company, 2012]

I.4.2.2 Le cas d'étude

Les cas d'étude des projets PLAY et SocEDA (la gestion d'une crise ayant pour origine un accident nucléaire) furent conçus quelques semaines avant le dramatique accident de Fukushima : pour le moment, aucun accident d'une telle gravité n'est à déplorer sur le sol français.

En France, le dernier événement le plus grave (au sens de l'échelle INES) fut un accident de niveau 4 (i.e. un accident n'entraînant pas de risques hors site) [IRSN, 2013] : en février 1980, dans la centrale de Saint-Laurent-des-Eaux (Loir-et-Cher), une défaillance technique a conduit à l'inflammation locale du combustible. L'accident a gravement endommagé l'installation. Le cas d'étude se voulant le plus réaliste possible, il a été conçu en s'inspirant de données réelles provenant tant des incidents et accidents français passés (Saint-Laurent-des-Eaux, Cadarache, Cruas entre autres) que des plans d'urgence définis par les autorités.

Ceci étant, l'accident de Fukushima nous montre à quel point il est impossible de prévoir les évolutions d'une situation de crise et par la même la nécessité de pouvoir rendre les processus de réponse agiles, ainsi que la difficulté de collecter, mettre en regard et analyser toutes les informations disponibles afin de prendre la meilleure décision possible à un instant t.

Le cas d'étude présenté ici est une synthèse des cas d'étude créés et implémentés dans les projets de recherche PLAY et SocEDA.

L'accident a lieu dans un REP (Réacteur à Eau Pressurisée). Le principe du fonctionnement d'un tel réacteur est schématisé par la Figure I.5 (d'après [Kuntoff, 2005]) :

1. Les barres de commande contenant le combustible sont immergées dans de l'eau (circuit primaire). Dans le circuit primaire, l'eau joue le rôle de fluide caloporteur qui circule au sein des assemblages entre les crayons où se produit la réaction en chaîne. Le contrôle de la réaction est assuré par la concentration de bore dans l'eau du circuit primaire et par la hauteur des barres de commande insérées dans les assemblages combustibles,
2. Les échanges de chaleur entre le circuit primaire et le circuit secondaire causent un dégagement massif de vapeur d'eau dans les générateurs de vapeur du circuit secondaire,
3. La vapeur d'eau est acheminée vers une turbine. La vapeur d'eau sous haute pression est détendue dans la turbine qui entraîne un alternateur qui produit l'électricité,
4. La vapeur d'eau contenue dans le circuit secondaire est ensuite piégée dans un condenseur qui permet le passage de l'état de vapeur à l'état de liquide,
5. La chaleur de l'eau du circuit secondaire est transférée à un circuit tertiaire, constitué principalement par une tour de réfrigération, dans laquelle l'eau est répartie en fines gouttelettes, ce qui permet un bon échange entre l'eau et l'air

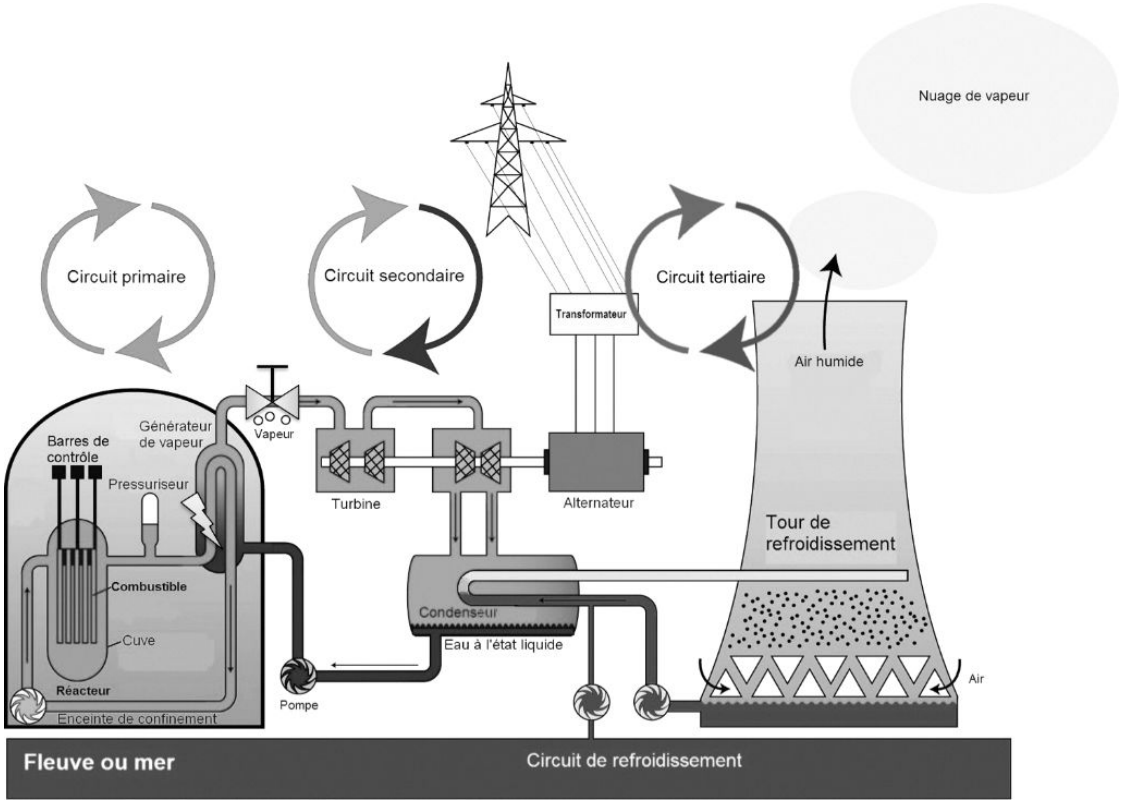


FIGURE I.5 – Principes généraux du fonctionnement d'un réacteur à eau pressurisée, et les incidents conduisant à la situation de crise

et donc ramène l'eau à une température voisine de celle de l'air ambiant. Une partie de cette eau s'évapore (ce sont ces fameux nuages de vapeur que l'on voit flotter au-dessus des tours des centrales), le reste est pompé dans le bassin situé en dessous de la tour et retourne refroidir le condenseur. L'eau évaporée est remplacée par de l'eau venant d'un fleuve, d'une rivière ou de la mer. L'eau qui sert au refroidissement du réacteur est pompée en amont de la tour de réfrigération.

Afin de faciliter les échanges de chaleur entre les deux circuits, la paroi les isolant est très fine. Cette finesse accélère le phénomène normale d'usure par l'érosion : une fuite se produit dans la partie isolant le circuit primaire du circuit secondaire dans le générateur de vapeur. De l'eau contaminée du circuit primaire se retrouve mêlée à celle du circuit secondaire, contaminant ainsi le circuit secondaire. Cette fuite a également pour conséquence l'augmentation de la pression dans le circuit secondaire. Pour éviter une explosion due à une surpression, la vanne de sécurité du circuit secondaire s'ouvre automatiquement. Les opérateurs tentent de refermer cette vanne en passant en commande manuelle. Malheureusement, la vanne reste bloquée en position ouverte, et la vapeur d'eau contaminée du circuit secondaire s'échappe dans l'atmosphère.

La description et la chronologie complète des événements est disponible en Annexe A.

I.4.2.3 La caractérisation de la crise

Le modèle de la situation de crise présentée ici a été réalisé en suivant le méta-modèle de la situation collaborative (Figure I.6) [Macé-Ramète et al., 2012] au niveau dédié à la crise (qu'elle soit civile, militaire, écologique, routière, etc.). Ce méta-modèle se compose d'un coeur et de couches thématiques englobant le coeur.

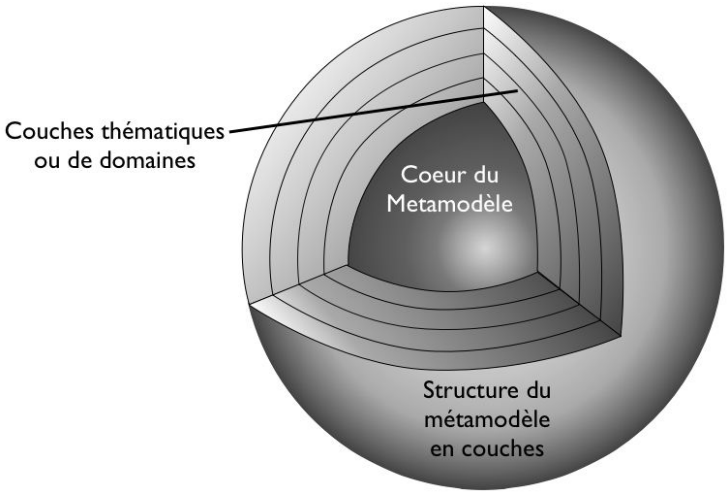


FIGURE I.6 – Structure du méta-modèle de la situation collaborative [Bénaben, 2012]

Le coeur représente les principes généraux relatifs à la collaboration d'organisations (objectif, partenaire, réseau, service, performance) qui sont communs à toutes les situations collaboratives. La Figure I.7 [Macé-Ramète et al., 2012] montre une version simplifiée du noyau qui caractérise les situations collaboratives en général, en faisant abstraction du domaine (crise nucléaire, crise routière, collaboration industrielle, etc.) .

On peut remarquer que plusieurs familles de classes gravitent autour du concept central qu'est l'objectif de la collaboration :

- L'objectif poursuivi par la collaboration concerne (éléments en gris clair) une opportunité ou une menace (inondation) soulevée par une caractéristique de l'environnement (tsunami) qui survient dans un environnement donné (littoral du Honshu) et qui impacte une composante de l'environnement (centrale nucléaire de Fukushima Daiichi).
- Le réseau collaboratif (gris foncé) composé des partenaires, de leurs compétences et du médiateur. Chaque acteur impliqué dans la collaboration partage ses données, ses fonctions et ses processus, i.e. ses compétences, pour atteindre l'objectif. Chaque compétence requiert des ressources pour pouvoir exécuter des instructions. La compétence créé des flux entrants et sortants. Les compétences peuvent être agrégées pour former une procédure.
- Chaque compétence répond à un objectif et est associée à un objectif de performance (élément blanc). Ce dernier est caractérisé par un indicateur de performance (KPI, Key Performance Indicator) qui évalue la compétence.

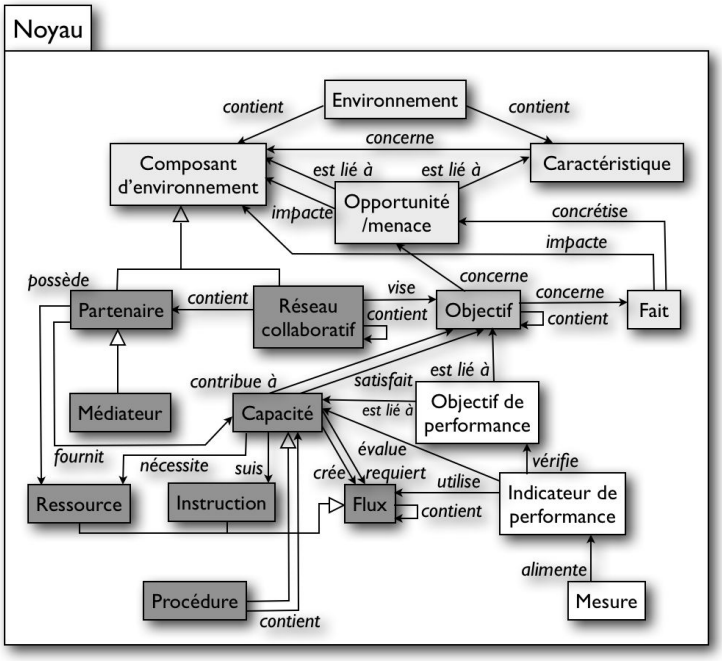


FIGURE I.7 – Structure du méta-modèle de la situation collaborative : le coeur [Bénaben, 2012] d'après [Macé-Ramète et al., 2012]

Une fois que les compétences sont définies, un processus collaboratif est mis en place pour les invoquer en vue d'atteindre l'objectif global. Ce processus est composé d'activités et est exécuté et orchestré par un médiateur (qui constitue ainsi un acteur particulier de la collaboration).

Les couches thématiques quant à elles, sont spécifiques à un domaine particulier et permettent de définir les principes relatifs à ce domaine. Ces principes spécifiques héritent de ceux (généraux) issus du coeur du méta-modèle. La couche qui nous intéresse ici est la couche spécifique au domaine de la gestion de crise (Figure I.8) [Macé-Ramète et al., 2012] .

Cette caractérisation se fait d'une part grâce aux informations collectées au préalable (dans la phase de préparation) sur les compétences matérielles et fonctionnelles des acteurs de la cellule de crise, et d'autre part grâce aux informations permettant de décrire la situation de crise elle-même (phase de réponse) : environnement, société civile, biens, risques, conséquences, facteurs aggravants, etc.

I.4.2.4 Le processus collaboratif

Les processus collaboratifs sont utilisés pour orchestrer l'enchaînement des activités des différents acteurs de la cellule de crise dans le cadre de la réponse à la crise. L'information portant sur la crise, son environnement, la cellule de crises, ses acteurs, ses activités est gérée à travers ces processus. Elle créé un lien symbiotique entre d'une part les activités et d'autre part les processus eux-mêmes.

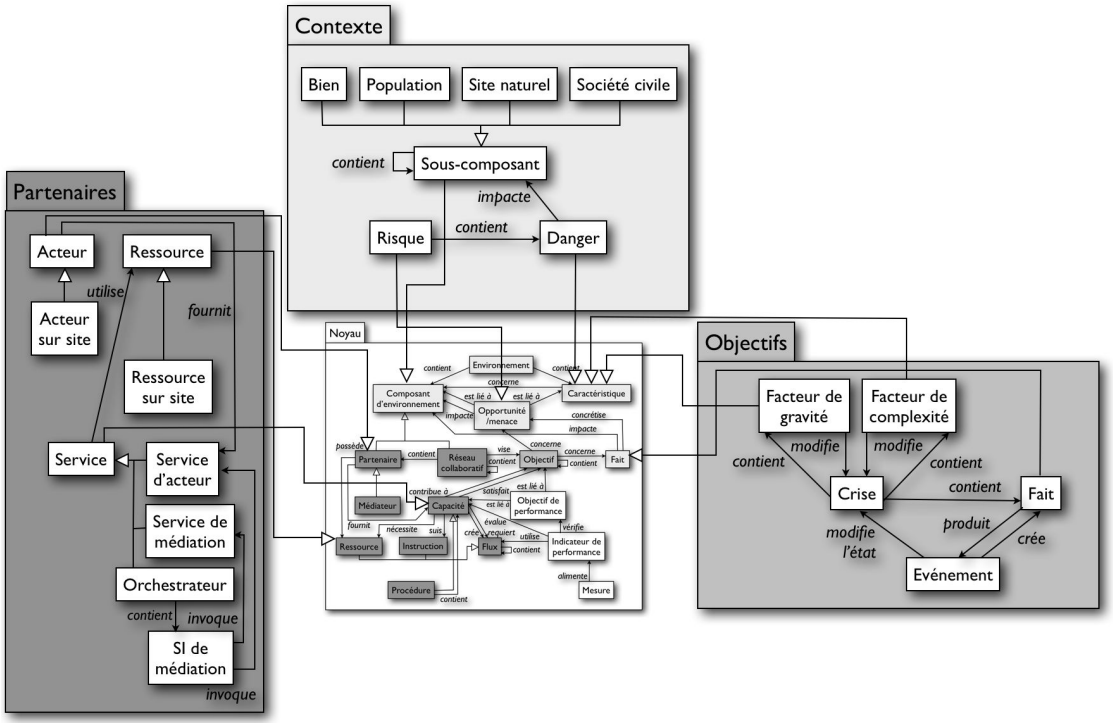


FIGURE I.8 – Structure du meta modèle de la situation collaborative, couche crise [Macé-Ramète et al., 2012]

Il est important de noter que dans le cadre de ces travaux, toute latitude est laissée aux acteurs de la crise en ce qui concerne la gestion interne et la maîtrise de leurs activités et procédures métier. Le détail d'une activité sous la responsabilité d'un acteur n'est donc pas exprimé dans le processus collaboratif, qui s'attache à orchestrer les relations, l'ordre et les entrées-sorties de chacune des activités concernées par le dit-processus.

Les processus collaboratifs sont classés sur trois niveaux d'après la norme ISO 9001-2000 [ISO, 2000] :

- Le niveau décisionnel. Les processus décisionnels sont en charge du choix de la stratégie et du pilotage des actions de l'organisation. Exemples : initier la cellule de crise, protéger la population, soutient scientifique, gérer la situation, s'abonner dynamiquement aux événements, revenir à l'état initial et clôturer la gestion de crise,
- Le niveau opérationnel. Les processus opérationnels contribuent directement à la réalisation du service ou du produit . Exemples : alerter et déployer les moyens, mettre en place le plan de circulation, mettre en place les mesures de sécurité, assister les victimes, évacuer la population, confiner la population, distribuer l'iode, communiquer (media), se retirer du terrain,
- Le niveau support. Les processus de support contribuent au bon déroulement des autres processus en leur fournissant les ressources nécessaires. Exemples : évaluer

la situation, fournir et gérer l'information (état des ressources, rapports, mesures), gérer les ressources.

Ces processus sont représentés suivant le formalisme BPMN (Business Process Model Notation) [OMG, 2011]. L'intégralité des processus BPMN du cas d'étude est disponible en Annexe B. Un exemple d'un tel processus est présenté dans la Figure I.9. Il décrit l'enchaînement des activités menés par la DIR (Direction Interdépartementale des Routes) en vue d'implémenter un plan de circulation spécifique pour évacuer les populations vivant autour de la centrale nucléaire. La DIR reçoit la décision de l'IRSN d'évacuer

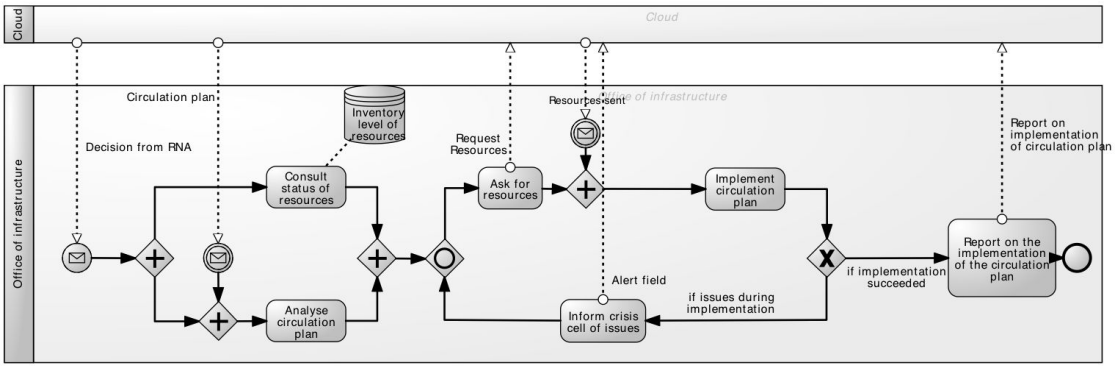


FIGURE I.9 – Exemple d'un processus collaboratif du cas d'étude : implémentation du plan de circulation

la population (et donc de mettre en place un plan de circulation). La DIR va donc en parallèle consulter l'état de ses ressources humaines et matérielles (camions, panneaux, etc.) et analyser le plan de circulation qui lui est transmis. Ensuite, sur la base de son analyse du plan et de son inventaire, elle va demander les ressources nécessaires pour déployer le plan de circulation. Une fois les ressources demandées confirmées, la DIR déploie effectivement le plan de circulation. En cas de problème durant la phase de déploiement, la DIR en informe la cellule de crise. Une fois que le plan de circulation a été implémenté avec succès, la DIR prépare un rapport et l'envoie au SIM.

I.4.2.5 Les services disponibles

Les processus collaboratifs déterminés dans ce cas d'étude font appel aux activités proposées par les partenaires de la situation collaborative. Le manque de maturité des SI des acteurs de la situation de crise, ainsi que la difficulté d'accès aux SI de ces acteurs (principalement pour des raisons de sécurité et de confidentialité) nous ont incité à développer nos propres interfaces utilisateurs et nos propres Services Web qui simulent les SI des partenaires. Dans un cadre industriel, la maturité des SI poserait a priori moins de problèmes avec l'utilisation désormais entrée dans les moeurs des logiciels de type ERP (Enterprise Resource Planning), des intranets, des extranets, etc. qui exposent des interfaces d'échanges avec le monde extérieur.

Un problème similaire s'est posé pour les capteurs. Pour des raisons évidentes, il n'a pas été possible de mettre en oeuvre un cas réel de crise nucléaire. De ce fait, nous n'avons pas pu nous intégrer dans une situation collaborative réelle de type gestion de crise nucléaire. Cependant, l'expérience passée d'un exercice NRBC nous a permis d'observer l'enchaînement des activités des partenaires de la cellule de crise et d'en tirer des données permettant de simuler les événements émis, avec des grandeurs (quantitatives et temporelles) réalistes. Des web services ont été développés afin d'émettre ces événements, auxquels le CEP s'abonne. L'ensemble de ces web services intègrent la norme WS-Notification, leur permettant d'émettre et recevoir des événements.

I.5 Positionnement dans l'écosystème de travail

En 2004, le projet MISE (Mediation Information System Engineering) [Bénaben, 2012] a été lancé par le département de Génie Industriel des Mines Albi-Carmaux. Le but du projet MISE est de fournir une architecture de médiation de systèmes d'information, ou SIM, dans le but d'être un support à la collaboration entre acteurs hétérogènes. Les présents travaux de thèse s'inscrivent dans la deuxième itération du projet.

I.5.1 Le projet MISE 1.0

La démarche MISE est décrite dans [Bénaben, 2012] comme une « approche IDM (Ingénierie Dirigée par les Modèles) [Bourey et al., 2007] de construction de cartographie BPM destinée à permettre la définition et la construction d'un SIM agile, positionné entre les SI des organisations et assurant l'interopérabilité, reposant sur le paradigme SOA et implémenté à l'aide d'un ESB ».

La première phase du projet, MISE 1.0, a été conduite avec succès de 2004 à 2010 et a donné lieu à la réalisation d'un prototype. De nombreux travaux de thèse (Jihed Touzi, Vatcharaphun Rajsiri, Sébastien Truptil) et stages de Master Recherche ont été impliqués durant la conception, la réalisation et l'évaluation de MISE 1.0. L'approche de MISE 1.0 est de type Ingénierie Dirigée par les Modèles (MDE pour Model-Driven Engineering). Outre les différents travaux de thèse qui ont permis d'articuler les différentes itérations du projet MISE, les projets de recherche européens et français ont constitué le deuxième pilier de ce projet. La première phase du projet, MISE 1.0, a pu s'appuyer sur les projets **JoNES** (ANR/05/RNTL/0015, 2005-2007 : Java Open ESB) et **ISyCri** (ANR/06/SECU/006, 2006-2010 : Interopérabilité des Systèmes en situation de Crise).

MISE 2.0 est supporté par le projet européen **PLAY** (ICT Call 5, topic ICT-2009.1.2, EU/FP7, 2010-2013) et les projets français **ISTA3** (FUI, 2008-2012 : Interopérabilité de 3ème génération des Sous-Traitants de l'Aéronautique) et **SocEDA** (ANR/2010/ARPEGE, 2010-2013).

Enfin, la troisième itération de MISE (MISE 3.0) s'appuie pour l'instant sur les projets **SIM-PeTra** (Ministère de l'Écologie, du Développement Durable, des Transports et du Logement, PREDIT G02 2011, 2011-2013 : Système d'Information de Médiation, application aux Perturbations dans les Transports), **OpenPAAS** (FUI/2012-2014 : Open

Platform As A Service) et **DRIVER** (FP7-SEC-2013.4.1-1/2014-2018, DRiving Inno-Vation in crisis management for European Resilience).

I.5.2 Le projet MISE 2.0

La seconde phase du projet, MISE 2.0 a été initiée en 2009 avec les travaux de thèse de Wenxin Mu et de Nicolas Boissel-Dallier. Ceux-ci se sont intéressés à la partie Design Time de MISE 2.0. Les autres travaux de thèse qui portent sur la partie Runtime [Barthe et al., 2011a] font partie de la suite du projet MISE 2.0 initiée fin 2010. Les travaux de thèse de Sarah Zribi, également liés à MISE 2.0 et débutés fin 2010, s'intéressent plus particulièrement au Design Time. Ses travaux portent sur l'aspect non-fonctionnel des services (confiance, taux de disponibilité, etc.) et complètent les travaux de Nicolas Boissel-Dallier en introduisant les critères non-fonctionnels comme éléments de décision pour la déduction des processus exécutables. La Figure I.10 présente l'imbrication de l'ensemble des travaux dans le contexte MISE 2.0. L'itération MISE 2.0 doit se terminer en fin d'année 2013.

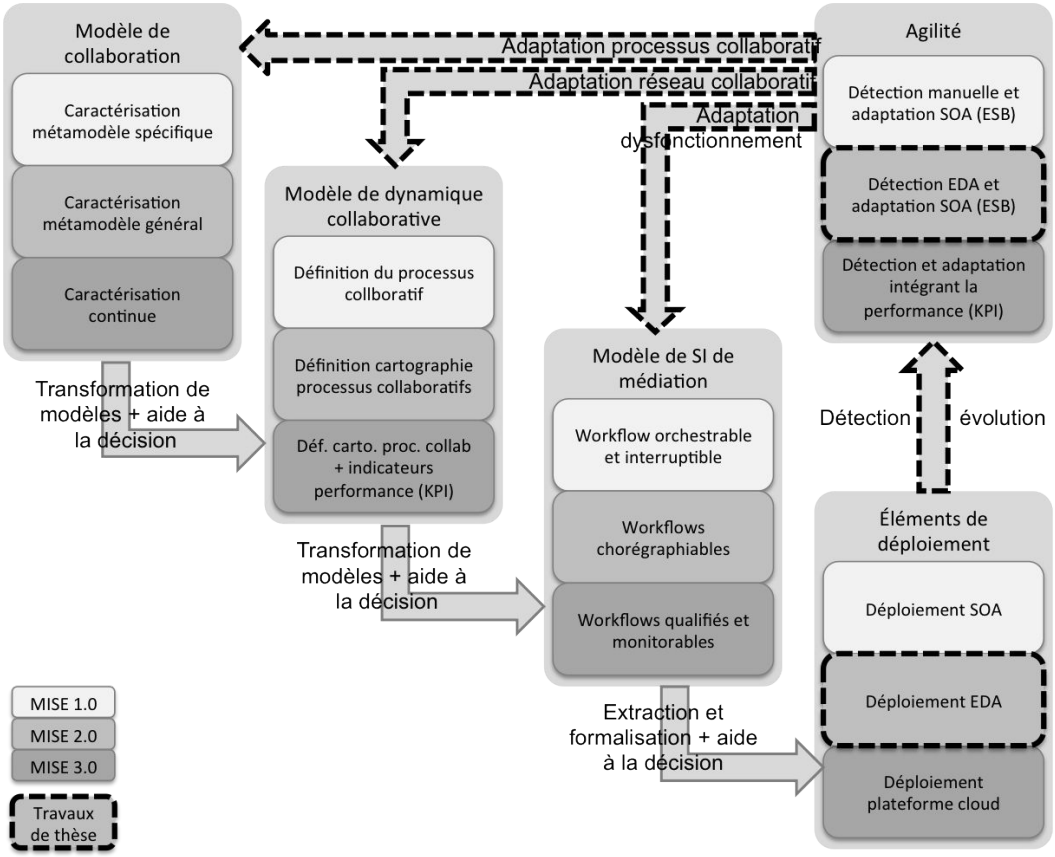


FIGURE I.10 – Positionnement de la thèse par rapport aux travaux menés dans le projet MISE

La démarche MISE 2.0 peut se résumer en cinq points (que l'on peut retrouver dans

la Figure I.10) :

- Le recueil de l'information et de la connaissance relative à la situation considérée : principalement en terme de contexte (i.e. l'environnement dans lequel se positionne la collaboration), d'objectifs (i.e. la raison d'être et les motivations collectives du réseau d'organisations) et de moyens (i.e. les compétences et ressources que les partenaires potentiels sont disposés à partager),
- Une première transformation de modèle : elle permet d'adresser la problématique de la découverte de la solution, c'est-à-dire comment passer de la description d'un contexte, d'objectifs et de moyens à une proposition de solution adaptée,
- Une deuxième transformation de modèles : elle adresse quant à elle la problématique de la concrétisation de la solution, i.e. le passage de la solution conceptuelle abstraite et tournée uniquement vers le domaine métier (issue de la première transformation) à une formalisation techniquement cohérente qui tient compte des exigences du contexte,
- L'extraction et la formalisation : il s'agit là de répondre à la problématique de l'implémentation et de l'exécution de la solution. Il faut passer de la description de la solution cohérente techniquement à sa réalisation concrète.
- La gestion de l'agilité. Nous avons vu précédemment que l'agilité des processus collaboratifs était désormais indissociable d'une démarche d'établissement et de maintien d'une collaboration. Les travaux de la présente thèse se positionnent à ce niveau de la démarche MISE, et visent à fournir l'outillage nécessaire au support de l'agilité.

Les trois premières étapes de la démarche MISE constituent les étapes de Design Time (conception) du SIM, et les deux dernières s'inscrivent dans le Runtime (exécution). Il est intéressant de noter que la mise en place de l'agilité va conduire le Runtime à déclencher l'une ou l'autre des étapes du Design Time, en vue d'adapter la logique opérationnelle collaborative (le SIM) aux nouvelles contraintes de l'environnement de la collaboration et/ou de la collaboration elle-même.

I.5.3 Le projet MISE 3.0

La troisième itération du projet, MISE 3.0 a débuté fin 2011 avec les travaux de thèse de :

- Guillaume Macé-Ramète, qui s'intéresse à la performance de la dynamique collaborative et à l'introduction d'indicateurs de performances dans la phase d'agilité du SIM. Il a de plus redéfini le méta-modèle de la collaboration afin de le rendre sensible au domaine métier considéré. Son cas d'étude porte sur la crise routière dans la région Grand Ouest de la France et ses travaux de thèse s'appuient sur le projet SIM-PeTra,
- Aurélie Montarnal, qui se penche vers la caractérisation continue de la dynamique collaborative dans le contexte d'une utilisation sur une plateforme ouverte et hautement distribuée. Elle met en place une couche sociale afin de faire émerger des opportunités de collaboration entre des partenaires qui ne se connaissent pas.

Ses travaux de thèse sont appuyés par le projet OpenPAAS,

- Loïc Bidoux, qui s'intéresse lui aussi à la déduction de la dynamique collaborative en y intégrant la prise en compte des ressources disponibles (hommes, véhicules, matériel, etc.). Pour cela, il se place dans le cadre de la gestion de crise, domaine où la gestion des ressources est capitale,
- Tiexin Wang, qui étudie l'amélioration conceptuelle et technique des différentes transformations de modèles mises en jeu dans MISE. Il vise ainsi la conception d'un atelier de définition automatique de transformation de modèles, ce qui permettra un développement rapide du SIM quelque soit le domaine métier ciblé,
- Alexandre Sirko, qui va s'intéresser aux notions de Big Data et Open-Data, en vue d'assurer une collecte « intelligente » de données afin de les transformer en information et connaissance exploitables. Les aspects de sécurité et de confiance dans les données vont faire partie intégrante de ses travaux.

I.6 Conclusion

Nous avons pu constater que l'agilité des processus collaboratifs est un enjeu majeur pour assurer la poursuite des objectifs définis par la dite collaboration. Ceci est valable quelque soit la situation collaborative considérée : sous-traitance industrielle, situation de crise, etc.

Afin de résoudre ce problème de l'agilité des processus collaboratifs, nous proposons un Service d'Agilité qui assurera les fonctionnalités principales suivantes (cf. Figure I.11) :

1. La détection de la divergence entre la situation que l'on espère avoir obtenue via l'exécution des processus collaboratifs (situation attendue) et la réalité du théâtre de la collaboration (situation terrain),
2. L'adaptation des processus collaboratifs, qui est décidée et réalisée suivant les conclusions tirées de la l'analyse de la divergence.

Ce Service d'Agilité s'appuie sur les résultats des travaux de thèse et projets de recherche menés précédemment au sein du laboratoire. Exécuté tout au long du cycle de vie de la collaboration, il prend en entrée la caractérisation de la situation et le modèle de la dynamique collaborative (issus des travaux menés sur la conception du système de traitement), ainsi que les événements générés par l'exécution des processus et le terrain de la collaboration. Les événements récoltés et analysés (à travers une architecture EDA) permettent de mettre à jour la caractérisation de la situation suivant deux points de vue : situation attendue, situation terrain. Le calcul de la divergence s'effectue en comparant les modèles de la situation attendue et de la situation terrain). Suivant le résultat de la détection (i.e. l'amplitude de la divergence, si elle existe), le Service d'Agilité propose la meilleure solution possible d'adaptation du système de traitement de la collaboration : redéfinition complète ou partielle des processus, remplacement ou re-exécution d'un service.

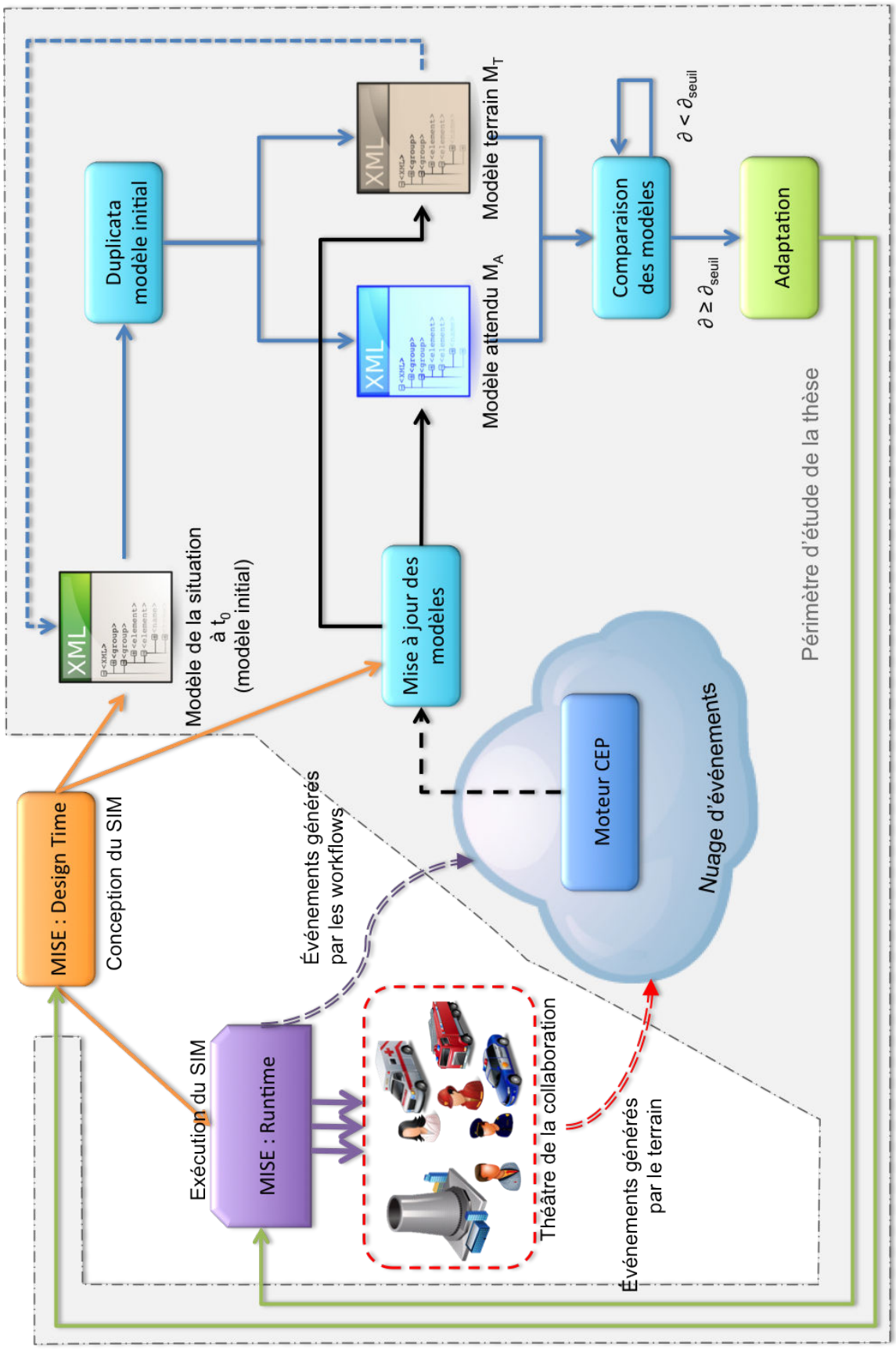


FIGURE I.11 – Schéma d'architecture haut niveau et de positionnement du Service d'Agilité

Chapitre II

État de l'art autour de l'agilité des processus

Après avoir défini le contexte de ces travaux, nous allons dresser un état de l'art relatif à l'agilité des processus. Nous allons procéder suivant deux points de vue : dans la première section (étude par rapport à l'objectif), nous mènerons une synthèse des différents travaux réalisés sur la définition de l'agilité des processus et les solutions existantes attachées aux différentes visions que nous rencontrerons. Nous extrairons de cette section la vision de l'agilité des processus que nous retiendrons dans la suite de ces travaux de thèse. Il s'agira de répondre à la question suivante : « Comment définir l'agilité des processus ? » Dans la seconde section, nous essaierons de répondre à la vision de l'agilité retenue en terme de moyens de mise en œuvre, c'est-à-dire « Comment mettre en place l'agilité des processus ? ». Cette étude par rapport aux moyens d'atteindre l'agilité des processus sera réalisée dans l'optique de créer un outil logiciel assurant l'agilité des processus. Enfin, la dernière section conclura ce chapitre et fera la synthèse de nos choix de concepts et de moyens.

II.1 Étude par rapport à l'objectif

II.1.1 Esquisse du paysage bibliographique

Depuis les années 1990, l'agilité, et plus particulièrement l'agilité des processus, est un sujet fréquemment discuté au sein de la littérature scientifique. Depuis le début du XXI^{ème} siècle, le nombre de publications portant sur ce thème a explosé, comme le montrent la Table II.1 et la Figure II.1, en synthèse des recherches menées sur les principales bases de données scientifiques académiques scientifiques que sont ScienceDirect, Web Of Knowledge et SpringerLink (recherches menées sur les champs auteur, titre, résumé, mots-clés). Ce tableau permet de voir quelle est la proportion d'articles concernant *l'agilité des processus collaboratifs* et *l'agilité collaborative* par rapport à *l'agilité des processus* : suivant les bases de données scientitiques, le rapport est de un article concernant l'agilité dans un contexte collaboratif pour six à vingt articles discutant de l'agitilité des

Mots-clés	**ScienceDirect**	**Web of Knowledge**	**SpringerLink**
Agility process	6437	1053	6996
Agility collaborative	1385	111	1743
Agility collaborative process	1341	55	1703

TABLE II.1 – Nombre d'articles trouvés lors des recherche sur les principales bases de données académiques

processus. Il apparaît clairement que l'étude de l'agilité des processus collaboratifs est un domaine d'actualité.

En ce qui concerne l'agilité des processus collaboratifs, une étude plus poussée des résultats nous fait remarquer un intérêt croissant de la communauté scientifique sur ce thème, notamment durant ces cinq dernières années (Figure II.1). Le pic correspondant à l'année 1994 se traduit par le regroupement des publications de l'année 1994 avec celles des années antérieures. Les données pour 2013 sont à revoir à la hausse d'ici la fin de cette année là : les recherches ayant été actualisées le 20 juin 2013, de nombreux articles sont encore en cours d'acceptation ou de parution et ne sont pas encore visibles dans la plupart des bases de données académiques scientifiques.

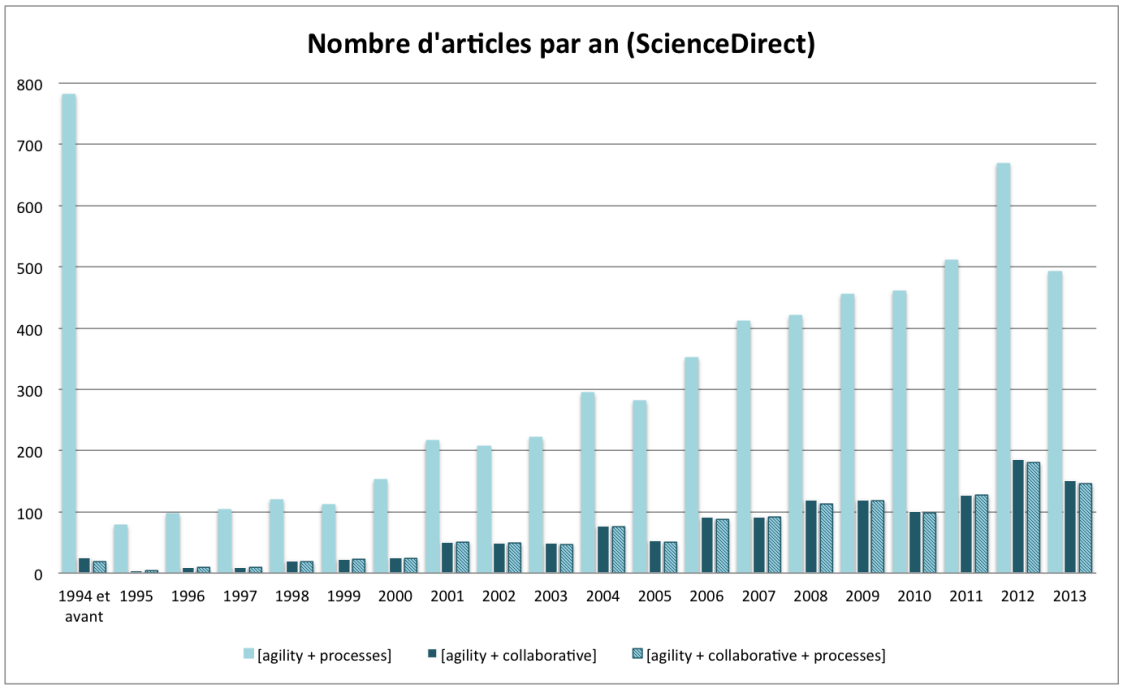


FIGURE II.1 – Résultat des recherches sur ScienceDirect sur les 20 dernières années

II.1.2 Définition de l'agilité

Le concept d'agilité a émergé avec l'évolution des relations industrielles, relations caractérisées comme la « transformation d'une structure cristalline vers un environnement fluide » [Bénaben et al., 2007] ou bien comme « la transformation d'une construction statique de Lego© vers un organisme vivant » [Luzeaux and Ruault, 2008].

Ces métaphores reflètent l'évolution des collaborations entre des entreprises et des organisations au sens large. Elles ne sont plus fondées sur une vision à long terme qui reste difficile à établir dans le contexte économique, politique et social actuel. De nos jours, elles sont basées sur des collaborations opportunistes, rapidement établies et tout aussi rapidement dissoutes.

L'agilité est prosaïquement définie comme la qualité de ce qui est agile, c'est-à-dire « qui manifeste de la promptitude et de l'aisance dans ses mouvements » [CNRS, 2013]. Si l'intérêt pour l'agilité des processus collaboratifs va croissant, il est intéressant de remarquer que très peu d'auteurs s'intéressent à définir ce qu'est cette agilité des processus collaboratifs (ou du moins à expliciter l'approche retenue) et se concentrent sur la mise en œuvre technique de cette dernière. Il nous faut donc adopter un angle d'attaque différent afin de définir cette agilité des processus collaboratifs. Le fait que les organisations mises en jeu dans les processus collaboratifs puissent être perçues comme constituant une même organisation virtuelle et éphémère, et que leurs fonctions soient mises à disposition de la collaboration et assemblées en vue d'atteindre le but commun (production de services essentiellement, dans le cas présent) nous amène à étudier les notions d'agilité de l'entreprise et d'agilité des chaînes logistiques [1]. Une étude de la littérature nous montre combien ces notions sont discutées depuis deux décennies. A travers les différentes définitions et visions présentées, nous chercherons à faire émerger une définition de l'agilité applicable aux processus collaboratifs.

II.1.2.1 Agilité de l'entreprise

Dans la littérature, nombre des définitions concernant l'agilité convergent vers le fait qu'une entreprise ne peut être qualifiée d'agile que si elle peut s'adapter aux variations du contexte dans lequel elle opère. Certains auteurs, tel Badot [Badot, 1998], évoquent la reconfiguration du système : l'entreprise agile tend vers « la vente à chaque client de solutions complètes combinant des biens, des services et de l'information, à un niveau élevé de différenciation et à des prix de série ». Il s'agit alors de pouvoir suivre une demande spécifique (différenciation) en minimisant le coût de cette réponse (prix de série). Lindberg évoque lui aussi la nécessité pour l'entreprise de travailler avec ses clients afin de les aider à identifier leurs problèmes et leurs exigences et d'acquérir les fonctionnalités nécessaires juste avant d'en avoir l'usage [Lindberg, 1990].

1. Une chaîne logistique est une réseau d'organisations —qui supporte des flux physiques, informationnels et financiers— impliquées par des relations en amont et en aval, dans différents processus et activités, qui fournissent un produit ou un service, dans le but de satisfaire le client [Christopher, 2010].

Pour Kidd [Kidd, 1994], l'agilité se traduit par une « adaptation rapide et proactive d'éléments de l'entreprise à des changements inattendus et imprévus ». Sharifi [Sharifi and Zhang, 1999] abonde dans ce sens en définissant l'agilité comme la « capacité à faire face à des changements inattendus, pour survivre à des menaces sans précédent dans l'environnement, et de faire de ces changements des opportunités ». Dans le domaine des systèmes d'information, l'agilité est perçue comme « l'habilité conjointe de créer et de répondre aux changements d'un environnement turbulent au profit de l'entreprise » [Highsmith, 2002].
Nous pouvons constater à travers la répétition de certains mots-clés (adaptation, changement) que la vision de l'agilité des entreprises est fortement liée à la production industrielle et à un besoin d'adapter l'outil de production à une demande fluctuante et de plus en plus exigeante en terme de personnalisation. Les auteurs discutent essentiellement du besoin de *réactivité* et d'*adaptabilité* dans un environnement instable.

D'autres articles font émerger la notion de *détection*. Ainsi, pour [Dove, 2002], l'agilité découle de deux capacités : d'une part la capacité de réponse, qu'il définit comme la capacité physique de réaction, et la capacité de détection, vue comme une capacité intellectuelle à trouver des éléments utiles sur lesquels agir. L'agilité est ainsi exprimée comme « la capacité à gérer et appliquer efficacement des connaissances, permettant à l'entreprise d'avoir le potentiel nécessaire pour prospérer dans un environnement changeant et imprévisible ». D'après [Overby et al., 2006] il s'agit de la « capacité des organisations à détecter un changement dans l'environnement et à y répondre aisément ». L'auteur complète sa définition en ajoutant que « l'agilité en entreprise se compose en deux volets : détection et réponse ».

Nous pouvons remarquer que cette notion de *détection* s'appuie sur la connaissance perçue par l'entreprise de son environnement, ce que l'on peut traduire par la sensibilité au contexte. Elle va se baser sur cette connaissance pour déterminer la nécessité de répondre à une éventuelle évolution de l'environnement.

II.1.2.2 Agilité des chaînes logistiques

Dans le domaine de la logistique, l'agilité est couramment définie comme la capacité à répondre à des changements imprévus [Sheffi, 2004] et se différencie de l'adaptation au fil de l'eau en réponse à un changement [McCullen et al., 2006]. Dans le Chapitre I, nous avons pu voir que les sources de changements imprévus étaient le théâtre de la collaboration ainsi que la collaboration elle-même.

[Charles, 2010] présente dans ses travaux de thèse une étude de l'agilité des chaînes logistiques. Après avoir croisé plusieurs articles discutant de la notion d'agilité et analysé les différentes facettes de l'agilité des chaînes logistiques, elle propose le cadre nommé « Maison de l'Agilité des chaînes logistiques » [Charles, 2010] [Lauras, 2012], présenté en Figure II.2. Ce cadre synthétise les principales capacités qui rendent une chaîne logistique agile : les définitions de ces capacités sont détaillées dans la Table II.2, également issue

des travaux de thèse de [Charles, 2010].

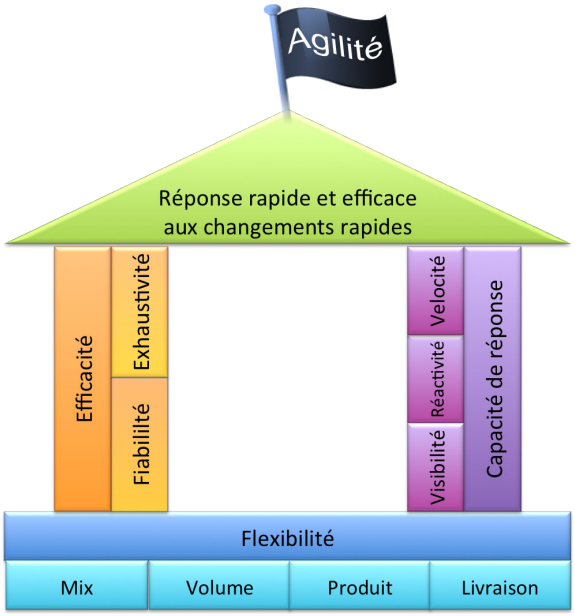


FIGURE II.2 – La Maison de l'Agilité, d'après [Charles, 2010]

	Capacités	**Définitions**	**CLC**	**CLH**
Flexibilité	Flexibilité volume	Capacité à changer le niveau de production totale [Slack, 2005]	+	+++
	Flexibilité livraisons	Capacité à changer les dates de livraisons planifiées ou présumées [Slack, 2005]	+	+++
	Flexibilité mix produit	Capacité à changer la gamme de produits livrés dans un temps donné [Slack, 2005]	++	+++
	Flexibilité livraison	Capacité à introduire de nouveaux produits ou modifier ceux existants [Slack, 2005]	++	+
Capacité de réponse	Réactivité	Capacité à rapidement identifier et évaluer les besoins	+	+++
	Vélocité	Capacité à couvrir rapidement les besoins	+	+++
	Visibilité	Capacité à connaître la nature, la localisation et le statut actuel et futur des entités en transit dans la chaîne logistique [Vernon, 2008]	++	+
Efficacité	Fiabilité (faire bien)	Capacité à livrer la bonne quantité du bon produit, au bon endroit, au bon moment, en bon état, avec la bonne documentation, à la bonne personne [Council, 2010]	+++	++
	Exhaustivité (faire tout)	Capacité à réaliser les objectifs	++	++

TABLE II.2 – Définition des concepts clés de l'agilité pour les chaînes logistiques, d'après [Charles, 2010] et [Lauras, 2012]

Pour construire ce cadre, [Charles, 2010] s'appuie sur les travaux de [Christopher and Towill, 2000] et de [Swafford et al., 2006] qui affirment que l'agilité d'une chaîne logistique repose sur la flexibilité. [Charles, 2010] étudiant plus particulièrement les chaînes logistiques humanitaires (qui fournissent une prestation de services), elle restreint ensuite la décomposition de la flexibilité donnée par [Zhang et al., 2003] aux capacités de « flexibilité en volume » et de « flexibilité en mix produit ». Elle s'appuie également sur la classification donnée par [Slack, 2005]. La flexibilité est ainsi vue comme la base indispensable pour obtenir une chaîne logistique agile, sur laquelle d'autres capacités nécessaires à l'agilité vont se positionner comme la capacité de réponse et l'efficacité.

Sur la base de cette étude, [Charles, 2010] propose la définition suivante : l'agilité est la « capacité de répondre rapidement et de façon adéquate aux changements à court terme concernant les demandes, l'approvisionnement ou l'environnement. Elle provient de la flexibilité, la capacité de réponse et l'efficacité des chaînes logistiques ».

La notion d'agilité dans les chaînes logistiques est très fortement liée à la capacité de production de biens. Or, dans le cadre de notre étude (agilité des processus collaboratifs), l'objectif commun des acteurs de la collaboration n'est pas nécessairement la production de biens. Il peut s'agit de la fourniture de services. D'autre part, le point de vue adopté dans ces travaux de thèse est celui de la situation collaborative : il s'agit pour chaque acteur de mettre ses fonctions à disposition de la collaboration, qui va sélectionner les services les plus pertinents afin de pouvoir exécuter un ou plusieurs processus collaboratifs. Que la fonction proposée par l'acteur et utilisée par le processus collaboratif produise un bien ou fournisse un service, du point de vue de la collaboration il s'agit dans tous les cas d'un service.
De fait, la notion de flexibilité de production n'étant pas pertinente dans le cadre d'une fourniture de services, elle ne peut s'appliquer au domaine des processus collaboratifs. Dans son étude, [Charles, 2010] évalue chacune des capacités pour le domaine de la chaîne logistique commerciale (CLC) et le domaine de la chaîne logistique humanitaire (CLH). De par sa nature même (prestation de service, réponse rapide au changement), la CLH se rapproche plus des processus collaboratifs que les CLC. Nous privilégions donc les aspects de l'étude de la flexibilité menée par [Charles, 2010] qui concernent les CLH.

Si l'on veut transposer la flexibilité présentée dans la *Maison de l'Agilité* au domaine des processus collaboratifs, il est possible de la voir comme la capacité de reconfiguration du réseau collaboratif (changer les activités des processus, changer leur ordre d'exécution par exemple) afin de prendre en compte les évolutions de l'environnement. La capacité de réponse est vue par [Charles, 2010] comme la combinaison de la réactivité, de la vélocité et de la visibilité. D'après les définitions attachées à ces capacités dans la Table II.2, nous pouvons constater que la réactivité et la vélocité sont favorisées dans le CLH : il faut en effet identifier, évaluer et répondre rapidement aux besoins. Nous trouvons ici la notion de détection du changement (qui a été abordée dans l'étude de l'agilité de l'entreprise) mêlée à la dimension temporelle (la réactivité). D'autre part, une collaboration n'étant

Composantes	**EA**	**CLA**
Détection / Identification d'un changement	+++	+
Adaptation / Réponse	+++	+++
Efficacité		++
Réactivité	+++	+++

TABLE II.3 – Synthèse des composantes apparaissant dans les définitions de l'agilité des entreprises et des chaînes logistiques

établie que dans le but d'atteindre un objectif commun, il semble pertinent de conserver la notion d'efficacité dans notre vision de l'agilité des processus collaboratifs afin de mesurer l'atteinte de l'objectif.

II.1.2.3 Synthèse

Des considérations précédentes, nous pouvons retenir les caractéristiques essentielles liées à la notion d'agilité des entreprises agiles (EA) et des chaînes logistiques agiles (CLA) et relever leur présence dans ces deux visions (Table II.3). Nous pouvons constater que l'adaptation et la réactivité sont les deux composantes qui apparaissent à niveau d'importance égale dans les définitions de l'EA et de la CLA Cependant, la définition de l'EA explicite la composante détection : la sensibilité au contexte dans lequel l'entreprise évolue est un point clé permettant de savoir à quel moment un changement nécessite la conduite d'une adaptation de l'existant, mais aussi de déterminer l'amplitude de l'adaptation à conduire sur la base de cette connaissance. De même, l'efficacité semble être une composante évoquée principalement dans le contexte de la CLA. L'efficacité constitue un bon indicateur de la performance de l'agilité des processus collaboratifs, permettant ainsi de mesurer la capacité de ces processus à évoluer en fonction du contexte afin de toujours satisfaire les objectifs de la collaboration. Cet indicateur peut également être utilisé à des fins d'apprentissage et de retour d'expérience afin de comprendre les implications des choix réalisés lors de l'adaptation vis-à-vis du résultat obtenu (en terme d'atteinte des objectifs).

Sur la base de cette synthèse des composantes essentielles de l'agilité des entreprises et des chaînes logistiques, nous proposons la définition de l'agilité des processus collaboratifs suivante : l'agilité des processus collaboratifs peut donc être vue comme la capacité d'un système à réaliser, dans le laps de temps le plus court possible *(*réactivité), la *détection* de son inadéquation à l'environnement dans lequel il évolue et la mise en œuvre de l'*adaptation* nécessaire.
Cette définition peut se synthétiser par la formule suivante :

Formule 1 *Agilité = (Détection + Adaptation) × (Réactivité + Efficacité)*

Où :

- Détection : elle consiste à détecter une évolution de la situation qui ne peut être traitée par le système en place et qui rend le système inadéquat vis-à-vis du positionnement courant,

- Adaptation : relevant de la flexibilité, elle transcrit la capacité du système à faire évoluer sa définition et sa structure afin de redevenir pertinent (en terme de moyens et d'objectifs) vis-à-vis du positionnement courant,
- Réactivité : elle relie les notions de détection et d'adaptation à la dimension temporelle. Détection et adaptation doivent être menées à bien le plus rapidement possible (idéalement, en temps réel).
- Efficacité : elle relie les notions de détection et d'adaptation à la dimension de la performance. La détection doit être juste et l'adaptation doit être pertinente.

Dans le cadre de ces travaux de thèse, nous nous attacherons à concevoir et mettre en œuvre les composantes de détection, d'adaptation et de réactivité. Si l'efficacité est une des composantes de l'agilité des processus collaboratifs, ce sont les travaux de thèse de Guillaume Macé Ramète [Macé-Ramète et al., 2012], actuellement en cours, qui s'attachent à ajouter des indicateurs de performance dans les modèles de situations collaboratives et à les exploiter.

Il est à noter que la frontière entre la détection et l'adaptation reste floue, de par la nature même de ces deux notions : la détection ne peut se faire qu'en étudiant les éléments traduisant l'évolution de la situation et conduisant au diagnostic d'adaptation. Or ces mêmes éléments traduisant l'évolution de la situation vont servir à définir la stratégie d'adaptation à mettre en œuvre. Par exemple, si les services d'un partenaire ne sont plus disponibles, le processus ne peut plus les appeler en vue de les exécuter. Une solution d'adaptation possible pourrait être de chercher chez ce partenaire d'autres services pour remplacer ceux manquants, ou de changer de partenaire (pour peu qu'il propose des services remplissant les fonctions attendues). La détection et l'adaptation se recouvrent en ce sens qu'elles travaillent sur la même matière première. En outre, les étapes de détection et d'adaptation ne sont pas nécessairement séquentielles et chronologiques (voir Figure II.3). S'il paraît évident que l'adaptation ne peut se faire sans avoir effectué la détection, cela ne signifie pas que la détection s'arrête lors de l'exécution de l'adaptation. Elle peut continuer à s'exécuter, mais elle doit prendre en compte le fait que si une action d'adaptation est réalisée, le modèle de référence de la situation est modifié. En effet, l'adaptation étant réalisée à partir de la connaissance extraite de la caractérisation de la collaboration et de son contexte (i.e. le modèle de la situation) à l'instant t où un besoin d'adaptation a été détecté, c'est ce modèle à l'instant t qui devient la référence pour les actions de détection futures.

II.1.3 De l'agilité des processus

Après avoir précisé la vision de l'agilité qui est adoptée dans ces travaux, il est possible d'étudier les différents types de mise en œuvre de l'agilité au niveau des processus et workflows (i.e. processus exécutables), tant dans le domaine académique que commercial.

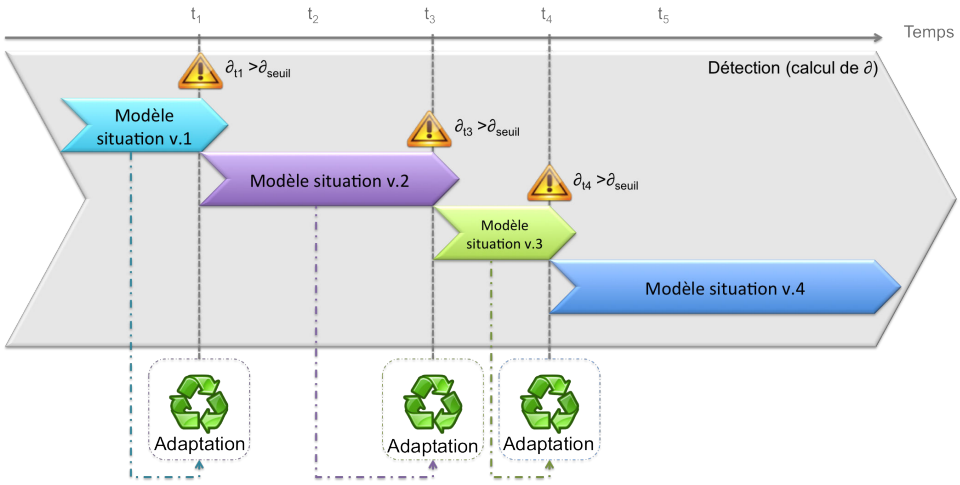


FIGURE II.3 – Positionnement des actions de détection et d'adaptation dans le temps

II.1.3.1 Concepts d'agilité des processus

L'adaptation et la réactivité constituent deux contraintes dans la mise en place de l'agilité au sein d'une collaboration réalisée à travers les systèmes d'information des partenaires. En effet, la partie *système informatique* (telle que définie par [Morley et al., 2007]) ne peut être modifiée (ou adaptée) étant donné la contrainte de temps qui est imposée : modifier le comportement des logiciels de l'organisation, redéfinir les connexions internes, etc. pose des problèmes de délai. Seule la partie *système d'information* peut être modifiée sans subir un coût temporel majeur. Dans le cadre de la situation collaborative, le SI concerné n'est autre que le SIM tel que défini dans le Chapitre I. Le SIM ayant en charge l'orchestration de l'enchaînement des activités des partenaires à travers les processus définis en amont, la mise en place de l'agilité va être réalisée au niveau des processus et des workflows.

Avant de pouvoir répondre à la problématique de l'agilité des processus, il nous faut définir notre approche de l'agilité –à savoir en conception et/ou en exécution– qui est liée à la nature même de nos processus et workflows. Dans [Hollingsworth, 1995], les fonctions sont classés en trois catégories : (i) les fonctions de conception des processus et workflows (Design Time), (ii) les fonctions de contrôle de l'exécution (Runtime) et (iii) les interactions lors de l'exécution avec les utilisateurs et les applications (Runtime). [Nurcan and Hicheur, 2005] propose une classification différente qui se base sur l'agilité en fonction de la capacité à intégrer des modifications dues aux changements dans les phases de Design Time ou de Runtime. Cette approche est également retenue dans les travaux de [Schonenberg et al., 2008], qui propose de classer les processus suivant leur approche de l'agilité. Sur la base de ces considérations, la Figure II.4 présente une vue d'ensemble du processus de conception d'un SI [Hollingsworth, 1995] et permet de positionner notre vision de l'agilité dans le SI. Les éléments en blanc représentent le SI collaboratif basé sur les caractéristiques de la situation collaborative et les activités

des acteurs qui sont utilisées pour définir les processus collaboratifs (phase de Design Time). Ces processus sont transformés en workflows qui sont exécutés durant la phase de Runtime. Les éléments en noir représentent les caractéristiques de l'agilité appliquée au SI collaboratif : la détection des évolutions (de façon automatique ou manuelle) et les approches possibles de l'adaptation (implémentées aussi bien en phase de Design Time qu'en phase de Runtime). Nous pouvons constater que la composante *détection* de

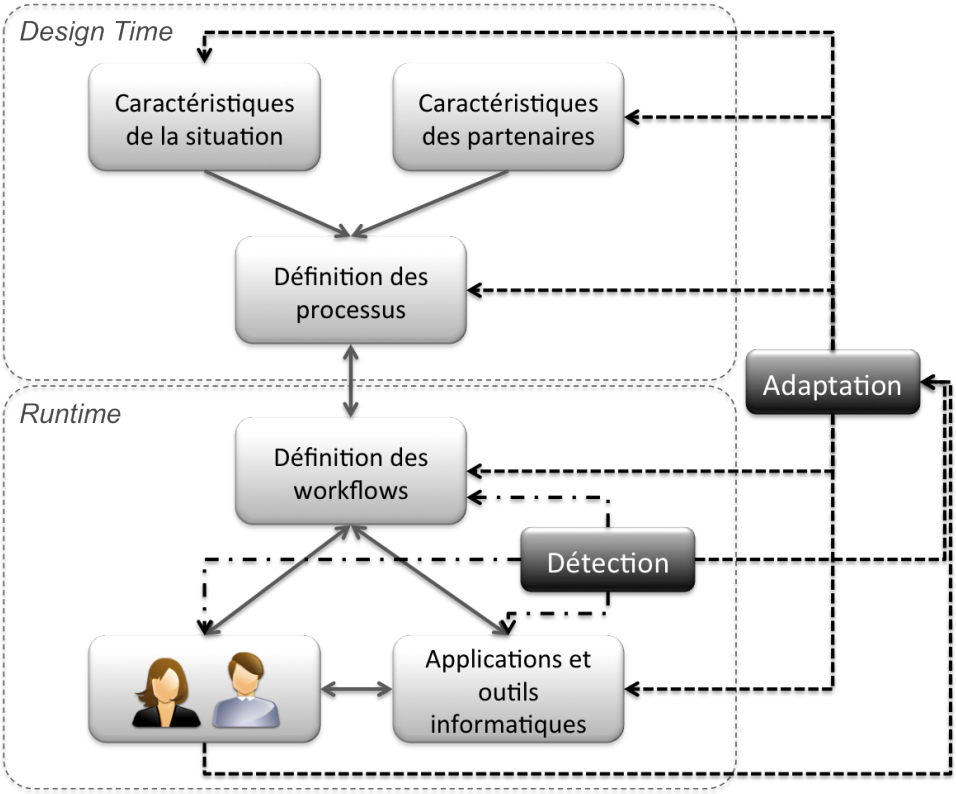


FIGURE II.4 – Vue d'ensemble des approches de l'agilité appliquée au SI collaboratif , d'après [Hollingsworth, 1995]

l'agilité est fortement liée aux données générées lors de l'exécution des workflows et leur environnement et ne porte donc que sur la partie Runtime. Ces données proviennent des activités concernées par les workflows, qu'elles soient exécutées par des êtres humains ou par des briques logicielles. Il nous faudra donc être en mesure de collecter les informations permettant de suivre l'exécution des workflows afin de détecter le besoin d'adaptation de ces workflows. La composante *adaptation* va influencer à la fois le Design Time et le Runtime des processus : elle peut ainsi demander à redéfinir ou compléter la connaissance à partir de laquelle les processus ont été définis et entraîner une refonte totale des processus et workflows (Design Time), modifier les processus existants (i.e. modifier le choix et l'enchaînement des activités métier — Design Time) ou encore modifier les workflows (i.e. modifier l'implémentation technique des processus ou re-exécuter une activité technique) — Runtime.

[Morley et al., 2007] expose deux vues de l'agilité des processus suivant l'horizon envisagé. Sur un horizon à court terme, l'agilité des processus consiste à répondre rapidement à des demandes sortant du cadre habituel. À long terme, il s'agit de la possibilité de modifier avec facilité la structure et les activités des processus. Morley indique que pour répondre au premier objectif (horizon à court terme), des aiguillages vers des chemins alternatifs peuvent être introduits dans la structure des processus. La création de ramifications pour couvrir les cas possibles permet de réagir rapidement à des fluctuations de l'environnement. [Morley et al., 2011] précise que ces aiguillages permettent aux acteurs des workflows de réaliser des activités manuellement : il s'agit donc plus d'une sorte d'issue de secours permettant de reprendre un contrôle manuel sur le processus automatisé et de court-circuiter le fonctionnement nominal de ce dernier pour répondre à des événements imprévus. Seule la composante *détection* de l'agilité est assurée : le choix de l'adaptation à mener et son exécution relèvent de la seule responsabilité d'un acteur désigné de l'organisation. Le second objectif (long terme) peut être réalisé en s'attachant, dès l'étape de conception des processus, à intégrer des techniques de généricité et de paramétrisation. Il s'agit alors de définir un processus générique pouvant évoluer vers différentes formes grâce à la présence d'alternatives dans la séquence et le choix des activités, les aiguillages, etc. Cette base est déclinée en spécialisations qui peuvent varier au cours du temps. Nous retrouvons ici une démarche proche de l'analyse des risques : pour chaque alternative identifiée, un ensemble de choix est mis en place à travers les branches alternatives du processus. Le but est alors d'être le plus couvrant possible (afin d'éviter de retomber sur la problématique de l'horizon à court terme). Cette agilité est pensée en amont, au moment de la conception du processus et relève donc du Design Time.

D'autres approches sont discutées dans la littérature, comme la gestion des incertitudes dans les systèmes de workflows. Nous pouvons mentionner en particulier les workflows lâches, les WIO (Workflows Inter-Organisationnels) adaptables [Andonoff et al., 2007] ou l'approche ADEPTflex [Reichert et al., 1999] ou, dans le même esprit, les travaux de van der Aalst [van der Aalst et al., 1998].

Les **Workflows Inter-Organisationnels** (WIO) ont pour objectif de faire coopérer des processus issus de différentes organisations en vue de fournir de nouveaux services et d'atteindre un but commun. Les WIO sont donc composés de plusieurs workflows individuels qui sont définis et exécutés au seins d'organisations qui sont par définition hétérogènes, autonomes et distribuées. L'enjeu des WIO est d'être le support de l'automatisation et de la coopération des tels processus métiers distribués et hétérogènes. Les WIO lâches correspondent à une collaboration occasionnelle et faiblement structurée. La particularité des WIO lâches est de définir cette collaboration et de rechercher les candidats (organisations) à la collaboration (ceux-ci n'étant pas nécessairement connus a priori). Les partenaires peuvent changer en pleine exécution des WIO. [Andonoff et al., 2009] présente une solution d'implémentation des WIO dans un environnement collaboratif. Ils se basent sur l'approche multi-agents pour prendre en charge l'agilité des

WIO, et utilisent deux ontologies pour rechercher puis sélectionner les organisations partenaires. Notons que cette solution d'implémentation s'appuie également sur les Web Services, facilitant ainsi l'intégration technique des services des partenaires dans le WIO.

Le but du projet **ADEPT** [Reichert and Dadam, 1998] est de développer un système de gestion de workflows adaptables. Il permet de dériver un workflow prédifini, c'est-à-dire qu'il autorise un utilisateur à intervenir directement sur les instances en cours d'exécution du workflow. L'utilisateur peut alors réaliser des modifications comme l'ajout ou la suppression d'activité, ou bien réordonner des séquences d'activités. Des règles de contrôle permettent de s'assurer que les instances des workflows restent cohérentes et exactes suite aux modifications apportées. Une première implémentation commerciale d'ADEPT (puis de sa version 2.0 nommée **ADEPTFlex**), appelée AristaFlow BPM Suite a vu le jour en 2008 [Dadam et al., 2010].

Une autre approche de l'agilité des workflows sont les **Worklets** [Adams et al., 2006]. Notons que dans le contexte des Worklets une activité est sous-divisée en un ensemble de tâches (actions). Un Worklet est un workflow atomique et autonome qui gère une tâche (action) spécifique dans un processus composite de plus grande ampleur (activité). Un modèle de processus est défini et son implémentation est réalisée durant son exécution en tant que workflow : pour chaque action du processus, un panel de worklets candidats est lié. Les worklets sont contextuellement choisis et appelés à partir du répertoire de chaque tâche lorsque l'instance de la tâche est activée lors de la progression de l'exécution du workflow. La flexibilité de cette approche réside dans le choix retardé : si la structure du processus est définie une fois pour toute, l'implémentation du workflow est choisie durant son exécution parmi un ensemble d'actions prédéfinies. Cette approche a été implémentée [Adams et al., 2006] dans un environnement **YAWL** [2] (Yet Another Workflow Language) [van der Aalst and ter Hofstede, 2005].

Case Handling est encore une autre approche, développée par [van der Aalst et al., 2005] s'appuie sur le pilotage des activités du point de vue des données échangées et non du processus (i.e. de l'enchaînement orchestrée des activités). Il s'agit dans cette démarche de choisir les activités à exécuter en fonction des données disponibles à un instant t et non en fonction de la dernière activité effectuée (i.e. de l'avancée dans l'exécution du processus) [Guenther et al., 2008]. En supposant qu'il possède les droits d'accès requis, le CHS (Case Handling System) met à disposition de l'utilisateur toutes les données issues du contexte. Le CHS orchestre l'exécution des activités sur la base des données liées à chaque cas et disponibles à un instant t. Cette approche a été implémentée dans **FLOWer** [Verbeek et al., 2002], qui a depuis été depuis renommée en **BPM|one** (il s'agit d'un produit commercial).

[Schonenberg et al., 2008] a proposé une synthèse de ces travaux via une taxonomie des approches de l'adaptation de workflows. Cette taxonomie, présentée dans la Figure

2. YAWL est un langage de workflow.

II.5, identifie quatre classes d'approches. Dans le cadre des présents travaux de thèse, la taxonomie utilisée est celle de [Bénaben, 2012] qui est largement inspirée de celle de Schonenberg. Nous y avons positionné les références aux travaux de recherche évoqués précédemment.

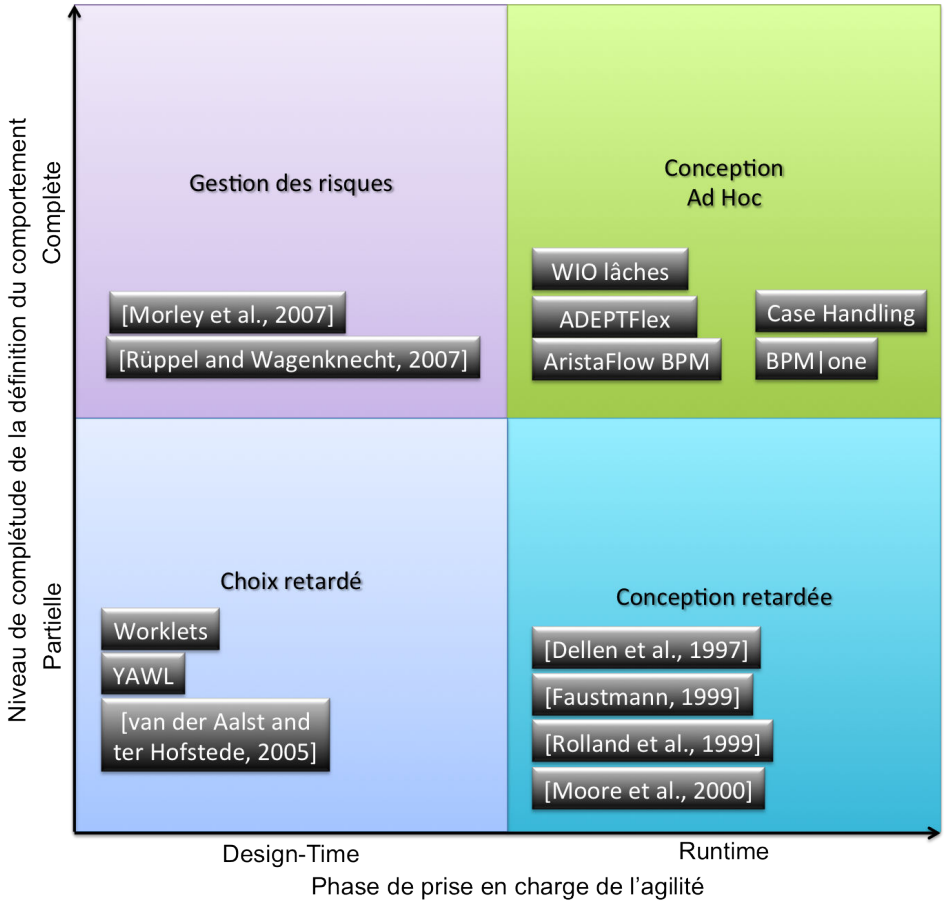


FIGURE II.5 – Référentiel d'adaptation du comportement, [Bénaben, 2012] d'après [Schonenberg et al., 2008] et positionnement des concepts d'agilité

Le choix retardé. Cette solution consiste à concevoir partiellement les processus (Design Time) et à les compléter durant leur exécution (Runtime). Il s'agit donc d'autoriser la construction de modèles de processus puis l'exécution des workflows pour lesquels certains fragments ou certaines activités ne sont pas décrits avec précision. Le choix final de l'implémentation des activités et sous-processus est réalisé au moment de l'exécution en choisissant un complément dans une liste [Faustmann, 1999] [Halliday et al., 2001] [Mangan and Sadiq, 2003] [Rolland et al., 1999]. Ce type d'approche de l'agilité est implémentée à travers le système YAWL (Yet Another Workflow Language) [van der Aalst and ter Hofstede, 2005]. Si la modélisation partielle des processus ne paraît pas problématique (les alternatives sont déjà définies), l'exécution de workflows partiellement définis semble

plus délicate à réaliser.

La conception retardée. De façon similaire au choix retardé, la conception retardée propose une modélisation partielles des processus. Elle propose en sus l'exécution partielle des workflows [Dellen et al., 1997] [Faustmann, 1999] [Moore et al., 2000] [Rolland et al., 1999], certains éléments des workflows n'étant pas identifiés au moment de la conception (Design Time) ni de l'exécution (Runtime). Ce type d'approche laisse penser qu'il faut pouvoir revenir en phase de Design Time pour pouvoir compléter les workflows en cours d'exécution dès que l'on se trouve confronté à une portion de workflow inachevée.

La gestion des risques permet de fournir de l'agilité au niveau de la conception des processus en incluant des chemins alternatifs pour faire face à un grand nombre de risques, menaces et opportunités. La sélection du chemin le plus approprié a lieu pendant l'exécution des processus. Par exemple, l'approche proposée par Rüppel [Rüppel and Wagenknecht, 2007] vise à définir le plus grand nombre de processus de réponse à une situation de crise. Ses travaux se focalisent spécialement sur le cas d'inondations en Allemagne. L'approche à court terme décrite par [Morley et al., 2007] entre également dans ce cadre. Les ramifications de ces processus sont très nombreuses [Sienou, 2009] mais ne peuvent couvrir la totalité des cas pouvant advenir. Cette limite est particulièrement palpable dans le cas des situations de crise.

La conception Ad Hoc. Cette dernière approche étend et généralise le principe de conception retardée. Elle apporte l'agilité au niveau de l'exécution en permettant de changer l'ordre d'exécution des activités sans modifier ces dernières (i.e. une tâche peut être annulée ou re-exécutée). Elle permet également de modifier la définition même des processus en cours d'exécution en insérant ou supprimant des tâches. Ce type d'approche est la plus commune : on y retrouve ADEPTflex par exemple [Reichert et al., 1999], ainsi que les travaux de [Borghoff et al., 1997], [Casati et al., 1998], [Reichert et al., 1999], [Sadiq and Orlowska, 2000], [Weske, 2001] et [van der Aalst and Jablonski, 2000].

II.1.3.2 Solutions existantes

Depuis une dizaine d'années, plusieurs solutions commerciales et projets de recherche tentent de proposer des fonctionnalités de conception et d'orchestration des workflows collaboratifs ainsi que la possibilité de rendre ces workflows agiles. Ce sont les WFMSs (Workflow Management Systems) [van der Aalst et al., 2009]. Du côté des solutions commerciales, les produits phares du secteur sont Bonita, les outils ARIS (Architecture of Integrated Information Systems) et TIBCO Business Studio.

Bonita Open Solution, qui est développé par Bonitasoft [Bonitasoft, 2013], offre une suite d'outils pour concevoir les processus, exécuter et surveiller les workflows. Bonita Open Solution présente l'intérêt de pouvoir simuler l'exécution des processus en fixant les paramètres suivants : coût, durée, consommation de ressources, calendrier. Dans sa

version gratuite (*Community Edition*), le logiciel ne conseille pas l'utilisateur sur les optimisations à mener (l'utilisateur doit réaliser lui-même l'analyse) et ne l'avertit donc pas sur la nécessité d'adapter les processus. Seule la version payante *Performance Edition* du logiciel autorise la surveillance des processus (monitoring) ainsi que la modification d'un processus en cours d'exécution (i.e. durant la phase de Run-time) en permettant de ré-exécuter une activité donnée d'un processus[3].

ARIS [Scheer and Nüttgens, 2000] est une approche de modélisation de l'écosystème de l'entreprise. Elle propose une suite d'outils qui permet de modéliser, exécuter, analyser, simuler et optimiser les processus de l'entreprise. Un point intéressant de la suite ARIS est la possibilité de combiner des fragments de processus entre eux suivant les événements reçus au cours de l'exécution du workflow afin d'en déterminer la suite à la volée. D'une certaine manière, ARIS permet d'adapter le workflow en cours d'exécution mais cette approche reste déterministe car il est impossible de sortir du cadre imposé par les processus unitaires existants. Cette approche reste fortement liée à l'étude des risques et à l'établissement de processus alternatifs en vue de couvrir un maximum de branches possible.

TIBCO iProcess Suite [TIBCO, 2013] propose un panel d'outils de création et de gestion de processus dont un logiciel de modélisation des processus, un orchestrateur ainsi que le moyen de simuler l'exécution des processus avant de les lancer en phase de production. Cette dernière fonctionnalité permet à l'utilisateur de déterminer quelles parties du processus peuvent être améliorées, de repérer les goulots d'étranglement et les manquements aux exigences en terme de coût et de qualité de service. La détection des goulots est réalisée à partir du monitoring des événements émis par les processus : l'outil avertit l'utilisateur de la présence de ces goulots. Des solutions d'adaptation des processus concernés sont alors proposées en puisant manuellement dans un recueil de processus pré-établis et qui permettent d'atteindre l'objectif concerné.

Du côté des résultats académiques, nous pouvons citer le projet européen **WORKPAD** [Catarci et al., 2011] qui permet de concevoir et développer une infrastructure logicielle dont le but est d'être le support de la collaboration dans des situations d'urgence ou de catastrophes. Ce projet vise à créer des communautés de PSSs (Public Safety Systems) et de permettre aux équipes sur le terrain d'exploiter ces PSSs au moyen des technologies mobiles, de la gestion de processus et de la géo-collaboration. WORKPAD propose une infrastructure de type pair-à-pair pour connecter les acteurs de la réponse à la crise entre eux (via le réseau de communication mobile et sans fil) et assurer le bon déroulement des processus mis en jeu dans la réponse à la crise. La plateforme résultant de ce projet propose d'adapter dynamiquement les processus en cas de panne ou d'indisponibilité (au sens réseau) d'un des services des acteurs de la crise. La détection

3. Cf. tableau comparatif des caractéristiques des éditions de Bonita Open Solution : `http://www.bonitasoft.com/products/product-comparison`

de la panne est réalisée en étudiant les fichiers de log des interactions humain-service et service-service. Sur la base de cette analyse et des liens sémantiques entre les acteurs et les services disponibles, de nouvelles solutions de remplacement sont proposées. Concernant l'adaptation des processus, WORKPAD se concentre sur le rétablissement des nœuds déconnectés du réseau via des tâches spécifiques.

Le projet européen **CRISIS** [Rooney, 2011] cherche à développer une plateforme de simulation pour de l'entraînement à la demande à destination des équipes de premiers secours et des gestionnaires de crise. Elle propose des cas d'entraînement réalistes (évacuation de stade, attaque terroriste dans un aéroport) et des niveaux de difficultés différents. La plateforme CRISIS est un support à la prise de décision sous conditions d'incertitude. Ces incertitudes peuvent varier suivant trois dimensions (augmentant ou diminuant la difficulté de la simulation) : la complexité de l'événement décrivant la situation, le nombre total d'événements et la probabilité de survenue de l'événement. Elle n'orchestre pas réellement les workflows et se concentre sur la prise de décision face à de nouveaux risques ou de nouvelles incertitudes. Cependant, la dimension de gestion de la connaissance et de retour d'expérience que présente la plateforme peut la positionner comme un outil d'adaptation a posteriori des processus de réponse à la crise.

Concernant la partie détection (une des composantes de l'agilité), le projet **PRONTO** [Marterer et al., 2012] vise à collecter puis déduire des événements complexes à partir de flux d'événements afin d'analyser les données dans des opérations de secours lors de situations de crise. Les événements proviennent du terrain : capteurs sur site, capteurs embarqués sur des véhicules ou des personnels d'intervention (position GPS, vitesse de déplacement), appels téléphoniques ou radio, images des caméras de surveillance. Ce projet fournit notamment un outil doté d'une interface web présentant une carte des opérations en cours (découpées en sections). Elle donne ainsi aux décideurs une image compréhensible de la situation de crise (d'un point de vue opérationnel) et permet de visualiser la localisation de leurs ressources ainsi que les événements associés (section créée, véhicule arrivé, véhicule entrant dans une section d'opérations, etc.). D'autres informations sont visibles comme les risques météorologiques (alerte orage, alerte pluie) pour la région des opérations. Le but recherché de PRONTO n'est pas de gérer l'exécution des workflows mais de fournir un support d'aide à la décision en analysant les événements entrants et en mettant en valeur les situations intéressantes pour la cellule de crise. Il diffère de CRISIS dans le sens où il n'a pas pour but d'être un support d'entraînement mais une véritable application de terrain.

Le projet **PACHUBE** [Haque, 2004] s'intéresse également au contexte [4] et aux données permettant de mettre à jour ce contexte. PACHUBE est une sorte de place de marché regroupant les données émises par des capteurs en temps réel et les met à dis-

4. Le contexte est représenté par toute information susceptible de caractériser la situation d'un sujet ou d'un système qui est considéré comme pertinent au regard de l'interaction entre l'utilisateur et l'application, incluant l'utilisateur et l'application eux-mêmes [Dey, 2001].

position des utilisateurs via Internet. Si ce projet ne s'intéresse pas à la gestion des processus collaboratifs et à leur adaptation au contexte, il montre l'intérêt croissant des problématiques type *Internet Of Things* [Atzori et al., 2010] où l'enjeu majeur est la collecte d'informations émises par des moyens variés (capteurs, personnes, réseaux sociaux) afin d'obtenir l'image la plus fidèle et pertinente possible du contexte dans lequel un système évolue et détecter des situations intéressantes (risques, opportunités, etc.). PACHUBE a été racheté en 2011 par la société LogMeIn et s'appelle désormais Xively.

Enfin, le projet européen **PLAY** [PLAY, 2010] propose une plateforme permettant non seulement l'exécution de workflows collaboratifs mais aussi la détection de risques et opportunités pour la collaboration. Cette détection est réalisée via une collecte et déduction d'événements, la plateforme jouant ainsi le rôle d'une place de marché en mettant à disposition des utilisateurs les données provenant d'événements émis par des capteurs, des réseaux sociaux, etc. PLAY fournit également un service d'adaptation des workflows appelé SAN (Situation-Action-Networks) [Verginadis et al., 2012]. Le principe de fonctionnement du SAN repose sur la décomposition des objectifs du processus en arbre. Si un objectif ne peut être atteint à cause d'une variation du contexte empêchant l'exécution de l'activité concernée, alors une autre activité est proposée. La proposition d'adaptation se fait grâce à une base de connaissances en cherchant l'activité la plus proche de celle à remplacer.

II.1.4 Conclusion sur les moyens existants

L'étude des différents moyens existants, passés au filtre de notre définition de l'agilité, nous permet d'obtenir le tableau de synthèse suivant (Tableau II.4).

Existant	**Détection**	**Adaptation**	**Réactivité**
Bonita (Gratuit)	Non	Non	Non
ARIS	Oui (Automatique, via événements)	Oui (Déterministe et automatique)	N/A
TIBCO	Oui (Manuelle, via lecture d'indicateurs)	Oui (Manuelle)	Non
WORKPAD	Oui	Oui (Uniquement réseau matériel)	Oui
CRISIS	Oui	Oui (Partielle, aide à la décision)	Oui
PRONTO	Oui	Non	Oui
PACHUBE	Oui	Non	Non
PLAY	Oui	Oui (Déterministe)	Oui

Table II.4 – Synthèse de la présentation des principales solutions commerciales et académiques permettant l'agilité des processus collaboratifs

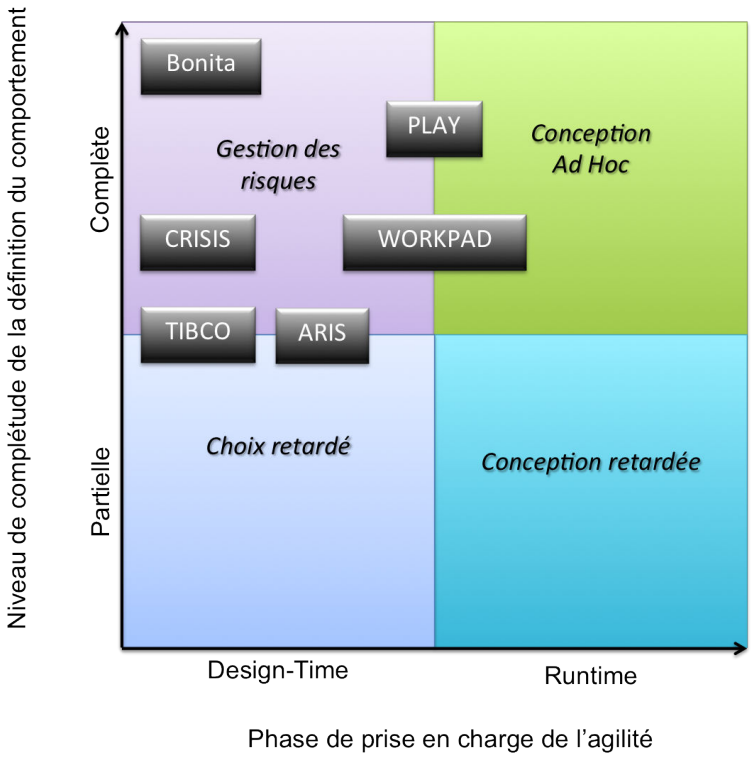


FIGURE II.6 – Positionnement des solutions étudiées dans le référentiel d'adaptation du comportement, [Bénaben, 2012] d'après [Schonenberg et al., 2008]

L'examen des travaux et outils de détection et d'adaptation a montré la diversité des solutions adoptées. En outre, il a mis en évidence certaines limites de ces solutions académiques et commerciales. Par exemple, l'approche déterministe par combinaison de fragments de processus occupe une place importante dans les techniques d'adaptation retenues. Or si nous retenons le fait que nous nous positionnons au niveau d'une adaptation ad-hoc, ainsi que définie par Schonenberg, à l'heure actuelle aucune solution ne satisfait pleinement les critères posés par la définition de l'agilité (voir Figure II.6). Deux solutions académiques (PACHUBE et PRONTO) n'apparaissent pas dans le positionnement dans le cadre de Schonenberg du fait de leur incapacité à prendre en charge la composante d'adaptation de l'agilité, malgré l'implémentation de la détection de situations nécessitant une prise de décision (telle qu'une adaptation des processus).

Il n'y a pour le moment pas de produit académique ou commercial qui propose les fonctions de conception, d'exécution et d'adaptation sensible au contexte des processus collaboratifs. Cependant certains proposent des pistes de réflexion quant au recueil des données et à leur analyse en vue de déterminer le besoin d'adapter les processus collaboratifs (essentiellement les projets PACHUBE, PRONTO et PLAY). Le projet WORKPAD se concentre plus sur la mise en œuvre matérielle via la reconfiguration du réseau que sur la réconfiguration des processus au sens métier : bien qu'intéressant, il nous semble moins pertinent vis-à-vis du sujet de ces travaux de thèse.

II.2 Étude par rapport aux moyens

II.2.1 Le recueil automatisé des données

Comme nous avons pu l'introduire dans le Chapitre I, le recueil des données est un enjeu majeur dans la réponse à la problématique de l'agilité. En effet, il s'agit de collecter toutes les données jugées pertinentes afin de pouvoir en extraire par la suite l'information nécessaire pour déclencher et guider la prise de décision. Si la connaissance de l'évolution du théâtre de la collaboration et de la collaboration restent inconnus, il sera impossible de détecter un besoin d'adaptation et de déterminer l'amplitude de l'adaptation à mener.

L'exhaustivité des données, la qualité des données, la rapidité avec laquelle elles sont récoltées et la qualité de l'analyse effectuée sont autant de facteurs qui influenceront la décision finale. Le rôle croissant des événements en tant que sources de données dans les applications d'entreprise apparaît clairement dès 2003, dans un rapport de Gartner [5] [Schulte, 2003].

II.2.1.1 Les événements

Tout changement, toute évolution, toute donnée qui peut influencer la pertinence des processus collaboratifs en cours d'exécution doit être collecté et analysé. D'après [Chandy and Schulte, 2009], [EPTS, 2008], [Etzion and Niblett, 2010] et [Luckham and Schulte, 2008] ces éléments qui adviennent et qui constituent des données peuvent être considérés (et gérés) comme des événements. Produits par le théâtre de la collaboration (ou terrain) mais également la collaboration elle-même, ces événements sont collectés via des mécanismes tels que ceux introduits dans le Chapitre I et détaillés dans le Chapitre III. Ils serviront à mettre à jour les modèles *terrain* et *attendu* de la situation collaborative.

[Morley et al., 2007] offre une vision de l'événement très attachée au domaine de la définition de processus métiers. Il s'agit pour elle d'un élément déclencheur d'une activité : l'événement est matérialisé par une information afin d'informer l'auteur de l'activité dudit événement. Le résultat d'une activité peut être soit un événement, soit une ressource pour une autre activité. Dans notre cas, nous envisageons l'événement comme une information dont les conséquences sont inconnues pour le producteur de l'événement. La consommation d'un événement peut engendrer l'émission d'un nouvel événement sans pour autant entraîner l'exécution d'une activité au sein du processus. Par exemple, la combinaison de deux événements « taux de radiation qui augmente de 20% » et « vent fort » consommé par un composant logiciel peut entraîner la création d'un nouvel événement « risque de contamination de la population ». Aucune activité d'un quelconque processus n'a été déclenchée dans ce cas. Autrement dit, la production d'un événement ne va pas de pair avec le déclenchement d'une activité.

5. Gartner Inc. est une entreprise américaine de conseil et de recherche dans le domaine des techniques avancées. Elle mène des recherches (portfolio de plus de 200 sujets), fournit des services de conseil, tient à jour différentes statistiques et maintient un service de nouvelles spécialisées.

La définition de l'événement que nous retiendrons est celle proposée Chandy, Etzion et Luckham. Il s'agit de la plus pertinente dans le cadre de ces travaux de thèse, afin d'atteindre notre but qui est la retranscription des évolutions se produisant non seulement au sein des processus collaboratifs mais également dans leur environnement d'exécution. Cette définition s'inscrit au-delà du cadre dans lequel se positionne Morley et où l'événement sert de marqueur d'état à l'activité métier. Cependant, nous continuerons à étudier le point de vue de Morley afin de compléter notre vision des événements, de leurs types et de leurs natures.

II.2.1.2 Les différents types d'événements

Dans la littérature, nous pouvons trouver différentes façons de typer les événements, de par leur nature et leur(s) effet(s) essentiellement. [Morley et al., 2007] type un événement par deux spécialisations (chacune exclusive) : *sa nature* et *son effet*.

La nature d'un événement est une spécialisation (exclusive) de l'événement, qui doit être prise en compte lors de la définition du processus :

- L'événement temporel. Il correspond à une échéance unique ou périodique. À cette échéance, on associe une réaction de l'organisation. Exemple : le versement des salaires le 26 de chaque mois,
- L'événement interne. Il résulte d'un choix d'organisation, d'une décision prise par un acteur de l'organisation. Par définition, on a une certaine maîtrise sur sa définition et son éventuelle modification. Il correspond à un stimulus généré à l'intérieur du périmètre du processus (ou dans le périmètre du domaine dans lequel s'inscrit le processus). Exemple : La décision d'un acteur qui provoque le déclenchement d'une activité est un événement interne,
- L'événement externe. Il s'agit d'un stimulus sur lequel on n'a pas (ou plus) de prise et qui est lié à un acteur ou à un système extérieur à l'organisation. L'événement externe nécessite d'être reconnu et d'être associé à une activité afin d'être pris en compte dans le processus. Exemple : si un système de vente en ligne n'a pas prévu de prendre en compte une réclamation client, la réclamation ne constitue pas un événement à traiter dans le cadre d'un processus. Elle sera traitée au cas par cas, de façon improvisée par les acteurs.

L'effet quant à lui est provoqué par l'événement et déclenche l'activité correspondante. Il peut être pris en compte au cours du déroulement de l'activité, conduisant à une deuxième spécialisation de l'événement suivant son effet sur l'activité (spécialisation considérée comme exclusive d'après [Morley et al., 2007] :

- L'événement déclencheur : il provoque l'exécution de la première tâche d'une activité [6],
- L'événement interrupteur : il conduit à interrompre une activité et à en sortir, même si toutes les tâches n'ont pas été effectuées. Dans ce cas, d'autres tâches

6. [Morley et al., 2007] définit l'activité comme un ensemble de tâches. La tâche est la décomposition la plus atomique de l'activité. Par exemple, l'activité « Préparer un cours » comprend les tâches « Réserver salle », « Affecter intervenant », « Reproduire supports de cours »

peuvent être effectuées,
- L'événement modificateur : il fait changer le cours du processus, conduisant à effectuer des tâches particulières ou à en ignorer certaines de l'activité (ces tâches sont donc soumises à condition).

Cette approche du type d'un événement est toujours très fortement liée au point de vue processus et à la frontière de l'organisation. L'événement ne peut être défini que dans le contexte de son utilisateur final, sa nature étant déterminée par sa place dans les frontières de l'organisation et son effet par rapport à l'activité à laquelle il est attaché. De plus, nous pouvons noter que cette définition est très limitante du fait de sa dépendance à l'activité et au processus. Par exemple, elle ne prévoit pas qu'un événement généré par un acteur de l'organisation (événement interne) soit imprévu (événement externe) : l'exclusivité de la nature de l'événement l'en empêche.

[Luckham and Schulte, 2008] a une vision plus globale du type d'événement. Il le définit comme une classe d'objets *événement* (exemple : le type de toutes les sorties d'un capteur pour n'importe quelle sorte de capteur). Tous les événements doivent être des instances d'un type d'événement. Un événement possède une structure définie par son type. Cette structure est représentée par un ensemble d'attributs (ou propriétés) d'événement. Cette définition rejoint celle proposée par [Etzion and Niblett, 2010].

[Luckham and Schulte, 2008] recommande que les types d'événements soient définis à l'aide de Schémas XML (.xsd) ou du langage Java. Au-delà de ces préconisations, [Etzion and Niblett, 2010], [Luckham and Schulte, 2008] et [Michelson, 2006] soulignent le fait que quel que soit le standard utilisé pour représenter les événements le type d'événement devrait posséder *à minima* certaines données prédéfinies comme :
- Un en-tête, contenant des meta-informations sur l'événement :
  - Un identifiant unique d'événement permettant de référencer l'événement,
  - Le type de l'événement,
  - L'horodatage de la création de l'événement,
  - La source d'origine de l'événement (propriété également évoquée par [Morley et al., 2007]).
- Un corps, contenant les données propres à l'événement. Ces données sont structurées suivant le type de l'événement.

L'étude de la définition et du typage des événements peut être faite suivant deux points de vue : le *processus*, défendu par [Morley et al., 2007] où l'événement marque un état d'une activité et est dépendant de l'organisation, et l'*environnement* (pouvant inclure le processus), proposé par [Chandy and Schulte, 2009], [EPTS, 2008], [Etzion and Niblett, 2010] et [Luckham and Schulte, 2008]. Dans ce dernier cas, l'événement représente toute donnée émise par un composant de la collaboration mais aussi de son environnement. Le type de l'événement ne dépend pas d'un quelconque lien à une activité ou d'une frontière organisationnelle. Un événement émis dans l'organisation et un événement émis à l'extérieur de l'organisation pourront avoir le même type. De plus, n'étant pas attaché à une activité donnée, l'événement n'a pas d'effet unique et prédéfini. Le consommateur de l'événement garde ainsi sa liberté de traitement et d'interprétation

des données portées par l'événement. Cette deuxième définition est beaucoup plus couvrante (compte tenu de nos objectifs) et permet de croiser les données provenant de sources différentes (théâtre de la collaboration, collaboration elle-même, etc.).

II.2.1.3 Le traitement des événements

Le traitement des événements (ou event processing) est un ensemble de méthodes et d'outils qui réalise des opérations sur les événements, incluant la lecture, la création, la transformation et la suppression d'événements en vue de réagir à des conditions changeantes, généralement sous certaines contraintes de temps.

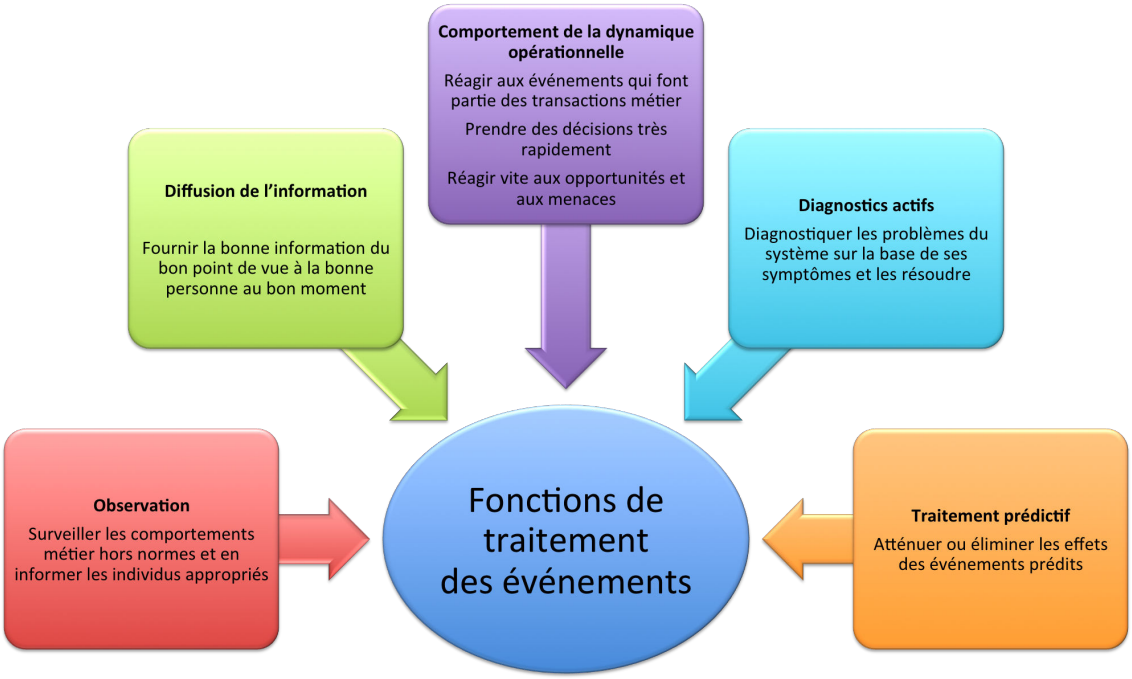


FIGURE II.7 – Les cinq catégories de traitement des événements [Etzion and Niblett, 2010]

Les types d'applications de traitement des événements peuvent être classés en cinq catégories (Figure II.7) [Etzion and Niblett, 2010] :

Observation. Le traitement des événements est utilisé pour surveiller un système ou un processus en traquant les comportements anormaux et en générant des alertes quand de tels comportements se produisent. Dans ce type de cas, la réaction à l'alerte, si elle existe, est laissé à la charge des individus ou systèmes consommant

ces alertes. Le rôle de l'application de traitement des événements se résume à produire des alertes. Exemples : un système de surveillance médicale d'un patient, un système d'acheminement des bagages dans un aéroport, un système de surveillance de zones à accès restreint dans une usine.

Diffusion de l'information. Une autre des raisons de l'utilisation du traitement des événements est la livraison de la bonne information au bon individu/système au bon moment suivant le bon point de vue (ou granularité). En d'autres termes, il s'agit d'une délivrance personnalisée de l'information. Exemples : un portail bancaire personnalisé, un système d'urgence qui contacte d'abord les équipes de premiers secours.

Comportement de la dynamique opérationnelle. Le traitement des événements est souvent utilisé pour diriger dynamiquement les actions réalisées par un système, de façon à réagir aux événements entrants (orchestration). Les sorties de l'application sont directement affectées par les événements entrants. Exemple : un système d'enchères en ligne.

Diagnostics actifs. Le but de l'application de traitement des événements est de diagnostiquer un problème, sur la base de l'observation des symptômes. Exemples : un système de détection de panne mécanique (basé sur le nombre et le type de défauts des pièces produites) dans une usine, un centre d'assistance (help-desk) qui va sur la base de la description des symptômes par l'utilisateur remonter à l'origine des problèmes.

Traitement prédictif. Il s'agit là d'identifier les événements avant qu'ils ne se produisent. Ainsi ils peuvent être éliminés et/ou leurs effets réduits. Exemples : les systèmes de détection de fraude, les systèmes utilisés par les traders pour passer leurs ordres.

Notons qu'une application de traitement des événements peut relever d'une ou de plusieurs de ces catégories à la fois. Quelle que soit la ou les catégories dont relève l'application de traitement des événements, elle doit réaliser un ensemble de fonctions élémentaires. Les systèmes de traitement d'événements réalisent les quatre fonctions principales suivantes (Figure II.8) :

1. **Recueillir**. Récupérer les données issues de multiples sources, en temps réel autant que possible,
2. **Détecter**. Agréger et analyser ces données pour détecter des motifs qui indiquent la présence de situations critiques requérant une réponse,
3. **Décider**. Déterminer la meilleure réponse pour de telles situations,
4. **Réagir**. Exécuter et suivre la réponse.

II.2.1.4 Les agents de traitement d'événements

Un EPA (Event Processing Agent, pour Agent de Traitement d'Événements) est un composant logiciel dont le rôle est de traiter les événements qu'il reçoit et d'en produire

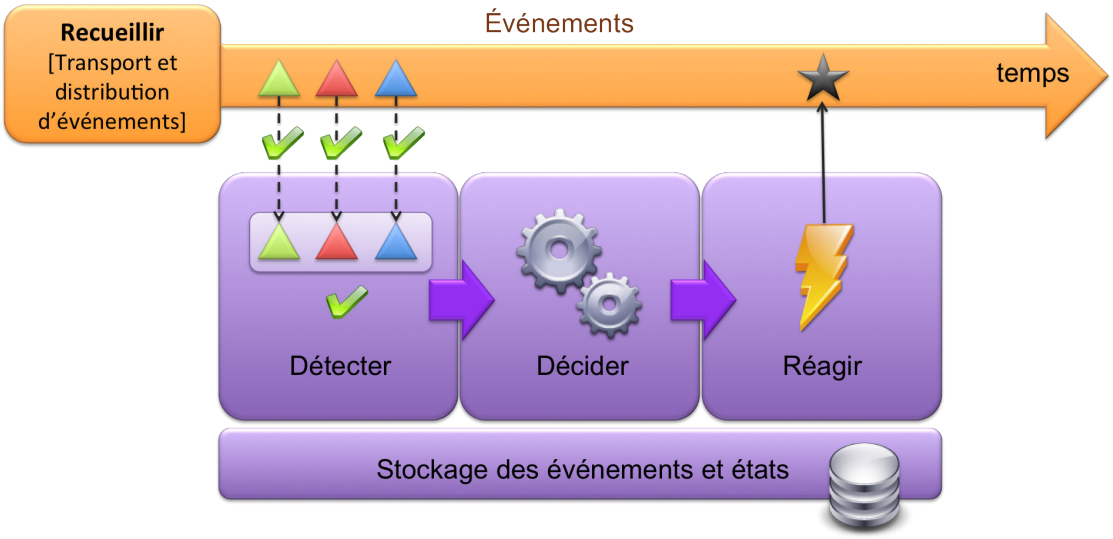


FIGURE II.8 – Les quatre fonctions principales du traitement des événements [TIBCO, 2011]

de nouveaux (résultant du traitement des événements entrants). A ce titre, un EPA est à la fois un consommateur et un producteur d'événements. C'est un composant de l'EDA. D'après [Etzion and Niblett, 2010] les EPA peuvent être classés en trois grandes familles suivant leur type :

- Filtrage (Filter) : ils filtrent les événements,
- Transformation (Transformation) : ils transforment les événements,
- Détection de motif (Pattern detect) : ils détectent des motifs d'événements.

Filtre Une application de traitement des événements met en jeu de nombreux producteurs d'événements qui génèrent un volume très important d'événements, qui ne sont pas forcément pertinents pour l'application, le domaine métier ou l'instant considéré. Un EPA de type *Filtre* peut être utilisé pour réduire ce volume en excluant les instances d'événements non désirées.

Un EPA Filtre (EPA Filter) est un EPA qui ne réalise que du filtrage et qui ne fera aucune activité de correspondance ou de dérivation (i.e. il ne transforme pas l'événement entrant). Il possède une expression filtre qui détermine quelles sont les instances d'événements sélectionnées (filtered-in) par l'EPA et celles qui sont rejetées (filtered-out) ou encore non interprétables (non-filterable).

Les trois sorties possibles d'un EPA Filtre sont donc les suivantes (Figure II.9) :

- Filtered-in : tout événement entrant qui satisfait à l'expression de filtrage passe par cette sortie,
- Filtered-out : tout événement entrant dont l'expression peut être évaluée, mais qui ne satisfait pas à l'expression de filtrage est dirigé vers cette sortie,
- Non-filterable : tout événement entrant dont l'expression ne peut être évaluée est

dirigé vers cette sortie.

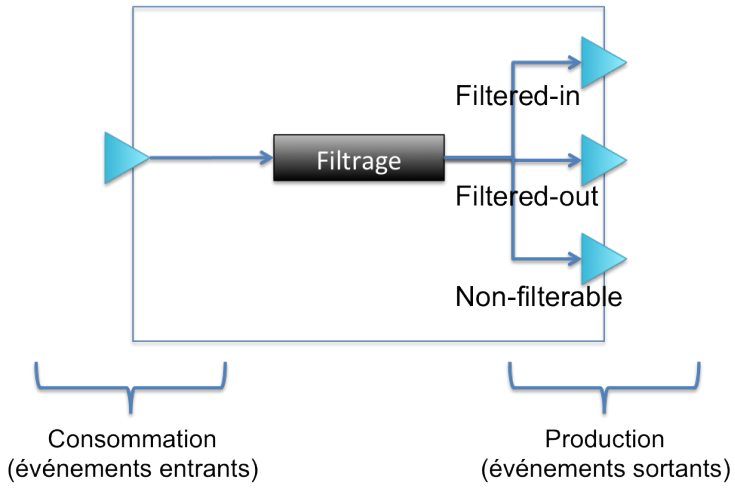


FIGURE II.9 – L'EPA Filtre [Etzion and Niblett, 2010]

Transformation Les agents de traitement des événements par transformation prennent un ou plusieurs événements en entrée et créent différents événements en sortie sur la base de ceux d'entrée. Ces événements sortants sont appelés événements dérivés. Exemples d'événements dérivés :

- Un événement rapportant qu'une entreprise B a conclu son offre de reprise de l'entreprise A avec une probabilité de 90% peut être dérivé de l'événement informant que le prix des actions de l'entreprise A a bondi de 15% dans les 5 dernières minutes,
- L'absence d'un événement attendu dans un laps de temps donné peut être considérée comme un événement. Cet événement dérivé va rapporter le fait que le premier événement évoqué n'a pas eu lieu.

Un EPA Transformation est un EPA qui inclut toujours une étape de dérivation et éventuellement une étape de filtrage (Figure II.10). Cette transformation peut être :

- **sans état**, c'est-à-dire que chaque événement entrant est traité indépendamment des instances d'événements précédentes et que chaque événement sortant est dérivé d'un unique événement entrant.
- **à états** : dans ce cas un événement sortant peut être dérivé de plusieurs événements entrants.

La Figure II.11 résume les différents types d'agents de transformation existants. Les différents types d'agents de transformation peuvent être classés suivant le type de la transformation (sans état ou à états) :

- Sans état : Projeter, Traduire, Enrichir,
- A états : Aggréger, Séparer, Composer.

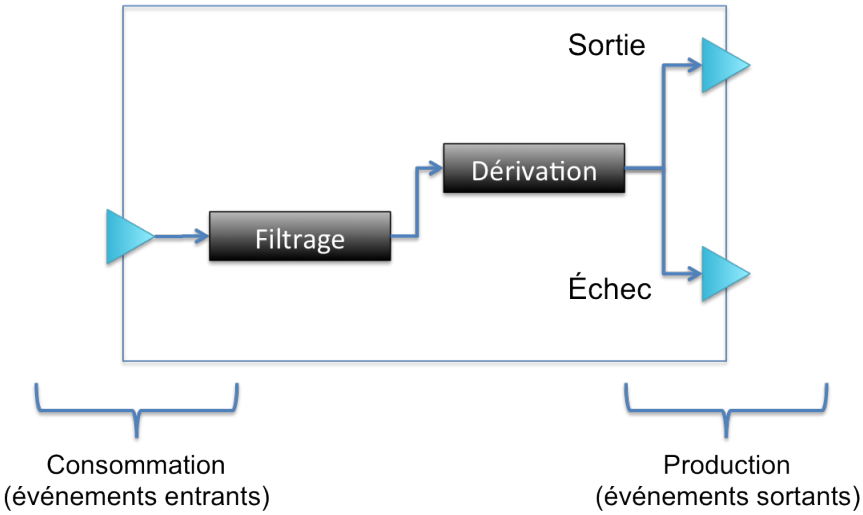


FIGURE II.10 – L'EPA Transformation [Etzion and Niblett, 2010]

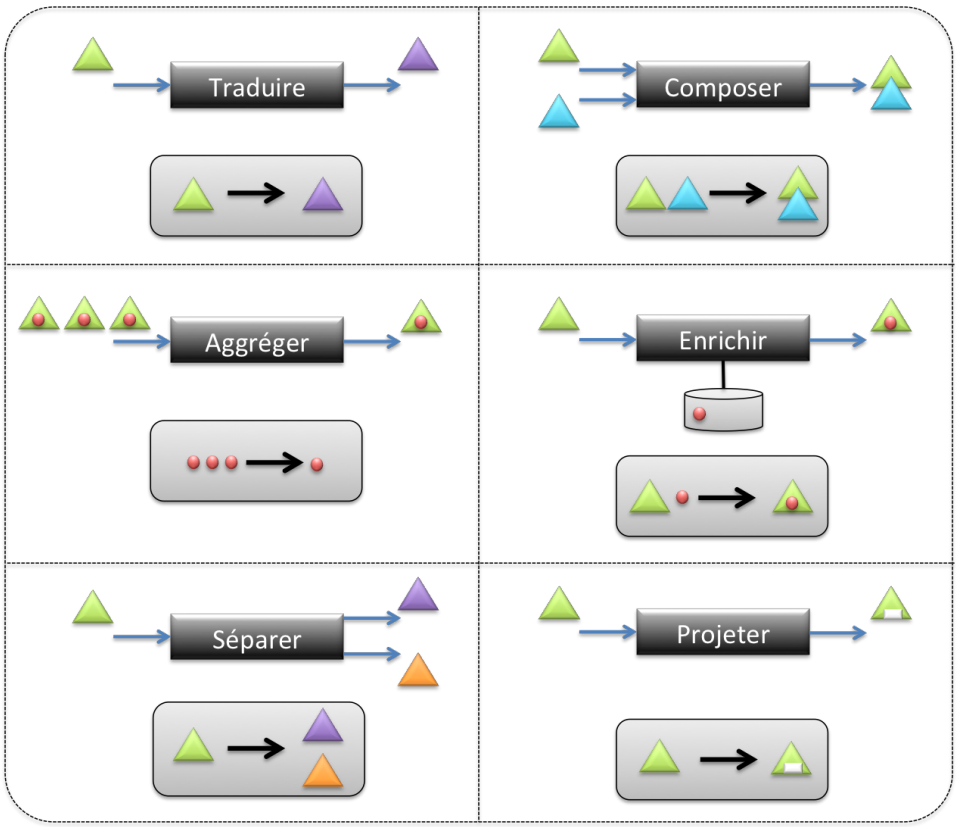


FIGURE II.11 – Classification des différents types d'agent de transformation : les triangles représentent les instances d'événements, les flèches montrent le nombre de flux entrants et sortants par type d'événements [Etzion and Niblett, 2010]

Détection de motif Un EPA Détection de motif (EPA Pattern detect) est un EPA qui réalise une fonction de correspondance de motif d'événements sur un ou plusieurs flux d'événements entrants. Il émet un ou plusieurs événements dérivés s'il détecte une occurrence du motif spécifié dans les événements entrants.

La structure logique d'un EPA Détection de motif (Figure II.12) est composée de trois parties : le filtrage, la correspondance et la dérivation. Le filtrage permet à l'EPA de ne sélectionner que les événements pertinents en analysant leurs types. Ces événements deviennent alors des candidats à la correspondance. L'EPA vérifie sur les événements candidats vérifient les motifs d'événements (définis par l'utilisateur). Si un motif est vérifié, alors l'EPA exécute l'étape de dérivation. Cette étape va émettre un ou plusieurs événements dérivés.

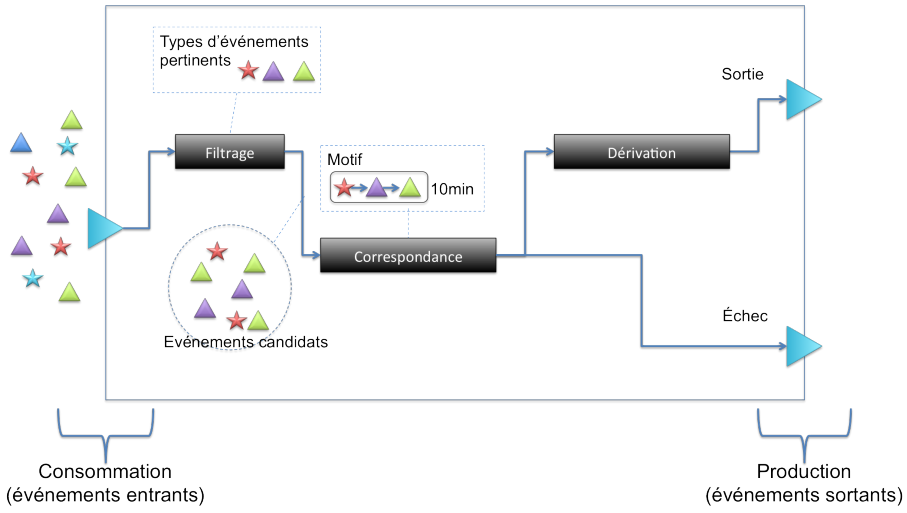


FIGURE II.12 – L'EPA Détection de motif d'événement [Etzion and Niblett, 2010]

Il est intéressant de noter que les motifs d'événements peuvent intégrer une dimension temporelle, à l'aide de fenêtre de temps. Ces fenêtres permettent de persister (au niveau de l'EPA) les événements reçus durant le temps indiqué et de réaliser des motifs plus complexes faisant appel à l'analyse de l'ordre d'arrivée, à des opérations (calcul de la moyenne, recherche du minimum/maximum, etc.). Par exemple, un motif d'événements permet de surveiller l'augmentation du taux de radioactivité dans l'air ambiant en faisant la moyenne de tous les taux reçus via les événements de type *mesure de radiation* dans les *10 dernières minutes*. Si cette moyenne dépasse un certain seuil, un événement de type *alerte* est émis par l'EPA. Deux stratégies temporelles peuvent s'appliquer [7] :

- La fenêtre fixe : pendant 10 minutes, l'EPA garde tous les événements candidats. Au bout de 10 minutes, il calcule la moyenne (et suivant le résultat, envoie ou non une alerte). Puis il vide sa mémoire, réinitialisant ainsi la fenêtre,
- La fenêtre glissante : le calcul de la moyenne s'effectue en continu sur les évé-

7. Notons que l'implémentation de ces stratégies temporelles ainsi que des opérations (moyenne, minimum, maximum, etc.) peut varier suivant les solutions logicielles.

nements candidats reçus par l'EPA (à l'étape de correspondance) dans les 10 dernières minutes. Tout événement candidat présent depuis plus de 10 minutes dans l'EPA est supprimé.

II.2.1.5 L'architecture orientée événements

L'EDA (Event Driven Architecture, pour Architecture Dirigée par les Événements) est un type d'architecture de SI dans lequel certains composants sont dirigés par les événements et communiquent au moyen des événements. Les architectures EDA ont été popularisées avec l'apparition de standards pour les places de marchés et les systèmes de vente aux enchères.

De par sa nature, une architecture dirigée par les événements est couplée de façon extrêmement lâche et est hautement distribuée [Michelson, 2006] [Josuttis, 2007]. Le créateur (source) de l'événement sait seulement que l'événement est apparu. Le créateur n'a aucune connaissance du traitement ultérieur de l'événement ou des parties intéressées, dans le sens où il ne sait pas quels sont les consommateurs finaux de l'événement qu'il a généré. D'autre part, les consommateurs de l'événement s'abonnent aux types d'événements qui les intéressent : or les instances de ces types d'événements peuvent être des événements dérivés. Les consommateurs n'ont aucun moyen de savoir si l'événement qu'ils consomment est un événement « original » simplement filtré ou s'il est dérivé. La traçabilité d'un événement à travers un réseau multi événements dynamique peut être difficile dans ce contexte. Ainsi, les architectures dirigées par les événements sont mieux utilisées pour les flux asynchrones de travail et d'information. Le couplage entre services est un couplage lâche et les communications sont toutes asynchrones.

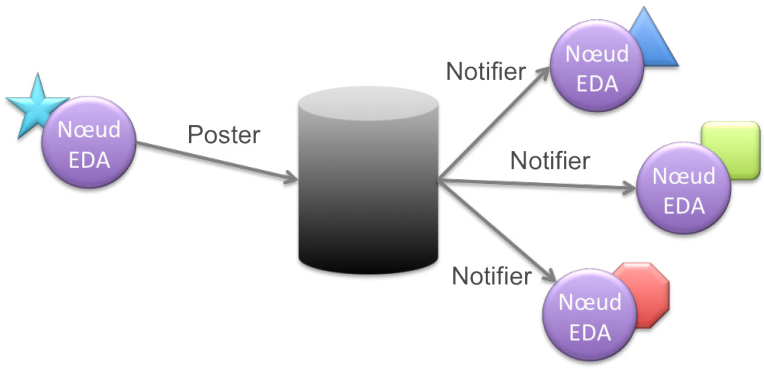


FIGURE II.13 – Principe du mécanisme Publication/Souscription [Maréchaux, 2006]

L'EDA utilise les échanges de messages peut permettre à deux processus ou plus de communiquer ensemble. La communication est initiée par un événement. Ce déclencheur correspond typiquement à une occurrence métier quelconque. Tous les abonnés au type d'événement correspondant à l'instance émise sont alors notifiés et donc activés. C'est ce qu'on appelle le mécanisme Publish/Subscribe (Publication/Souscription) de l'EDA, représenté dans la Figure II.13.

La Table II.5 résume les propriétés fondamentales d'une architecture dirigée par les événements.

Propriété	**Description**
Couplage lâche	Les producteurs (sources) d'événements ne sont pas au courant de l'existence de leurs abonnés
Communication n-m	Le mécanisme Publication/Souscription permet qu'un seul événement spécifique puisse avoir un impact sur plusieurs abonnés
Déclencheur basé sur les événements	Le flux de contrôle est déterminé par le destinataire, sur la base de l'événement reçu
Asynchrone	Les opérations asynchrones sont rendues possibles via les événements

TABLE II.5 – Principales caractéristiques de l'EDA

Le Complex Event Processing Le CEP (Complex Event Processing, pour Traitement d'Événements Complexe) est une technologie de réseau émergente [8] qui déclenche des actions et créé de la connaissance à propos de la situation à partir de systèmes distribués basés sur les messages, de bases de données et d'applications en temps réel. Le CEP s'occupe de l'évaluation d'une confluence d'événements et d'une prise de mesures, sur des périodes temporelles (ou fenêtres) de durées variables. Cette corrélation d'événements peut être causale, temporelle, ou spatiale. Le CEP requiert l'emploi d'interprètes d'événements sophistiqués, de définition de motifs et de correspondance d'événements. Le CEP est communément utilisé au sein des organisations pour détecter et répondre aux anomalies métiers, aux menaces et opportunités en surmontant la complexité et l'hétérogénéité de leurs réseaux.
Notons que le CEP fait généralement référence au traitement des événements qui suppose un nuage d'événements [9] comme entrée, et ne peut donc faire aucune hypothèse sur l'ordre d'arrivée des événements, contrairement à un flux d'événements [10].

La Figure II.14 résume les principaux concepts du CEP [Fülöp et al., 2010]. Les mêmes concepts peuvent s'appliquer à l'Event Stream Processing, en remplaçant le nuage d'événements par un flux continu d'événement, et un traitement continu des événements. Le traitement des événements s'applique sur un Système d'Information, qui est capable de recevoir des événements. Ces événements proviennent du monde extérieur à travers des capteurs, un ERP, une CRM, etc. (exemples : décollage d'un avion, arrivée d'un

8. Définie pour la première fois comme telle par David Luckham dans les années 90.

9. Un nuage d'événements est un ensemble d'événements partiellement ordonné, qu'il soit borné ou non, où les ordres partiels sont imposés par la causalité, le temps ou tout autre relation entre les événements.

10. Un flux d'événements une séquence ordonnée et linéaire d'événements (de type potentiellement différents). Habituellement, les flux sont ordonnés par le temps, par exemple l'heure d'arrivée.

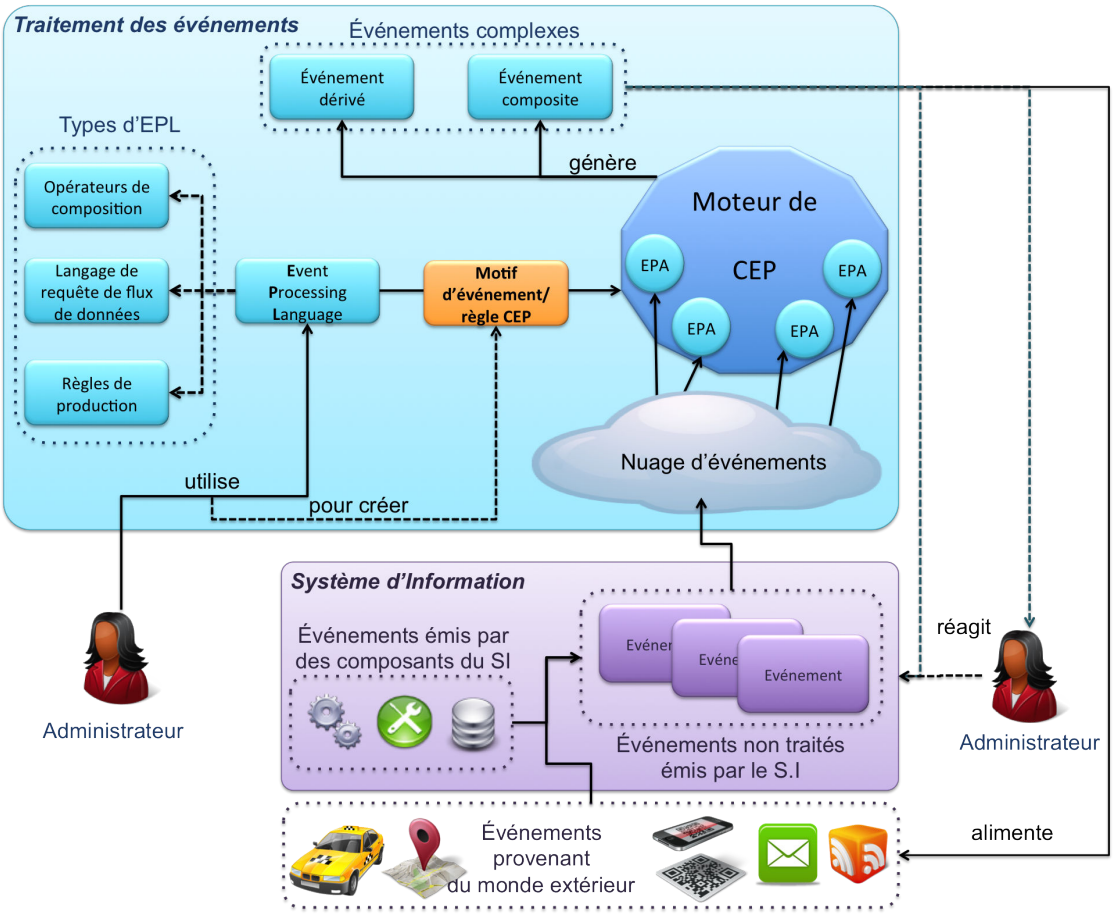


FIGURE II.14 – Concepts de CEP d'après [Fülöp et al., 2012]

bus, confirmation d'une commande, une alerte aux orages dans une prévision météorologique, etc.) mais aussi du SI lui-même (exemples : réalisation d'une activité, production d'un rapport, etc.). Ces événements sont récupérés sous forme de nuage par les EPA qui vont (par exemple) les filtrer avant de les transmettre au moteur qui va traiter les événements. Si un motif d'événements ou une règle sont satisfaits par les événements entrants, le moteur le notifie à l'administrateur (ou à un composant du SI) qui réagira en conséquence. Cette notification (dans notre cas) se fait grâce à l'émission de nouveaux événements émis par le moteur de CEP, suite au déclenchement des motifs d'événements ou des règles.

L'intérêt de la mise en place d'une architecture EDA faisant appel à un moteur CEP est donc de pouvoir découvrir de nouvelles données en identifiant les événements significatifs dans le nuage d'événements générés par le SI et le monde extérieur. Par déduction, analyse et corrélations de ces événements, le CEP peut créer des événements complexes qui mettent en valeur cette découverte de données. Ces événements sont dits complexes

car on ne peut pas les détecter directement dans la situation observée : c'est la combinaison d'autres événements qui permet de les révéler.
Par exemple, dans une forêt, plusieurs stations de mesures météorologiques sont disséminées : elles mesurent la température de l'air ambiant, la vitesse et la direction du vent, et ont chacune une position GPS précise. En début de matinée, 10 capteurs situés au nord de la forêt génèrent les événements : « Température air ambiant supérieure à 80 degrés Celsius », « Vent fort direction sud-ouest ». La combinaison de ces événements —température anormale dans le secteur nord à cette heure de la journée, vent fort— permet de détecter deux nouveaux événements qui sont « Incendie de forêt au nord » et « Risque de propagation au sud-ouest ». Ces deux événements sont des événements complexes. Même s'il n'y a pas d'observation directe du fait que les flammes commencent à dévorer le nord de la forêt (comme pourrait le permettre une observation par un être humain sur les lieux, une photo, une vidéo surveillance, etc.), il a été possible de déduire la situation par la combinaison d'autres événements et de faire état de cette situation en générant des événements complexes.

Ces événements complexes sont mis à disposition des parties intéressées (les utilisateurs et le SI) qui vont exploiter ces données afin de réagir en temps réel.

L'EDA et le CEP trouvent donc tout naturellement leur intérêt dans le cas de nos travaux. La capacité de déduction d'une nouvelle situation à partir d'événements simples permet d'obtenir des données plus intéressantes et pertinentes au regard des besoins des utilisateurs et du SI concerné. Cet apport de nouvelles données qui sont de fait filtrées, triées, analysées permet d'alimenter les représentations de la situation collaborative à un instant t, à travers la mise à jour des modèles *terrain* et *attendu*.

Les Event Processing Languages et les moteurs associés Qu'il s'agisse d'ESP ou de CEP, habituellement un EPE (Event Processing Engine, pour Moteur de Traitement des Événements) est utilisé dans l'infrastructure. Les événements sont fournis à l'EPE par les EPA, qui vont par exemple filtrer les événements. Ensuite, l'EPE va traiter les événements et éventuellement notifier à l'administrateur de l'EPE l'existence de correspondances avec des motifs ou des règles (établies au préalable par l'administrateur). Ce motif ou cette règle peut être définie dans un langage de traitement des événements et via lequel on peut s'abonner à certains motifs d'événements. Quand il est notifié par l'EPE de la détection de certains motifs d'événements, l'administrateur exécute les actions appropriées sur le système d'information. Ces actions peuvent être automatisées en utilisant le modèle d'ECA (Event Condition Action), dans lequel l'action est automatiquement réalisée sur le système d'information si la condition est vraie.

Les EPAs peuvent être exprimés dans de nombreux EPL (Event Processing Language, pour Langage de Traitement d'Événements). A l'heure actuelle, de nombreux types d'EPL coexistent. On y trouve [Etzion and Niblett, 2010] :

- Les langages de règles. Ici, plusieurs types de règles sont recouverts par ces langages :

  - Les règles de production, ou d'inférence [Vianu, 1997]. Ce type de règles est utilisé pour représenter les comportements de type *SI condition ALORS action*. Les règles s'enchaînent en cascade et la conclusion (action) résultant d'une règle n devient la condition de la règle $n + 1$ (principe du chaînage avant). Si ces règles sont normalement basées sur les changements d'état et non sur les événements, certains langages les étendent et permettent le traitement des événements. Les événements sont alors modélisés comme des parties des conditions des règles. Le traitement des événements se fait à travers un processus d'inférence. Exemple : Drools Rule Language (qui se base sur l'algortihme RETE), TIBCO Business Events.
  - Les règles actives, aussi connues sous le nom de règles ECA (Event Condition Action). Les moteurs de règles actives détectent et réagissent aux événements entrants et exécutent des motifs d'événements. Exemple d'EPL basé sur les règles ECA : IBM WebSphere Business Events, RAPIDE,
  - Les règles basées sur la programmation logique. La programmation logique est un style de programmation basé sur les assertions logiques (Prolog est le langage de ce type le plus connu). Exemple d'EPL relevant de cette catégorie : ETALIS.
- Les langages de programmation impératifs : les opérations sont décrites en termes de séquences d'instructions élémentaires exécutables par le processeur pour modifier l'état du programme. Il s'agit du type de programmation le plus répandu parmi l'ensemble des langages existants. Exemples : FORTRAN, COBOL, BASIC, Pascal, C, Ada, Perl, Python, PHP, Java (liste non exhaustive). Ces langages peuvent servir d'EPL,
- Les langages orientés flux qui sont des extensions de SQL (et souvent vaguement basés sur SQL), aussi appelés *Streaming SQL*. Ils permettent d'effectuer des requêtes sur les flux d'événements. Exemples : Esper EQL (Event Query Language), Oracle CQL (Continuous Query Language), CCL (Continuous Computation Language).
- D'autres langages orientés flux mais non typés SQL. Exemple : SPADE ([Gedik et al., 2008]).

Les langages de type Streaming SQL représentent la plus grand part des EPL à l'heure actuelle.

Le choix d'un EPL n'est pas trivial. Si certains spécialistes tels Luckham persistent et arguent qu'un seul EPL est valable parmi ceux existants (ex. : Rapide a la préférence de Luckham) [11], d'autres spécialistes tels qu'Etzion sont beaucoup plus nuancés et préfèrent penser que le choix qu'un EPL dépend essentiellement des besoins et des contraintes de l'environnement concerné par le choix de l'EPL [Etzion, 2008]. Il faut également que l'EPL soit compatible avec un moteur de CEP. Cela va relativement de soi quand on choisit un EPL commercial (on achète le moteur de CEP qui est livré avec son EPL),

11. On peut douter de l'impartialité de Luckham dont le choix est quelque peu biaisé puisque Luckham a dirigé le groupe de travail de Stanford qui a développé Rapide.

mais est moins évident dans le cas des EPL académiques où certaines initiatives n'ont pas été implémentées.

Dans notre cas (travaux de thèse), un EPL open source et gratuit, qui soit compatible avec un CEP lui-même libre et gratuit, et toujours maintenu sont nos critères majeurs de choix. Nous avons effectué un comparatif des solutions existantes (voir Annexe C). En regardant de plus près les différents EPL et CEP cités dans notre comparatif, seul Esper EQL et son implémentation dans Esper Engine répondaient à ces critères au moment du lancement du développement du prototype en 2011. En outre, la possibilité de décrire les événements à l'aide d'objets Java ou de flux XML était un atout appréciable afin de faciliter l'intégration du CEP avec les autres composants du prototype (ESB, abonnement aux événements, etc.).

II.2.1.6 SOA et EDA

Dans le Chapitre I, nous avons positionné notre problématique dans le contexte d'un SI construit suivant le paradigme SOA, qu'il s'agisse des projets européen PLAY ou ANR SocEDA, ou encore du projet interne au laboratoire Centre de Génie Industriel de Mines Albi-Carmaux. La question majeure qui se pose alors est la compatibilité entre les architectures SOA et EDA. L'usage de l'une exclut-elle de facto l'autre ? D'après Maréchaux [Maréchaux, 2006], ce choix n'a pas lieu d'être du moment qu'un ESB (Enterprise Service Bus) est utilisé dans l'architecture du SI, car il permet l'implémentation tant des concepts SOA que des concepts EDA.

L'EDA ne remplace pas, mais au contraire, complète la SOA [Maréchaux, 2006] [Michelson, 2006] [Etzion, 2007] [Josuttis, 2007]. Alors que la SOA est généralement une bonne solution pour les échanges synchrones de type requête/réponse, l'EDA introduit les processus asynchrones de longue durée. Etzion [Etzion, 2007] [Etzion and Niblett, 2010] souligne que la SOA n'est pas définie par les échanges de type requête/réponse. La SOA est avant tout une architecture de modularisation des fonctions de l'entreprise et plus particulièrement de son SI [Etzion, 2007], introduisant ainsi la notion de services. Le mode d'interaction entre les services est secondaire : ils peuvent interagir de manière synchrone ou asynchrone. L'EDA est quant à elle une architecture fortement liée au style de communication publication/souscription, communication qui peut s'appliquer aux services. EDA et SOA sont alors combinées dans ce cas.

Pour [Josuttis, 2007], l'EDA permet également une décentralisation totale du contrôle de l'exécution des services au sein de processus BPM. En cela, l'EDA diffère radicalement de la SOA. Au lieu de considérer les services comme des entités « esclaves » que l'on doit appeler (via un orchestrateur) pour qu'elles s'exécutent, le processus devient un ensemble de traitements séquentiels et/ou parallèles déclenchés par les événements. La fin d'exécution d'un service est marquée par l'envoi d'un événement qui va déclencher l'exécution de tous les services concernés par cet événement, sans qu'un composant tiers —tel qu'un orchestrateur— intervienne. Cet affranchissement vis-à-vis de l'orchestrateur permet d'abaisser encore le couplage entre les services.

Mais là encore, il est possible de combiner les apports de l'EDA et de la SOA en ce qui concerne l'exécution de processus. Le revers du mécanisme asynchrone de publication/souscription utilisée par l'EDA est l'incapacité de l'émetteur à savoir quels sont ses consommateurs et si ces derniers ont bien reçu les événements émis. Que se passe-t'il en cas de panne de l'une ou de l'autre partie ? Dans le cas d'une panne de l'émetteur, une règle de gestion du traitement des événements (côté récepteur) pourrait surveiller l'absence anormale d'événements entrants au-delà d'un laps de temps donné et déclencher une alerte en conséquence. Mais dans le cas d'une panne côté récepteur, l'émetteur ne peut être mis au courant : il continue à envoyer ses événements sans savoir si ces derniers sont consommés. Or la SOA, dès lors qu'elle s'appuie sur les échanges synchrones de type requête/réponse, permet de suivre les échanges de messages et donc d'assurer cette traçabilité.

Si nous nous positionnons dans le contexte d'une collaboration, il paraît tout à fait envisageable de conserver une architecture SOA (avec des échanges synchrones de type requête/réponse) pour orchestrer les services à l'intérieur de chacun des processus intra-organisationnels et de mettre en place une EDA pour chorégraphier les processus inter-organisationnels dans le cadre de la collaboration. Ainsi, chaque organisation est capable de tracer ses échanges en interne (en cas de besoin : panne, sécurité, etc.) et d'orchestrer ses processus métiers propres. La collaboration utilise quant à elle certains des processus de chaque organisation et les combine entre eux afin d'obtenir les processus inter-organisationnels nécessaires. Ceci est réalisable via la chorégraphie apportée par l'EDA : le couplage extrêmement lâche autorise la création de ces processus inter-organisationnels éphémères réalisés par assemblage (via les événements) de briques élémentaires constituées par les processus intra-organisationnels.

Nous retiendrons ici que EDA et SOA ne s'excluent pas mais peuvent se combiner harmonieusement en une architecture ED-SOA [12] (Event-Driven SOA) [Etzion, 2007], où la SOA apporte la notion de services et la possibilité d'assurer une traçabilité des échanges entre services (dans le cadre de l'orchestration de processus) et où l'EDA fournit une capacité de communication beaucoup plus souple entre les processus à travers les échanges et l'analyse d'événements.

La mise en place d'une EDA pourrait amener à compléter la traditionnelle orchestration des services (apportée par la SOA) par la chorégraphie des processus. L'apport principal de la chorégraphie par l'EDA serait l'affranchissement du composant d'orchestration et d'un couplage extrêmement lâche entre les processus (ou les services, suivant la granularité adoptée), caractéristique recherchée dès lors que l'on s'intéresse à l'agilité des

12. On peut trouver plus couramment le terme SOA 2.0 dans la littérature pour décrire une combinaison de l'EDA et de la SOA, mais bien souvent cette combinaison s'arrête à la combinaison des services entre eux et non pas à la combinaison des processus entre eux. Nous avons donc décidé de garder le terme ED-SOA, le trouvant particulièrement adapté à cette notion de chorégraphie (EDA) des orchestrations (SOA).

processus. Nous pouvons ainsi définir une architecture ED-SOA qui permet la chorégraphie des processus inter-organisationnels composés de processus intra organisationnels qui sont quant à eux orchestrés afin de garantir la fiabilité/traçabilité des processus dans l'organisation.

II.2.2 La transformation de données en information

Une fois que les modèles *terrain* et *attendu* sont mis à jour, il faut extraire l'information provenant de la comparaison de leur données. En effet, que l'un ou l'autre des modèles nous apprenne que 18 blessés sont qualifiés d'Urgence Absolue, qu'un hélicoptère est disponible, que les équipes de pompiers viennent d'arriver sur le terrain, etc. ne nous indique rien quant à la pertinence des activités et moyens mis en place pour résoudre la crise à un instant t.

II.2.2.1 Comparer les données

Il faut pouvoir confronter les données du modèle *terrain* avec les données du modèle *attendu*. C'est cette comparaison qui pourra informer les acteurs de la collaboration d'un éventuel écart entre le planifié et le réalisé.

Notre hypothèse de travail est que les modèles *terrain* et *attendu* sont décrits sous forme d'instances du modèle de la crise. Or nos modèles étant des instances d'ontologies au format OWL (OWL Web Ontology Language) [McGuinness and Van Harmelen, 2004], cela signifie que nos modèles sont écrits en XML (eXtensible Markup Language) [W3C, 2008]. Il nous faut donc pouvoir comparer des arbres de données hiérarchisées.

Dans le cas présent, nous recherchons un moyen de comparer des nœuds XML entre eux et de conclure à leur ajout, suppression ou modification entre les deux modèles. Notons que l'ordre des nœuds dans une même branche n'est pas une propriété intéressante dans cette étude. En effet, les fichiers XML des modèles de situation ont vocation à stocker les données : ces arbres peuvent être désordonnés, i.e. l'ordre d'écriture entre des nœuds frères et soeurs importe peu. Par contre l'ordre des relations parent-enfant est toujours crucial. Le cas aurait été différent si ces fichiers XML servait de base à une restitution graphique des données (site web) : l'ordre des relations parent-enfant et l'ordre des relations frères-soeurs auraient été aussi important l'un que l'autre.

Au sens strict, nous cherchons donc les similarités entre les modèles *terrain* et *attendu* plus qu'une identité comme l'indique la Figure II.15.

Dans ce domaine, la littérature scientifique s'est d'abord intéressée à la comparaison de données contenues dans des fichiers plats (type texte). C'est ce qui a amené à la création du bien connu algorithme et utilitaire *diff*. Il est embarqué dès 1974 dans la cinquième distribution d'Unix et un article le présentant est publié en 1976 [Hunt and McIllroy, 1976]. *diff* compare les documents ligne à ligne et relève les différences. Il se base sur l'algorithme de Hunt–McIlroy qui est une résolution d'un problème de plus longue sous-séquence commune (LCS, Longuest Common Subsequence). Son usage premier était

```
<library>
    <book id="1" title="Comment voyager avec un saumon" author="Umberto Eco"/>
    <book id="2" title="Les cavernes d'acier" author="Isaac Asimov"/>
</library>
```

```
<library>
    <book id="2" title="Les cavernes d'acier" author="Isaac Asimov"/>
    <book id="1" title="Comment voyager avec un saumon" author="Umberto Eco"/>
</library>
```

FIGURE II.15 – Similarité entre deux documents XML

dédié au suivi des évolutions du code source des programmes informatiques ainsi que les modifications apportées à des documents techniques.

II.2.2.2 La comparaison d'arbres XML

De nos jours, le succès de XML en tant que format pivot pour l'échange de données a renouvelé l'intérêt des travaux de recherche dans le domaine de la comparaison de données : vérification de l'intégrité de fichiers, gestion de version de documents, surveillance de données, etc. Il s'agit désormais de tirer parti de la nature même des documents XML –à savoir des données organisées de façon arborescente– pour détecter les différences [13] entre deux documents écrits en XML et de ne plus comparer les fichiers ligne à ligne.

Nous en profitons pour noter que lorsque nous parlons de supprimer un nœud d'un document XML, il s'agit de supprimer l'intégralité du nœud : le nœud courant et ses descendants, comme expliqué dans le modèle de Selkow [Selkow, 1977].

Nous avons pu trouver dans la littérature scientifique deux études poussées des différents algorithmes et outils de comparaison de documents XML : l'une menée par [Cobéna et al., 2000], l'autre par [Peters, 2005] (qui actualise l'étude menée par Cobéna). Après avoir éliminé les algorithmes et outils menant une comparaison ligne à ligne, ceux ne permettant pas la comparaison sur des arbres désordonnés et ceux n'étant plus utilisés, nous pouvons étudier les solutions suivantes.

DeltaXML Core DeltaXML Core est un produit commercialisé à l'origine par Monsell EDM Ltd., puis par DeltaXML [DeltaXML, 2013]. Il permet de détecter les différences entre des arbres désordonnés ainsi que de fusionner les documents entre eux, qu'il s'agisse d'arbres ordonnés ou désordonnés. Il permet la détection des opérations d'insertion, de modification, de suppression mais pas de déplacement. Cette dernière n'est pas vitale dans le cadre de nos travaux, d'autant que nous ne nous attachons pas à respecter l'ordre des nœuds au niveau frères-soeurs.

13. Ajout, suppression, modification

En sortie, DeltaXML fournit un fichier δ qui résume des différences détectées entre les deux documents. Bien que puissant [Cobéna et al., 2000], son défaut majeur est le format de sortie du récapitulatif des différences. Il est écrit dans un format propriétaire qui rend l'exploitation du récapitulatif ardu ce qui, associé à sa licence commerciale, rend son usage rédhibitoire dans le cadre de travaux de recherche.

XyDiff XyDiff a été développé par Cobéna au sein de l'INRIA [Cobéna et al., 2004] [Cobéna et al., 2000]. Son algorithme permet de détecter des sous-arbres identiques entre les deux documents ce qui améliore son efficacité en terme de performance et de gestion de la mémoire. Outre la gestion des quatre opérations (ajout, suppression, mise à jour, déplacement), il permet d'ignorer les espaces entre les balises ainsi que les caractères de formatage.

La sortie de XyDiff fournit un fichier XyDelta qui représente les différences entre les deux documents. Le contenu du fichier indique tous les nœuds concernés par une différence et les références en leur affectant un identifiant unique XID. Le fichier de sortie est peu lisible sans l'aide du logiciel et n'en permet pas une exploitation en dehors du cadre de XyDiff.

XMLUnit XMLUnit [Bacon and Martin, 2013] est une extension du framework JUnit [14] [Beck and Gamma, 2013]. Il facilite les tests unitaires portant sur les documents XML afin d'en vérifier l'intégrité au niveau de la structure et du contenu [Glover, 2006]. Ceci est particulièrement utile dans le développement et le test de logiciels qui génèrent du contenu XML, ou qui sérialisent/désérialisent des documents XML.

La comparaison effectuée par XMLUnit entre deux flux XML (contrôle et test) a pour résultat : identique, similaire, différent. L'unité de mesure utilisée par la comparaison est la différence. Les différences peuvent être récupérables ou irrécupérables. Deux flux XML peuvent être :

- Identiques s'il n'y a strictement aucune différence entre eux,
- Similaires s'il n'y a que des différences récupérables entre eux (exemple : l'ordre des nœuds au sens d'un même nœud parent),
- Différent s'il y a la moindre différence irrécupérable entre eux (exemple : absence d'un nœud, valeur différente pour le même attribut d'un même nœud).

XMLUnit permet également de récupérer la liste de toutes les différences identifiées entre le document XML de contrôle et le document XML de test en mémorisant l'expression XPath permettant de remonter au nœud incriminé. La liste des différences est très simplement accessible via un objet et une méthode Java mis à disposition par l'API de XMLUnit.

Le tableau ci-dessous (II.6) récapitule les différentes caractéristiques étudiées dans la couverture des outils de comparaison de documents XML. Il apparaît clairement que XMLUnit est le candidat qui satisfait le plus grand nombre de nos critères de choix.

14. JUnit est un Framework de test unitaire pour le langage de programmation Java.

Solution	**Désordonné**	**Opérations** + - ∼	δ **exploitable**	**Licence**
DeltaXMLCore	Oui	Oui	Non	Commerciale
XyDiff	Oui	Oui (+ déplacement)	Non	Open Source
XMLUnit	Oui	Oui	Oui	Open Source

TABLE II.6 – Synthèse de la couverture des critères pour le choix d'un outil de comparaison de documents XML

II.2.3 La transformation d'information en connaissances

Détecter une ou plusieurs divergences entre les modèles de la situation collaborative n'est pas suffisant pour savoir (i) si une adaptation des workflows ou processus est nécessaire, (ii) à quel niveau cette adaptation doit avoir lieu et à quelle profondeur. Il faut connaître la nature de ces divergences pour pouvoir les identifier (source de la divergence), les classer (laquelle a le plus d'impact sur la collaboration). En effet, savoir que le nombre de véhicules sanitaires légers acheminés sur le terrain n'est pas le même que celui attendu (eût égard au processus de réponse à la crise) ne nous renseigne que sur l'existence de la divergence.

Savoir si le nombre réel de véhicules disponibles est supérieur ou inférieur au nombre attendu peut modifier le comportement de la cellule de crise et l'adaptation des workflows : dans un cas il y a pénurie, dans l'autre excédent. Dans un cas, on cherchera à trouver des solutions alternatives (faire faire des allers-retours aux véhicules, appeler des renforts, etc.), dans l'autre, on pourra remettre en jeu ces ressources excédentaires.

Pour cela, il faut introduire la notion de coût d'une divergence suivant deux axes :

- Sa nature : s'agit-il d'une suppression ? D'un ajout ? D'une modification ? Par exemple : suppression d'un acteur de la cellule de crise, ajout de nouvelles compétences pour un acteur de la cellule de crise, modification d'un risque.
- Sa gravité : il s'agit de qualifier plus finement la divergence. Il paraît en effet évident que l'ajout d'un risque « état grippal » est moins important que l'ajout d'un risque « état grave » à nombre d'individus concernés égal. Il faut arriver à traduire une classification des instances des classes du modèle de la situation collaborative.

Si nous nous tournons vers le domaine de la gestion de projet, nous pouvons nous apercevoir qu'elle intervient alors pour contrôler la continuité des objectifs. Lorsqu'il y a dérive, elle développe les actions d'ajustement en réaction aux changements ou risques avérés. La notion de risque est donc naturellement présente dans tout projet et la gestion des risques va de pair avec la gestion de projet.

L'étude et le management des risques se base principalement sur une approche statis-

tique de la fréquence et de l'importance des dommages d'un risque. De nombreux travaux et normes s'attachent à la phase d'identification des risques, tels que ceux de [Smith, 2007], [ISO, 2009] et [Hopkin, 2010]. Le but de cette phase est d'énumérer les risques du projet : elle constitue une tâche d'extrapolation car il faut anticiper et imaginer les risques que le projet peut être amené à traverser du fait de sa nature et de son environnement. Cette activité nécessite donc la présence d'experts-projet capables de disposer d'une vision globale quant aux données relatives aux sources de risques (les composantes du projet).

Un outil classique de la gestion des risques est la matrice de criticité. La criticité est définie comme le produit de la probabilité d'occurrence d'un accident par la gravité de ses conséquences :

Formule 2 *Criticité = Probabilité × Gravité*

Une explication plus détaillée de la matrice de criticité est disponible en Annexe D.

II.3 Conclusion

L'étude bibliographique par rapport à l'objectif nous a amené à définir la notion d'agilité en nous appuyant sur différents domaines tels que la gestion de chaines d'approvisionnement, la gestion de systèmes d'informations, etc. Les contraintes dues au contexte particulier qu'est la réponse à la situation de crise nous ont tout naturellement amenées à positionner l'adaptation des processus collaboratifs par rapport à une approche de conception ad-hoc.

La connaissance du contexte de la collaboration est essentielle pour suivre les évolutions du théâtre de la collaboration et de la collaboration elle-même. Il est crucial de pouvoir collecter toutes les données pertinentes qui peuvent traduire un changement dans la caractérisation de la situation collaborative. L'EDA répond à ce besoin en proposant tout d'abord une collecte des données intelligente via le principe d'abonnement aux événements puis en permettant la mise en place d'un CEP. Ce CEP permet de découvrir les événements complexes, par déduction, analyse et corrélation d'événements élémentaires.

Etant donné la volumétrie des données récupérées, l'EDA convient parfaitement pour cette fonction en permettant un pré-tri au moment de la collecte elle-même puis la possibilité de filtrer les événements recueillis via le CEP. Ce CEP va filtrer, transformer et reconnaître des motifs d'événements par le biais de règles métiers. Les événements sortants (émis par le CEP) serviront à mettre à jour les modèles *terrain* et *attendu*. Après comparaison des solutions existantes, Esper dans sa version open source et gratuite est le CEP retenu pour ces travaux de thèse.

L'étude des moyens de mise en œuvre de l'agilité des processus dans la Section II.1 a mis en valeur le besoin de transformer les données issues des modèles *terrain* et *attendu*

en information permettant de conclure à un besoin d'adaptation —et le cas échéant à son amplitude—. Ce besoin est traduit par la comparaison des modèles *terrain* et *attendu* qui sont au format XML. Le choix d'un outil capable de supporter cette fonctionnalité doit répondre aux exigences suivantes :

- Structure
  - Pouvoir tirer parti de la nature d'un document XML : comparer les nœuds entre eux, ne pas effectuer un traitement ligne à ligne,
  - Ne pas tenir compte de l'ordre des nœuds dans une même branche : seule la distance par rapport à la racine compte.
- Contenu
  - Pouvoir accéder facilement au contenu du fichier δ,
  - Détailler la différence trouvée entre les deux modèles : sur quel aspect du modèle porte la différence, quels sont les attributs impactés, etc.

Nous pouvons en conclure que seul l'algorithme de XMLUnit répondait à la plupart de nos exigences.

Si le résultat de la comparaison entre les modèles *terrain* et *attendu* permet de donner une photographie globale de la divergence entre les modèles, cela n'est pas suffisant pour décider puis choisir l'adaptation des processus collaboratifs à mener. Il faut analyser chaque instance concernée par la divergence afin qu'il soit possible de déterminer (i) l'existence d'un besoin d'adaptation des processus collaboratifs, (ii) l'amplitude de l'adaptation à mener le cas échéant. Cette analyse peut être menée dans le même esprit que les principes de gestion des risques en projet. L'évolution étant avérée, il ne s'agit pas de la prévenir et de mener des actions préventives mais plutôt de l'intégrer dans la réalité de la situation collaborative et d'en déduire les actions correctives à mener si nécessaire. Si la gestion des risques permet d'évaluer notamment la probabilité de la survenue d'un risque et sa gravité, dans notre cas il s'agit plutôt de définir l'impact d'une instance et son poids dans la prise de décision. A cette fin, nous introduisons :

- La notion d'opération qui permet de qualifier la nature de la divergence : ajout, suppression ou modification d'une instance entre les modèles *terrain* et *attendu* de la situation collaborative.
- La notion de concept, qui permet de déterminer la classe à laquelle l'instance appartient : acteur, service, risque, etc.

Pour chacune des instances, le croisement de ces informations permet de déterminer sa criticité pour la pérennité de la situation collaborative et donc son poids dans la décision finale d'adaptation. Le classement des instances suivant les axes *opérations* et *concepts* est matérialisé par une matrice. Cette matrice est définie lors de la conception des processus collaboratifs par des experts métier. Elle devra pouvoir évoluer au cours du cycle de vie des processus collaboratifs afin de pouvoir corriger des erreurs d'appréciation ou de mieux refléter le domaine métier ou la réalité du terrain.

Chapitre III

Mise à jour des modèles : Détection (partie 1)

La détection est une des quatre composantes de l'agilité telle que nous l'avons définie dans le Chapitre II (détection, adaptation, réactivité et efficacité) et l'une des trois que nous conservons dans ces travaux. Cependant, avant de pouvoir détecter quoi que ce soit, il faut définir l'objet sur lequel porte la détection : son périmètre, ses caractéristiques. Afin de répondre à nos exigences de suivi de l'évolution de la situation collaborative, il faut également permettre le recueil et le traitement des données pertinentes tout au long du cycle de vie de la collaboration (voir Figure III.1).

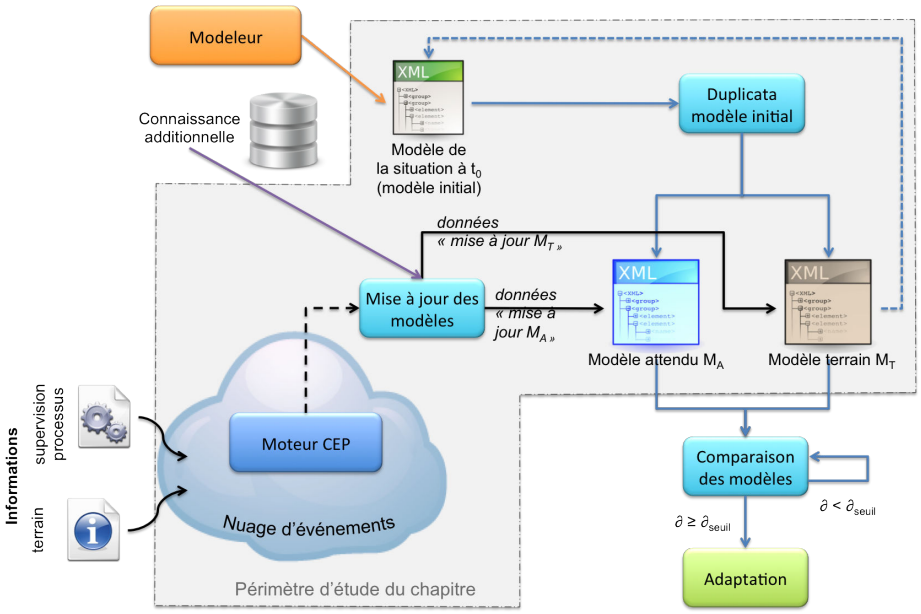


FIGURE III.1 – Architecture simplifiée du Service d'Agilité et périmètre de l'étude dans le Chapitre III

Le présent chapitre vise à expliquer d'une part le modèle de la situation collaborative sur lequel la détection va s'effectuer, d'autre part les mécanismes de mise à jour de ce modèle afin de suivre en temps réel l'évolution les changements qui surviennent dans la collaboration ou dans le théâtre de son déroulement. La Figure III.1 représente le périmètre de l'étude de ce chapitre, à savoir :

- La structure et le contenu du modèle de la situation collaborative (obtenu via le modeleur issu des travaux de thèse de [Mu, 2012]),
- Le mécanisme de mise à jour des modèles *terrain* et *attendu* grâce aux événements générés par le moteur de CEP (par filtre et/ou dérivation d'événements émis par le terrain et la supervision des processus) et à la connaissance additionnelle provenant de la correspondance entre les activités métier et les services techniques [Boissel-Dallier, 2012].

III.1 Le modèle de la situation collaborative

Nous avons évoqué le méta-modèle de la situation collaborative et la sur-couche dédiée à la gestion de crise dans le Chapitre I. Nous allons présenter plus en détails le contenu des modèles qui l'instancient ainsi que l'outil qui, sur la base de ce méta-modèle, permet d'obtenir un modèle de la situation de crise et de son environnement (i.e. le système étudié) ainsi que le système de traitement.

III.1.1 Le modèle de la crise

Nous allons détailler ici les notions de modèle de la crise, de modèles terrain et attendu, ainsi que les éléments de connaissance additionnelle permettant d'exploiter ces modèles.

III.1.1.1 Le modèle à l'instant t_0

Le modèle de la situation de crise à t_0 correspond à la caractérisation de ladite situation au tout début de la phase de réponse à la crise. Ce modèle sert notamment à construire la cartographie de processus collaboratifs (processus de réponse à la crise ici), qui sera ensuite traduite en un ensemble de workflows exécutables.

III.1.1.2 Le modèle à l'instant t

Le modèle à l'instant t doit refléter les évolutions de la situation collaborative pour en constituer une photographie instantanée. Nous avons pu voir dans le chapitre précédent que les sources d'évolution de la situation collaborative étaient regroupées en trois grandes familles : celles dues au caractère instable de la situation (i.e. du système étudié), celles dues aux variations des acteurs de la collaboration et celles dues aux processus collaboratifs (i.e. le système de traitement).

De ces trois familles d'évolution, deux points de vue émergent pour caractériser la situation collaborative à un instant t :

La photographie : sur la base des événements émis par théâtre de la collaboration (et analysés par le moteur de CEP), le modèle représente une photographie instantanée à t des caractéristiques de la situation collaborative. Cette photographie retranscrit la situation telle qu'elle est réellement,

Le portrait-robot : compte tenu des processus établis et des workflows mis en place, à un instant t de l'exécution des workflows il est possible d'esquisser un portrait-robot de la situation collaborative sur la seule connaissance de l'état d'avancement des services des workflows. Ce portrait-robot retranscrit un état supposé de la situation.

Ces points de vue donnent ainsi respectivement naissance aux :

- Modèle terrain M_T, qui représente le modèle de la situation telle qu'elle se déroule réellement, grâce à la prise en compte des événements émis par le terrain,
- Modèle attendu M_A, qui représente le modèle de la situation telle qu'on l'attend, compte tenu des événements reçus de la part de la surveillance de l'exécution des workflows.

Notons que le modèle terrain peut également servir à réinitialiser la caractérisation de la situation de crise suite à la réalisation d'une adaptation. En effet, si suite à l'étape de détection menée à un instant t donné —sur les modèles terrain et attendu— une adaptation est décidée, alors le modèle terrain à l'instant t est considéré comme étant l'image la plus réaliste de la situation collaborative. Il est donc utilisé comme modèle initial (à t) et il est à son tour dupliqué pour donner naissance aux modèles terrain et attendu. Ces modèles seront mis à jour et permettront de détecter des évolutions de la situation de crise qui nécessitent une adaptation du système de traitement de la crise.

III.1.2 Le modeleur de la crise

L'outil de modélisation de la situation de crise est issu des travaux de thèse de [Mu, 2012]. Il permet de décrire facilement une situation de crise (i.e. le système étudié), à l'aide de différents concepts issus directement du méta-modèle de la collaboration (du point de vue de la sur-couche dédiée à la crise). Une capture d'écran de l'utilisation du modeleur est visible en Figure III.2. Nous pouvons remarquer que nous retrouvons dans la palette (menu situé à gauche dans la Figure III.2) des objets qui ont les mêmes noms que les classes présentes dans le méta-modèle : risque, conséquence. Ceci est tout à fait logique, car le modeleur sert à créer un modèle de la situation de crise sur la base du méta-modèle.

D'un coup d'oeil, on sait que le rejet de matière radioactive dans l'atmosphère *(conséquence)* induit un *risque* de contamination de l'atmosphère *(environnement)*, du personnel de la centrale nucléaire *(personnes)*, ainsi que de la population habitant dans le voisinage *(personnes)*. Certains *biens* tels que la centrale nucléaire, les habitations et une école primaire sont menacés. La population est également concernée par le *risque* de panique, ce qui n'est pas le cas du personnel de la centrale.

Le modèle de la situation de crise est ensuite exploité via le fichier XML dans lequel il est enregistré (fichier que nous appelons ici *ModelSituation.xml*).

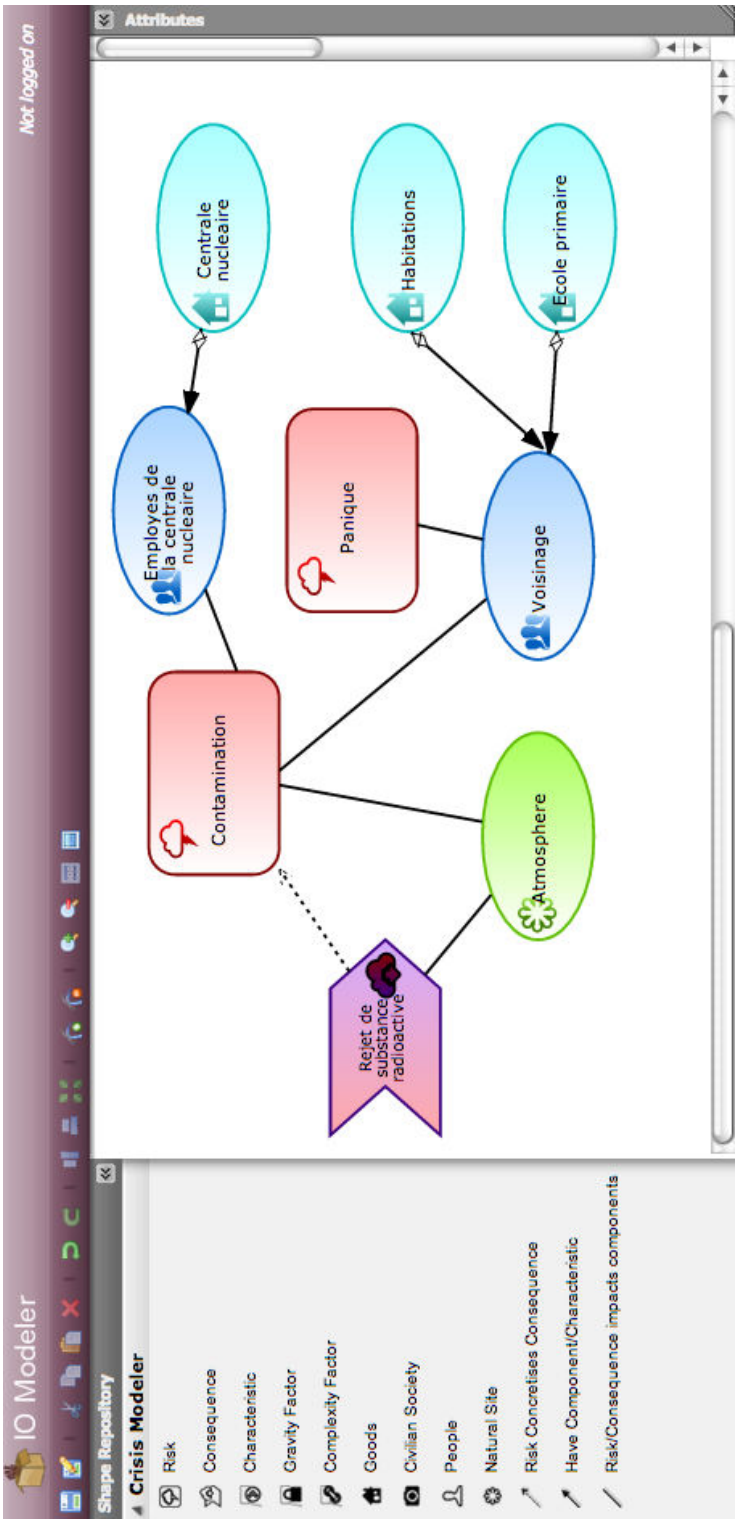


FIGURE III.2 – Exemple très simple de l'utilisation du modeleur de la situation de crise

III.1.3 Les liens entre les concepts

D'après le méta-modèle de la crise, nous pouvons constater qu'un risque est traité par une activité, qui elle-même est réalisée par un acteur de la cellule de crise. Les liens entre risque et activité ainsi qu'entre partenaire et activité sont définis via des mécanismes exposés dans les travaux de thèse de [Mu, 2012] et [Boissel-Dallier, 2012] dans le cadre du projet MISE. La Figure III.3 positionne l'étape de préparation et de réponse à la crise par rapport au projet MISE. Nous pouvons voir qu'en phase de préparation les ontologies et la doctrine fournissent des éléments de réponse *théoriques* à des problèmes : soit sous forme de correspondance {activité ; problème}, soit sous forme de processus. En phase de réponse, le modèle de la crise permet de définir les problèmes réels ainsi que les composants du système étudié.

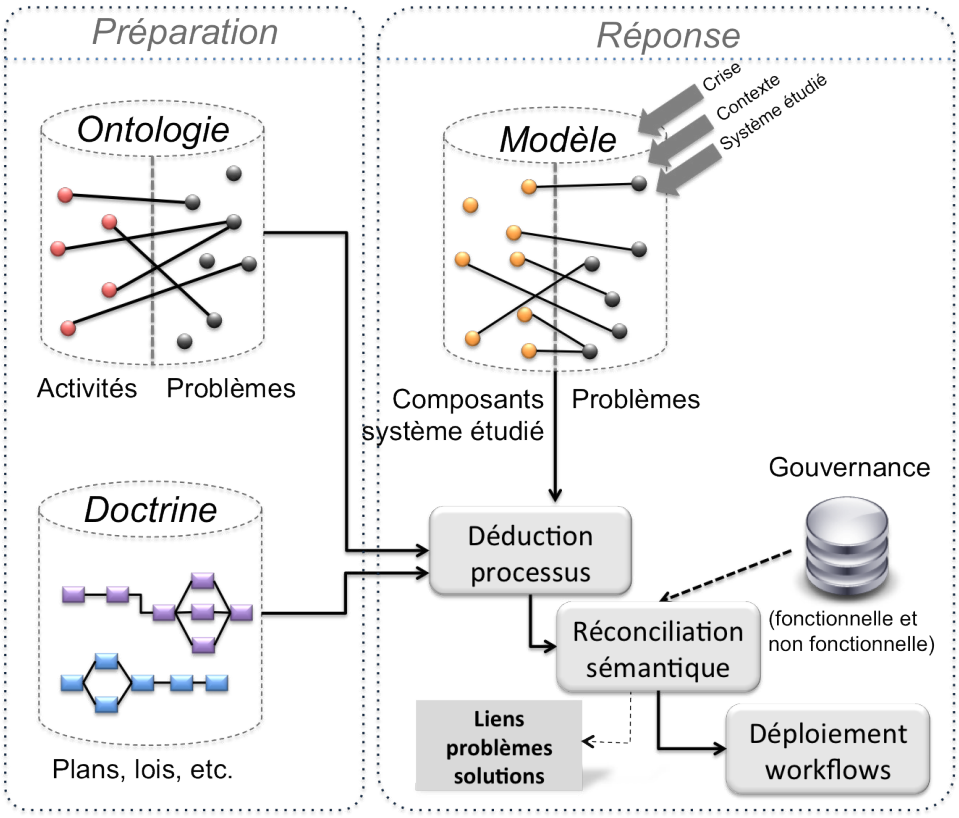


FIGURE III.3 – Positionnement de l'étape d'obtention des liens entre *problèmes* et *solutions* dans MISE

A partir des données provenant du modèle et de la phase de préparation, les processus collaboratifs de réponse à la crise sont déduits. Puis le passage des processus *métier* aux workflows *exécutables* est rendu possible grâce à la réconciliation sémantique [1]. Cette

1. Il s'agit ici d'un abus de langage, car les travaux de [Boissel-Dallier, 2012] portent sur la réconciliation syntaxo-sémantique.

dernière s'appuie sur la gouvernance fonctionnelle et non-fonctionnelle des services, afin de réaliser le meilleur choix possible pour affecter un (ou des) service technique à une (ou plusieurs) activité métier (correspondance *n-m* entre les activités et les services). A ce stade là de la conception du SI de médiation, nous connaissons les liens entre les services et les activités, ainsi que les liens entre les activités et les problèmes identifiés dans la situation de crise. Il est donc possible de connaître les liens entre les risques/conséquences (problèmes) et les services (ou solutions) qui vont tenter de les résoudre.

La définition de tous ces liens est accessible à tout moment à partir du fichier XML *ServiceChoose.xml* dans laquelle elle est enregistrée (voir Code III.1).

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<serviceChoose xmlns="http://www.enstimac.fr/mise/demio/crisis/servicechoose">
    <problemes>
        <probleme id="Contamination" name="Contamination" type="Risk" state="ko"
            priority="1"/>
        <probleme id="Intoxiques_Lourds" name="Intoxiques_Lourds" type="Consequence"
            state="ko" priority="2"/>
        </problemes>
    <solutions>
        <solution id="Decontaminer" name="Decontaminer" state="selected" actor="SAMU"
             tx_acc="1.0" wip="wait"/>
        <solution id="Soigner" name="Soigner" state="possible" actor="SAMU" tx_acc="
            1.0" wip="wait"/>
    </solutions>
    <linkPBSols>
        <linkPBSol id="DecontaminerContamination" service="Decontaminer" probleme="
            Contamination" confiance="1" state="possible"/>
        <linkPBSol id="DecontaminerIntoxiques_Lourds" service="Decontaminer" probleme
            ="Intoxiques_Lourds" confiance="1" state="selected"/>
    </linkPBSols>
    <linkSolSols>
        <linkSolSol id="Decontaminer&amp;Soigner" from="Decontaminer" to="Soigner"
            type="post"/>
    </linkSolSols>
    <linkSolConds>
        <linkSolCond id="DecontaminerVictime(s) evacuee(s)" service="Decontaminer"
            condition="Victime(s) evacuee(s)" type="needs"/>
        <linkSolCond id="DecontaminerVictime(s) decontaminee(s)" service="
            Decontaminer" condition="Victime(s) decontaminee(s)" type="fulfills"/>
     </linkSolConds>
</serviceChoose>
```

Code III.1 – Exemple de fichier des liens entre risque, conséquence et service

- *<problemes>* : il regroupe toutes les instances de *<probleme>* qui représentent les *risques* et *conséquences* à traiter. L'attribut *state* permet de savoir si une solution a été trouvée pour répondre au problème de façon couvrante (*ok*), de façon partielle (*okcould*) ou si aucune solution n'a été trouvée (*ko*). Par défaut, en sortie du moteur d'inférence, la valeur est égale à *ko* [Truptil, 2011],

- *<solutions>* : il contient les éléments *<solution>* qui décrivent les services des partenaires via un nom, un état de sélection (sélectionné, possible), le partenaire propriétaire de ce service, un état d'avancement de l'exécution du service et un taux d'accessibilité. L'accessibilité traduit l'existence de pré-requis et/ou de post-requis pour la solution. S'il n'y en a pas, la valeur est fixée à 1.
- *<linkPBSols>* : les éléments *<linkPBSol>* qu'il contient permettent de faire le lien entre un problème donné et une solution donnée (le problème et la solution doivent exister dans le fichier ServiceChoose.xml). L'élément *<linkPBSol>* permet de savoir si l'association problème-solution a été retenue (attribut d'état) et la confiance que l'on donne dans cette association.
- *<linkSolSols>* : il permet de spécifier les pré-requis et post-requis d'une activité. Par exemple, on ne peut exécuter la solution *Soigner* que si l'on a exécuté la solution *Décontaminer* au préalable,
- *<linkSolConds>* : il détaille les conditions ou pré-requis (*needs*) et les réalisations ($fulfills$) d'une solution.

L'obtention du fichier ServiceChoose.xml se fait de manière semi-automatique (voir les travaux de [Truptil, 2011], [Mu, 2012] et [Boissel-Dallier, 2012]). Le modèle de la situation de crise (qui représente le système étudié) ainsi que la liste des acteurs et de leurs compétences sont injectés dans une ontologie. L'exécution du moteur d'inférence permet d'obtenir le fichier *ServiceChoose.xml*. Ce dernier propose à chaque acteur une liste de problèmes à trier en fonction de sa criticité ainsi qu'une liste de solutions, c'est-à-dire une liste de compétences pouvant résoudre les problèmes identifiés. Les acteurs de la cellule de crise (les décideurs plus précisemment) affinent les propositions en attribuant une criticité à chaque problème et en validant ou non les propositions de solution. Une fois ce fichier paramétré par les acteurs, il sert d'entrée pour les outils de déduction des processus collaboratifs. Le fichier XSD définissant le format du fichier *ServiceChoose.xml* est disponible en Annexe E.

III.1.4 Conclusion sur les structures des modèles

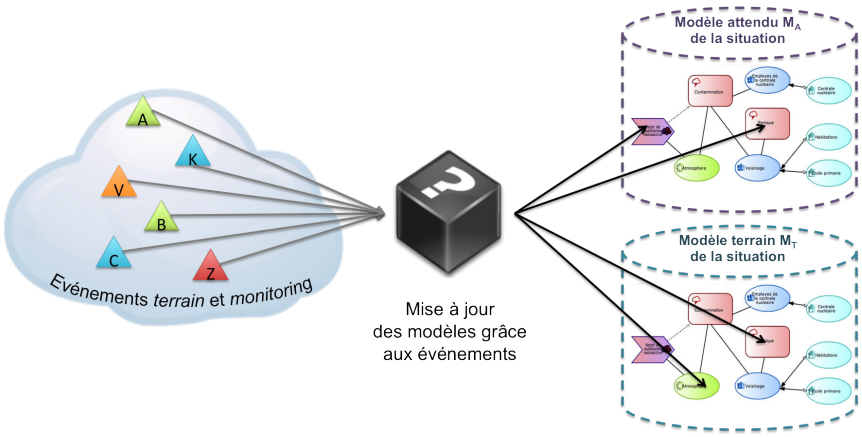


FIGURE III.4 – Mise à jour des modèles : état des lieux

Tel qu'indiqué sur la Figure III.4, actuellement nous disposons :

- D'une part d'événements émis par l'environnement de la collaboration et la collaboration elle-même,
- D'autre part de deux modèles (représentant chacun un point de vue de la situation collaborative) qu'il faut mettre à jour pour tenir compte des évolutions de la collaboration et de son contexte.

Connaissant la structure de ces modèles de la situation ainsi que celle de la connaissance additionnelle portée par les liens entre problèmes et solutions, nous pouvons nous poser la question suivante : *comment le modèle de la situation de collaboration est-il mis à jour ?*

III.2 Contexte technologique de la mise à jour

Pour répondre à la question précédente, à savoir de quelle façon réaliser la mise à jour du modèle. Nous avons pu voir dans les chapitres précédents qu'une de nos hypothèses de travail est que l'architecture de notre système de traitement de la crise est *orientée services*. Avant d'aller plus loin dans la définition de notre service d'agilité et dans la réalisation technique, revenons sur quelques notions fondamentales de SOA qui nous serviront de base pour implémenter certains principes de l'EDA.

III.2.1 L'architecture orientée service

La SOA (Service Oriented Architecture, pour Architecture Orientée Service) est une forme d'architecture de médiation qui met en oeuvre des composants logiciels appelés services. Ces services disposent d'interfaces qui sont accessibles à travers le réseau [Maréchaux, 2006]. L'interface (cf. Section III.2.3) est définie de façon neutre : elle est indépendante de la machine, du système d'exploitation et du langage de programmation dans lesquels le service est implémenté.

L'avantage majeur de la SOA est la capacité à sortir des pratiques isolationnistes en développement logiciel. Traditionnellement, chaque département développe son propre système logiciel sans avoir connaissance de ce qui existe déjà dans l'organisation (niveau supérieur du département), et le cloisonnement entre les départements est fort (pas d'échanges inter-départementaux) : c'est ce qu'on appelle l'approche en silos [Maréchaux, 2006]. Cette approche devient vite coûteuse et inefficace quand la même fonctionnalité est développée en plusieurs exemplaires. Outre le coût du développement, elle entraîne également le déploiement et la maintenance des multiples variantes de cette fonctionnalité. La philosophie SOA va se baser sur un répertoire de services partagé à travers l'ensemble de l'organisation. Elle fournit ainsi un moyen efficace de réutiliser et d'intégrer les ressources existantes.

La SOA est basée sur un mécanisme classique de requête/réponse (Figure III.5) : le service est une action exécutée par un fournisseur (provider) à l'intention d'un consom-

mateur (consumer). Cependant l'interaction entre consommateur et fournisseur n'est pas directe. Elle est faite par le biais d'un mediateur comme un ESB (Enterprise Service Bus, pour Bus de Services d'Entreprise). Un ESB est un logiciel tiers qui crée un réseau d'échange d'informations entre différentes applications informatiques.

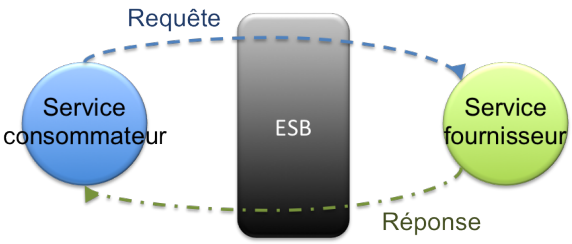


FIGURE III.5 – Principes du mécanisme d'une architecture SOA

Ce type d'architecture logicielle est caractérisée par :

- *Des couplages externes lâches*, par l'utilisation d'une couche d'interface interopérable. Les services sont invoqués quelque soit leur technologie ou leur localisation,
- *Une communication un-à-un* : un service donné est invoqué par un consommateur à la fois. Les communications sont bidirectionnelles,
- *Un déclencheur basé sur le consommateur* : le flux de contrôle est initié par le client (i.e. le service consommateur),
- *Des échanges synchrones* : les réponses sont envoyées au consommateur de manière synchrone.

III.2.2 Le SOAP

Le protocole SOAP (Simple Object Access Protocol) [Gudgin et al., 2007] est au départ une recommandation du W3C datant de juin 2003. Ce protocole permet de transmettre des messages entre des objets distants. Cela signifie que SOAP autorise un objet à invoquer des méthodes d'objets physiquement situés sur un autre serveur. Le transfert se fait le plus souvent à l'aide du protocole HTTP, mais peut également se faire par un autre protocole, comme SMTP.

Un message SOAP est basé sur une grammaire XML et peut être formaté de plusieurs façons en fonction de son style et de son type d'encodage. Il existe deux styles de formatage des messages SOAP : RPC (Remote Procedure Call) et Document.

Le style *RPC* possède une structure prédéfinie et fixe. Le corps du message, i.e. ce que contient l'élément *<soap :body>*, contient un élément qui a pour valeur le nom de l'opération que l'on souhaite invoquer dans le service. Cet élément comporte également un éléments fils pour chaque paramètre d'appel nécessaire. La réponse comporte simplement une valeur.

Le style *Document* quant à lui n'a pas de structure imposée. Le corps du message *<soap :body>* est constitué d'un document XML dont le contenu peut être contraint via un *Schema XML* (XSD). La réponse renvoyée par l'opération du service invoqué prend elle aussi la forme d'un document XML.

III.2.3 Le WSDL

Le WSDL (Web Service Description Language) [Christensen et al., 2001] sert à décrire l'interface publique d'accès à un Service Web, notamment dans le cadre d'architectures de type SOA. La version 2.0 [Chinnici et al., 2007] a été approuvée le 27 juin 2007 et est désormais une recommandation officielle du W3C. Dans ces travaux de thèse, nous utilisons la version 1.1, pour des raisons de compatibilité avec les outils servant à développer notre prototype.

Le WSDL est une description basée sur le XML qui indique de quelle façon communiquer pour utiliser le service. Le WSDL sert à décrire :

- Le protocole de communication (SOAP RPC ou SOAP orienté message),
- Le format de messages requis pour communiquer avec ce service,
- Les méthodes que le client peut invoquer ainsi que la localisation du service.

Un fichier WSDL 1.1 est structuré en plusieurs modules répartis entre une partie abstraite et une partie concrète comme l'explique la Figure III.6).

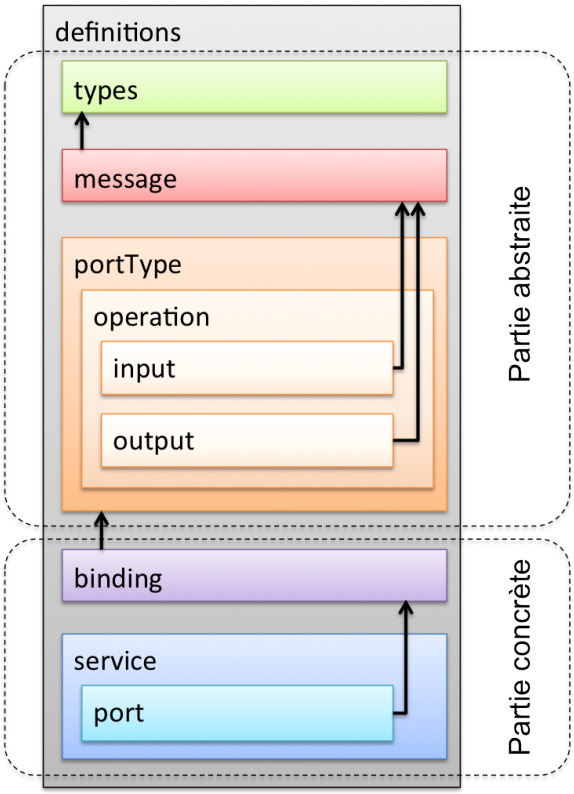


FIGURE III.6 – Structure du WSDL 1.1

Les services sont ainsi définis par six éléments majeurs [Christensen et al., 2001] :

- *<types>* qui définit les types de données qui seront utilisés dans le corps du message. Les types de données sont exprimés préférentiellement avec une XSD,
- *<message>* qui représente une définition abstraite des données qui sont échangées

entre les services. Les différents types de requêtes ainsi que les différents types de réponse y sont mentionnés,

- *<portType>* qui spécifie un ensemble d'opérations abstraites du Service Web. Chaque opération fait référence à un message entrant et à un message sortant,
- *<binding>* qui détaille l'implémentation, c'est-à-dire les protocoles de transport et le format des messages à utiliser pour chaque opération et message définis pour un portType particulier,
- *<port>* qui est associé à un point terminal (une adresse réseau ou une URL). C'est à partir de ce point terminal que l'accès au Service Web pourra être effectué,
- *<service>* qui est utilisé pour agréger un ensemble de ports.

Un exemple d'implémentation de WSDL est visible en Annexe F. On y retrouve le nom des six parties majeures dans le nom des balises de ce fichier.

Nous verrons plus tard dans ce chapitre que le WSDL joue un rôle important dans l'implémentation de l'émission et de la réception d'événements (au sens EDA du terme).

III.2.4 Les événements

Nous avons pu voir précédemment que les données permettant de caractériser la situation collaborative constituent les éléments de base de notre démarche d'agilité. En effet, grâce à ces données, il est possible de mettre à jour les modèles terrain et attendu et de reporter les nouvelles caractéristiques de la situation. Ces caractéristiques retranscrivent l'évolution du théâtre de la collaboration ou de la collaboration elle-même et peuvent amener à la détection d'une divergence et à l'adaptation du système support de la collaboration.

III.2.5 La typologie d'événements

Il faut donc déterminer le type de données qu'il est nécessaire de surveiller, récolter et analyser : il s'agit de définir les types d'événements auxquels le Service d'Agilité va s'abonner afin de pouvoir mettre à jour les modèles terrain et attendu. Par exemple, dans le cas d'une activité, plusieurs données sont indispensables telles que les relations entres les activités, l'objectif poursuivi par l'activité, etc.

La détection d'une évolution peut être réalisée via la mise à jour des modèles, qui s'appuient sur la connaissance recueillie à propos :

- De la situation,
- De l'état des activités,
- De l'état des ressources,
- Du résultat des activités.

Dans le cadre de notre architecture orientée événement, cette connaissance est apportée par les événements. Par conséquent, quatre grandes familles de types d'événements ont été définies. Le Service d'Agilité va s'abonner aux types d'événements des familles [Barthe-Delanoë et al., 2012b] :

1. Situation : il permet de recueillir l'information à propos de la situation. Cette dernière provient de capteurs ou d'une description faite par un être humain. Exemple : un capteur de radiation émet des événements de type *Mesure* (qui relève de la situation) pour transmettre les valeurs qu'il mesure,
2. Activité : ce type fournit des informations à propos de l'état d'un service : en attente, en cours, terminé, en panne. Exemple : lorsqu'une activité d'un processus se termine sans faute, elle émet un événement de type *Activité*, qui comporte la valeur *terminé*,
3. Ressource : il est utilisé pour informer de l'état des ressources (conseils et plans d'intervention, besoins matériels et en ressources humaines). Exemple : quand le plan d'évacuation de la population d'un secteur géographique donné est établi, il est diffusé à travers un événement *Ressource*,
4. Consequence : ce type d'événements transmet le résultat d'une décision (instruction) ou d'une activité (rapport de fin d'activité). Exemple : la décision d'évacuer la population, le rapport de fin de mise en place du périmètre de confinement.

Le contenu de l'enveloppe du message porté par l'événement ainsi que son type sont dépendants du contexte métier considéré. Dans le cas des présents travaux de thèse, le domaine étudié est celui de la réponse à la crise nucléaire. Des quatre familles précédemment présentées, huit types d'événements ont été dérivés [Barthe-Delanoë et al., 2012b].Le diagramme de classes présenté en Figure III.7 détaille le contenu des propriétés portées par chacun des types d'événements.

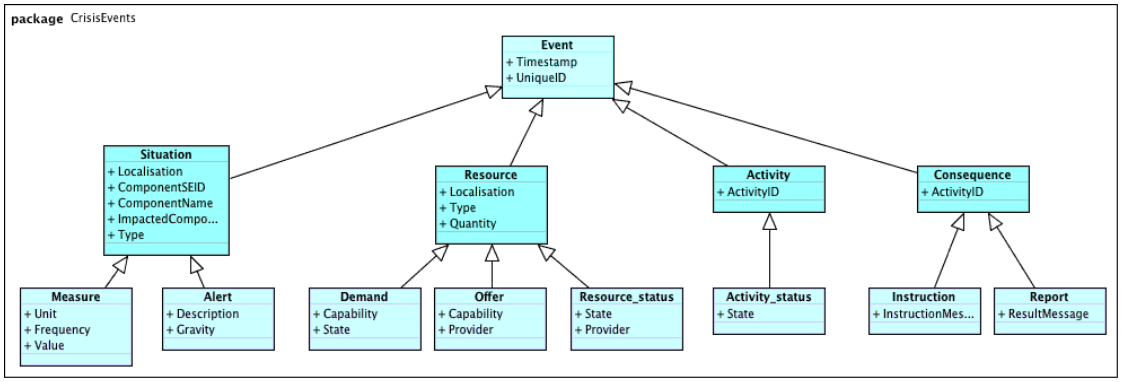


FIGURE III.7 – Les huit types d'événements couvrant une situation de crise

Le type *Mesure* concerne la mesure d'une caractéristique de la situation : par exemple, un taux de radiation en un point donné, la vitesse du vent, le nombre de blessés. *Alerte* permet de signifier un problème. L'événement de type *Demande* (respectivement *Offre*) est produit quand un acteur de la cellule de crise effectue (respectivement répond à) une demande de ressources. *Statut de Ressource* traduit le statut d'une ressource donnée. Il en va de même pour *Statut d'activité*. L'événement de type *Instruction* est produit quand une décision est prise. *Rapport* est généré quand une activité génère un rapport

en fin d'exécution.

III.2.6 L'abonnement aux événements

Le mécanisme de messagerie publication/abonnement sert à distribuer un message à plusieurs abonnés. Dans ce modèle, une application –dite productrice ou émettrice– publie un message (i.e. une notification d'événement) dans le système de messagerie et l'a marqué avec un sujet qui indique l'objet du dit-message. Les applications –dites consommatrices ou destinataires– abonnées au sujet en question, et qui disposent des droits appropriés pour recevoir les messages, reçoivent toutes une copie du message qui a été publié par l'application émettrice. Toute application destinataire peut filtrer les messages d'un sujet donné à l'aide du filtre de contenu de message évalué pour le contenu du message XML du corps du message.

Une explication plus détaillée de la norme et des concepts qu'elle met en jeu est disponible en Annexe I.

III.2.6.1 La norme WS-Notification

WS-N est une recommandation développée sous le contrôle d'OASIS [2] et est reconnue par la plupart des industriels et académiques. WS-N est décrite dans une famille de trois normes : WS-BaseNotification [OASIS, 2006c], WS-BrokeredNotification [OASIS, 2006b] et WS-Topic [OASIS, 2006a].

WS-BaseNotification WS-BaseNotification définit les rôles de base des applications émettrice/destinataire. Un producteur de notification (NotificationProducer) est une application qui envoie des notifications d'événement dans le système, alors qu'une application consommatrice de notifications (NotificationConsumer) reçoit les notifications d'événement publiées par une application tiers (généralement, un NotificationProducer). Techniquement, la spécification de notification de base des Services Web définit *des types de port WSDL* pour les applications qui souhaitent jouer le rôle de producteur ou de destinataire de notification. Les applications qui veulent consommer les notifications d'événements de façon asynchrone doivent exposer un nœud final (endpoint) de Service Web. Celui-ci doit implémenter un port de type *NotificationConsumer*. Les applications localisent ensuite le *NotificationProducer* [3] qui produit les notifications d'événements qu'elles souhaitent recevoir et appellent l'opération d'abonnement (*subscribe*) sur cette application NotificationProducer. L'application NotificationProducer est chargée des tâches liées à l'expédition de notifications d'événements. Elle fonctionne avec un service *SubscriptionManager* pour gérer la durée de vie d'un abonnement. La méthode *notify* du consommateur (exposée à travers l'interface NotificationConsumer) est invoquée à chaque fois qu'une notification est disponible pour un sujet auquel il s'est abonné (et que la notification a passé le filtre avec succès).

2. Organization for the Advancement of Structured Information Standards
3. Ou le NotificationBroker dans le cas où la communication n'est pas directe, cf. Annexe I.2.

WS-BrokeredNotification WS-BrokeredNotification définit l'interface de web services pour le NotificationBroker (courtier de notifications). De même que dans WS-BaseNotification, le courtier doit supporter les méthodes *notify* et *subscribe*. Cette dernière lui permet de router ou de produire une séquence de 0 à N notifications. Le courtier permet de découpler les fournisseurs des consommateurs, réduisant ainsi le nombre d'interconnexions entre les services émetteurs et récepteurs d'événements. Il permet également aux souscripteurs de s'abonner à des événements (via son intermédiaire) sans avoir à connaître l'identité des fournisseurs.

WS-Topic WS-Topic est une spécification qui fournit une méthode pour organiser et catégoriser les notifications d'événements auxquels il est possible de s'abonner par *sujet* (topic, dans la versions originale de la norme). Un *sujet* est décrit au format XML. Les sujets sont groupés au sein d'*Espaces de nom de Sujet* (Topic Namespaces). Les sujets sont utilisés par un abonné pour identifier un centre d'intérêt qu'il souhaite couvrir à travers son abonnement.

III.2.6.2 Implémentation de WS-N

Comme nous avons pu le voir précédemment, WS-N permet de définir de nouveaux ports dans l'interface WSDL d'un Service Web, afin de permettre l'émission et/ou la réception d'événements. Cela implique l'ajout de la définition des dits ports dans le WSDL du Service Web, mais aussi l'ajout de fichiers complémentaires pour définir les sujets ainsi que l'implémentation de la norme au sens technique.

Nous avons utilisé les bibliothèques open sources développées par PeTALS Link[4] (packagées sous le nom *com.ebmwebsourcing.wstar.notification.service*) afin d'ajouter la surcouche WS-N à nos Services Web.

A titre d'exemple, nous nous servirons d'un Service Web permettant l'envoi d'emails ainsi que d'événements à l'envoi d'un mail. Nommé *WebServiceSendMailEDA*, son code source complet est à disposition à l'adresse `http://perso.mines-albi.fr/~barthe/these/ws-eda/WebServiceSendMailEDA`.

L'*Espace de nom de Sujet* (Topic Namespaces) est défini via un fichier XML nommé *EventTopicNS.xml* (Code III.2).

```
<wstop:TopicNamespace name="CrisisEvents"
        targetNamespace="http://www.enstimac.fr/mise/event"
        xmlns:data="http://www.enstimac.fr/mise/event"
        xmlns:wstop="http://docs.oasis-open.org/wsn/t-1"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://docs.oasis-open.org/wsn/t-1 http://docs.oasis-
            open.org/wsn/t-1.xsd http://http://www.enstimac.fr/mise/event/ ./
            EventCloud.wsdl">
```

4. Editeur de logiciels libres français, racheté en 2012 par Linagora.

```
    <wstop:Topic name="situationalEvent" messageTypes="data:measureEvent data:
        alertEvent"/>
    <wstop:Topic name="resourcesEvent" messageTypes="data:demandEvent data:offerEvent
        data:resourcesStatusEvent"/>
  </wstop:TopicNamespace>
```

Code III.2 – Définition de l'espace de noms des sujets

Les différents espaces de nom présents dans l'élément racine *<TopicNamespace>* permettent de faire le lien avec les définitions des éléments provenant de grammaire externes qui sont utilisés dans *EventTopicNS.xml*. Typiquement, *<TopicNamespace>* est préfixé par *<wstop>* dont l'espace de nom est *<http ://docs.oasis-open.org/wsn/t-1>*, et dont le schéma XML qui définit le vocabulaire autorisé par la norme WS-Notification pour définir un Espace de nom de Sujet est disponible à l'URL `http://docs.oasis-open.org/wsn/t-1.xsd`. On trouve également un attribut *name* dans l'élément racine : celui-ci indique le nom de notre *Espace de nom de Sujet*, soit *CrisisEvents* ici. *targetNamespace* indique l'espace de nom pour le Topic Namespace en cours de définition. Il est exprimé sous forme d'URL (`http://www.enstimac.fr/mise/event`). Ensuite les différents *sujets* sont listés sous forme d'éléments *<Topic>* qui contiennent un attribut *name* pour nommer le sujet, ainsi qu'un attribut *messageTypes* qui indique le format des messages des notifications liées à ce sujet. Ici, nous n'avons que deux sujets déclarés (*situationalEvent* et *resourcesEvent*), qui attendent les formats d'événement de type *measureEvent* et *alertEvent* (pour le sujet *situationalEvent*) et *demandEvent*, *offerEvent* et *resourcesStatusEvent* (pour le sujet *resourcesEvent*).

L'ensemble des sujets est ensuite défini à son tour dans le fichier *EventTopicSet.xml* (Code III.3). Il sert à définir l'ensemble des types d'événements supportés par le Web Service et permet de recevoir toutes les notifications des événements concernés par l'ensemble de sujets en ne souscrivant qu'un seul abonnement..

```
<?xml version="1.0" encoding="UTF-8"?>
<wstop:TopicSet xmlns:wstop="http://docs.oasis-open.org/wsn/t-1" xmlns:tns1="http://
    www.enstimac.fr/mise/event/">
  <tns1:situationalEvent wstop:topic="true" />
  <tns1:resourcesEvent wstop:topic="true" />
</wstop:TopicSet>
```

Code III.3 – Définition de l'ensemble des noms des sujets

L'élément racine *TopicSet* possède deux espaces de nom : le premier (*wstop*) correspond à la norme WS-Notification, le second (*tns1*) à l'espace de nom défini précédemment. Les types d'événements (sujets) concernés cet ensemble de sujets sont simplement listés à l'intérieur de l'élément racine. L'attribut *wstop :topic* valué à *true* permet d'indiquer que le type d'événement est bien un sujet. L'intérêt d'un tel attribut n'apparaît pas de façon flagrante dans notre exemple, car nous n'avons que deux sujets « simples ». En effet, il est possible d'avoir un sujet composé (« sujet parent ») lui-même d'autre sujets

(« sujets enfants »). L'attribut *wstop :topic* permet d'inclure des « sujets enfants » (en valuant l'attribut à *true*) sans pour autant inclure le « sujet parent » (attribut à *false*) dans ce TopicSet.

Le fichier WSDL (Code III.4) subit quelques ajouts afin de faire figurer les ports ainsi que les interfaces des applications permettant *l'émission* et la *réception* des événements (import des WSDL). Afin de rendre la lecture plus aisée, les parties concernant l'envoi des messages à l'invocation d'une opération ainsi que les déclarations des espaces de nom (dans l'élément *<wsdl :definitions>*) ont été retirées.

```
1  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2  <wsdl:definitions>
3    <!-- Import des WSDL WS-Notification -->
4    <wsdl:import location="./eventservice/wsn-producer.wsdl"
5      namespace="http://www.petalslink.com/wsn/service/WsnProducer" />
6    <wsdl:import location="./eventservice/wsn-consumer.wsdl"
7      namespace="http://www.petalslink.com/wsn/service/WsnConsumer" />
8
9    <!-- Partie envoi de messages par invocation d'operations retiree-->
10
11   <!-- Producteur d'evenements -->
12   <wsdl:service name="webserviceSendMailEDAProducer">
13     <sawsdl10:attrExtensions
14       sawsdl10:modelReference="http://www.petalslink.org/wsoui/topicnamespace#
             classpath://EventTopicNS.xml http://www.petalslink.org/wsoui/topicset#
             classpath://EventTopicSet.xml" />
15     <wsdl:port binding="wsn:NotificationProducerSoapBinding"
16       name="webserviceSendMailEDAProducerEndpoint">
17       <soap:address
18         location="http://localhost:5500/webserviceSendMailEDAProducerEndPoint" />
19     </wsdl:port>
20   </wsdl:service>
21
22   <!-- Consommateur d'evenements -->
23   <wsdl:service name="webserviceSendMailEDAConsumer">
24     <sawsdl10:attrExtensions
25       sawsdl10:modelReference="http://www.petalslink.org/wsoui/topicnamespace#
             classpath://EventTopicNS.xml http://www.petalslink.org/wsoui/topicset#
             classpath://EventTopicSet.xml" />
26     <wsdl:port binding="wsnc:NotificationConsumerSOAPBinding"
27       name="webserviceSendMailEDAConsumerEndpoint">
28       <soap:address location="http://localhost:5501/webserviceSendMailEDAEndpoint" />
29     </wsdl:port>
30   </wsdl:service>
31 </wsdl:definitions>
```

Code III.4 – WSDL avec les ports permettant l'émission et la réception d'événements

Nous pouvons constater tout d'abord l'import de contrats WSDL externes : *wsn-producer.wsdl* et *wsn-consumer.wsdl* (lignes 4 à 7). Ils prennent respectivement en charge la description des interfaces pour l'émission d'événements et la consommation d'évé-

nements, notre Web Service *WebServiceSendMailEDA* étant capable d'émettre et de s'abonner à des événements. Ces WSDL exposent les différentes opérations décrites dans la norme WS-Notification (*subscribe*, *unsubscribe*, *notify*, etc.). Ensuite, nous pouvons trouver la déclaration de deux *services* exposant les points finaux :

- *webserviceSendMailEDAProducer* (lignes 12 à 20) expose *webserviceSendMailEDAProducerEndpoint*, sur lesquel les événements seront émis (et auquel un service tiers pourra s'abonner),
- *webserviceSendMailEDAConsumer* (lignes 23 à 30) expose *webserviceSendMailEDAConsumerEndpoint*, qui permet de collecter les événements émis par des services tiers.

Pour chacun des services, on constate que l'*Espace de nom des Sujets* (Topic Namespace) ainsi que l'*Ensemble de Sujets* (Topic Set) que nous avons définis précédemment sont référencés.

Diverses classes Java sont ensuite créées sur la base de ce WSDL afin d'assurer l'implémentation des fonctions *subscribe* et *notify* (par exemple) en plus de la logique métier du Web Service. Pour des raisons de clarté, nous ne les présenterons pas ici : leurs sources sont disponibles à l'adresse `http://perso.mines-albi.fr/~barthe/these/ws-eda/WebServiceSendMailEDA/`.

Les événements sont exprimés au format XML et leur structure est contrainte par un Schéma XML qui respecte la topologie des événements définie précédemment (cf. Section III.2.5). Le fichier XSD du Schéma XML est disponible en Annexe G.

Quand l'événement est émis par un Service Web qui suit la norme WS-N, il est enveloppé dans la sur-couche provenant de l'implémentation de WS-N. Techniquement parlant, les données de l'événement sont contenues dans l'élément <Message> de la notification.

Un exemple de notification au format WS-N est visible ci-après (Code III.5) : elle embarque un événement de type Demande, où une demande en ressources (10 voitures en l'occurrence) est effectuée. Pour des raisons de lisibilité, nous avons à nouveau supprimé les informations qui concernent les différents espaces de nom utilisés dans la notification (elles se trouvent normalement dans les éléments *Notify* et *demandEvent*).

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <ns14:Notify>
3    <ns14:NotificationMessage>
4      <ns14:SubscriptionReference>
5        <ns13:Address>http://localhost:9803/
             OfficeOfInfrastructureEventProducersSOAPEndpoint</ns13:Address>
6        <ns13:ReferenceParameters/>
7      </ns14:SubscriptionReference>
8      <ns14:Topic Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">top:
           resourcesEvent</ns14:Topic>
9      <ns14:ProducerReference>
10       <ns13:Address>http://localhost:9300/OfficeOfInfrastructureSOAPEndpoint</ns13:
```

```
            Address>
11        <ns13:ReferenceParameters/>
12      </ns14:ProducerReference>
13      <ns14:Message>
14        <ns2:demandEvent>
15          <ns2:uid>DemandEvent-0</ns2:uid>
16          <ns2:timestamp>2013-06-23T09:46:24.399+01:00</ns2:timestamp>
17          <ns2:capability/>
18          <ns2:state/>
19          <ns2:uncertainty>
20            <ns2:unit/>
21            <ns2:value/>
22            <ns2:origin/>
23          </ns2:uncertainty>
24          <ns2:resources>
25            <ns2:type>car</ns2:type>
26            <ns2:localisation/>
27            <ns2:quantity>10.0</ns2:quantity>
28          </ns2:resources>
29        </ns2:demandEvent>
30      </ns14:Message>
31    </ns14:NotificationMessage>
32  </ns14:Notify>
```

Code III.5 – Exemple de notification d'événement WS-N

La notification de l'événement consiste en un document XML composé du message de notification (*NotificationMessage*) qui contient :

- *SubscriptionReference*, la référence à l'abonnement (lignes 4 à 7),
- *Topic*, la référence du sujet auquel le service s'est abonné (ligne 8),
- *ProducerReference*, la référence à l'émetteur de l'événement ((lignes 9 à 12),
- *Message*, le corps du message qui contient l'événement proprement dit (lignes 13 à 30). L'événement est structuré sous la forme d'un document XML (lignes 12 à 29), suivant le Schéma XML précédemment mentionné (cf. Annexe G). On retrouve les composants d'un événement de type *Demand* : identifiant unique (ligne 15), horodateur (ligne 16), capacité (ligne 17), état (ligne18), incertitude —unité, valeur et origin— (lignes 19 à 23), ressources —type, localisation et quantité— (lignes 24 à 28).

III.2.7 Conclusion du contexte technologique

Nous avons pu voir que l'architecture retenue est de type ED-SOA (définie dans le Chapitre II), implémentant la norme WS-Notification pour assurer l'émission et la réception d'événements. Nous avons également défini les huit types d'événements [5] qui seront utilisés dans le contexte de la réponse à une crise civile. Maintenant que ces choix architecturaux ont été présentés, nous pouvons nous pencher sur leur mise en œuvre.

5. *Mesure*, *Alerte*, *Demande*, *Offre*, *Statut de Ressource*, *Statut d'Activité*, *Instruction* et *Rapport*.

III.3 Mise en œuvre de la mise à jour des modèles

Les huit types d'événements présentés dans la Section III.2.5 vont être utilisés pour réaliser la mise à jour des modèles *terrain* et *attendu*. Cependant, suivant le point de vue adopté (i.e. suivant le modèle à mettre à jour), certains types d'événements ne seront pas utiles.

Le rôle du moteur de CEP va être d'effectuer un tri entre les événements reçus, de les traiter et d'émettre des événements concernant uniquement la mise à jour du modèle *attendu* et d'autres événements dédiés à la mise à jour du modèle *terrain*. Ce tri va être rendu possible grâce à l'application de règles métier. Nous verrons que parmi les huit types d'événements définis, un type sera simplement filtré par le moteur de CEP. Les sept autres seront analysés et agrégés par le moteur de CEP de façon à obtenir de nouvelles données sur la base de celles reçues.

III.3.1 Les événements de suivi de l'exécution des processus

Les événements de suivi de l'exécution des processus sont des événements émis directement par les services présents dans les workflows exécutés par l'orchestrateur. Ces événements permettent de connaître l'état d'un service :

- *wait* : le service est en attente, il n'a pas encore été exécuté,
- *wip* : le service a été appelé, il est en cours d'exécution,
- *finish* : le service est terminée, de façon nominale,
- *ko* : le service est terminé, suite à une panne ou à un dysfonctionnement.

Ces événements correspondent aux événements de type *Activité* décrits précédemment. Dans le cas présent, le moteur de CEP va simplement filtrer les événements sur la base de leur type et les émettre dans un sujet dédié à la mise à jour du modèle attendu. Il est possible d'éviter à l'outil de mise à jour d'être submergé par le nombre de notifications de changement des statuts d'exécution des services en mettant en place des règles métier. Par exemple, dans le cas où un service envoie régulièrement des événements à propos de son état d'avancement (sans que celui-ci aie nécessairement évolué), les règles métier permettent au moteur de CEP de n'envoyer de nouvelle notification à l'outil de mise à jour que si la valeur du statut pour cette activité a changé depuis la dernière notification reçue (par le moteur de CEP) à ce sujet. La règle métier suivante illustre cet usage :

SI Activite_precedente.ActiviteID == Activité_courante.ActiviteID
ET Activite_precedente.Statut != Activite_courante.Statut
AVEC Activite(ActiviteID, statut).Activite_courante **SUIT** Activité(ActivitéID, statut).Activite_precedente
==>Activite{Activite_courante.ActiviteID ; Activite_courante.Statut}

Si pour deux notifications d'événement de type *Activité* portant sur la même activité (ActiviteID identiques dans les deux notifications) mais que le statut de l'activité diffère, alors le CEP émet un événement décrivant le nouvel état de l'activité. Dans la cas où les états sont identiques, aucun événement n'est émis par le CEP.

III.3.2 Les événements complexes générés par le CEP

En plus des événements émis par les activités au niveau des workflows, le moteur de CEP va « écouter » les événements émis par le terrain.

Son premier rôle va être de *filtrer les événements reçus* et d'en analyser le contenu, afin de ne transmettre que l'information pertinente et nécessaire à l'outil de mise à jour des modèles. Les événements émis par le terrain étant très nombreux et les informations qu'ils transportent pouvant traduire aucune évolution des caractéristiques de la situation vue du *terrain*, il est nécessaire de filtrer les événements entrants afin d'en réduire la quantité et d'en assurer la pertinence (transmettre au service de mise à jour pour la douzième fois consécutive la présence d'un risque d'incendie n'apporte rien quant à la mise à jour du modèle *terrain*, puisqu'après la réception de la première notification allant dans ce sens, le modèle l'intègre déjà).

Face à ces événements plus variés et surtout potentiellement plus nombreux, la première tâche du moteur de CEP va être de les filtrer. Ce premier rôle du CEP va éviter à l'outil de mise à jour des modèles d'être submergé avec des événements transmettant des données redondantes ou non pertinentes.

La mise en place de la notion de temporalité (à travers des fenêtres de temps) dans les règles métier permet aussi de limiter le nombre de notifications envoyées en attendant qu'un certain nombre et type d'événements soient reçus. En général, des fonctions d'agrégation (comme une moyenne) sont utilisées pour effectuer un traitement sur les données reçues dans l'intervalle de temps de la fenêtre.

Tirée de notre cas d'étude, une telle règle métier peut concerner la surveillance des niveaux de radioactivité. Dans notre exemple, nous souhaitons connaître par tranche de 2 minutes le taux moyen de radioactivité d'un certain endroit de la centrale au début de la réponse à la crise. Or les capteurs de radioactivité de ce secteur mesurent et transmettent leurs relevés toutes les 20 secondes ou toutes les 30 secondes (suivant les réglages des capteurs). On souhaite donc connaître la moyenne des valeurs contenues dans les événements *Mesure* émis par les capteurs.

La règle métier est la suivante :

POUR TOUT Mesure{Radioactivité ; CapteursOuest})
PENDANT 2 min
==>**Moyenne(**Mesure{Radioactivité ; CapteursOuest}**)**

Le deuxième rôle du moteur de CEP va être de *déduire de nouveaux événements* sur la base d'événements reçus. Pour cela, il va s'appuyer sur des règles métiers (définies en amont). Par exemple, si la mesure de la radioactivité ambiante dépasse la valeur seuil V_+ ou qu'elle dépasse le seuil V_- tout en augmentant fortement, alors le moteur de CEP émet un événement de type *Alerte* concernant le comportement de la radioactivité :

SI Mesure{Radioactivité} $> V_+$
OU (Mesure{Radioactivité} $> V_-$ **ET** δRM/δt $> s$ **)**
==>Alerte{Risque ; Radioactivité ; Mesure.localisation}

D'autres règles métier seront décrites dans la description et l'implémentation de notre cas d'étude.

Ces nouveaux événements, appelés *événements complexes*, traduisent l'évolution de la collaboration et du théâtre de son exécution. C'est la raison pour laquelle ils servent à mettre à jour le modèle terrain. Ces événements sont émis par le moteur de CEP sur un sujet concernant la mise à jour du modèle terrain. Notons que la connaissance mise en jeu dans l'établissement des règles métier n'est pas figée : fortement liée à l'expérience des acteurs de la cellule de crise (et plus généralement de la situation collaborative), elle doit refléter son évolution. A cette fin, il est possible :

- Pour le moteur de CEP de souscrire de nouveaux abonnements à de nouvelles sources d'événements (sans avoir besoin d'arrêter l'exécution du moteur),
- D'injecter de nouvelles règles métier dans le moteur de CEP au cours de son exécution.

En outre, il faut souligner le fait que les événements reçus ne sont pas conservés de façon durable. Leur existence (une fois collectés) est limitée à la « durée » spécifiée dans la requête de l'EPA qui utilise ces événements. Si une fenêtre est utilisée dans la règle portée par l'EPA et suivant son type —temporelle [6] ou quantitative [7], un événement sera conservé plus ou moins longtemps dans l'EPA.

Conclusion sur le rôle du CEP dans la mise à jour des modèles

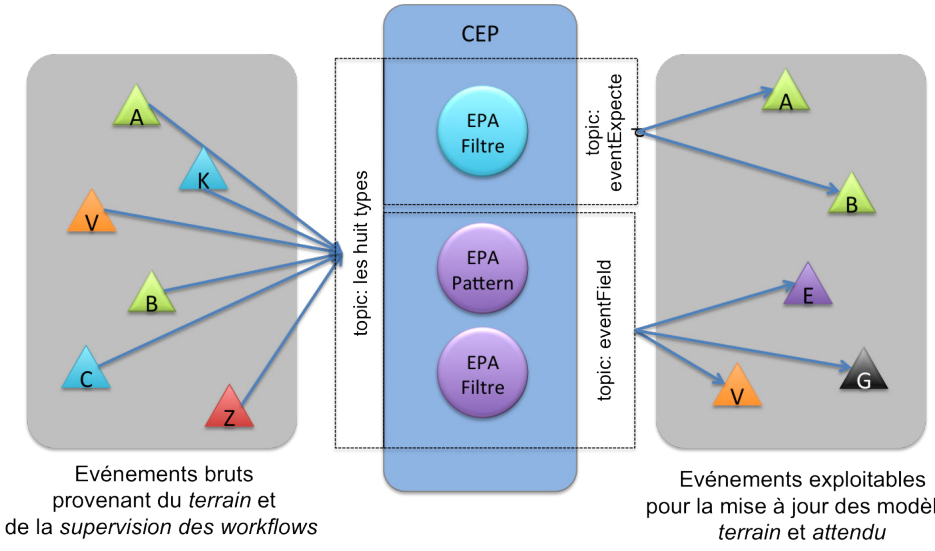


FIGURE III.8 – Le rôle du CEP dans le traitement des événements en vue de la mise à jour des modèles

Le moteur de CEP s'abonne aux huit types d'événement (dans le cas de la réponse à la crise). Il joue deux rôles vis-à-vis des événements entrants (Figure III.8) :

6. Exemple : on conserve les événements pendant un temps donné.
7. Exemple : on conserve les N derniers événements.

- Le CEP filtre les événéments de type *Activité* (événements A et B sur la figure), et les ré-émet sur le sujet *eventExpected*. Les événements ne subissent aucune opération de dérivation,
- Le CEP créé de nouveaux événements, sur la base des événements dont les types sont différents de *Activité*. Ces événements sont filtrés mais aussi passés au crible de motifs d'événement (règles métier injectées dans le CEP).
  Si l'un de ces motifs est identifié, alors un nouvel événement (dérivés des événements déclencheurs du motif) est émis. Ainsi sur la Figure III.8 nous pouvons voir que l'événement V (dont le type n'est pas *Activité*) a été filtré mais il a aussi servi (avec les autres événéments) à créer les événements émis par le CEP sur le sujet *eventField*.

III.3.3 Préparation à la mise à jour des modèles

III.3.3.1 La structure du fichier contenant le modèle

Le modèle de la situation est automatiquement écrit au format XML par le modeleur dans un fichier *SituationModel.xml*. Le format de ce fichier est donné par le Schema XML *CrisisModel.xsd* disponible en Annexe H.

Nous allons donc travailler sur la mise à jour de cet arbre XML à partir des informations générées par le CEP et des données faisant le lien entre problème, service et acteur issues du fichier *ServiceChoose.xml*.

III.3.3.2 La préparation des modèles

Le fichier *SituationModel.xml* issu du modeleur contient deux types d'informations : celles concernant le contenu (instances des concepts, liens entre instances), celles à propos de la présentation (formes, flèches, objet texte). Il serait donc plus judicieux du point de vue du traitement des données du fichier de ne conserver que celles qui sont pertinentes au vu de nos objectifs.

Il se trouve que l'outil de déduction de la cartographie des processus collaboratifs (outil issu du projet MISE) retire les informations de présentation graphique.

Nous allons donc utiliser le fichier du modèle initial de la situation collaborative *après son passage dans l'outil de déduction*, réduisant ainsi significativement le nombre de données embarquées dans le fichier et sa longueur (voir Figure III.9 en page suivante) avant de le dupliquer pour créer les modèles *terrain* et *attendu* initiaux (qui seront mis à jour).

III.3.3.3 Choix du modèle à mettre à jour

Suivant le type de sujet auquel appartient l'événement reçu par l'outil de mise à jour des modèles, la méthode correspondant au modèle *ad hoc* est appelée.

FIGURE III.9 – Le fichier du modèle de la situation avant et après retrait des données concernant l'affichage graphique du modèle

L'algorithme de mise à jour des modèles terrain et attendu $MAJModeles()$ se présente comme suit.

MAJModeles()

Algorithme principal de la procédure de mise à jour des modèles

Paramètres : mT : modèle terrain
mA : modèle attendu
event : objet événement

Début
  Si (event.getTopic()=="expectedEvent") **Alors**
    Action : MAJModeleAttendu(mA, event)

  Si (event.getTopic()=="fieldEvent") **Alors**
    Action : MAJModeleTerrain(mT, event)

  Fin Si
Fin

Si l'événement reçu provient de l'abonnement réalisé sur le sujet *expectedEvent* (respectivement *fieldEvent*), il s'agit d'un événement concernant la mise à jour du modèle

attendu (respectivement *terrain*). Les fonctions *MAJModeleAttendu* et *MAJModeleTerrain* auxquelles cet algorithme principal fait appel sont détaillées dans la suite de cette section. La fonction *getTopic()* est un accesseur permettant de récupérer le nom du topic sur lequel l'événement a été émis.

III.3.4 La mise à jour du modèle attendu

Le modèle attendu M_A est mis à jour grâce aux données provenant des événements *issus de l'exécution des services des workflows* (cf. Section III.3.1). L'outil de mise à jour s'abonne à ces événements émis par le CEP via le sujet *eventExpected*.

L'état de chacun des services est conservé dans le fichier du modèle *attendu*. Suivant la valeur de l'événement reçu (à savoir l'identifiant du service et la valeur de son état), le service concerné verra son statut d'exécution mis à jour avec les valeurs *wait*, *wip*, *finish* ou *ko* dans le modèle *attendu*.

Une fois cette mise à jour de l'état d'avancement du service effectuée, il est nécessaire d'analyser les impacts d'une telle mise à jour sur le modèle de caractérisation de la situation. En effet, nous avons vu précédemment que chaque problème avait une à plusieurs solutions qui lui étaient attachées. Ces liens sont conservés dans le fichier *ServiceChoose.xml*. Si toutes les solutions (services) affectées à un même problème voient leur état passer à *terminé* sans erreur (*finish*) alors le problème est légitimement considéré comme résolu. Dans ce cas, le problème pourra être supprimé du modèle de la situation. Ce sont ces mécanismes que l'algorithme de mise à jour du modèle M_A propose de mettre en place (Algorithme $MAJModeleAttendu(mAttendu, evenement)$).

$MAJModeleAttendu(mAttendu, evenement)$

Cet algorithme permet de mettre à jour le modèle attendu

Paramètres : mAttendu : le modèle attendu de la situation collaborative
evenement : l'objet événement, transmis par le CEP

Variables : instanceId : l'identifiant de l'instance d'Activité dans le modèle attendu, concerné par l'événement
tabLiens : le tableau d'objets lien, qui contient les liens entre les solutions et les problèmes
tabProblemesLiesIds : le tableau de chaînes, qui contient les identifiants des problèmes liés
tabSolutionsLieesIds : le tableau de chaînes, qui contient les identifiants des solutions liées
drapeau : booléen permettant de vérifier si on a trouvé une activité à un état différent de 'finish'

Début
instanceId ← **Action** : Récupérer l'identifiant de l'instance porté par l'événement
drapeau ← FAUX
tabLiens ← Récupérer les liens entre problèmes et solutions

...à suivre...

```
...suite...
    Pour i variant de 1 à taille(tabLiens) Faire
        Si (instanceId == tabLiens[i].solution) Alors
            tabProblemesLiesIds ← Action : Ajouter(tabLiens[i].probleme)
        Fin Si
    Fin Pour
    Pour j variant de 1 à taille(tabProblemesLiesIds) Faire
        tabSolutionsLieesIds ← Vider(tabSolutionsLieesIds)
        Si (tabProblemesLiesIds[j] == tabLiens[j].probleme) Alors
            tabSolutionsLieesIds ← Action : Ajouter(tabProblemesLiesIds[j].solution)
        Fin Si
        Pour k variant de 1 à taille(mAttendu) Faire
            Pour m variant de 1 à taille(tabSolutionsLieesIds) Faire
                Si ((mAttendu[k].service.id == tabSolutionsLieesIds[m].solution) &&
                (mAttendu[k].service.etat <> "finish")) Alors
                    drapeau ← VRAI
                Fin Si
            Fin Pour
        Fin Pour
        Si (drapeau == FAUX) Alors
            Action : Supprimer(tabProblemesLiesIds[j].solution, mAttendu)
        Fin Si
    Fin Pour
Fin
```

L'algorithme va parcourir le fichier *ServiceChoose.xml* contenant les liens entre les solutions (services) et les problèmes (risques, conséquences...). Pour un service donné (celui concerné par l'événement reçu), il va mettre son état à jour (suivant la valeur transmise par l'événement) puis, si cet état est égal à *finish*, rechercher tous les problèmes liés. Puis, pour chaque problème lié trouvé, l'algorithme va rechercher les solutions qui sont liées à ce problème (étant donné la correspondance de type *n-m* entre les solutions et les problèmes). Pour chacune des solutions liées trouvées, son état va être recherché (dans le fichier du modèle *attendu* ici). Si pour l'ensemble de ces solutions liées leur état est égal à *finish* (et dans ce cas le booléen de contrôle reste à égal FAUX), alors cela signifie que le problème est résolu. L'algorithme va donc supprimer l'élément du modèle *attendu* qui possède le même identifiant unique que le problème résolu.

III.3.5 La mise à jour du modèle terrain

Le modèle terrain M_T est maintenu à jour uniquement grâce aux événements provenant du *terrain* : Mesure, Alerte, Demande, Offre, Statut de ressource, Instruction, Rapport. Le CEP va analyser, filtrer et/ou dériver ces événements entrants et émettre des événements sortants. Ces événements sortants sont construits en fonction des règles métier décrites dans le CEP (ces règles sont dépendantes de la situation collaborative

considérée) et contiennent les informations nécessaires pour modifier le modèle *terrain*. Nous avons vu que le modèle *terrain* est un flux XML contenant une liste hiérarchisée d'éléments tels que des partenaires (*partner*), des services (*services*), des biens matériels (*goods*), etc. Chacun de ces éléments (que nous appellerons *instance* par la suite) contient à minima deux informations : un identifiant unique, un nom. Or les événements émis par le CEP (sur le sujet *eventField*) ont pour but la mise à jour du modèle *terrain* et donc de modifier ou supprimer des instances existantes ou de créer de nouvelles instances dans ce modèle. Chaque événement émis sur le sujet *eventField* va contenir l'identifiant unique de l'instance contenue dans le modèle.

Ces événements sont récupérés par l'outil de mise à jour à travers son abonnement au sujet *eventField* mis à disposition par le CEP. L'outil de mise à jour va pour chaque événement reçu rechercher l'existence de l'instance du concept correspondante dans le modèle, grâce à son identifiant unique (transmis par l'événement). Si l'instance existe, cela signifie que ses caractéristiques seront à mettre à jour (modification ou suppression) en fonction des données portées par l'événement. Si l'instance n'est pas retrouvée, alors elle n'existe pas dans le modèle *terrain* et l'outil de mise à jour des modèles va la créer.

L'identifiant unique de l'instance provient des données portées par l'événement. Il ne faut surtout pas le confondre avec l'identifiant unique de l'événement (qui permet de l'identifier dans le nuage des événements, lors d'un envoi ou d'une réception, etc.). Par exemple, dans le cas d'un événement Alerte (qui hérite de Situation), l'identifiant de l'instance est la propriété ComponentSEID (Identifiant du composant du Système Etudié). Si l'instance est concernée par un événement de type Instruction ou Rapport, son identifiant sera la propriété ActivityID.

L'algorithme de mise à jour du modèle M_T est le suivant (Algorithme *MAJModeleTerrain(mTerrain, evenement)*). On suppose qu'à la réception de la notification de l'événement, ce dernier est extrait de « l'enveloppe » de la notification (qui est au format XML) et est transformé en l'objet Java correspondant. La manipulation de l'objet Java facilite l'accès aux propriétés de l'événement, telles que son type (*Risque*, *Conséquence*, ...), etc.. Ces informations sont nécessaires pour créer ou mettre à jour l'instance concernée par l'événement.

$MAJModeleTerrain(mTerrain, evenement)$

Cet algorithme permet de mettre à jour le modèle terrain

Paramètres : mTerrain : le modèle terrain de la situation collaborative
evenement : l'objet événement, transmis par le CEP

Variables : tablInstancesIds : le tableau de chaînes, qui contient les identifiants de tous les éléments composant le modèle terrain
instanceId : l'identifiant de l'instance d'un élément du modèle terrain, concerné par l'événement
drapeau : booléen permettant de vérifier si on a retrouvé l'instance (indiquée dans l'événement) à l'intérieur du modèle terrain

Début
| instanceId ← **Action** : Récupérer l'identifiant de l'instance porté par l'événement

...à suivre...

...suite...
tablInstancesId ← **Action** : Récupérer les identifiants des éléments du modèle terrain
drapeau ← FAUX
Si (tablInstancesIds <> vide) **Alors**
Pour i **variant de** 1 **à** taille(tablInstancesIds) **Faire**
Si (instanceId == tablInstancesIds[i]) **Alors**
Action : MAJElement(instanceId, evenement, mTerrain)
– ∴ – *On change la valeur du drapeau à VRAI*
drapeau ← VRAI
Fin Si
Fin Pour
Si (drapeau<>VRAI) **Alors**
Action : CreerElement(instanceId, evenement, mTerrain)
Fin Si
Fin Si
Fin

Cet algorithme prend donc en entrée le modèle *terrain* et l'événement reçu. Il va tester l'existence de l'élément concerné par l'événement dans le modèle terrain. Par exemple, si l'événement concerne une instance d'identifiant *H2G2_42*, l'algorithme va chercher l'existence d'une instance d'identifiant *H2G2_42* dans le modèle *terrain*. Si l'instance existe, alors l'action à mener est de type mise à jour (ce qui signifie modification ou suppression, suivant les autres données contenues dans l'événement). Si l'instance n'existe pas, cela signifie qu'il faut l'ajouter au modèle, et donc la créer.

Afin de savoir sur quel concept (Service ? Risque ? Consequence ? etc.) s'appuyer pour créer une nouvelle instance dans le modèle *terrain* M_T, ou pour modifier les propriétés d'une instance existante dans le modèle *terrain* M_T, les méthodes *MAJInstance()* et *CreerInstance()* prennent en paramètre l'objet *evenement*.

Le paramètre *instanceId* sert à indiquer l'identifiant de l'instance à mettre à jour ou à créer, selon le cas. Enfin, on passe également en argument le modèle *terrain* tel qu'il est juste avant la mise à jour.

A partir de cet objet, il est aisé de retrouver le type de l'événement et par là même le concept dont hérite l'instance concernée par l'événement. Les méthodes d'accès aux propriétés de l'objet événement permettent de récupérer les valeurs de ses attributs et de les affecter aux propriétés pertinentes de l'instance considérée.

Une table de correspondance entre les caractéristiques de l'événement entrant et les composants du modèle a été réalisée. Elle permet de retranscrire les informations portées par l'événement afin de mettre à jour les éléments concernés dans le modèle. Eût égard à son volume, l'ensemble de cette table de correspondance n'est pas présentée dans le corps de texte du manuscrit. Elle est disponible en Annexe J. Seul un extrait est présenté ci-dessous (Figure III.10).

			Modèle impacté : fieldModel.xml			
			Élements du modèle (et leurs caractéristiques) concernés			
			Risk	**Consequence**	**G/P/C/N**	**link**
Alert	Timestamp					
	UniqueID					
	Localisation	=	localisation	localisation		
	ComponentSEID	=	id	id		from *ou* to
	ComponentName	=	name	name		
	Description	=	description	description		
	Gravity	=	gravity	gravity		
	Type	=	*nom_element*	*nom_element*		
	ImpactedComponentSEID	=			id	from *ou* to

FIGURE III.10 – Correspondance entre l'événement *Alert* généré par le CEP et les éléments du modèle *terrain*

Cet extrait de la table de correspondance montre les équivalences entre les propriétés de l'événement *Alert* et les propriétés des éléments du modèle *terrain* qui sont concernés, à savoir *Risk* ou *Consequence* et *link*. La valeur de *ComponentSEID* portée par l'événement correspond à la valeur de l'élément concerné côté modèle *terrain*. Si jamais cet élément n'existe pas dans le modèle, cela signifie que l'événement va provoquer la création de cet élément dans le modèle *terrain*. Dans ce cas, pour savoir quel type d'élément créer, il suffit de récupérer la valeur de la propriété *Type* de l'événement (qui prend pour valeur soit *Risk* soit *Consequence*).

En outre, un *Risk* ou une *Consequence* s'appliquant sur un composant du système étudié, un lien doit être créé entre le nouveau *Risk* (ou *Consequence*) créé et le *Goods* (ou *People*, *NaturalSite*, *CivilianSociety*) impacté. Pour cela, un élément *link* sera également créé dans le modèle *terrain* et ses attributs *from* et *to* prendront les valeurs de *ComponentSEID* et de *ImpactedComponentSEID*.

Conclusion intermédiaire Compte tenu des considérations architecturales précédentes et de la présentation des moyens mis en œuvre pour réaliser la mise à jour des modèles, nous pouvons compléter la Figure III.4 vue en Section III.1.4 par la Figure III.11

Les événements émis par la collaboration et son environnement sont collectés par le moteur de CEP (via abonnement aux huit types d'événements propres au contexte de la crise). Les EPA du CEP se partagent deux rôles : celui de simplement filtrer les événements typés *Activité*, et celui de filtrer puis créer de nouveaux événements sur la base de ceux entrants (de type autre qu'*Actitivé*). Ces événements filtrés ou produits par le CEP sont respectivement émis sur les sujets *eventExpected* et *eventField*.

L'outil de mise à jour des modèles s'abonne à ces deux sujets. Suivant le sujet dont est issu l'événement reçu, l'outil va mettre à jour soit le modèle *attendu* (événement dont le sujet d'abonnement est *eventExpected*) soit le modèle *terrain* (événement dont le

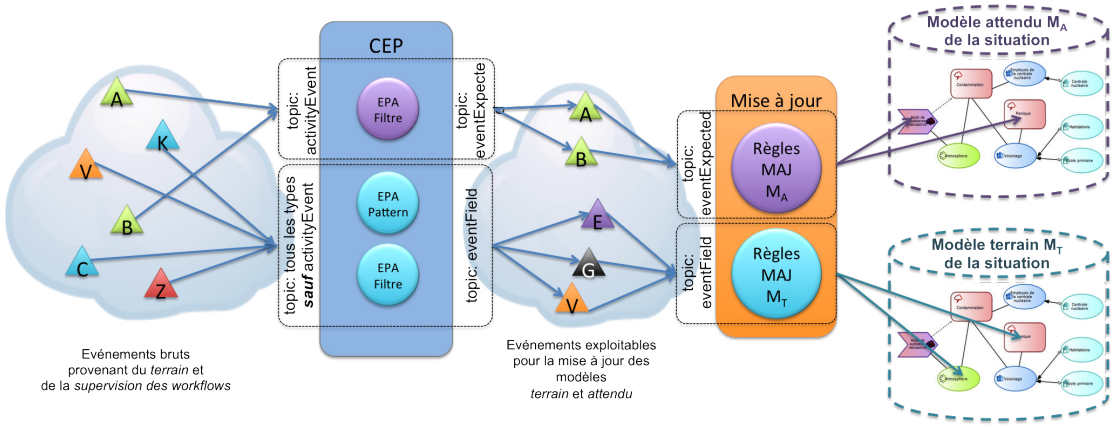


FIGURE III.11 – Des évéments à la mise à jour des modèles : un traitement de type CEP et une analyse des événements émis par le CEP

sujet d'abonnement est *eventField*). Pour chaque modèle, l'outil de mise à jour dispose de règles de traitement qui font appel à la correspondance entre les informations portées par les événements entrants et le contenu du modèle.

III.3.6 La notation des règles métier

Les règles métier introduites dans le moteur de CEP sont définies par les partenaires de la collaboration : issues de leur retour d'expérience ou de décisions pré-établies, elles permettent de soulager la tâche de la cellule de crise en détectant automatiquement des situations intéressantes à travers l'analyse des événements reçus par le moteur de CEP.

Afin de capitaliser l'expérience des collaborations passées, un catalogue de règles peut être créé et mis à disposition du moteur de CEP. Lorsque les événements reçus par le CEP correspondent à un motif d'événements permettant de déclencher une règle, le moteur de CEP ne l'exécute pas immédiatement. Il propose à l'utilisateur d'appliquer ou non cette règle. En effet, suivant le contexte de la situation collaborative et l'instant dans lequel elle s'inscrit, certaines règles seront ou non pertinentes. On peut donc mettre en place un système de note des règles métier : plus une règle est utilisée, plus elle est pertinente.

Si l'utilisateur décide d'appliquer la règle (respectivement refuse de l'appliquer), la note de la règle sera augmentée (respectivement diminuée). En plus de la note, un horodatage de la décision est enregistré : il permet de replacer la décision dans le contexte où elle a été prise, facilitant ainsi le retour d'expérience.

La mise en place du système de notation permet à terme de mettre en avant certaines règles suivant le domaine métier et le contexte de la collaboration. Elle permet aussi d'avoir un autre point de vue du cycle de vie de la collaboration au moment de la formalisation du retour d'expérience à la fin de la collaboration.

III.4 Conclusion

L'utilisation conjointe de l'EDA et de la SOA autorise un couplage lâche entre les composants mis en jeu dans le maintien à jour des modèles : facilité de communication entre le CEP et l'outil de mise à jour, facilité d'abonnement/désabonnement à des sources d'événements, déduction de nouveaux événements pertinents en vue d'une mise à jour du modèle terrain, etc. La mise en place de cette architecture ED-SOA permet de rendre la représentation de la situation collaborative sensible au contexte.

La notation des règles métier permet quant à elle d'évaluer la pertinence des règles proposées pour déduire de nouveaux événements. Ces événements participent à la mise à jour des modèles : ils ont donc une influence directe sur la qualité des données sur lesquelles l'étape de détection sera réalisée et sur son résultat, à savoir la décision d'adaptation du système de traitement. Support d'aide à la décision, la notation des règles peut devenir support au retour d'expérience à la fin de la collaboration : telle règle aurait-elle dû être appliquée ? Telle règle était-elle bien définie ?

La mise à jour des modèles de la situation collaborative fait aussi appel aux informations récoltées au sujet du système de traitement (partenaires, activités, conditions, etc.). Cette connaissance est également sujette à des changements traduits par les événements : un partenaire qui s'en va de la collaboration, et c'est l'ensemble des compétences qu'il propose à la collaboration qui devient indisponible. La résolution des éventuels problèmes causés par un changement et sa retranscription dans la connaissance du système de traitement sera faite dans l'étape d'adaptation du Service d'Agilité.

Chapitre IV

Comparaison des modèles : Détection (partie 2)

L'étape de détection se structure en deux parties : tout d'abord la mise à jour des éléments caractérisant la situation collaborative (telle qu'expliquée dans le chapitre précédent), et enfin la comparaison proprement dite. Dans ce chapitre, le périmètre de

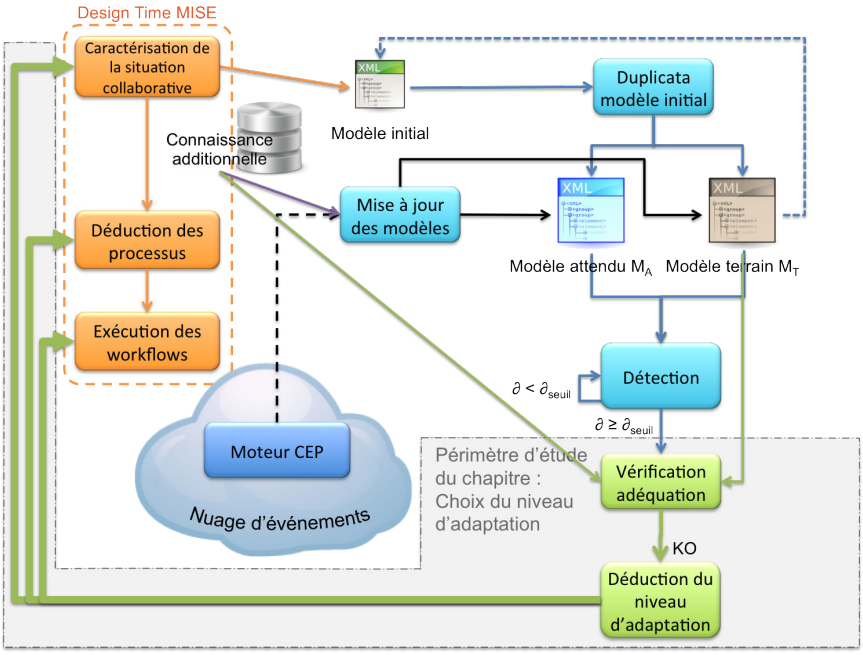


FIGURE IV.1 – Architecture simplifiée du Service d'Agilité et périmètre de l'étude dans le Chapitre IV

l'étude (Figure IV.1) porte dans un premier temps sur la présentation des mécanismes de comparaison des modèles *terrain* et *attendu* entre eux, puis sur l'étude des résultats de la comparaison effectuée afin de pouvoir qualifier l'amplitude de la différence entre les deux modèles. En effet, la comparaison des modèles ne peut s'arrêter à la simple

détection d'une différence. Il faut également analyser en profondeur les causes de cette différence, afin de pouvoir en recueillir la connaissance nécessaire pour, plus tard, diriger une adaptation des processus collaboratifs qui soit la plus pertinente possible compte tenu de la caractérisation de la situation collaborative à l'instant t.

Pour cela, nous allons dans un premier temps préparer les modèles de la situation, i.e. retirer toutes les données superflues et réorganiser les données (afin de faciliter le travail de détection). Cette préparation des modèles va être effectuée via deux transformations XSL (exécutées l'une à la suite de l'autre). Ensuite, nous allons définir notre méthode de calcul de la divergence δ, qui passe par la mise en place de matrices de coût —au sens de poids, **w**eight— de l'opération **w** (ajout, suppression, modification) et d'i**m**portance **m** (pour qualifier un élément du modèle par rapport aux autres issus du même concept). Enfin, nous détaillerons les mécanismes des algorithmes permettant la détection des différences et le calcul de la divergence δ.

IV.1 Détection des différences entre modèles

La phase de détection se déroule en deux temps : tout d'abord, les fichiers XML des modèles sont nettoyés des informations inutiles à travers un jeu de transformations. Ensuite, la recherche d'une (ou des) différence(s) entre les modèles est effectuée.

IV.1.1 La préparation des modèles

Comparer les modèles *terrain* et *attendu* entre eux implique de s'intéresser au contenu des modèles d'un point de vue technique, c'est-à-dire de se pencher sur le format et la façon dont les données sont écrites dans les fichiers XML des modèles. Afin de faciliter le travail de comparaison des modèles, nous allons retirer toutes les données qui ne sont pas pertinentes des fichiers XML correspondants.

Les modèles *terrain* et *attendu* sont issus à l'origine de la duplication du modèle initial de la situation. Ce dernier étant réalisé via un modeleur graphique, le fichier du modèle contient non seulement les données du modèle lui-même, mais aussi toutes les informations nécessaires à la représentation graphique du modèle. Nous avons vu en Section III.3.3.2 que le modeleur prenait en charge le retrait de ces informations là. Nous travaillons donc sur des modèles exempts de données concernant l'interface graphique.

Les fichiers XML des modèles de situation contiennent des informations qui ne nous seront pas utiles dans la suite des traitements. Après étude du contenu des fichiers, nous remarquons que les espaces de noms des éléments utilisés pour décrire les différentes instances des concepts sont les mêmes pour le modèle *terrain* et *attendu*. Ceci provient du fait que ces deux modèles sont issus de la duplication du modèle *initial* de la situation collaborative et qu'ils sont tous les deux issus du même méta-modèle (l'ensemble de la grammaire XML définie par la mise en oeuvre du méta-modèle appartient au même espace de nom). Il n'est donc pas nécessaire de conserver ces informations sur les espaces

de noms, afin de ne pas gaspiller de ressources dans le traitement de la détection de différence. De plus, l'organisation des données à l'intérieur des fichiers XML ne nous permet pas d'optimiser notre traitement de la détection des différences (et de le rendre générique pour qu'il s'applique à des workflows collaboratifs autres que ceux de la réponse à la crise). Pour cela nous allons réaliser deux transformations XSL : la première a pour but de retirer les données superflues, la seconde réorganise les données d'une façon plus exploitable pour réaliser la comparaison des modèles.

IV.1.1.1 La transformation XSL

Avant de détailler la réalisation des deux transformations XSL, nous allons effectuer un bref rappel de ce qu'est XSL et avoir un aperçu des outils disponibles pour réaliser les transformations (dans le cadre d'une programmation en langage Java).

XSL signifie *eXtensive Stylesheet Langage*, ou langage extensible de feuille de style. Cette abréviation recouvre en fait trois langages :

- XPath signifie *XML Path Language* [Clark, 1999] désigne un moyen d'accéder à un noeud quelconque de l'arborescence d'un document XML à partir d'un autre noeud quelconque,
- XSLT signifie *eXtensible Stylesheet Langage Transformation* [Berglund, 2006]. Il en existe deux versions (1.0 et 2.0). La version 3.0 est à l'heure actuelle à l'état d'étude. Il s'agit du langage de transformation XSL, qui permet de transformer un document XML en un autre document XML ou en un document HTML.
- XSL-FO signifie *eXtensible Stylesheet Langage-Formating Objects* [Berglund, 2006], et désigne un langage permettant le contrôle de la mise en page finale de la transformation. Ce langage est particulièrement destiné à la production de contenus au format PDF. Ce langage ne nous intéresse pas dans le contexte de ces travaux, n'ayant pas de transformation vers le format PDF à effectuer.

Comme son nom l'indique, le langage XSLT ne sert pas uniquement de langage de feuille de style. Il est aussi un très puissant manipulateur d'éléments. Il permet de transformer un document XML source en un autre, permettant ainsi, à l'extrême, d'en bouleverser la structure. C'est ce que l'on appelle des transformations XSL ou XSLT. Le fonctionnement du XSLT est fondé sur les manipulations de modèles (templates). Les éléments du document XML d'origine sont remplacés (ou modifiés) par ces modèles. Un modèle contient ainsi le texte (éléments, attributs, texte...) de remplacement d'un élément donné.

Tout élément pouvant être remplacé dans le fichier de sortie par tout type de contenu texte, XSL est un outil privilégié de production de fichiers XML ou HTML à partir de sources XML. PHP fait ainsi appel à des bibliothèques de procédures de type XSL comme libxslt quand il doit gérer l'interface avec des bases de données XML. Java propose aussi plusieurs implémentations de XSLT à travers ses bibliothèques, telles que :

1. JAXP (Java API for XML Processing) est l'API embarquée par défaut dans le JDK (Java Development Kit). Elle permet de manipuler, créer et traiter des documents XML. Il s'agit de l'API que nous utilisons ici,

2. JDOM : JDOM est une API qui permet de travailler avec des documents XML. Son fonctionnement est comparable à DOM (Document Object Model) [1],
3. DOM4J est une API Open Source Java permettant de travailler avec XML, XPath et XSLT,
4. StAX (Streaming Api for XML) propose des fonctionnalités pour parcourir et écrire un document XML mais ne permet pas de manipuler le contenu d'un document,
5. JAXB : Java Architecture for XML Binding est l'API Java d'Oracle qui permet de créer des classes Java à partir de Schémas XML et inversement. Elle facilite l'utilisation du XML en Java en compilant un schéma XML dans une ou plusieurs classes.

Dans ces travaux de thèse, nous utiliserons les API JAXP (pour réaliser les transformations XSLT) et JAXB (pour accéder plus aisément aux données contenues dans les flux XML). Notons au passage qu'un fichier XSL est un fichier XML, et qu'il doit par conséquent respecter les normes de syntaxe de ce format.

IV.1.1.2 La suppression des espaces de nom

La première transformation XSL fait appel à la feuille de style suivante (cf. Code IV.1). Cette feuille de style utilise comme condition d'exécution des modèles (*template*) d'une part les expressions XPath « étoile » * (i.e. tous les éléments), et d'autre part les expressions XPath @* (i.e. tous les attributs). Cela indique que tous les éléments et tous les attributs des éléments du document XML sont concernés par les instructions contenues dans les modèles.

Les instructions de la feuille XSL étudiée permettent de cloner les éléments et attributs du document XML d'entrée tout en supprimant les espaces de noms dont ils héritaient du modeleur.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3      <xsl:output method="xml" indent="yes"/>
4      <xsl:template match="*">
5          <xsl:element name="{local-name(.)}">
6              <xsl:apply-templates select="@* | node()"/>
7          </xsl:element>
8      </xsl:template>
9      <xsl:template match="@*">o
10         <xsl:attribute name="{local-name(.)}">
11             <xsl:value-of select="."/>
```

1. DOM permet de construire une arborescence de la structure d'un document et de ses éléments. Il est donc préférable de parcourir et de mémoriser l'intégralité du document avant de pouvoir effectuer les traitements voulus. Pour cette raison, les programmes utilisant DOM ont souvent une consommation de mémoire volumineuse en cours de traitement. À l'inverse, à partir d'un arbre DOM donné, il est possible de générer des documents dans le langage de balisage voulu, qui pourront à leur tour être manipulés par l'interface DOM.

```
12          </xsl:attribute>
13      </xsl:template>
14  </xsl:stylesheet>
```

CODE IV.1 – Feuille XSL pour la transformation permettant de retirer les espaces de nom inutiles

Le premier modèle (template) se positionne à la racine du document et se charge de récupérer tous les éléments existants(lignes 4 à 8). Le nom de chaque élément récupéré permet de créer un élément de même nom que celui récupéré mais sans *l'espace de nom* (ligne 5). Ensuite (ligne 6), on regarde s'il contient des éléments enfants (qui subissent le même traitement que leur parent) ou des attributs (@*). Dans ce cas, un second modèle est déclenché (lignes 9 à 13). Il créé un nouvel attribut, qui prend pour nom celui de l'attribut récupéré, toujours sans *l'espace de nom* (ligne 10), et on lui affecte la même valeur que celle de l'attribut récupéré (ligne 11). La Figure IV.2 nous montre le résultat de la transformation : nous obtenons un fichier XML identique à celui d'origine, exception faite de la déclaration des espaces de nom (les lignes entourées dans le document source ne sont pas retrouvées dans le document résultat.

FIGURE IV.2 – Résultat de la première transformation XSL : les espaces de nom ont été supprimés

IV.1.1.3 Le formatage des données

Cependant, après cette première transformation, nous pouvons constater que l'organisation des données dans le document XML ne facilite pas son exploitation : certaines instances de concepts (i.e. certains éléments qui portent le même nom) sont parfois regroupés dans un élément parent, parfois non.

Afin de pouvoir rendre le traitement de ce fichier le plus générique possible, c'est-à-dire que n'importe quelle représentation d'une situation collaborative s'appuyant sur le méta-modèle cœur et l'une de ses sur-couches dédiées à un domaine précis puisse être analysé par l'outil de comparaison défini au sein du Service d'Agilité (nous voulons traiter aussi bien les modèles de situation de crise nucléaire que de crise routière, ou encore les situations de sous-traitance industrielle, etc.). Pour cela, nous souhaitons baser notre

travail d'étude des données en nous affranchissant de la connaissance exacte des noms des concepts utilisés dans le modèle, c'est-à-dire du nom des éléments présents dans le fichier XML. A cette fin, nous avons décidé de formater les fichiers XML des modèles *terrain* et *attendu*. Toutes les instances d'un même concept se trouvent regroupées dans un même élément père (voir Figure IV.3). Ainsi, quel que soit le nombre et l'intitulé des concepts portés par le méta-modèle du domaine considéré (crise en général, crise nucléaire, sous-traitance, etc.), le mécanisme de récupération des données des deux fichiers XML et de leur comparaison est générique.

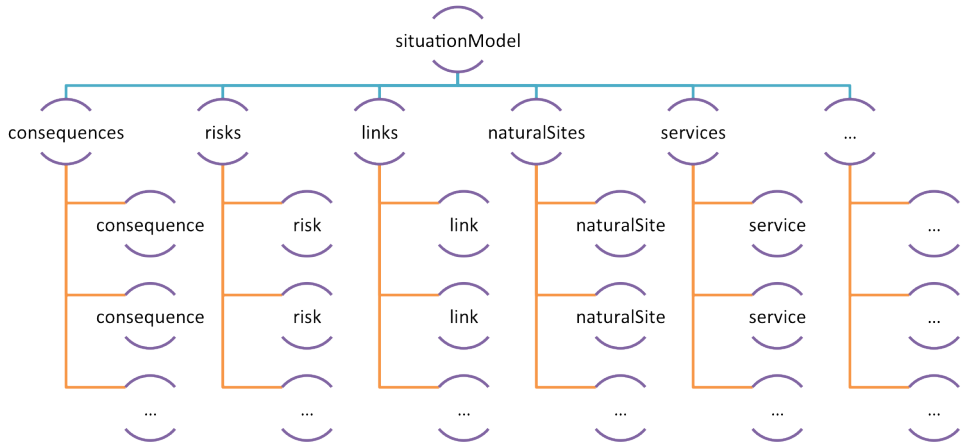


FIGURE IV.3 – Principe de la nouvelle structure des données des fichiers XML des modèles terrain et attendu

Le formatage des modèles *terrain* et *attendu* peut également se faire via une transformation XSL. Il est à noter que dans le cas de cette transformation, la feuille de style est dépendante du domaine étudié. En effet, il est nécessaire de connaître le nom des balises du modèle et leur hiérarchie afin de décrire les règles de transformation. Nous avons fait le choix de reporter la dépendance au domaine sur cette transformation plutôt que sur l'algorithme de comparaison en lui-même : en effet, il est plus aisé de modifier une feuille XSLT plutôt que de changer la logique d'un programme. La feuille de style de cette seconde transformation est disponible en Annexe K. Cette feuille XSL est dépendante du domaine de la gestion de crise, compte tenu du cas d'étude abordé dans ces travaux. Elle permet de récupérer chacune des instances présentes (*risk, consequence*, etc.) dans le fichier XML des modèles obtenus après la suppression des espaces de noms, et les encapsule dans un élément père (*risks, consequences*, etc.). Le fichier résultat de cette transformation se présente de la façon suivante (voir Code IV.2).

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <situationModel>
3      <consequences>
4          <consequence id="C_000124" description="Firemen report #3: fighting fire on
               plant 3 requires more human means"/>
5      </consequences>
```

```
 6      <risks>
 7          <risk id="R_000123" description="Fire" gravity="HIGH" localisation="Plant 3"/
                >
 8          <risk id="R_000126" description="Contamination" gravity="HIGH" localisation="
                Plant 3"/>
 9          <risk id="R_000128" description="Contamination" gravity="LOW" localisation="
                Plant 4"/>
10          <risk id="R_000186" description="Fire" gravity="MEDIUM" localisation="Plant 4
                "/>
11      </risks>
12      <services>
13          <service id="S_000456" description="Extinguish fire" state="FAILED"/>
14          <service id="S_000467" description="Evacuate people" state="IN_PROGRESS"/>
15      </services>
16      <linkService_Risks>
17          <linkService_Risk id="LSR_123" serviceId="S_000456" riskId="R_000123"/>
18          <linkService_Risk id="LSR_128" serviceId="S_000467" riskId="R_000126"/>
19      </linkService_Risks>
20  </situationModel>
```

Code IV.2 – Extrait du modèle *terrain* après formattage

Nous pouvons constater dans le Code IV.2 que chacun des éléments du modèle de la situation est enfant d'un élément de plus haut niveau. Les différents éléments sont ainsi triés par concept : tous les *<service>* sont enfants d'un unique élément *<services>* (lignes 12 à 15), tous les *<risk>* ont pour seul parent *<risks>* (lignes 6 à 11), etc. Cette hiérarchisation et cette organisation des données permettra par la suite de faciliter le travail de détection d'une différence entre les modèles *terrain* et *attendu* (i.e. entre les arbres de données représentant ces deux modèles).

IV.1.2 XMLUnit

Parmi les outils implémentant un algorithme de type *diff*[2] adapté aux arbres XML, nous avons pu voir dans le Chapitre II que XMLUnit était le meilleur candidat vis-à-vis de nos attentes : accès au détail des différences identifiées, prise en compte de la similarité entre documents, etc. Nous allons ici détailler notre « détournement » de cet outil de test unitaire dédié au XML pour réaliser notre outil de détection des différences entre les fichiers XML des modèles *attendu* et *terrain*.

IV.1.2.1 Principe de fonctionnement

Par défaut, l'algorithme de comparaison de deux flux XML considère ces deux derniers comme :

1. *Identiques* si aucune différence n'a été trouvée,

2. *diff* est un algorithme de comparaison de deux fichiers, initialement implémenté dans le système Unix.

2. *Similaires* si les différences trouvées sont « récupérables », c'est-à-dire qu'elles n'ont pas d'impact sur le sens des données. Les différences entrant dans ce cas sont au nombre de deux :
   - Deux éléments dans les deux documents XML contiennent les mêmes éléments enfants mais dans un ordre différent,
   - Un attribut dont la valeur par défaut (décrite dans le Schéma XML des deux documents XML) a été explicitée dans l'un des documents mais pas dans l'autre.
3. *Différents* dès qu'une des différences identifiées n'est pas « récupérable ».

IV.1.2.2 Le moteur de comparaison

La documentation de XMLUnit étant des plus succintes en ce qui concerne le fonctionnement du moteur de comparaison, nous avons dû nous immerger dans le code source de cet outil programmé en Java pour comprendre ses mécanismes.

Il apparaît dans un premier temps que si XMLUnit propose une classe *Diff* grâce à laquelle on appelle la méthode de comparaison, celle-ci n'offre pas un *diff* au sens UNIX traditionnel du terme (cf. Section II.2.2.1). En effet, elle permet d'évaluer la comparaison de documents XML en termes d'identité, de différence mais aussi de similarité et retourne un message décrivant la première différence trouvée (par exemple : nombre de noeuds différents, élément ajouté/supprimé, etc.).

Si l'on souhaite pouvoir accéder au détail des différences trouvées entre les deux documents, il faut utiliser la classe *DetailedDiff*. Cette classe *DetailedDiff* hérite de *Diff* et en étend les fonctionnalités. Elle permet de comparer deux documents XML et la comparaison ne s'arrête pas à la première différence non récupérable trouvée (contrairement à *Diff*). La classe permet d'accéder par la suite à la liste détaillée de toutes les différences rencontrées. Dans l'implémentation de notre prototype, nous utiliserons *DetailedDiff* afin de pouvoir réaliser l'analyse de chaque différence identifiée.

Diff et *DetailedDiff* font tous les deux appel au coeur de la fonctionnalité de comparaison des documents XML de XMLUnit qui se trouve implémenté dans la classe *DifferenceEngine*. Cette classe compare les noeuds XML (objets Node) entre eux et notifie l'écouteur *DifferenceListener* de toute différence ou dissimilarité trouvée.

La classe *DifferenceEngine* fait appel principalement à la méthode *compare()* qui prend quatre paramètres en entrée : le noeud XML de référence (*control*), le noeud XML à tester (*test*), l'écouteur (*listener*) qui est notifié de toute différence détectée durant la comparaison des noeuds entre eux, et une interface *elementQualifier* qui indique au moteur de comparaison les types de données sur lesquelles effectuer la comparaison (uniquement les éléments, uniquement les attributs, les éléments et les attributs, etc.). Dans notre usage de XMLUnit, l'*elementQualifier* est paramétré à *ElementNameAndAttributeQualifier()* : il permet de réaliser la comparaison sur les noms et les valeurs de

tous les éléments et attributs présents dans les documents tout en autorisant la similarité entre les documents (l'ordre entre les éléments frères et soeurs n'a pas d'importance).

Dans un premier temps cette méthode vérifie que les noeuds XML sont comparables entre eux via les méthodes *compareNode()* et *compareNodeBasics()* : ces méthodes vont comparer le type du noeud issu du flux de référence et le type du noeud provenant du flux de test. Ceci permet de garantir que l'on compare des éléments avec des éléments, des attributs avec des attributs, des commentaires avec des commentaires, etc.

compareNodeBasics() va vérifier que le noeud de référence et le noeud de test contiennent des données de type texte ou CDATA [3].

compareNode() prend en paramètres le noeud XML de référence (*control*), le noeud XML à tester (*test*), l'écouteur (*listener*) l'interface *elementQualifier*. Cette méthode va vérifier les types des noeuds. Les valeurs des types de noeuds utilisés dans XMLUnit suivent la recommandation du W3C [Le Hors et al., 2004].

Ensuite, si les types sont identiques, la méthode ad hoc va être appelée pour effectuer la comparaison sur ces noeuds (nom et valeur des éléments et attributs) :

- compareElement(control, test, listener) dans le cas de deux noeuds de type *Element*,
- compareText(control, test, listener) dans le cas de deux noeuds de type *Text* ou *CDATASection*,
- compareComment(control, test, listener) dans le cas de deux noeuds de type *Comment* (commentaire),
- compareDocumentType(control, test, listener) dans le cas de deux noeuds de type *DocumentType* (interface pour les entités définies dans le document XML),
- compareProcessingInstruction(control, test, listener) dans le cas de deux noeuds de type *ProcessingInstruction* (instruction de traitement),
- compareDocument(control, test, listener) dans le cas de deux noeuds de type *Document* (document complet, i.e. le nœud racine de l'arborescence DOM),

Ensuite, la méthode *compareNode()* vérifie si les noeuds ont éventuellement des enfants : si c'est le cas, la méthode s'appelle elle-même pour comparer les noeuds enfants entre eux.

Dans les autres cas de types de noeuds, la comparaison n'est pas effectuée car XMLUnit n'implémente pas les méthodes correspondantes. Cette limite technique n'affecte pas le bon déroulement de la comparaison dans le cadre de nos travaux étant donné la structure de nos fichiers XML de modèles de situation qui ne contiennent que des

3. CDATA signifie Character Data. Il s'agit d'un balisage particulier qui permet de définir une section dont le contenu ne sera pas analysé par le compilateur XML lors de la lecture du fichier. Le texte sera donc affiché tel qu'il est écrit dans cette section. Ce type de section est très utile si l'on doit utiliser beaucoup de caractères réservés —&, >, <, etc.— et que l'on souhaite réduire l'écriture répétitive des échappements de ces caractères

éléments ayant éventuellement des éléments fils et des attributs.

IV.2 Coût de l'opération et importance de l'instance

Dans cette section, nous allons introduire les principes du calcul de la divergence. Dans le Chapitre II, nous avons choisi l'outil XMLUnit afin de détecter les différences entre deux arbres XML non ordonnés. Cependant, calculer une divergence ne se limite pas à détecter des différences. Il faut pouvoir donner un sens à ces différences afin de pouvoir obtenir la distance « réelle » entre les deux arbres XML (i.e. les modèles *attendu* et *terrain*). Le calcul de la distance entre les modèles *attendu* et *terrain* fait à la fois appel aux notions de coût, c'est-à-dire au coût minimal des opérations pour passer du modèle *attendu* au modèle *terrain*, et d'importance, c'est-à-dire à l'importance que l'on peut donner à une différence détectée plutôt qu'à une autre.

IV.2.1 Le calcul de la divergence

Une approche du calcul de la divergence peut consister à ne comptabiliser que le nombre d'opérations nécessaires sur les caractères pour passer d'une chaîne A à une chaîne B, comme le propose la distance de Levenshtein [Levenshtein, 1966]. Cependant, dans notre cas, cette approche peut se révéler fastidieuse pour rechercher les différences entre des modèles et exploiter le résultat du calcul de la distance. Une autre approche du calcul de la *distance d'édition* a été généralisée aux arbres et formalisée par [Tai, 1979]. La distance entre deux arbres[4] *ordonnés* est définie comme étant le coût minimal des opérations (insertion, suppression, modification d'un nœud) à effectuer pour passer d'un modèle à l'autre. Cette expression de la distance est présentée dans le Lemme 1. Plus la distance est petite, plus les arbres sont similaires.

Lemme 1

$$\delta(F,G) = min \begin{cases} c_{suppression}(\Gamma_F) + \delta(F - \Gamma_F, G); (1) \\ c_{suppression}(\Gamma_G) + \delta(F, G - \Gamma_G); (2) \\ c_{modification}(\Gamma_F, \Gamma_G) \ + \ \delta(T(\Gamma_F) - \Gamma_F, T(\Gamma_G) - \Gamma_G) \\ \ + \ \delta(F - T(\Gamma_F), G - T(\Gamma_G)); (3) \end{cases}$$

Où :

- F et G sont deux forêts,
- Γ_F (respectivement Γ_G) est le nœud racine de l'arbre situé le plus à droite de la forêt F (respectivement G),
- F-$T(\Gamma_F$ (respectivement G-$T(\Gamma_G)$ est la forêt obtenue après entière suppression de l'arbre le plus à droite de F (respectivement G),
- $c_{suppression}(v)$ est de le coût de la suppression du nœud v,

4. Un arbre est un graphe acyclique organisé de manière hiérarchique à partir d'un nœud appelé racine.

- $c_{modification}(a,b)$ est le coût du renommage d'un nœud a pour le transformer en un nœud b,
- $\delta(F,G)$ est la distance d'édition entre les arbres F et G. Elle est évaluée récursivement,
- Les trois opérations de la distance d'édition (1), (2) et (3) représentent respectivement la suppression du nœud le plus à droite de F, du nœud le plus à droite de G et du renommage du nœud le plus à droite de F par le nœud le plus à droite de G.

Nos arbres de données XML ne sont pas ordonnés donc cette démarche de calcul de la distance d'édition par la droite et toutes celles qui en découlent (comme la stratégie de décomposition par la gauche de [Zhang and Shasha, 1989]) ne nous intéressent pas pour une application directe. En revanche, nous allons nous inspirer de cette démarche en intégrant de coût de l'opération pour toute différence de type ajout, suppression ou modification de nœud. L'aspect désordonné de nos arbres ne pose pas de problème dans la recherche de différences car XMLUnit permet d'ignorer les différences d'ordre dans les relations frères-soeurs lors de la recherche d'écart entre deux arbres XML.

IV.2.2 Le coût de l'opération

Si le coût d'une insertion, suppression ou modification est généralement égal à 1, cette évaluation du coût d'une opération ne convient pas dans le cadre de ces travaux. En effet, compte tenu du contexte dans lequel ce calcul de distance est effectué (à savoir déterminer les évolutions du contexte d'exécution des processus), il serait naturel de différencier ces coûts suivant le type de propriété du contexte concerné (risque, conséquence, service, etc.). En effet, l'ajout d'un risque dans le modèle *terrain* indique que la situation de la collaboration est potentiellement en train de se dégrader et a donc un impact plus important dans le calcul de la distance que l'ajout d'un service, élément potentiellement bénéfique et avec une portée sur la pertinence des processus collaboratifs a priori plus réduite.

Autrement dit, le coût de l'opération unitaire (insertion, suppression, modification) doit s'exprimer en fonction de l'élément sur lequel l'opération porte.

On peut ainsi définir le coût de l'opération de la façon suivante :

Formule 3 $coût_{opération} = f(operation, concept)$

Où :

- L'*opération* est le type d'édition permettant de passer d'un document à l'autre. Les valeurs possibles sont *insertion*, *suppression*, *modification*.
- Le *concept* est le type de caractéristique de l'instance du modèle de la situation qui est impacté. Il prend pour valeur l'un des concepts existant dans le modèle de la situation considéré. Dans le cas de la situation de crise nucléaire, les valeurs peuvent être : risque, conséquence, facteur de gravité, facteur de complexité, service, etc.

Par convention, le $coût_{opération}$ n'est jamais égal à 0 car même si la différence identifiée est considérée comme mineure, elle existe et ne doit pas être annulée dans le calcul du *delta*. L'évaluation se fait sur une échelle ouverte débutant à 1 (coût minimum), le pas étant de 1.

Les valeurs sont définies par les partenaires de la collaboration, au moment de la caractérisation de la situation collaborative. Ces valeurs peuvent être modifiées au cours du cycle de vie de la collaboration si nécessaire. La liste des coûts peut être représentée par une matrice {operation × concept} contenant l'ensemble des valeurs des coûts en fonction des concepts et des opérations, de la manière suivante :

TABLE IV.1 – Matrice {operation × concept}

		Opération		
		Insertion	Suppression	Modification
Concept	Partner	1	4	3
	Resource	1	3	1
	Service	1	3	2
	Risk	3	1	2
	...	....	...	...

Cette matrice devient un paramètre d'entrée dans la détection des différences entre les deux documents : dès qu'une différence est détectée par le moteur de XMLUnit et que cette différence est validée par notre algorithme, le coût de l'opération (vis-à-vis du concept impacté) est lu dans la matrice et enregistré pour qualifier la différence.

Dans notre prototype, la matrice est implémentée sous forme de fichier XML, conçu de la façon suivante (Code IV.3).

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <parameters>
3  <threshold>5</threshold>
4  <weight id="partner" added="1" deleted="4" updated="3"/>
5  <weight id="resource" added="1" deleted="3" updated="1"/>
6  <weight id="service" added="1" deleted="3" updated="2"/>
7  <weight id="risk" added="3" deleted="1" updated="2"/>
8  </parameters>
```

CODE IV.3 – Extrait du fichier XML contenant les valeurs de la matrice {operation x concept} ainsi que la valeur du seuil δ_{seuil}

L'élément *weight* possède quatre attributs : *id* qui prend pour valeur le nom du concept, *added*, *deleted* et *updated* qui prennent respectivement les valeurs du coût de l'opération pour un ajout, une suppression et une modification.

IV.2.3 L'importance de l'instance

La nature même des données sur lesquelles nous travaillons nous amène au constat suivant : deux instances d'un même concept peuvent avoir une importance différente, qui exprime leur poids dans le calcul de la distance entre les deux modèles.

Par exemple, prenons deux risques qui sont ajoutés dans le modèle *terrain* : le premier concerne un risque de contamination de la population par le virus Ebola, le second un risque de contamination de la population par un simple rhume. Il est évident que le risque de contamination par Ebola est plus grave que le risque de contamination par un rhume. Son critère d'importance sera plus élevé que celui du risque de rhume.

L'introduction d'un critère d'importance m_i permet de représenter le caractère indispensable ou grave d'une instance de concept, et de classer les instances de concepts entre elles. Ce critère m_i a une valeur comprise entre 0 (valeur exclue) et 1 (valeur inclue).

Cependant, cette matrice de l'importance des instances est sensible à l'environnement de la collaboration — plus exactement à deux de ses composantes : le temps et le contexte. Elle n'est pas figée une fois pour toutes. Il est possible de la faire évoluer au fur et à mesure du cycle de vie de la collaboration, pour refléter les évolutions des appréciations des éléments caractéristiques de la situation collaborative.

IV.2.3.1 Sensibilité au temps

La sensibilité au temps retranscrit le fait qu'entre l'instant t_0 et l'instant t, la façon dont une instance est perçue peut changer. Par exemple, au début de la réponse à une crise nucléaire, le partenaire *IRSN* a une valeur plus importante que le partenaire *Police municipale* : expert scientifique, le rôle de l'IRSN est indispensable pour aider à la prise de décision concernant l'établissement d'un périmètre de sécurité. Par contre, à la fin de la gestion de la crise, l'importance de la Police municipale devient plus importante (pour refléter son rôle de protection de la population pour la ramener dans ses habitations, réguler les problèmes de trafic) et celle de l'IRSN décroît.

IV.2.3.2 Sensibilité au contexte

La sensibilité au contexte s'intéresse à la perception de l'instance par rapport aux conditions dans lesquels la collaboration s'effectue. Comme tenu du contexte, une instance peut alors être considérée comme neutre (sans effet notable sur la collaboration et ses workflows), soit positive, soit négative. Prenons l'exemple d'un *Risque* de pluie. Dans le cas de la réponse à un incendie de grande ampleur, ce risque va avoir une faible importance. Par contre, dans le cas de la réponse à des inondations, cette instance de risque n'est pas négligeable. En effet, l'arrivée de ce nouveau risque peut compromettre la poursuite de la réponse à la crise, voire empirer la situation de crise actuelle.

IV.2.3.3 Stockage

La valeur de l'importance pourrait être indiquée directement dans la définition de l'instance lors de son ajout au modèle. L'utilisation d'une ontologie permettrait de capitaliser la connaissance sur les critères d'importance en fonction des estimations effectuées par les partenaires de la collaboration lors des collaborations passées et de déduire la valeur de la gravité des nouvelles instances si aucune information sur leur importance n'a été donnée.

Typiquement, si un nouveau service *Eteindre un feu de maison* est introduit dans le modèle terrain et qu'aucune information n'a été donnée quant à la valeur de son importance, le raisonnement sur l'ontologie permet de retrouver le service le plus proche de *Eteindre un feu de maison* (comme *Eteindre un feu domestique*) et d'en récupérer la valeur du critère d'importance pour l'affecter à *Eteindre un feu de maison*.

Ceci étant, la solution du stockage des importances dans une ontologie rend difficile cette prise en compte du temps et du contexte. Dans un premier temps, une solution est de définir manuellement un fichier contenant les tuples {instance ; importance}. Lors du calcul du δ, l'outil de détection ira lire les valeurs des importances pour chacune des instances présentes dans le modèle de la situation

IV.2.4 Le calcul de la distance

Compte tenu des considérations précédentes, nous pouvons proposer la formule de calcul de la distance entre deux modèles suivantes :

Formule 4

$$\delta = \sum_{i=0}^{n} \delta_i \ , \ avec \ \delta_i = w_i \times m_i$$

Où :

- δ est la distance globale entre les deux modèles,
- δ_i est la distance en un point donné entre les deux modèles,
- w_i est le coût de l'opération compte tenu du concept auquel appartient l'instance (i.e. le noeud XML) concernée par la différence détectée, avec $w_i \in \mathbb{N}^*$,
- m_i est l'importance donnée à l'instance, avec $0 < \delta_i \leqslant 1$,
- n est le nombre de différences constatées entre les deux modèles.

IV.3 Algorithme de calcul de la distance entre les modèles

En ce qui concerne le fonctionnement de la comparaison des noeuds de deux arbres XML entre eux, nous nous appuyons sur les principes de l'algorithme implémenté dans l'outil XMLUnit. Le modèle de référence est le modèle *attendu*, car il s'agit du modèle de la situation espérée compte tenu des processus collaboratifs exécutés. Le modèle de test

est le modèle *terrain* qui représente la situation réelle que l'on compare à celle espérée (i.e. notre référence).

IV.3.1 Quelques précisions concernant la détection des différences

Compte tenu de la nature de nos modèles (éléments de bas niveau représentant les instances et ne contenant donc pas de texte et dont les attributs sont obligatoires), les types de différences qui peuvent exister entre deux modèles sont :

1. La suppression d'élément de bas niveau, représentant la suppression d'une instance dans le modèle,
2. L'ajout d'élément de bas niveau, signifiant l'ajout d'une instance dans le modèle,
3. La modification de la valeur des attributs d'un élément de bas niveau, représentant la mise à jour d'une instance existante dans le modèle.

Cela signifie que si une différence est relevée entre les modèles *attendu* et *terrain* et qu'elle est de type :

- Suppression : l'élément concerné existe dans le modèle *attendu* mais n'est pas présent dans le modèle *terrain*,
- Ajout : l'élément concerné n'existe pas dans le modèle *attendu* mais est présent dans le modèle *terrain*,
- Modification : la valeur d'un ou plusieurs attributs a été modifiée dans le modèle *terrain* par rapport aux valeurs existant dans le modèle *attendu*,

Cependant, l'algorithme de XMLUnit ne répond pas complètement à nos exigences : il ne prend pas en compte la nature particulière des attributs qui sont typés *xsd :ID*. Un attribut typé *xsd :ID* implique que sa valeur est unique dans le document XML. Ceci sert à créer des attributs permettant d'identifier de manière unique un élément au sein d'un document XML. Dans notre cas, tous les éléments de bas niveau de nos modèles de situation collaborative embarquent un attribut *id* de type *xsd :ID*.

Or, dans le cas où nous avons un élément de bas niveau dans chacun des modèles qui ont le même nom, les mêmes attributs et les mêmes valeurs d'atttribut *à l'exception* de l'attribut *id*, qui prend la valeur *A01* dans le modèle *attendu* et la valeur *B01* dans le modèle *terrain*, XMLUnit va interpréter cette différence comme une modification d'une valeur d'attribut (la valeur de *id* passe de *A01* à *B01*).

Pour correspondre à nos attentes, XMLUnit devrait comprendre qu'il s'agit de la suppression du noeud identifié par *A01* et l'ajout du noeud identifié par *B01*. Pour combler ce manque, nous avons introduit dans notre algorithme de détection (présenté en Section IV.3.3) une analyse de la différence identifiée spécifique au traitement du cas *id*.

Si XMLUnit relève une différence de type *modification* de la valeur de l'attribut *id* entre deux éléments, cela signifie que les éléments possèdent le même nom. Donc une différence de valeur d'attribut *id* implique que les deux nœuds concernés par cette différence ont un *id* unique à chacun. Cela entraîne la requalification de la différence

de type *modification* en deux différences *ajout* et *suppression*— de nœuds : l'un des deux nœuds n'existe que dans le modèle *terrain*, l'autre n'est présent que dans le modèle *attendu*.

Il faut donc introduire cette notion d'identifiant unique dans l'algorithme de détection des divergences et mettre en place une analyse du contenu du résultat de la détection de différences effectuée par XMLUnit afin de traduire la nature particulière de l'attribut *id*.

Comme nous avons pu le voir dans la Section IV.1.2, XMLUnit renvoie un objet *DetailedDiff* qui est une liste contenant l'ensemble des différences (objets *Difference*) trouvées entre le document de référence et le document de test. Pour chaque objet *Difference*, il est possible d'accéder au détail des éléments suivants :

- L'identifiant du type de différence (exemple : présence d'un noeud enfant, existence d'un attribut supplémentaire, valeur de l'élément, etc.),
- Sa description,
- Sa nature (récupérable ou non),
- Le détail du noeud concerné par la différence du côté du document de référence (via un objet *NodeDetail*[5]),
- Le détail du noeud concerné par la différence du côté du document de test (via un objet *NodeDetail*).

Nous allons donc utiliser ces propriétés pour déterminer la nature de la différence détectée, à savoir l'ajout, la suppression ou la mise à jour (modification) d'un noeud, tout en vérifiant la valeur de l'attribut *id*. La stratégie adoptée est la suivante : si XMLUnit détecte deux nœuds différents et que la différence porte sur la valeur de l'attribut *id*, alors nous en concluons qu'iil s'agit non pas d'une mise à jour[6] mais de l'ajout d'un nœud ans un des deux modèles et la suppression d'un nœd dans l'autre modèle. Ces mécanismes seront détaillés dans l'algorithme de recherche des différences.

IV.3.2 L'objet DetailledDifference

Afin de calculer la divergence et de procéder à son analyse (pour émettre des recommandations d'adaptation), il nous faut avoir un accès direct aux informations caractérisant chaque différence détectée entre les deux modèles. Pour cela, nous allons introduire un nouveau type d'objet qui permet, *pour chaque différence détectée*, de conserver :

- L'ensemble des informations concernant l'objet *Difference* (résultat de la détection réalisée par le moteur de XMLUnit),
- L'expression XPath permettant de localiser l'élément concerné par la différence dans le fichier du modèle de la situation,
- Le type de l'opération (insertion, suppression, modification),
- Le concept concerné (risque, conséquence, facteur de gravité, etc.),

5. A travers un *NodeDetail*, on peut récupérer le noeud, la valeur du noeud ainsi que l'expression XPath permettant d'atteindre le noeud dans le document XML d'origine.

6. En vertu de l'unicité de l'identifiant et de son appartenance définitive à l'élément du modèle sur lequel il porte.

- Le coût,
- Et l'importance.

Nous avons créé une classe *DetailledDifference*[7] qui possède les propriétés suivantes :

- *difference*, de type Difference (type défini par XMLUnit),
- *xpath*, de type chaîne de caractères,
- *operation*, de type chaîne de caractères,
- *concept*, de type chaîne de caractères,
- *cost*, de type réel,
- *importance*, de type réel.

Pour chaque différence trouvée par le moteur XMLUnit et filtrée par notre algorithme (pour la prise en compte de l'attribut *id* des éléments contenus dans les modèles), une instance de *DetailledDifference* est créée et peuplée à l'aide des informations remontées par *DetailedDiff* (pour les propriétés *difference*, *xpath*) et par notre algorithme (pour les propriétés *operation*, *concept*, *cost* et *importance*).

IV.3.3 L'algorithme principal de XMLCompare

Dans cette section, nous présentons l'algorithme de détection de divergence et de calcul de la distance dont la structure est basée sur les éléments présentés dans les sections précédentes : reconnaissance et interprétation de l'attribut *id*, prise en compte du coût de l'opération et de l'importance du concept compte tenu du domaine métier et de la situation collaborative considérés.

$Identifier_Differences(detDiff, couts, importances)$

Cet algorithme permet d'analyser chacune des différences trouvées par le moteur DiffEngine de XMLUnit, de les analyser

Paramètres : detDiff : DetailedDiff, tableau résultat de la comparaison par XMLUnit
couts : entier, tableau des coûts
importances : entier, tableau des importances

Variables : differences : Difference, le tableau contenant les différences de detDiff
difference : Difference, la différence analysée
describedDifference : DetailledDifference, différence enrichie
differencesEnrichies : DetailledDifference, le tableau des différences enrichies

Début
differences ← getDifferences(detDiff)
Pour i **variant de** 1 **à** getTaille(differences) **Faire**
difference ← detDiff[i]

...à suivre...

7. *DetailledDifference* et non *DetailedDifference* afin d'éviter la confusion avec *DetailedDifference* qui existe déjà dans XMLUnit. La classe développée jouant le rôle d'une surcouche à XMLUnit, nous avons nommé notre classe en essayant de se rapprocher de l'existant en la matière.

...suite...

Cas de l'ajout d'un noeud dans le document test

Si (difference.type=="Ajout noeud" && difference.nodeT != null && difference.nodeA == null) **Alors**

AddDetailledDifference(differencesEnrichies, difference, "ADDED", nodeT)

– ∴ – *Cas de la suppression d'un noeud dans le document test*

Si (difference.type=="Ajout noeud" && difference.nodeT == null && difference.nodeA != null) **Alors**

AddDetailledDifference(differencesEnrichies, difference, "DELETED", nodeA)

– ∴ – *Cas de l'ajout d'un noeud et de la suppression d'un noeud*

Si (difference.type=="valeur attribut" && difference.nodeT.endsWith("id") && difference.nodeA.endsWith("id")) **Alors**

AddDetailledDifference(differencesEnrichies, difference, "ADDED", nodeT)
AddDetailledDifference(differencesEnrichies, difference, "DELETED", nodeA)

– ∴ – *Cas de la mise à jour d'un noeud*

Si (notAnalysed(difference)) **Alors**

differencesEnrichies ← **Action** : Ajouter(DetailledDifference(difference, getXPath(nodeT), "UPDATED", difference.nodeT.name, getCout(nodeT,cout),1))

Sinon

Action : "Exception"

Fin Si

Fin Pour describedDifferences ← **Action** : getImportances(differencesEnrichies, importances)

RetourdifferencesEnrichies

Fin

Dans cet algorithme, nous allons comparer les types de différence trouvées par XMLUnit (suppression d'un nœud, modification d'un attribut, etc.) avec les constantes décrites dans la classe *DifferenceConstants* de XMLUnit. Nous tenons à rappeler que l'objet Difference (résultat de la détection d'une différence par XMLUnit) embarque non seulement le type de différence, mais aussi le nœud provenant du document Test (nodeT) et le nœud provenant du document Control (node1) qui ont été comparés.

Cas de l'ajout d'un nœud dans le document test Ce test est le premier que nous effectuons dans l'algorithme. Si les trois conditions suivantes sont remplies, alors on entre dans le cas d'un ajout de nœud dans le document Test (i.e. le modèle *terrain*) :

- Le type de différence relevé par XMLUnit doit être de type *DifferenceCons-*

tants.CHILD_NODE_NOT_FOUND_ID. Cette constante signifie que la présence d'un nœud enfant a été détectée,
- Le nœud provenant du modèle *terrain* **n'est pas** nul,
- Le nœud provenant du modèle *attendu* **est** nul.

Si les conditions sont remplies, on appelle la fonction *AddDetailledDifference()*.

Cas de la suppression d'un nœud dans le document Test Ce test est le deuxième effectué dans l'algorithme. Si les trois conditions suivantes sont remplies, alors on entre dans le cas de la suppression d'un nœud dans le document Test (i.e. le modèle *terrain*) :
- Le type de différence relevé par XMLUnit doit être de type *DifferenceConstants.CHILD_NODE_NOT_FOUND_ID*. Cette constante signifie que la présence d'un nœud enfant a été détectée,
- Le nœud provenant du modèle *terrain* **est** nul,
- Le nœud provenant du modèle *attendu* **n'est pas** nul.

Si les conditions sont remplies, on appelle la fonction *AddDetailledDifference()*.

Cas de l'ajout d'un nœud et de la suppression d'un nœud dans le document Test Ce troisième test réalisé par l'algorithme relève les différences portant sur la valeur de l'attribut **id** entre un nœud du document Test (modèle *terrain*) et un nœud du document Control (modèle *attendu*). En effet, l'attribut **id** étant l'attribut permettant d'assurer l'unicité des éléments au sein du modèle, si le même élément existe côté modèle *attendu* et côté modèle *terrain*, aucune différence sur la valeur de l'attribut id ne devrait être relevée. Nous avons paramétré XMLUnit de façon à ce que le critère de test entre un nœud (Test) et un nœud (Control) soit l'attribut nommé **id** si les nœuds ont le même nom d'élément et des attributs similaires (même nom mais pas forcément la même valeur). Dans ce cas, XMLUnit relève une différence de type *modification* sur la valeur de l'attribut **id** (qui est prioritaire sur les autres attributs portés par les nœuds). Ceci est normal car XLMUnit n'interprète pas l'attribut **id** comme étant un identifiant unique. Nous avons donc décrit la logique d'interprétation d'un tel cas : en lieu et place d'une seule différence de type *modification*, nous avons deux différences portant sur l'*ajout* d'un nœud et la *suppression* d'un nœud dans le modèle *terrain*. Et de la même façon que pour les deux cas précédents, les objets *DetailledDifference* correspondants sont créés et ajoutés au tableau *differencesEnrichies*.

Cas de la modification de la valeur (i) d'un élément ou (ii) d'un attribut qui n'est pas *id* Cet avant-dernier cas permet de tester les nœuds présentant une valeur de l'élément ou d'un attribut (autre que **id**) différente entre les deux documents Test et Control (la différence apparaît comme étant de type *DifferenceConstants.NUMBER_OF_CHILD_NODES*. Avant toute chose, nous allons vérifier que cette différence n'agit pas en doublon avec une différence déjà traitée. Cela peut arriver si un même élément possède plusieurs attributs dont toutes les valeurs —sauf celle de l'id— sont différentes d'un document l'autre. Si le nœud n'a pas déjà été traité, alors on créé

un objet *DetailledDifference* qui prendra *UPDATED* pour valeur d'*opération*. Les valeurs des autres propriétés de l'objet sont alimentées par nodeT, comme dans le cas d'un ajout de nœud au document Test. Enfin l'objet *DetailledDifference* est ajouts au tableau *differencesEnrichies*.

Cas de l'exception Enfin, si la différence détectée par XMLUnit porte sur un nœud qui n'est ni un élément, ni un attribut, ni du texte, une exception est levée. Compte tenu de la structure de nos fichiers de modèle de situation, ce cas ne devrait jamais se produire si le Schema XML du fichier du modèle a été respecté.

Retour du résultat Enfin, les valeurs des importances pour chaque *DetailledDifference* du tableau *differencesEnrichies* sont récupérées avant de retourner le tableau *differencesEnrichies*.

Nous allons maintenant détailler l'algorithme de la fonction *AddDetailledDifference* qui est appelée à plusieurs occasions dans l'algorithme de la fonction *Identifier_Differences* que nous venons d'étudier.

AddDetailledDifference(*differencesEnrichies*, *difference*, *operation*, *nodeT*)

Cet algorithme permet d'analyser un noeud avant de l'ajouter dans la table des différences enrichies

Paramètres : detDiff : DetailedDiff, tableau résultat de la comparaison par XMLUnit
couts : entier, tableau des coûts
importances : entier, tableau des importances

Variables : differences : Difference, le tableau contenant les différences de detDiff
difference : Difference, la différence analysée
describedDifference : DetailledDifference, différence enrichie
differencesEnrichies : DetailledDifference, le tableau des différences enrichies

```
Début
    Si (difference.nodeT.longueur == 0) Alors
        describedDifference ← Action : Ajouter(DetailledDifference(difference, getX-
        Path(nodeT), operation, difference.nodeT.name, getCout(nodeT,cout),1))

    Sinon
        Pour j variant de 1 à getTaille(nodeT) Faire
            describedDifference ← Action : Ajouter(DetailledDifference(difference,
            getXPath(nodeT[j]),operation, difference.nodeT[j].name, get-
            Cout(nodeT[j],cout),1))

        Fin Pour
    Fin Si
Fin
```

L'algorithme *AddDetailledDifference*, appelé par *Identifier_Differences* effectue en premier lieu une vérification pour étudier la nature du nœud transmis en paramètre :

s'agit-il d'un simple nœud ? Ou bien d'un sous-arbre ? En effet, cette distinction est primordiale, car nous allons créer un objet *DetailledDifference* pour chaque nœud ajouté. Dans le cas du nœud simple, un seul objet *DetailledDifference* sera créé et stocké dans le tableau *differencesEnrichies*. Dans le cas du sous-arbre, il y aura autant d'objets *DetailledDifference* créés et ajoutés à *differencesEnrichies* que de nœuds enfants.

Dans tous les cas, l'objet *DetailledDifference* recevra les valeurs suivantes dans le cas de l'ajout d'un nœud dans le modèle *terrain* (la propriété concernée est indiquée entre parenthèses [8]) :

- L'objet Difference provenant du résultat de XMLUnit et qui est à l'origine sa création (difference),
- Le chemin XPath du nœud provenant du modèle *terrain* (xpath),
- Le type d'opération *ADDED* pour signifier l'ajout du nœud (opération),
- Le nom du nœud provenant du modèle *terrain* (concept),
- Le coût retrouvé via le tableau des coûts en fonction de l'opération et du concept (coût),
- L'importance est pour l'instant valuée à 1 (importance).

S'il s'agit de la suppression d'un nœud dans le modèle *terrain*, l'objet *DetailledDifference* recevra les valeurs suivantes (la propriété concernée est indiquée entre parenthèses :

- L'objet Difference provenant du résultat de XMLUnit et qui est à l'origine sa création (difference),
- Le chemin XPath du nœud provenant du modèle *attendu* (xpath),
- Le type d'opération *DELETED* pour signifier la suppression du nœud (opération),
- Le nom du nœud provenant du modèle *attendu* (concept),
- Le coût retrouvé via le tableau des coûts en fonction de l'opération et du concept (coût),
- L'importance est pour l'instant valuée à 1 (importance).

L'algorithme *Calculer_Delta(D)* permet de calculer la divergence globale, en appliquant la Formule 4.

Calculer_Delta(D)

Cet algorithme permet de calculer la distance entre les deux modèles une fois les différences collectées, analysées et enrichies

Paramètres : D : DetailledDifference, le tableau contenant les différences enrichies

Variables : delta : réel, contient le résultat du calcul de la distance
tailleD : entier, la taille du tableau D
i : entier, compteur

Début
tailleD ← Taille(D) delta ← 0 **Pour** i **variant de** 1 **à** tailleD **Faire**
delta ← delta + D[i].cost*D[i].importance
Fin Pour
Retour delta
Fin

8. La structure de l'objet *DetailledDifference* est décrite en Section IV.3.2.

Suite à l'exécution de l'algorithme *Identifier_Differences*, un tableau d'objets *DetailledDifference* a été obtenu. Pour chaque objet *DetailledDifference*, les valeurs de ses propriétés *coût* et *importance* sont récupérées afin de calculer la valeur de δ_i (soit le produit du *coût* et de l'*importance*). Les résultats de ces produits sont cumulés, permettant d'obtenir la valeur de δ, une fois que tous les objets *DetailledDifference* ont été analysés.

IV.3.4 L'implémentation de l'algorithme

L'algorithme présenté dans la section précédente a été implémenté à l'aide du langage Java. Il a permis la création de l'outil XMLCompare, qui réalise la comparaison des deux modèles *attendu* et *terrain*.

IV.4 Conclusion

Nous pouvons conclure cette section en rappelant que la détection de la distance entre les deux modèles *attendu* et *terrain* est réalisée en trois temps : le premier temps concerne la mise à jours desdits modèles (voir Chapitre III). La seconde étape (étudiée dans ce chapitre) s'intéresse à la mesure de la distance entre les modèles *attendu* et *terrain*.

Pour cela, la détection se concentre sur la comparaison de chacun des noeuds des fichiers XML des modèles entre eux. Le calcul de la distance globale se fait en étudiant chaque distance δ_i relevée entre les deux modèles. Pour chaque différence, le coût w_i de l'opération associée (insertion, suppression ou modification) et l'importance m_i de l'instance concernée par la différence permet de calculer la distance entre les deux modèles pour cette différence donnée. La somme de ces distances permet d'obtenir une distance globale entre les deux modèles.

Nous proposons la formule suivante pour caractériser la distance globale δ entre les deux modèles représentant la situation collaborative du point de vue de la surveillance de l'exécution des processus (modèle *attendu*) et du point de vue des acteurs et systèmes présents sur le théâtre de la collaboration (modèle *terrain*).

$$\delta = \sum_{i=0}^{n} \delta_i \text{ , avec } \delta_i = w_i \times m_i$$

Le chapitre suivant vise à détailler l'analyse le contenu de cette distance globale afin d'en extraire la connaissance nécessaire pour proposer une solution d'adaptation des processus collaboratifs en cours d'exécution.

Chapitre V

Recommandations d'adaptation : Adaptation

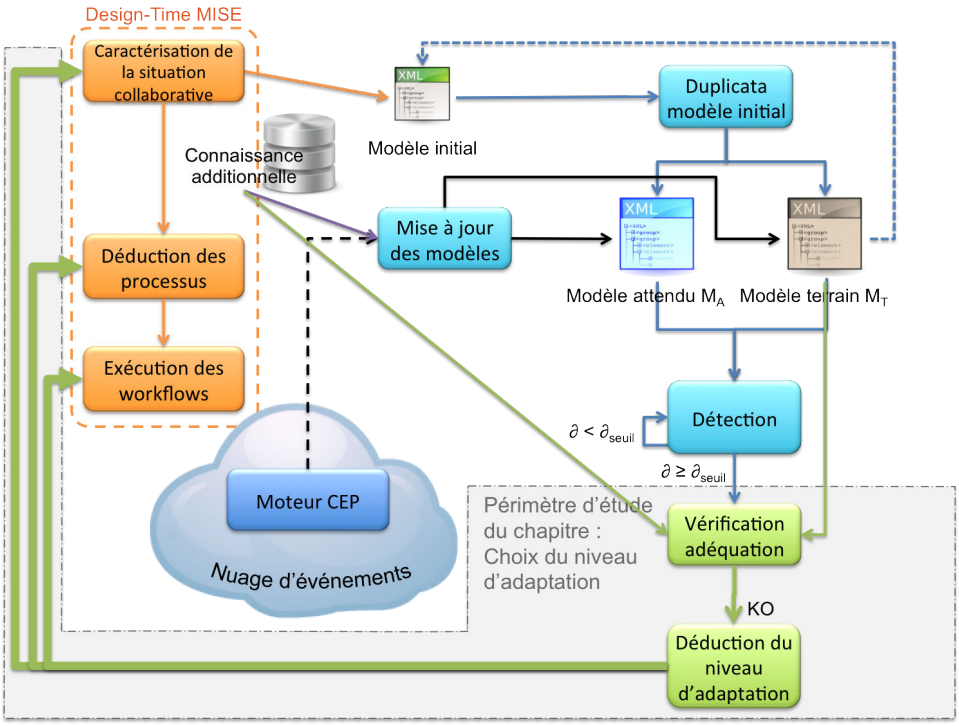


FIGURE V.1 – Architecture simplifiée du Service d'Agilité et périmètre de l'étude dans le Chapitre V

Suite au déclenchement de la détection et au calcul d'une divergence δ supérieure au seuil défini par les partenaires de la collaboration, le Service d'Agilité propose des recommandations d'adaptation (Figure V.1). Dans ce chapitre, nous allons détailler les mécanismes permettant de définir ces recommandations. Cette étape d'adaptation se décompose en deux parties : tout d'abord une vérification de l'adéquation des processus

collaboratifs avec la situation courante —malgré la détection d'un écart— sera réalisée via une analyse rapide du modèle terrain. Puis, le cas échéant, la déduction et la définition du mécanisme d'adaptation des processus collaboratifs le plus adapté à la situation seront effectuées via une analyse du δ (dont le calcul a été présenté dans le chapitre précédent).

La vérification de l'adéquation des processus avec la situation courante est réalisée en recherchant l'existence d'un service (i) déjà appelé dans les workflows courants et (ii) couvrant le nouveau problème à résoudre. Pour cela, nous nous aidons du modèle *terrain* ainsi que de la connaissance apportée par la définitions des liens entre les services et les problèmes (risques, conséquences, ...) et les conditions d'antériorité des services les uns par rapport aux autres. Dans un second temps, nous détaillerons l'analyse du δ afin d'en extraire les caractéristiques qui vont déterminer le mécanisme d'adaptation des processus collaboratifs adapté à la divergence détectée (en accord avec les trois niveaux qui composent la chaîne de déduction des processus collaboratifs tels que définis dans MISE —voir Chapitre I.5.2).

V.1 Contexte et limites

Avant de définir plus en avant les types d'adaptation possibles pour les processus collaboratifs (à la lumière des différences détectées entre les modèles terrain et attendu), ainsi que les mécanismes de recommandation d'adaptation, nous allons aborder les contraintes et pré-requis de l'algorithme dédié à la recommandation d'adaptation détaillé dans les sections suivantes.

V.1.1 Dépendance par rapport à la situation collaborative

L'algorithme de la recommandation du choix d'adaptation est dépendant du domaine considéré. En effet, le conseil en matière d'adaptation est dépendant du « métier » dans lequel s'inscrit la collaboration, c'est-à-dire des concepts et des relations entre les concepts mis en jeu. Cela se traduit par une forte dépendance au méta-modèle de la situation collaborative considérée : partenariat industriel, gestion de crise, gestion de crise nucléaire, gestion de crise routière, etc.

Cette dépendance se traduit par des jeux de conditions différentes pour choisir l'amplitude de l'adaptation des processus collaboratifs (totale, partielle ou réexécution). Dans le cas de ces travaux de thèse, le mécanisme de choix du niveau d'adaptation est décrit pour le domaine de la gestion de crise civile (englobant le cas particulier de la crise nucléaire).

V.1.2 Pertinence des processus en cours d'exécution

Avant d'appeler les mécanismes de choix d'adaptation, il est nécessaire de vérifier s'il existe éventuellement une adéquation entre les processus en cours d'exécution et la situation collaborative telle que définie à l'instant t. En effet, le calcul du δ a permis de

mettre en lumière l'existence d'une divergence entre les modèles *terrain* et *attendu* et d'en définir l'amplitude. Mais dans certains cas de divergence, il est possible d'obtenir ce qu'on pourrait qualifier de *divergences neutres*.

Typiquement, dans le cas d'un risque, d'une conséquence, d'un facteur de complexité ou de gravité dans le modèle *terrain*, il est possible qu'un service des processus en cours d'exécution puisse le prendre en charge sans qu'il soit nécessaire de modifier les processus collaboratifs. Cette capacité d'un service à prendre en charge un problème (risque, conséquence, etc.) est définie par les liens sémantiques tels que détaillés dans les travaux de Wenxin Mu [Mu, 2012] et Nicolas Boissel-Dallier [Boissel-Dallier, 2012] (voir Section V.2). En combinant ces données avec celles indiquant les liens établis entre les problèmes et les solutions (services), il est possible de déterminer si une solution actuellement mise en place au sein des processus pourrait répondre au nouveau problème, tant en terme de nature (identique à, similaire à, etc.) que d'ordre d'exécution (existence de contraintes d'antériorité pour l'exécution de la solution).

Si tel est le cas pour l'ensemble des δ_i du δ calculé, alors la divergence détectée est de type neutre. La conduite d'une adaptation n'est en soi pas nécessaire.

Il s'agit donc d'évaluer la pertinence des processus existants en mettant en regard les nouveaux besoins (risques, conséquences....) et les services prévus (i.e. les processus combinés à l'ontologie qui fait le lien entre l'activité et le problème qu'il peut traiter).

V.1.3 Sources d'adaptation

Si l'adaptation des processus collaboratifs se révèle nécessaire après exécution des mécanismes de détection, il faut alors déterminer le type d'adaptation à mener sur lesdits processus. Le choix de l'adaptation retenue est guidé par la nature de l'évolution de la situation collaborative, telle que nous l'avons présentée dans le Chapitre I d'après [Pingaud, 2009b] :

- L'évolution du contexte,
- L'évolution du réseau,
- L'occurrence du dysfonctionnement.

Dans le Chapitre I, nous avons démontré que les trois niveaux de rebouclage correspondaient parfaitement aux trois niveaux de conception (Design Time) de MISE. L'étude de ces évolutions, mises en regard avec les différents niveaux de conception et de déploiement du SIM, va permettre de conseiller l'utilisateur dans le choix de l'adaptation à mener vers l'une des trois étapes du Design Time (caractérisation de la situation, réconciliation syntaxo-sémantique ou réexécution d'un service). Nous allons donc détailler ces trois étapes du Design Time dans la suite de ce chapitre, afin de mieux comprendre les raisons qui pousseront le Service d'Agilité à proposer tel niveau d'adaptation plutôt qu'un autre.

V.2 Rappels sur la conception du SIM

Dans le chapitre I, nous avons précisé que ces travaux de thèse se positionnaient dans plusieurs projets de recherche et notamment le projet MISE [1] dont le but est de fournir un Système d'Information de Médiation (SIM). Ce projet se base sur de nombreux travaux de thèse qui ont chacune produit des briques constituant l'architecture permettant la conception puis l'exécution du SIM (cf. Figure V.2) suivant trois niveaux qui s'enchaînent permettant respectivement l'obtention du modèle de collaboration, de la dynamique collaborative et du modèle du SIM. Ces trois niveaux sont regroupés sous l'appellation Design Time.

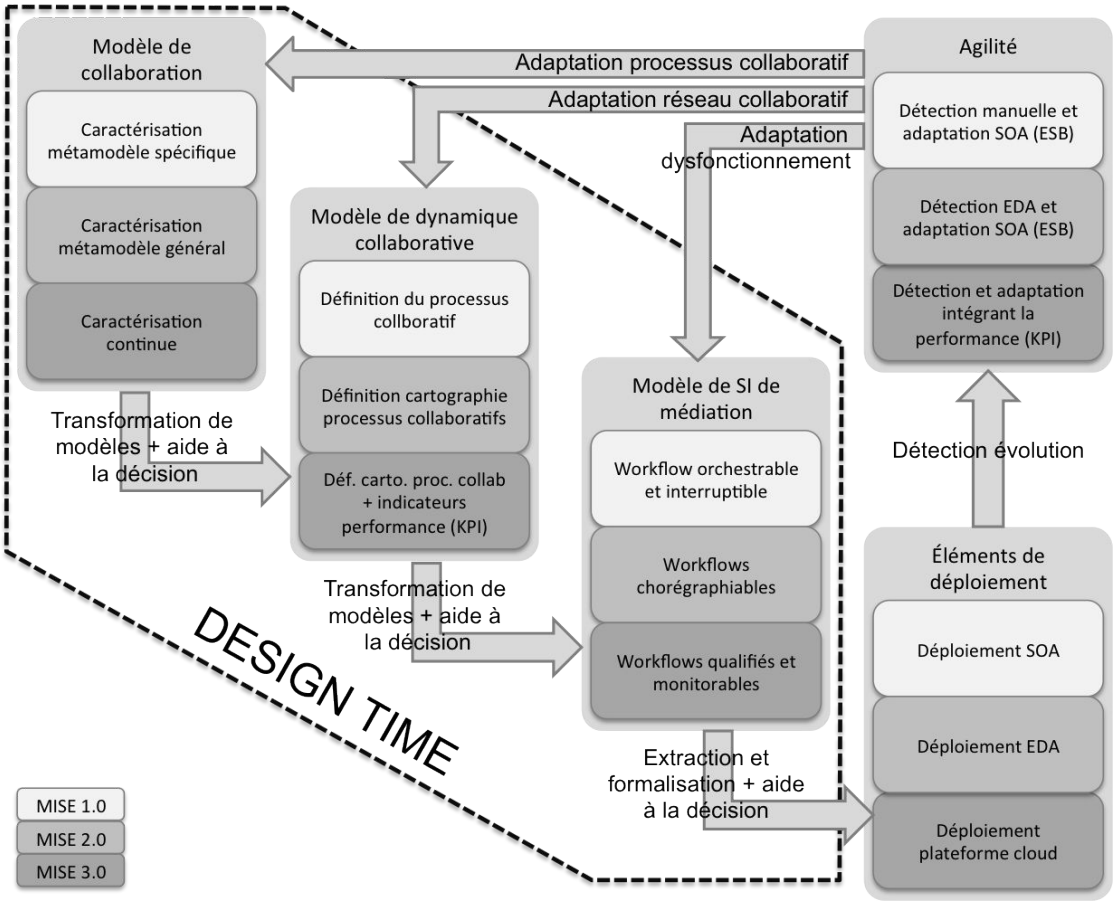


FIGURE V.2 – Les trois niveaux de conception du SIM dans MISE : Design Time

Ces trois étapes de conception font écho aux trois types de sources d'évolution des processus collaboratifs (évolution du contexte, évolution du réseau, occurrence du dysfonctionnement). Il est donc possible de redéfinir tout ou partie des processus collaboratifs afin de les adapter au contexte de la collaboration à l'instant t. Ce choix est légitimé par le fait que la détection du besoin d'adaptation se fait sur la base de l'ana-

1. Dans sa version 2.0 dans le cadre des présents travaux de thèse.

lyse des informations provenant à la fois du terrain de la collaboration, du système de traitement (les partenaires de la collaborations et le SIM) et du suivi de l'exécution des processus collaboratifs et de leurs effets réels. Avant d'étudier les conditions menant à un niveau d'adaptation en particulier, nous allons rappeler les principes et les mécanismes fondamentaux des trois niveaux du Design Time de MISE.

V.2.1 Modèle de collaboration

Dans cette première étape de la conception du SIM, il s'agit de recueillir et exploiter la connaissance relative à la collaboration que l'on souhaite établir. Comme évoqué en Section I.4.2.3 du Chapitre I, un méta-modèle intégrant les principaux concepts d'une situation collaborative au sens général (méta-modèle coeur) ou spécialisée (B2B, gestion de crise, etc.) est utilisé pour recueillir et formaliser cette connaissance. L'instanciation de ces concepts permet de caractériser la situation étudiée [Mu, 2012]. Ces instances servent à peupler l'« ontologie collaborative », dont la structure (concepts et relations entre les concepts) respecte le méta-modèle. Ainsi, il est possible de collecter et analyser les données permettant de décrire le contexte de la collaboration, les partenaires de la collaboration et les objectifs de la collaboration.
Comme présenté plus avant dans ce manuscrit, un éditeur a été créé afin de permettre la caractérisation d'une situation collaborative, suivant les concepts du méta-modèle [Mu, 2012]. L'introduction de nouvelles instances permet d'enrichir l'ontologie collaborative.

V.2.2 Dynamique collaborative

La cartographie de processus collaboratifs a été détaillée par Wenxin Mu dans ses travaux de thèse [Mu, 2012]. Cette cartographie regroupe l'ensemble des processus métiers collaboratifs, permettant d'atteindre les objectifs de la collaboration (définis dans le précédent modèle collaboratif), à l'aide des fonctions métier mises à disposition par chacun des partenaires de la collaboration. L'établissement de la cartographie de processus se déroule en deux phases : (i) la sélection des combinaisons de fonctions métier parmi l'ensemble des fonctions disponibles chez les acteurs de la collaboration, et (ii) l'agencement des fonctions retenues dans le but de créer la dynamique collaborative qui sera par la suite exécutée.

La sélection des services exploite la notion de transitivité apportée par les liens sémantiques entre les instances des concepts dans l'ontologie collaborative. Afin d'éclaircir cette notion de transitivité, nous pouvons l'illustrer par un exemple issu du quotidien : « en train de préparer un repas de famille, M. fait fondre de la graisse de canard dans une casserole, qui prend soudainement feu. M. doit faire face à un *feu de casserole*. A sa disposition, elle a une *planche à découper en bois*. Or, un *feu de casserole* contenant de la graisse présente les mêmes risques qu'un *feu de friteuse*. Le meilleur moyen d'éteindre un *feu de friteuse* est de le priver d'air en couvrant ladite friteuse d'un *couvercle de casserole*... élément que notre cuisinière n'a pas à portée de main ! Elle estime que sa *planche à découper en bois* aura les mêmes propriétés occultantes qu'un couvercle de

casserole. Elle se sert donc de sa *planche à découper en bois* pour éteindre son *feu de casserole.* »

Les relations de proximité ou de similarité entre les instances permettent ainsi de sélectionner un ensemble d'activités (parmi celles proposées par les partenaires) qui peuvent répondre (totalement ou partiellement) aux objectifs de la collaboration.

La définition de l'enchaînement des fonctions métier passe dans un premier temps par l'étape nécessaire de priorisation des objectifs de la collaboration et par conséquent leur priorité de traitement. Dans un second temps, l'étude des entrées et sorties des services métier (définies dans l'ontologie de collaboration) permet d'établir une structuration entre les services grâce aux liens sémantiques qu'ils entretiennent [Mu, 2012] . Par exemple, soient les services B et A respectivement d'entrée E_b et de sortie S_a. Si S_a est sémantiquement proche de E_b, alors A peut être considéré comme un possible prédécesseur de B du point de vue du flux de messages entre les services.

Par ailleurs, il faut noter que la sélection des services métier a été réalisée du fait de leur couverture des objectifs de la collaboration. De fait et grâce aux liaisons *assure* et *contribue à* entre les services métier et les objectifs, il est possible de savoir si les services sélectionnés couvrent tout (*assure*) ou partie (*contribue à*) des objectifs pour lesquels ils ont été retenus. Il est alors possible de déterminer la nature des relations entre les services sélectionnés et la façon dont ils doivent être appelés : concurrents, il faut permettre le choix entre les services ; complémentaires, ils doivent s'exécuter en série ou en parallèle. Dans ce dernier cas, pour choisir entre série et parallèlisation, de nombreux critères sont étudiés comme (liste non exhaustive) la mobilisation des ressources, l'appartenance au partenaire, les effets sur l'environnement collaboratif, etc. Sur la base de ces critères, un ensemble de règles a été établi afin de pouvoir déduire le choix de séquencement entre des activités complémentaires.

L'utilisation conjointe de ces trois principes permet de dégager une structuration des processus en vue de répondre aux objectifs de la collaboration. Il faut souligner quand dans le cadre du projet MISE, les processus collaboratifs sont décrits suivant le formalisme BPMN.

La cartographie s'occupe également de classer les processus suivant les niveaux décisionnel, opérationnel et support [ISO, 2005a] [ISO, 2005b]. Si une fonction principale est une fonction de type décisionnel, alors elle sera placée dans le pool (conteneur au sens BPMN) des processus décisionnels. La démarche reste la même s'il s'agit d'une fonction de type support ou opérationnel. Dans le cas où une fonction principale inclut deux fonctions de type décisionnel, une de type opérationnel ou support, la fonction est alors placée dans le pool général (voir Figure V.3).

L'ensemble des fonctions des différents pools communique à l'aide de messages « objectif », « rapport », « information ».

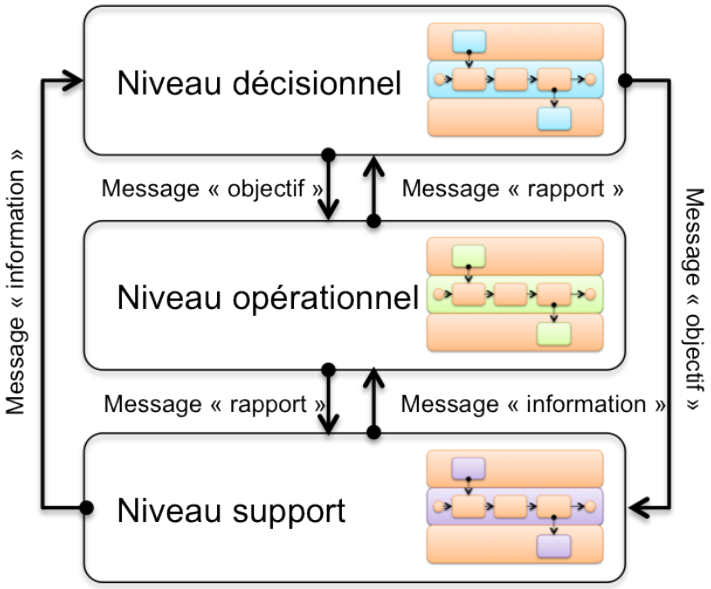


FIGURE V.3 – La cartographie de processus dans MISE 2.0, d'après [Mu, 2012]

Message « objectif » : il contient le résultat d'une décision prise au niveau décisionnel. Ce message peut être envoyé vers les niveaux opérationnels et support, qui doivent atteindre les buts définis dans le message,

Message « rapport » : il transmet le résultat d'une action réalisée au niveau opérationnel. Il sert à envoyer un rapport d'exécution au niveau décisionnel et/ou support : erreur d'exécution, exception, résultat, etc.,

Message « information » : ce type de message peut contenir n'importe quelle sorte d'information (ressource, rapport, erreur, alerte, etc.).

V.2.3 Modèle du SIM

L'objectif de cette étape est l'obtention du SIM exécutable, suivant deux étapes : d'abord la sélection des combinaisons de services techniques répondant à la couverture fonctionnelle des activités métier parmi les services proposés par les acteurs de la collaboration, puis l'agencement des services entre eux en respectant la logique imposée par la cartographie de processus BPMN obtenue précédemment, en vue de déployer des workflows exécutables.

La sélection des services se fait en utilisant les principes de la réconciliation syntaxo-sémantique (au niveau des services mais aussi des messages reçus/émis par ces services), telle qu'expliquée et détaillée dans les travaux de thèse de Nicolas Boissel-Dallier [Boissel-Dallier, 2012]. Chaque partenaire possède une ontologie de domaine qui représente ses activités, produits et données métier. Les ontologies des partenaires sont ensuite rapprochées afin de créer des liens d'équivalence entre les concepts sémantiquement proches. Ce rapprochement permet la création d'une ontologie commune à l'ensemble des partenaires

de la collaboration et servira de base à la réconciliation des services. Une autre ontologie va quant à elle servir à la réconciliation des messages.

Le savoir-faire de chaque acteur est représenté par un ensemble de Web Services. L'utilisation du standard SAWSDL [Kopecký et al., 2007] au niveau des contrats des Web Services permet au partenaire dont il dépend de les annoter sémantiquement. En effet, SAWSDL ajoute des attributs aux différentes balises des fichiers WSDL et XML Schema (XSD) qui sont déjà présentes dans le contrat de service. Les annotations sémantiques sont sélectionnées depuis l'ontologie que nous avons évoquée précédemment. Les processus métier BPMN 2.0 (résultant de la caractérisation de la dynamique collaborative) sont eux aussi annotés sémantiquement grâce aux champs disponibles.

La réconciliation syntaxo-sémantique est alors exécutée (via le moteur de réconciliation) pour proposer, pour chacune des activités des processus, un ou plusieurs Web Services (Figure V.4). Chaque proposition fait l'objet d'une note (ou score) permettant de guider l'utilisateur dans le choix final du ou des Web Services réalisant une activité donnée. Il faut souligner le caractère *n-m* de cette réconciliation sémantique : plusieurs activités peuvent être assurées par plusieurs Web Services.

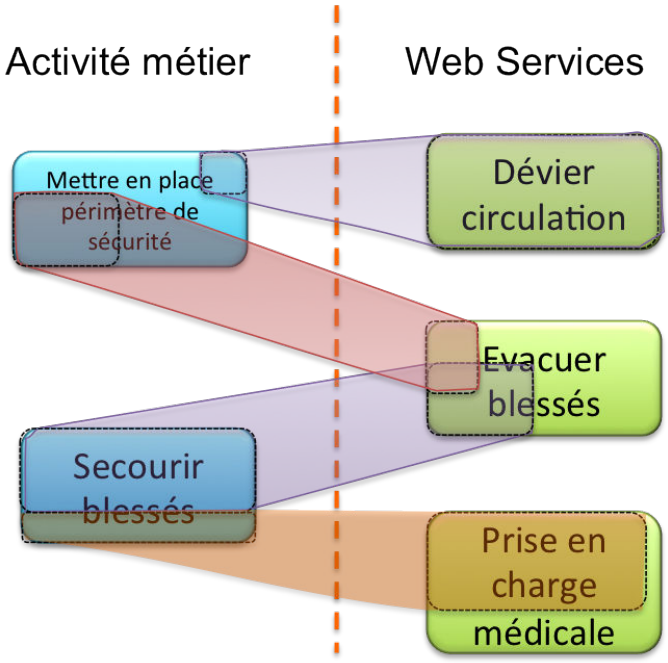


FIGURE V.4 – Principes de la réconciliation n-m des services

Une fois la réconciliation des services effectuée, l'étape de réconciliation des messages est réalisée (Figure V.5). Il s'agit ici de s'assurer de la compatibilité des données entrante et sortantes de chaque service vers les services suivants/précédents (dans la dynamique collaborative). Cette réconciliation va générer tous les fichiers XSLT en charge de réaliser les transformations de données nécessaires pour chaque échange de message entre les

services.

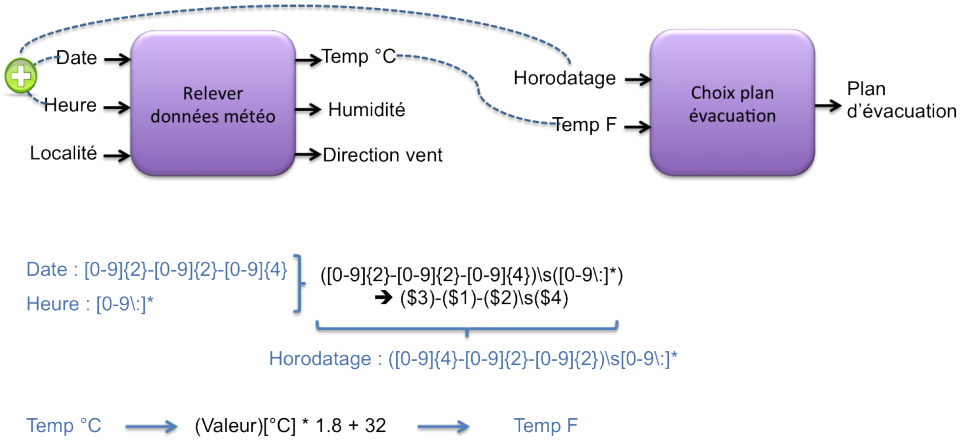


FIGURE V.5 – La réconciliation des formats de données des messages échangés entre les services : données liées et règles de transformation associées, d'après [Boissel-Dallier, 2012].

Cette étape terminée, les workflows exécutables sont générés et déployés sur un ESB embarquant un orchestrateur.

Maintenant que nous avons détaillé le contenu des trois étapes du Design Time de MISE et le principe de fonctionnement des outillages développés par [Mu, 2012] et [Boissel-Dallier, 2012], nous allons pouvoir examiner les trois recommandations d'adaptation qui en découlent.

V.3 Différentes recommandations d'adaptation candidates

Comme nous l'avons souligné dans le chapitre précédent, le résultat δ de la comparaison des modèles *terrain* et *attendu* contient des informations sur les origines de la divergence entre les deux modèles. Il faut donc analyser le contenu du δ pour pouvoir conseiller une adaptation possible des processus collaboratifs, en vue d'atteindre le but de la collaboration.

Dans cette section, nous allons présenter les trois types de recommandations d'adaptation candidates : la caractérisation de la situation collaborative, la redéfinition du réseau collaboratif et la réexécution d'un service. Pour chaque type d'adaptation, nous exposerons les conditions dans lesquelles il est recommandé. Nous détaillerons également les algorithmes d'appel des fonctions associées aux trois types de recommandation.

Ce travail préparatoire nous permettra d'aborder la Section V.4 en ayant les clés de compréhension nécessaires pour définir les mécanismes de choix d'une recommandation d'adaptation compte tenu des évolutions détectées dans les modèles *terrain* et *attendu*.

V.3.1 Caractérisation de la situation collaborative

A ce niveau là de l'adaptation, il convient de caractériser la situation collaborative sur la base des éléments transmis par le terrain afin d'en avoir une photographie réaliste représentant la situation collaborative telle qu'elle est à l'instant t.

Il est tout à fait possible de réutiliser le modèle *terrain* obtenu lors de l'exécution du Service d'Agilité (dans la phase de mise à jour des modèles). En effet, ce modèle est mis à jour à l'aide des événements émis par le terrain de la collaboration et se veut être l'image la plus fidèle possible de la réalité. Ce modèle *terrain* est injecté en entrée du modeleur.

Adaptation_Caracterisation(mTerrain)

Cet algorithme permet de redéfinir le modèle de la collaboration
Paramètres : mTerrain : le modèle terrain de la situation collaborative
Début
    Si (mTerrain <> vide) **Alors**
        Action : TransformerModele(mTerrain)
        Action : OuvrirModeleur(mTerrain)
    Sinon
        Action : Afficher('Le fichier est vide, merci de changer de fichier')
        mTerrain ← **Action** : Action : valeur_saisie()
        Adaptation_Caracterisation(mTerrain)
    Fin Si
Fin

L'algorithme *Adaptation_Reseau* est appelé si la recommandation d'adaptation retenue correspond à la redéfinition totale du réseau collaboratif. Dans ce cas, le Service d'Agilité bascule vers le modeleur de la situation collaborative afin que l'utilisateur puisse éventuellement enrichir ou modifier la caractérisation de la situation. L'utilisateur ne part pas d'un modèle vide (le Service d'Agilité vérifie ce point en amont). Le modeleur est initialisé avec le modèle *terrain* existant à l'instant où la détection d'une divergence a déclenché une adaptation (*OuvrirModeleur(mTerrain)*).

Il est à noter que le modèle *terrain* –qui est un flux XML respectant le formalisme défini dans le Chapitre III– doit être transformé (*TransformerModele(mTerrain)*) de façon à respecter le format d'entrée du modeleur (format JSON), permettant ainsi l'affichage des éléments graphiques symbolisant les données contenues dans le modèle *terrain*.

V.3.2 Redéfinition du réseau collaboratif

Compte tenu de sa structure, la dynamique collaborative peut être affectée par les modifications apportées aux acteurs et/ou à leurs services. Par exemple, la disparition ou l'ajout d'un partenaire de la collaboration a un impact sur la situation collaborative par suppression ou apport de services mis à disposition de la dynamique collaborative.

L'évolution des compétences mises à disposition de la collaboration peut donc se traduire par une remise en cause partielle ou totale des choix effectués quant à la sélection des activités et leur ordre d'exécution. Il faut noter que l'évolution du catalogue des activités disponibles peut avoir des impacts :

- Négatifs : typiquement avec la disparition d'une activité utilisée dans les processus. Il faut donc trouver une activité ou une combinaison d'activités permettant d'atteindre les objectifs (dans le cadre de la gestion de crise, prévenir un risque ou traiter un fait),
- Mélioratifs : la mise à disposition d'une nouvelle activité peut conduire au remplacement d'une activité sélectionnée auparavant par cette nouvelle activité, si celle-ci présente une couverture plus étendue ou plus adaptée quant au problème qu'elle doit traiter.

Adaptation_Reseau($mTerrain, services, ontologie, liensMT$)

Cet algorithme permet de redéfinir la dynamique collaborative

Paramètres : mTerrain : le modèle terrain de la situation collaborative
services : la liste des Web Services à remplacer
ontologie : l'ontologie de collaboration
liensMT : la liste des liens entre les activités métier et les Web Services

Début
  Si (mTerrain <> vide ET ontologie <> vide ET liensMT <> vide ET services <> vide) **Alors**
    Action : OuvrirModeleur(mTerrain, services, ontologie, liensMT)
  Sinon
    Action : Afficher('Les fichiers sont vides, merci de changer de fichiers')
  Fin Si
Fin

L'algorithme *Adaptation_Reseau* est appelé si la recommandation d'adaptation retenue correspond à la redéfinition partielle du réseau collaboratif. Dans ce cas, le Service d'Agilité bascule vers l'outil de réconciliation syntaxo-sémantique (*OuvrirModeleur(mTerrain, services, ontologie, liensMT)*). Cet outil prend en entrée les informations concernant la caractérisation de la collaboration (modèle *terrain*), l'ontologie de collaboration, la liste des Web Services qu'il faut éventuellement remplacer, et enfin la liste des liens entre activités métier et Web Services (issue de la réconciliation syntaxo-sémantique précédente).

V.3.3 Réexécution d'un service

Dans ce type d'adaptation, il ne s'agit pas de modifier la dynamique collaborative établie, mais d'en exécuter à nouveau une certaine partie afin de répondre à un dysfonctionnement dans le déroulement des workflows de réponse à la crise. Le dysfonctionnement recouvre deux notions :

- L'inefficacité d'un service : celui-ci s'est bien déroulé, mais il n'a pas permis

d'obtenir le résultat escompté (cas du largage d'eau censé refroidir les réacteurs de Fukushima Daiichi),

- La panne d'un service : ce dernier n'a pu s'exécuter correctement, pour des raisons diverses.

La différenciation entre la panne et l'inefficacité est possible en regardant l'état du service incriminé. Si le service est à l'état *terminé*, et que l'élément qu'il devait traiter (risque, conséquence, facteur aggravant ou de complexité) est toujours présent dans le modèle *terrain*, alors on peut en déduire qu'il s'agit d'une inefficacité du service. La cause peut en être une description initiale partielle de la situation collaborative ou une évolution majeure de la situation collaborative, conduisant ainsi au choix d'un service inadapté au contexte réel. Si le service est à l'état *failed*, la cause est la panne du service. Cette différenciation est importante pour permettre la capitalisation de connaissances et l'apprentissage sur la réponse à la crise.

Adaptation_Execution(*mTerrain*, *services*)

Cet algorithme permet de redéfinir le modèle de la collaboration

Paramètres : mTerrain : le modèle terrain de la situation collaborative
services : tableau contenant les clés-valeurs (identifiant, état) des services à réexécuter

```
Début
    Si (services <> vide) Alors
        Action : Afficher('Les services suivant sont à l'état de :')
        – ∴ – Tous les services et leurs états respectifs sont listés à l'utilisateur
        Action : Afficher('Pour exécuter à nouveau les services, taper 1, sinon taper
        2') choix ← Action : Action : valeur_saisie()
        Si (choix == 1) Alors
            Action : Administration_Processus(services)
        Sinon
            Action : Adaptation_Reseau(mTerrain)
        Fin Si
    Sinon
        Action : Afficher('Le tableau des Services est vide, adaptation interrompue')
    Fin Si
Fin
```

L'algorithme *Adaptation_Execution* présente le comportement du Service d'Agilité quand la recommandation d'adaptation retenue est de type *réexécution de service*. Après avoir vérifié qu'il existe au moins un service à réexécuter, le Service d'Agilité présente à l'utilisateur la liste des services concernés et lui propose de les exécuter à nouveau. Si l'utilisateur accepte cette solution, le Service d'Agilité va appeler l'outil de suivi et d'administration des workflows [2] . Cela permet à l'utilisateur de demander une seconde

2. Cet outil est proposé par l'orchestrateur fractalisé [Faure et al., 2009] et lui fournit la liste des services concernés à travers la fonction *Administration_Processus(services)*. Il permet non seulement de

exécution des services concernés dans les processus collaboratifs.

Si l'utilisateur refuse cette possibilité (car il y a un nombre trop important de services inefficaces ou en panne, par exemple), le Service d'Agilité va alors proposer de revenir au niveau de la caractérisation de la dynamique collaborative du SIM en appelant la fonction *Adaptation_Reseau(mTerrain)* (dont nous avons détaillé le fonctionnement précédemment). L'utilisateur reprendra la main sur la sélection des services et leur ordre d'exécution, afin de modifier les workflows déployés. Ces workflows exécutables suivent la logique décrite dans les processus BPMN et ne la modifient pas.

V.4 Décision de la recommandation d'adaptation

Le choix de la recommandation d'adaptation amène donc à réinitier la conception et/ou le déploiement du SIM en rappelant l'exécution de l'une des trois étapes du Design Time. Pour choisir l'étape de Design Time la plus pertinente à exécuter, il faut donc étudier les différences relevées entre les modèles *terrain* et *attendu* et choisir les actions correctives à mener.

V.4.1 Facteurs de choix

Il est intéressant de noter que si les différences (seules ou combinées entre elles) peuvent amener à plusieurs solutions d'adaptation, le pire cas (celui qui demande le plus d'effort d'adaptation) sera proposé en priorité, en vertu du principe de précaution. En effet, si l'étude des différences amène à considérer une réexécution d'un service donné et une recaractérisation de la situation collaborative, le Service d'Agilité ne proposera que la recaractérisation de la situation collaborative.

Cette décision peut être discutée, notamment en ce qui concerne les effets des processus collaboratifs sur la situation collaborative. La réexécution du service aurait peut-être permis de supprimer les facteurs ayant conduit à proposer également une recaractérisation de la situation collaborative, rendant cette dernière finalement inutile. Dans ce cas, la réexécution du service avait un coût moindre, en terme de temps et de ressources, que la recaractérisation de la situation collaborative qui entraîne à sa suite les deux autres étapes de conception de MISE (i.e. la redéfinition des processus collaboratifs et la réconciliation syntaxo-sémantique qui permettent d'obtenir les workflows collaboratifs) . Cette facette de l'analyse des facteurs de choix d'adaptation –qu'on pourrait presque qualifier de prédictive– n'est pas abordée dans le cadre de ces travaux mais peut constituer une piste de travail future.

V.4.2 Nuance de l'adaptation

La notion de nuance de l'adaptation est également introduite, afin de traduire deux points de vue dans la décision d'adapter les processus. En fonction de la divergence

visualiser les workflows en cours d'exécution mais aussi d'agir sur les workflows

détectée, l'adaptation peut être menée dans le but de corriger les processus (car un problème ou un objectif n'a pas été atteint) ou bien dans afin d'améliorer le fonctionnement des processus pour éviter le gaspillage de ressources par exemple.

En effet, suivant l'évolution de la collaboration, il est possible de se trouver dans une situation où les processus collaboratifs engagent des moyens désormais disproportionnés ou appellent l'exécution de tâches inutiles : envoyer une équipe secourir des blessés qui n'existent finalement pas, appeler une équipe de pompiers spécialisés dans les incendies nucléaires pour un feu qui est finalement banal, demander un engin de démolition pour abattre un bâtiment qui s'est écroulé entretemps.

La collaboration peut également accueillir de nouveaux partenaires qui proposent des services ayant une meilleure couverture des activités que ceux appelés par les workflows établis. La détection de la divergence (due à cette arrivée de nouveaux partenaires et de leurs services) peut amener à une recommandation d'adaptation dont le but est de sélectionner les meilleurs services.

Nous avons choisir d'introduire la nuance de l'adaptation comme un paramètre de choix qui pourra guider l'utilisateur dans son choix d'adaptation en classant les recommandations par ordre de priorité (Table V.1). Les recommandations à but correctif étant prioritaires sur celles à but mélioratif, leur rang commence par 1. Le classement du niveau d'adaptation reste identique à celui évoqué précédemment : d'abord le retour à la caractérisation (x.1), puis la redéfinition du réseau (x.2) et enfin la réexécution d'un service (x.3). Une recommandation qui possède un rang égal à 1.3 est prioritaire sur une recommandation de rang 2.2.

	Nuance	
Niveau d'adaptation	Correction	Amélioration
Caractérisation situation	1.1	2.1
Définition du réseau	1.2	2.2
Réexécution service	1.3	2.3

TABLE V.1 – Table des rangs de classement des recommandations d'adaptation

Voici un exemple montrant l'intérêt de la nuance. Le Service d'Agilité a détecté deux adaptations possibles : un retour à la caractérisation (pour amélioration de l'utilisation des ressources humaines et matérielles) et une réexécution d'un service (pour atteindre les objectifs). D'après la règle du pire cas possible évoquée précédemment, le Service d'Agilité devrait proposer l'adaptation relevant de la caractérisation de la situation. Or cette adaptation n'a pas d'action directe sur une correction des processus en vue d'atteindre l'objectif collaboratif, contrairement à la réexécution. D'après la Table V.1, la réexécution d'un service à but correctif possède un rang 1.3 qui est supérieur à celui du retour à la caractérisation en vue d'améliorer les workflows 2.1.

V.4.3 Algorithme du choix de l'adaptation

Jusqu'ici, nous avons vu comment *calculer la divergence* δ entre les modèles *terrain* et *attendu* et nous avons étudié *quelles étaient les recommandations d'adaptation* possibles. Il nous faut désormais mettre en correspondance la divergence et les recommandations d'adaptation. Comme le montre la Figure V.6 nous allons donc rechercher des points de liaison entre la divergence et les recommandations d'adaptation.

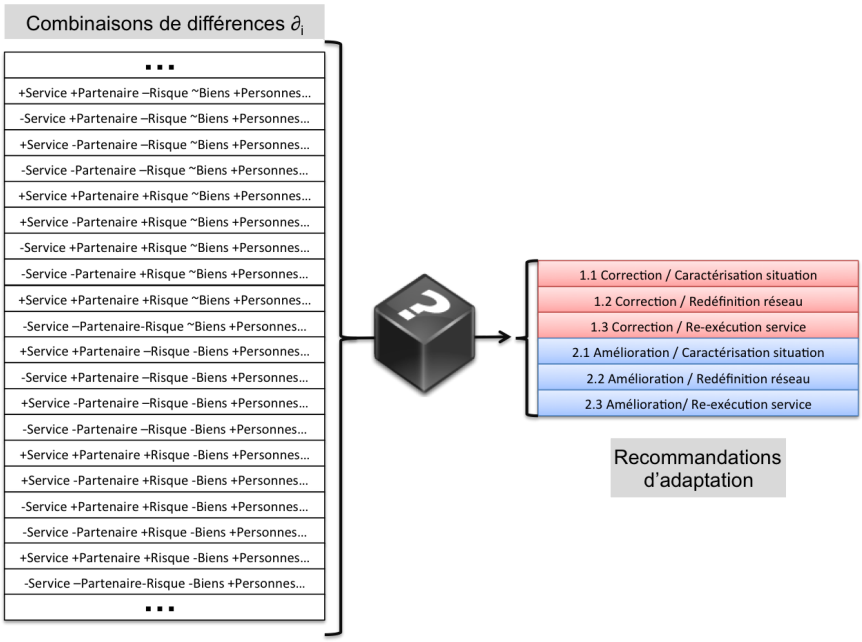


FIGURE V.6 – Problématique actuelle : effectuer la correspondance entre la divergence et les recommandations d'adaptation

Si nous revenons à la définition de la divergence, nous savons qu'elle se compose de la somme de toutes les différences atomiques δ_i détectées entre les deux modèles. Or, une différence atomique δ_i est le produit d'un *coût* et d'une *importance*. Le *coût* correspond au coût engendré par l'opération effectuée sur un élément (ou instance) du modèle *terrain* pour être isomorphe avec le modèle *attendu*.

Comme nous l'avons présenté dans le Chapitre IV, nous avons une matrice {concept ; opération} qui liste les différents coûts existants pour les trois opérations (ajout, suppression et modification) menées sur tous les concepts du méta-modèle utilisé (dans le cas de la couche crise du méta-modèle, on retrouve les concepts de partenaire, service, biens, personnes, risques, etc.).

L'*importance* quant à elle permet de classer les instances de chaque concept par ordre d'importance (un *risque* de *contamination par le virus Ebola* est plus important qu'un *risque* de *contamination par le virus H1N1*). Nous pouvons remarquer que la composante « coût » d'une différence permet de connaître l'origine et l'effet de la différence, autrement dit elle nous renseigne sur le concept de l'instance qui a subit une évolution.

Concernant les recommandations d'adaptation, nous avons pu voir que ces recommandations étaient guidées par la nature de la divergence, à savoir l'évolution du contexte de la collaboration, l'évolution du réseau collaboratif ou encore un dysfonctionnement à l'exécution. On s'intéresse dans ce cas plus particulièrement aux types des éléments concernés (personnes, biens, risque, facteur de gravité, etc.) qu'aux individus en eux-mêmes[3]. Or ces évolutions sont retranscrites par l'ensemble des δ_i détectés, et plus particulièrement par la composante de coût, à travers le tuple {concept, operation}.

Nous pourrions être tentés de décrire tous les tuples {concept, operation} possibles et de lister toutes les combinaisons pouvant se réaliser lors de la détection d'une divergence. Et ensuite, pour chaque combinaison, nous étudierions ses effets sur le contexte de la collaboration, le réseau collaboratif et l'exécution des workflows afin de déterminer la recommandation d'adaptation la plus juste. Cette tâche est cependant difficilement réalisable, en raison de la combinatoire explosive. Par exemple, dans le cadre de la situation de crise, la couche crise du méta-modèle nous propose 10 concepts : *People*, *Natural Site*, *Goods*, *Civilian Society*, *Gravity Factor*, *Complexity Factor*, *Risk*, *Consequence*, *Actor*, *Service*. Ces 10 concepts combinés aux 3 opérations (ajout, suppression, modification) nous amènent au minimum à 3^{10} soit *59049* combinaisons possibles entre les tuples {concept, operation}.

Afin de réduire cette combinatoire, nous avons pensé à créer des *classes d'équivalence de nos tuples* en étudiant les relations entretenues entre les classes du méta-modèle de la crise. Avec, nous réduisons drastiquement le nombre de combinaisons possibles de tuples conduisant aux six recommandations d'adaptation.

Si nous revenons à notre méta-modèle (couche crise), nous constatons qu'il s'agit d'un diagramme de classes UML. Les classes sont reliées entre elles à travers des relations qui possèdent une signification (association, héritage, composition, agrégation, dépendance). Nous avons ainsi une représentation conceptuelle des données que nous pouvons utiliser, suivant le formalisme UML. Si nous éliminons les classes qui ne sont pas représentées dans le modèle de la situation collaborative (qu'elle soit de type terrain ou attendu), nous obtenons les classes et les relations présentées dans la Figure V.7.

Les *Biens*, les *Personnes*, la *Société Civile* et les *Sites naturels* héritent de la classe *Sous Composant* du package *Système étudié*. Il en est de même pour les acteurs, mais ceux-là font partie du package *système de Traitement* : ils ont donc une logique comportementale différente de celles des éléments et individus décrits à travers les classes *Biens*, les *Personnes*, la *Société Civile* et les *Sites naturels*. Un *Sous Composant* possède une *Caractéristique* dont héritent les *Facteur de Complexité* et *Facteur de Gravité*. Un *Risk* compose une *Caractéristique*. Il impacte un *Sous Composant* et est concrétisé par une *Conséquence* qui agit sur le *Sous Composant*. Un *Acteur* fournit des *Service* qui

3. Sauf dans le cas particulier de la réexécution d'un ou plusieurs services : il est nécessaire de connaître exactement leurs identifiants uniques afin de pouvoir rappeler leur exécution.

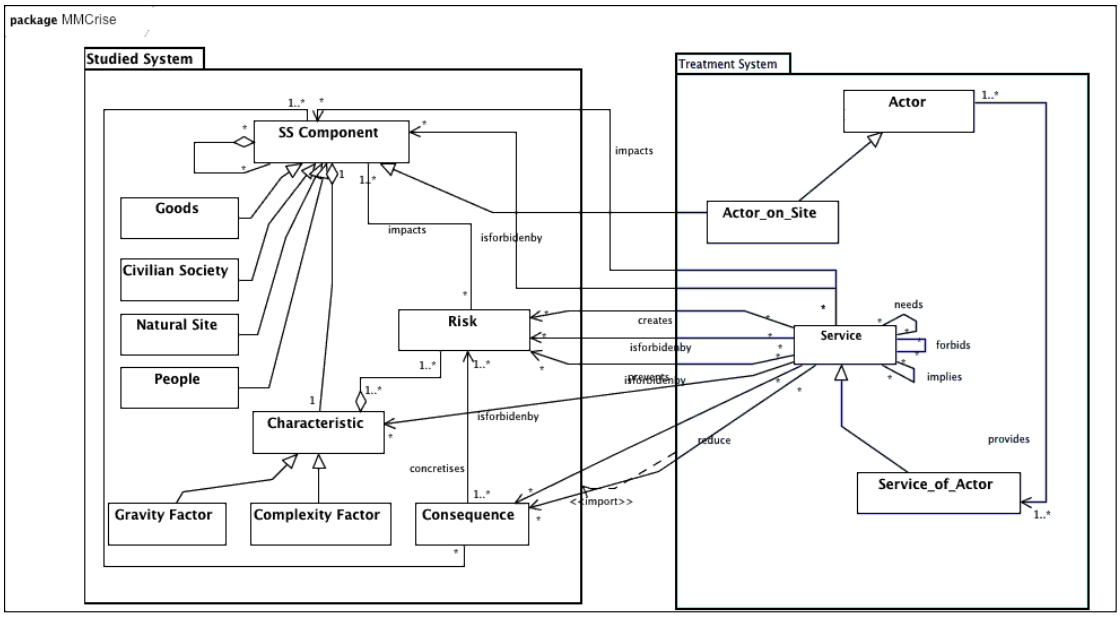


FIGURE V.7 – Le méta-modèle (réduit) de la crise

répondent aux *Caractéristique*, aux *Risque* et aux *Consequence*.

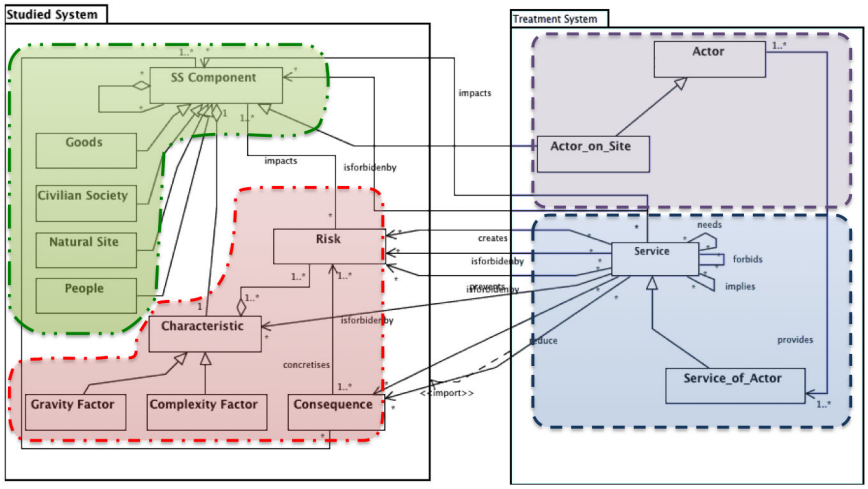


FIGURE V.8 – Le diagramme de classes simplifié du modèle de la situation de crise

D'après les considérations précédentes, nous voyons que des classes d'équivalence se détachent nettement ainsi que le montre la Figure V.8 :

- La classe d'équivalence *Sous Composant* du système étudié, qui contient les *Biens*, les *Personnes*, la *Société Civile* et les *Sites naturels*,
- La classe d'équivalence *Perturbation* qui impacte la classe d'équivalence *Sous Composant*. Elle contient les *Caractéristique*, les *Risque*, les *Consequence*, les *Facteur de Complexité* et les *Facteur de Gravité*,

- La classe d'équivalence *Service* qui contient uniquement la classe *Service*. Ceci est dû caractère particulier des *Service* : ce type d'élément doit pouvoir être exécuté à nouveau. Son comportement dans le système étant différent de celui de l'*Acteur*, ils ne peuvent être dans la même classe d'équivalence même s'ils partagent des liens très forts (exemple : la disparition de l'acteur entraîne la disparition des services qu'il fournit). La classe d'équivalence *Service* répond à la classe *Perturbation*,
- La classe d'équivalence *Partenaire* qui répondent à la classe *Perturbation*.

Considérant ces 4 classes d'équivalence, nous leur avons appliqué les opérations d'ajout, suppression et modification . Puis nous avons noté les impacts sur la situation collaborative en termes d'évolution du contexte de la collaboration, d'évolution du réseau collaboratif ou encore de dysfonctionnement des services. L'étude des liens entre les classes d'équivalence permet de trouver les conditions minimum à remplir pour suggérer telle ou telle recommandation d'adaptation. Nous avons ainsi déterminé plusieurs cas (qui seront repris dans le même ordre dans l'algorithme *Choix_Adaptation*) qui amènent chacun à l'une des 6 recommandations d'adaptation existantes. Dans la mesure du possible, nous avons représenté graphiquement ces cas :

Cas 1

- L'ajout ou la mise à jour d'un élément de la classe d'équivalence *Perturbation* : il s'applique sur un ou plusieurs éléments de la classe d'équivalence *Sous Composant* ou *Service*. Il faut donc prévoir un *Service* pour y faire face. L'adaptation recommandée est de rang **1.1**,
- La mise à jour d'un élément de la classe d'équivalence *Sous Composant* et la suppression a minima des liens qui le lie à un élément de la classe d'équivalence *Perturbation* (si l'élément *Perturbation* est supprimé, le cas s'applique aussi) comme représenté dans la Figure V.9. L'évolution d'un composant de l'environnement de la collaboration impose une recommandation d'adaptation de rang **1.1**.

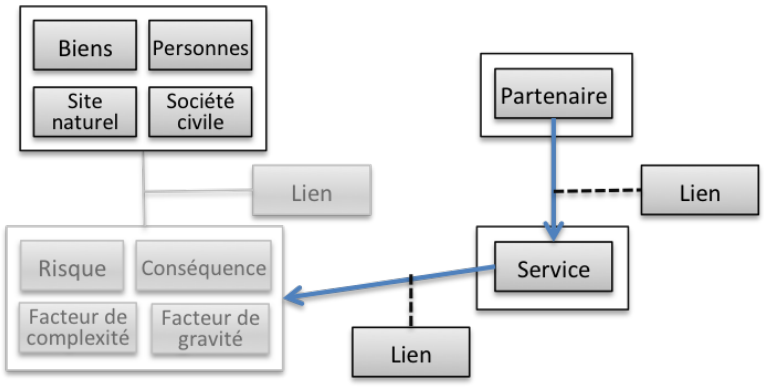


FIGURE V.9 – Le cas de la mise à jour d'un *Sous Composant* et de la suppression son lien avec une *Perturbation* (voire la suppression de la *Perturbation*)

Cas 2

- La suppression d'un élément de la classe d'équivalence *Perturbation* et la suppression d'un élément de la classe d'équivalence *Sous Composant* qui lui est lié entraîne une réduction de l'environnement impacté par la crise (Figure V.10). Les autres éléments *Perturbation* qui étaient liés à cet élément *Sous Composant* n'ont plus d'impact sur ledit élément *Sous Composant*. Les *Services* qui répondaient à l'élément *Perturbation* supprimé n'ont plus lieu de s'exécuter (si la suppression de l'élément *Perturbation* n'est pas dû à une exécution de ces *Services*). Nous sommes dans le cas d'une recaractérisation de la situation collaborative, mais à but mélioratif : les workflows en cours d'exécution répondent toujours aux objectifs fixés. La recommandation d'adaptation est donc de rang **2.1**,

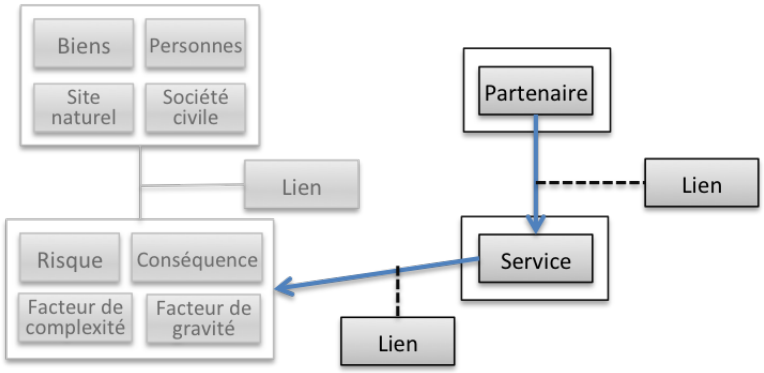


FIGURE V.10 – Le cas de la suppression d'une *Perturbation* et d'un *Sous Composant* qui lui est lié

- L'ajout d'un élément de la classe d'équivalence *Sous Composant* : cela peut éventuellement avoir un impact, mais s'il n'y a pas d'élément de la classe d'équivalence *Perturbation* qui le lie (Figure V.11), l'adaptation proposée sera simplement à but mélioratif. Compte tenu de la nature de l'évolution (évolution du contexte), une caractérisation de la situation sera recommandée. L'adaptation recommandée est de type **2.1**,

Cas 3

- La suppression d'un élément de la classe d'équivalence *Perturbation* entraîne la suppression du lien avec les éléments de la classe d'équivalence *Service* (Figure V.12). Dans l'hypothèse où ce *Service* n'avait pas encore été exécuté pour traiter la *Perturbation*, son appel par le workflow devient peu pertinent. Mais s'il est appelé, cela ne compromet pas la bonne marche des workflows (il s'agit plus ici d'éviter les gaspillages de ressources). La recommandation d'adaptation proposée est donc de rang **2.2**, afin d'optimiser la définition du réseau collaboratif,
- L'ajout d'un élément de la classe d'équivalence *Partenaire* indique une évolution de la cellule de crise, mais avec un impact positif (un nouvel acteur rejoint la collaboration). Cependant, son intégration dans la collaboration n'a de sens que

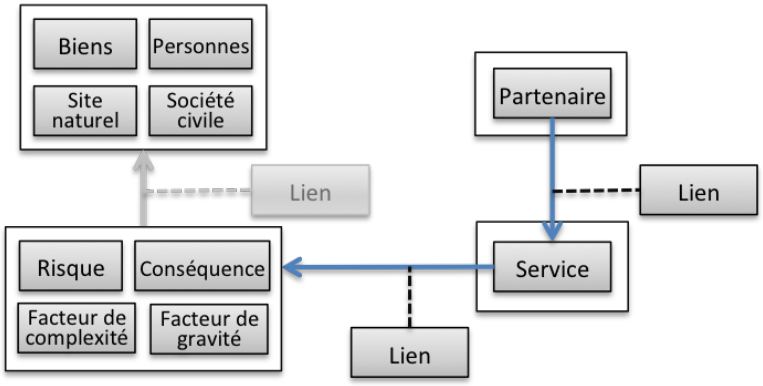


FIGURE V.11 – Le cas de l'ajout d'un *Sous Composant* sans lien avec une *Perturbation*

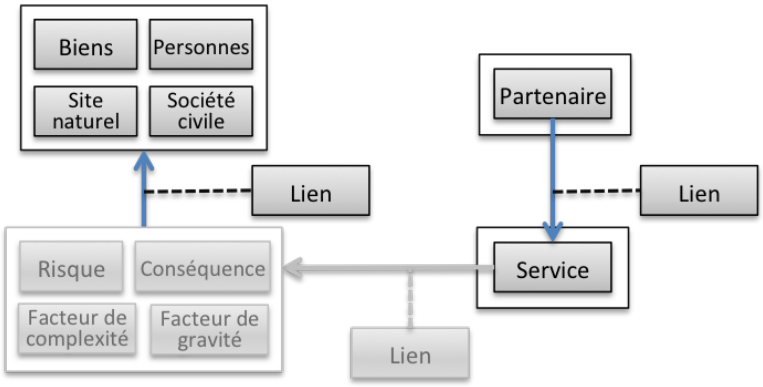


FIGURE V.12 – Le cas de la suppression d'une *Perturbation* et de son lien avec un *Service*

si ses services sont proposés à la collaboration. Pour cela il faut que des éléments de la classe d'équivalence *Service* lui soient liés (et donc qu'un nouvel élément *Service* soit ajouté). Or la mise à disposition de nouveaux services peut conduire à la redéfinition de la logique collaborative afin de vérifier que les nouveaux services techniques disponibles ne couvrent pas mieux une ou des activités métier que ceux intégrés dans les workflows. La recommandation d'adaptation pour ce cas est de rang **2.2**, afin d'optimiser la définition du réseau collaboratif.

- Enfin l'ajout d'un élément de la classe d'équivalence *Service* indique qu'un nouveau service est mis à disposition de la collaboration. Or la mise à disposition d'un nouveau service peut conduire à la redéfinition de la logique collaborative afin de vérifier que ce nouveau service technique disponible ne couvre pas mieux une ou des activités métier que ceux intégrés dans les workflows. La recommandation d'adaptation pour ce cas est de rang **2.2**, afin d'optimiser la définition du réseau collaboratif.

Cas 4

- Si un un élément de la classe d'équivalence *Service* est supprimé et que tous les

éléments *Perturbation* auxquels il était lié sont eux aussi supprimés (par résultat de l'action du service ou par évolution de l'environnement de la crise) (voir Figure V.13), cela signifie que les objectifs poursuivis par le service ont été atteints. Du coup, cette évolution de la collaboration n'a pas d'impact sur la poursuite de l'exécution des workflows. On peut éventuellement vouloir redéfinir la logique collaborative afin de supprimer les appels à ce service. Dans ce cas, la recommandation d'adaptation est de rang **2.2**.

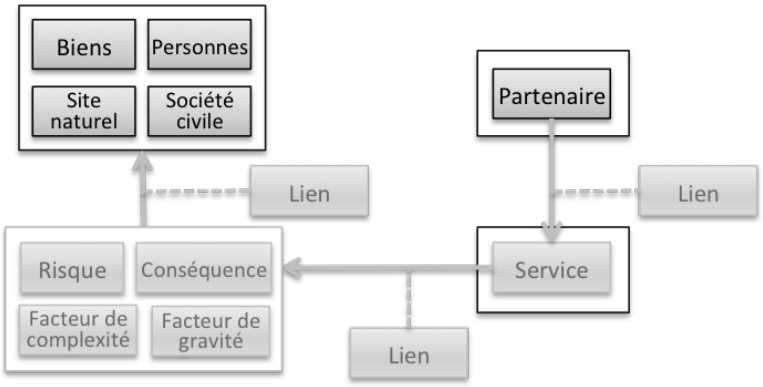


FIGURE V.13 – Le cas de la suppression d'un *Service* et de toutes ses *Perturbations* liées

- Par contre, si un un élément de la classe d'équivalence *Service* est supprimé mais qu'il lui reste au moins un élément *Perturbation* lié (Figure V.14), alors le traitement de cette *Perturbation* est compromis. Il va falloir choisir un nouveau *Service* pour traiter la *Perturbation*. L'adaptation recommandée dans ce cas est de rang **1.1**.

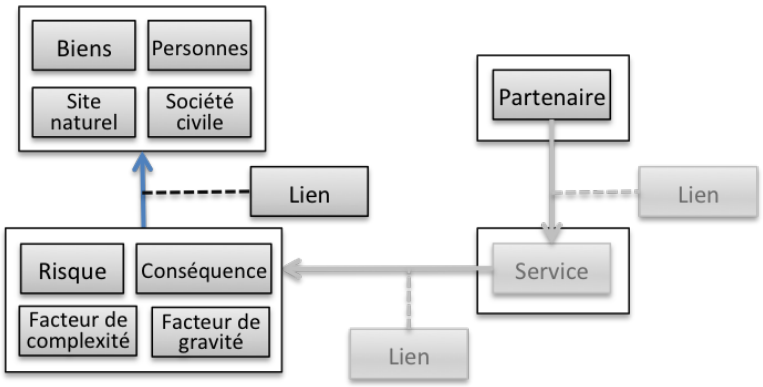


FIGURE V.14 – Le cas de la suppression d'un *Service* mais pas de toutes ses *Perturbations* liées

Cas 5 Ce cas ressemble au Cas 4. En effet, nous allons retrouver les mêmes conditions mais augmentées par la condition de la suppression d'un élément de la classe d'équivalence *Partenaire*.

- Si un élément de la classe d'équivalence *Partenaire* est supprimé, les éléments *Service* qui lui sont liés sont également supprimés (suppression en cascade). Si pour tous ces *Service* supprimés l'ensemble des *Perturbation* qui leur sont liés sont également supprimées (Figure V.15), alors la poursuite de l'exécution des workflows collaboratifs n'est pas compromise. Une redéfinition du réseau collaboratif peut être envisagée, à titre d'amélioration. La recommandation d'adaptation est alors de rang **2.2**,

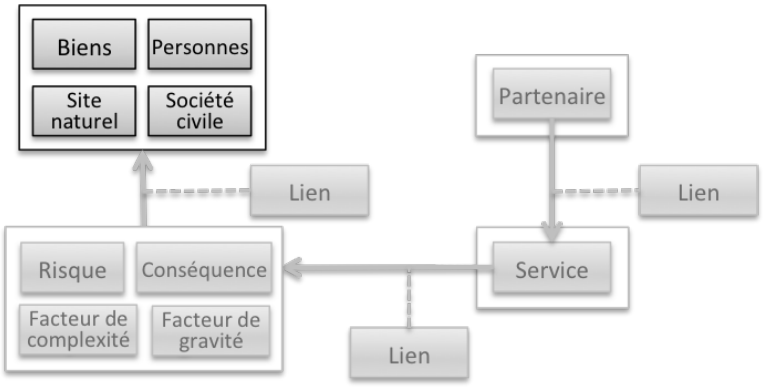


FIGURE V.15 – Le cas de la suppression d'un *Partenaire*, de tous ses *Service* et de toutes les *Perturbations* liées à ses *Service*

- Dans le cas où un élément de la classe d'équivalence *Partenaire* est supprimé, les éléments *Service* qui lui sont liés sont également supprimés (suppression en cascade). Mais s'il existe encore au moins une *Perturbation* qui était liée à un des *Service* supprimés, alors elle n'est plus traitée (Figure V.16). Il faut trouver un autre *Service* qui puisse prendre en charge cette *Perturbation*. Pour cela, l'adaptation recommandée est de rang **1.2**.

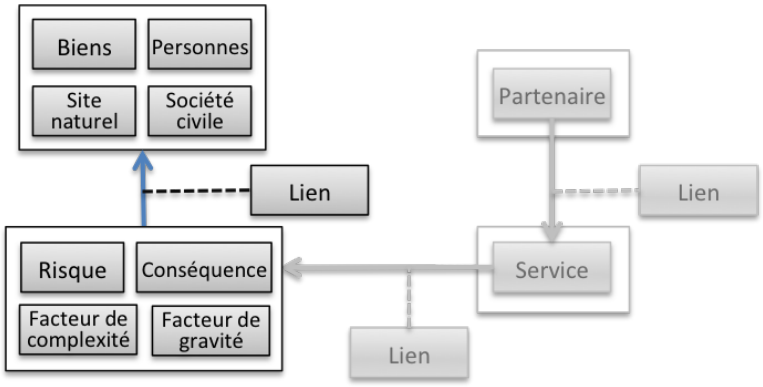


FIGURE V.16 – Le cas de la suppression d'un *Partenaire*, de tous ses *Service* mais pas de toutes les *Perturbations* liées à ses *Service*

Cas 6 La mise à jour d'un élément de la classe d'équivalence *Partenaire* entraîne de fait une évolution de la cellule de crise : il n'est pas possible d'ignorer l'évolution d'un tel élément. Si cette évolution se traduit par une dégradation de sa disponibilité, de ses ressources, etc., cela peut entraîner à terme une indisponibilité totale, rendant inaccessibles les services qu'il propose et qui sont utilisés pour traiter des *Perturbation*. Cela signifie que l'adaptation proposée sera de rang **1.1** afin de revenir à la caractérisation de la collaboration et d'étudier les conséquences de cette évolution.

Cas 7 La mise à jour d'un élément de la classe d'équivalence *Service* et l'ajout (ou la mise à jour) d'un élément de la classe d'équivalence *Perturbation*. L'état du service a été modifié mais les *Perturbation* liées n'ont pas été supprimées (Figure V.17). On peut donc en déduire que le service est soit tombé en panne, soit son résultat n'a pas été celui escompté. D'après l'étude des solutions candidates à l'adaptation en cas de défaillance d'un *Service*, il faut ré-exécuter ledit *Service*. Dans ce cas, une adaptation de rang **1.1** est donc recommandée. De plus, il faut alimenter cette recommandation avec l'identifiant unique du *Service* afin d'indiquer à l'outil de supervision de l'orchestration le *Service* qui doit être exécuté de nouveau.

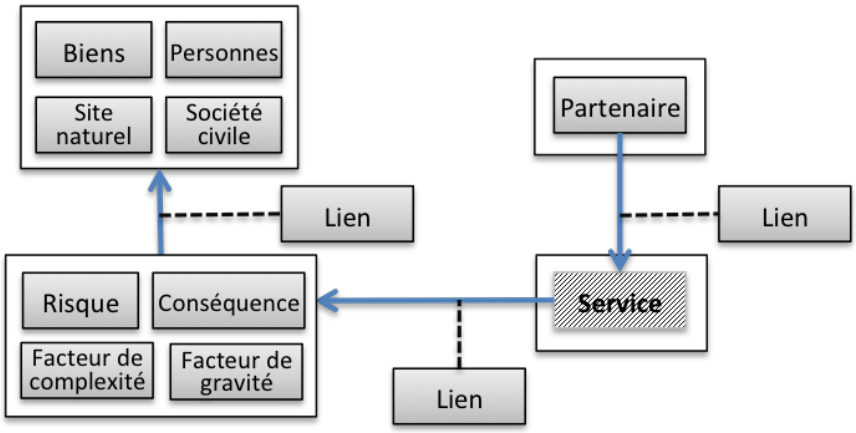


FIGURE V.17 – Le cas de la panne de service

Notons que plusieurs cas peuvent être vérifiés simultanément. Dans cette configuration là, le Service d'Agilité va classer les recommandations d'adaptation par ordre de priorité décroissante et les présenter ainsi triées à l'utilisateur.

Nous pouvons en conclure que la connaissance portée par le méta-modèle nous a permis de réduire drastiquement le champ des combinaisons possibles de tuples {concept ; opération} par la mise en place de classes d'équivalence. De plus, l'étude des liens entre les classes d'équivalence nous a permis de définir les conditions minimales à satisfaire pour choisir le niveau d'adaptation le plus pertinent au vu de l'évolution de la situation collaborative.

Nous allons maintenant présenter l'algorithme de choix d'adaptation. Afin d'en simplifier la lecture, les conventions suivantes ont été adoptées :

- **GPNC** symbolise un objet DetailledDifference (provenant du δ) dont le concept est de type *Goods*, *People*, *Natural Site*, ou *Civilian Society*. Il représente la classe d'équivalence *Sous Composant*,
- **RCGC** symbolise un objet DetailledDifference (provenant du δ) dont le concept est de type *Risk*, *Consequence*, *Gravity Factor*, ou *Complexity Factor*. Il représente la classe d'équivalence *Perturbation*,
- la **nuance** de l'adaptation prend la valeur **1** pour une adaptation corrective, **2** pour une adaptation méliorative.

Cet algorithme reprend chacun des 7 cas détaillés précédemment.

Choix_Adaptation(*delta*, *mTerrain*)

Cet algorithme permet de choisir le niveau d'adaptation à proposer à l'utilisateur
Paramètres : delta : objet contenant l'ensemble des différences détectées entre les modèles terrain et situation
mTerrain : fichier du modèle terrain
Variables : choix : tableau contenant les différentes possibilités de choix de l'adaptation, sous forme de 4-uplet besoin d'adaptation [booléen], type d'adaptation conseillé [chaine], nuance de l'adaptation [entier], argument pour l'adaptation [chaine]

Début
  Si ((delta contient un RCGC ajouté ou mis à jour) OU (delta contient un GPCN mis à jour ET le(s) RCGC lié(s) sont supprimés)) **Alors**
    choix ← Ajouter({VRAI, "CARACTERISATION", 1, ""})
  Si ((delta contient un RCGC supprimé ET contient un GPCN supprimé ET le RCGC est lié au GPCN) OU (delta contient un GPCN ajouté sans RCGC lié)) **Alors**
    choix ← Ajouter({VRAI, "CARACTERISATION", 2, ""})
  Si ((delta contient un RCGC supprimé) OU (delta contient un Acteur ajouté) OU (delta contient un Service ajouté)) **Alors**
    choix ← Ajouter({VRAI, "RESEAU", 2, ""})
  Si (delta contient un Service supprimé) **Alors**
    Si ((delta contient un RCGC supprimé) ET (le Service est lié au RCGC)) **Alors**
      choix ← Ajouter({VRAI, "RESEAU", 2, ""})
    Sinon
      choix ← Ajouter({VRAI, "RESEAU", 1,""})
    Fin Si

...à suivre...

```
...suite...
    Si (delta contient un Acteur supprimé) Alors
        Si ((delta contient un RCGC supprimé) ET (delta contient un Service supprimé)
        ET (le Service est lié à l'Acteur) ET (le Service est lié au RCGC)) Alors
            choix ← Ajouter({VRAI, "RESEAU", 2, ""})
        Sinon
            choix ← Ajouter({VRAI, "RESEAU", 1, ""})
        Fin Si
    Si (delta contient un Acteur mis à jour) Alors
        choix ← Ajouter({VRAI, "RESEAU", 1, ""})
    Si ((delta contient un Service mis à jour) ET (delta contient un RCGC ajouté ou
    mis à jour) ET (le Service et le RCGC sont liés)) Alors
        – ∴ – On mémorise l'identifiant du service concerné pour pouvoir le ré-
        exécuter
        argsExe← updatedService
        choix ← Ajouter({VRAI, "EXECUTION", 1, argsExe})
    Sinon
        choix ← Ajouter({VRAI, "CARACTERISATION", 2, ""})
    Fin Si
    Retour choix
Fin
```

V.4.4 Algorithme de proposition d'adaptation à l'utilisateur

Il est important de souligner que l'adaptation est une proposition faite par le SIM à l'utilisateur. Il s'agit là d'un support d'aide à la décision : le SIM ne va pas prendre la décision d'adapter ou non les processus, cette dernière revient à l'acteur de la collaboration qualifié pour ce type d'action.

C'est la raison pour laquelle nous avons introduit un algorithme de proposition d'adaptation : si une adaptation est reconnue comme pertinente par le Service d'Agilité, celle-ci est proposée à l'utilisateur, qui accepte ou refuse la proposition. Dans le cas du refus, une notification est affichée et le choix d'adaptation suivant dans la liste est proposé. Dans le cas d'une acceptation, l'appel à l'algorithme d'adaptation correspondant est effectué.

Les propositions d'adaptation corrective sont présentées en priorité à l'utilisateur. Les conseils d'adaptation à but mélioratif sont proposés dans un second temps.

Proposer_Adaptation(choix)

Cet algorithme permet de proposer un niveau d'adaptation à l'utilisateur
Paramètres : choix : tableau contenant la liste des adaptations possibles
Variables : adapter : booléen, qui détermine si une adaptation est conseillée ou non
typeAdaptation : chaine, qui indique le type d'adaptation recommandée
granularitéAdapt : eniter, la nuance de l'adaptation menée
decisionUtilisateur : booleen, qui détermine la décision de l'utilisateur

```
Début
    Action : Trier le tableau "choix" suivant la granularité ("1" puis "2") puis le type
    d'adaptation ("CARACTERISATION", "RESEAU" et enfin "EXECUTION")
    compteurTableau ←-1
    Si (choix <> vide) Alors
        Répéter compteurTableau ← compteurTableau + 1
        Si (choix[compteurTableau].adapter == VRAI) Alors
            Action : Afficher('Une adaptation des processus collaboratifs est conseillée,
            de type ' + choix[compteurTableau].typeAdaptation)
            Action : Afficher('Voulez-vous l'exécuter ? Si oui, tapez 1, sinon tapez 0')
            decisionUtilisateur ← Action : Action : valeur_saisie()
        Fin Si
        Jusqu'à choixUtilisateur == 1
        Si (decisionUtilisateur) Alors
            Action : Afficher('L'adaptation de type '+ typeAdaptation +' des
            processus collaboratifs est en cours d'exécution')
            Si (choix[compteurTableau].typeAdaptation="CARACTERISATION")
            Alors
                Adaptation_Caracterisation()
            Si (choix[compteurTableau].typeAdaptation="RESEAU") Alors
                Adaptation_Reseau()
            Si (choix[compteurTableau].typeAdaptation="EXECUTION") Alors
                Adaptation_Execution()
            Sinon
                Action : Afficher('Erreur : mauvais argument pour le type d'adapta-
                tion')
            Fin Si
        Sinon
            Action : Afficher('Vous avez choisi de ne pas adapter les processus')
        Fin Si
    Sinon
        Action : Afficher('Aucune action d'adaptation des processus collaboratifs n'est
        préconisée actuellement')
    Fin Si
Fin
```

L'algorithme Proposer_Adaptation(choix) se charge de proposer à l'utilisateur les recommandations d'adaptation déterminées par le Service d'Agilité. Elles sont passées en paramètre d'appel de l'algorithme sous forme d'un tableau de recommandations d'adap-

tation. Les recommandations d'adaptations sont déjà triées par ordre de priorité décroissant. Tant que l'utilisateur ne valide par la recommandation d'adaptation que le Service d'Agilité lui propose, ce dernier continue à les lui présenter. Une fois que l'utilisateur a fait son choix, le Service d'Agilité analyse le type d'adaptation (*CARACTERISATION*, *RESEAU*, *EXECUTION*) lié à la recommandation d'adaptation retenue et appelle la fonction d'adaptation pertinente (respectivement *Adaptation_Caracterisation()*, *Adaptation_Reseau()*, *Adaptation_Execution()*, que nous avons détaillées précédemment).

V.4.5 Retour d'expérience

Afin de faciliter le retour d'expérience, un historique des décisions de l'utilisateur vis-à-vis des conseils d'adaptation du Service d'Agilité est alimenté comme suit : horodatage, choix de l'utilisateur (granularité d'adaptation et type d'adaptation retenu), données des modèles *terrain* et *attendu*, de la matrice des coûts des opérations et de l'importance de l'instance, du delta résultant et de la liste des propositions d'adaptation déterminées. Ceci permet d'étudier la pertinence des propositions du Service d'Agilité à un instant t afin d'en améliorer les mécanismes de calcul du δ et de choix de l'adaptation mais aussi de « rejouer » la situation de crise et de simuler les conséquences de choix différents de ceux effectués lors de la crise qui a réellement eu lieu.

V.5 Conclusion

L'adaptation des processus collaboratifs, si elle est nécessaire, est conduite à l'un des trois niveaux de conception du Système d'Information de Médiation. Le choix du niveau de conception à partir duquel la conception du SIM est re-exécutée est déterminée par l'étude des tuples {concept, opération} contenus dans le δ calculé par le Service d'Agilité. Il faut noter que l'adaptation est intrinsèquement liée au domaine métier considéré, à travers l'étude des tuples {concept, opération}.

La combinatoire explosive générée par le nombre de tuples possibles peut être drastiquement réduite en se servant des propriétés des classes et des relations entre les classes du méta-modèle de la crise dont sont issus les concepts, générant ainsi des *classes d'équivalence*. Sur la base des combinaisons retenues entre ces classes d'équivalence, des propositions d'adaptation sont formulées. Si le δ vérifie plusieurs de ses combinaisons, c'est la solution d'adaptation la plus coûteuse qui est retenue. Ceci permet de faire face au pire cas d'évolution de la situation collaborative qui a amené à cette proposition.

Il est intéressant de noter que si l'adaptation est menée initialement dans un esprit correctif (afin d'atteindre les objectifs de la collaboration) compte tenu des évolutions de la situation collaborative, l'adaptation peut aussi être menée à un niveau plus fin afin d'éviter un gaspillage de moyens, tant humains que matériels et temporels.

Dans ce cas, laisser les processus en l'état ne gène pas la poursuite des objectifs en soi. Mais monopoliser des ressources pour exécuter une tâche qui n'a plus lieu d'être

n'est pas pertinent : par exemple, faire intervenir une équipe de pompiers pour éteindre un incendie qui n'existe plus (le feu n'était plus alimenté et s'est éteint de lui-même) empêche peut-être la mise à disposition de ces ressources humaines et matérielles pour intervenir sur un autre site du théâtre de la crise. Il faudrait prendre en compte cette dimension d'optimisation de l'utilisation des ressources, en s'aidant d'indicateurs métiers et d'une analyse des processus collaboratifs exécutés pour détecter les branches rendues obsolètes par effet de bord de ces même processus de réponse à la crise ou de par la nature de la crise.

Chapitre VI

Illustration et implémentation

Nous allons maintenant nous intéresser à l'implémentation du Service d'Agilité décrit dans les précédents chapitres, ainsi qu'à un exemple d'utilisation extrait du cas d'étude présenté dans le Chapitre I. Dans un premier temps, nous étudierons l'architecture technique et les différents composants du Service d'Agilité. Nous verrons que ce Service d'Agilité s'appuie sur plusieurs types d'architecture (SOA, EDA et Web). Puis nous illustrerons le fonctionnement de ce Service d'Agilité à l'aide d'un exemple de type crise nucléaire, inspiré de ceux créés dans le cadre des projets de recherche PLAY et SocEDA. Ce cas d'étude simplifié nous permettra d'illustrer les phases de mise à jour, de détection et de recommandation d'adaptation. Les différents événements étudiés dans cet exemple vont nous amener à rencontrer plusieurs types de recommandation d'adaptation.

VI.1 La réalisation du Service d'Agilité

Avant de développer le paramétrage du Service d'Agilité dans le but de jouer l'exemple extrait des cas d'études réalisés dans les projets de recherche PLAY [PLAY, 2010] et SocEDA [SocEDA, 2010], nous allons nous intéresser à l'architecture logicielle du Service d'Agilité : présentation des outils et services utilisés, agencement avec l'existant (héritage du projet MISE). Nous examinerons également le type de licence dont bénéficiera le prototype, compte tenu de ses dépendances avec des librairies et des composants tiers.

VI.1.1 Considérations préliminaires à la réalisation technique

Dans cette section, nous allons d'abord nous intéresser aux *outils utilisés* pour développer le code de notre prototype (langages, serveurs, outils de génération automatique, etc.). Puis nous présenterons les *différents Web Services* que nous utilisons, à savoir ceux du Design Time du SIM et celui développé afin d'assurer la partie de Complex Event Processing de notre Service d'Agilité.

VI.1.2 Outils utilisés

WSDL Pour les raisons exposées dans le Chapitre I, nous avons dû simuler les systèmes d'information (SI) des partenaires et les capteurs disséminés dans l'environnement. A cette fin, nous avons développé des Web Services qui simulent ces SI et ces capteurs. La réalisation des différents Web Services a été facilitée par l'utilisation d'un outil de génération de Web Service WSDL (développé au sein du Centre de Génie Industriel de Mines Albi), et de la librairie EasyWSDL développée par Petals Link/Linagora qui gère les descriptions de WSDL.

Cette librairie EasyWSDL implémente la norme *WS-Notification* contrairement à l'outil de génération de Web Services. Ce dernier impose donc une modification manuelle des données du WSDL généré afin d'ajouter le port d'émission et/ou le port d'abonnement aux événements (ainsi que les dépendances aux fichiers liés comme la description des *sujets* via lesquels les événements sont émis/collectés). L'outil de génération de Web Service (Figure VI.1) propose à l'utilisateur une interface minimaliste, lui permettant d'entrer le nom du service, les opérations et les paramètres typés de ces opérations.

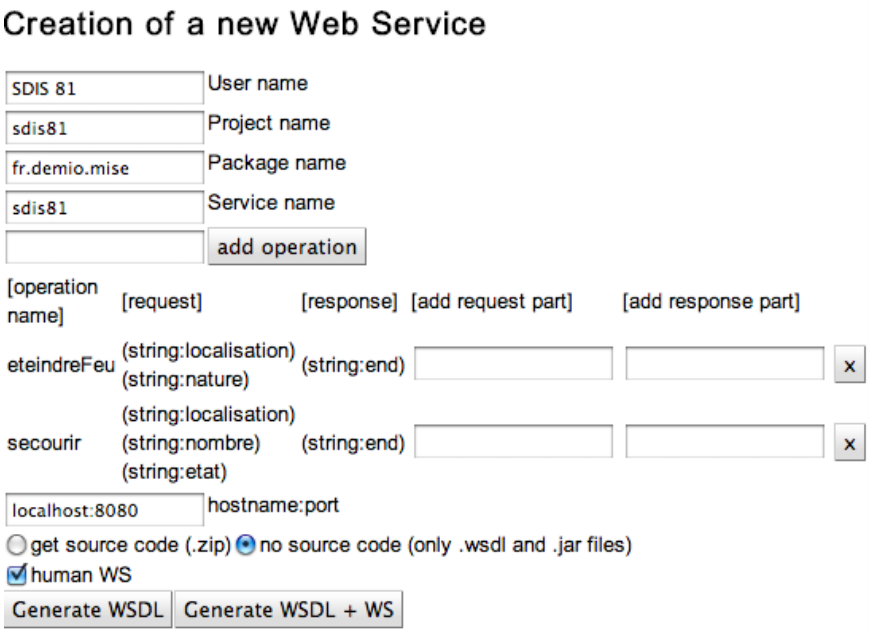


FIGURE VI.1 – Exemple d'utilisation de l'outil de génération des Web Services

La Figure VI.1 montre la création d'un Web Service du SDIS 81, qui comporte deux opérations *eteindreFeu* et *secourir*. Dans *eteindreFeu*, les informations d'entrée sont la *localisation* et la *nature* de l'incendie. Cette opération attend ensuite une réponse indiquant que l'opération a été exécutée. Nous avons ainsi doté une opération « humaine » (i.e. un traitement non logiciel) d'une interface de Web Service. C'est ce que nous appelons un Web Service d'Interface Humaine, qui permet d'intégrer toute activité humaine dans un workflow. La nature particulière de ce Web Service est mentionnée en cochant la case *human WS* sur l'interface de création. Dans le cadre de cette création de Web

Service, nous choisissons de ne générer que le WSDL.

Maven Une fois les informations nécessaires à la prise en charge de l'émission/abonnement aux événements manuellement ajoutées au WSDL, nous générerons le squelette du code source Java du Web Service à l'aide du plugin Apache Maven [1] WSOUI (réalisé par Petals Link/Linagora). Ensuite, charge à nous d'implémenter la logique métier propre à chaque Web Service i.e. la mécanique de chaque méthode, mais aussi l'alimentation en données des événements émis et/ou le traitement des événements reçus.

Bus de services (Petals ESB) Les Web Services des partenaires mais également ceux du Design Time ou celui assurant le CEP sont déployés sur le bus de services Petals ESB. La Figure VI.3 détaillée dans la section suivante montre l'intégration de ce bus dans l'architecture globale du prototype permettant la mise en œuvre du Service d'Agilité.

Langages Durant le développement de notre prototype, nous avons utilisé principalement les langages Java (1.6), XML (et ses dérivés XSD, XSL et XPath), WSDL ainsi que l'EQL (le langage de requête d'Esper).

VI.1.3 Services utilisés

Moteur CEP (Esper) Le moteur de CEP utilisé est *Esper 4.6.0* (dans sa version gratuite). Il propose une API Java : cela nous a permis de l'encapsuler dans un Web Service et de le doter d'une interface d'abonnement et de réception des événements au format WS-Notification. Nous avons à nouveau utilisé la librairie EasyWSDL pour ce faire. Ce WebService baptisé *MISECep* est embarqué (comme les autres Web Services) sur le bus de services Petals ESB.

Le code de l'encapsulation d'Esper dans un Web Service capable de produire et de recevoir des événements au format WS-Notification et de les faire traiter par Esper est disponible sur la forge du projet MISE, à l'adresse : `https://svn.mines-albi.fr/mise/documents/ambarthe/MISECEP/` [2].

ModelsUpdate Nous avons développé un autre Web Service (qui ne se base sur aucun applicatif tiers), nommé *ModelsUpdate*, qui est en charge de la collecte et de l'interprétation des événements filtrés et générés par MISECep. Une fois les données portées par ces événements analysées, *ModelsUpdate* va s'en servir pour mettre à jour les modèles *terrain* et *attendu* de la situation collaborative.

Desgin-Time MISE Dans le cadre de ces travaux, nous allons appeler les services de Design Time du Système d'Information de Médiation (SIM) qui implémentent les trois niveaux d'adaptation présentés dans le Chapitre V. Ces services du Design Time sont le

1. Apache Maven est un outil libre de gestion et d'automatisation de production de projet Java.
2. Une authentification est nécessaire : nous contacter si vous n'avez pas de compte utilisateur.

fruit des travaux de thèse de [Mu, 2012] et [Boissel-Dallier, 2012] qui se sont intéressés à la partie conception (Design Time) et déploiement (Runtime) du SIM, dans le cadre du projet MISE.

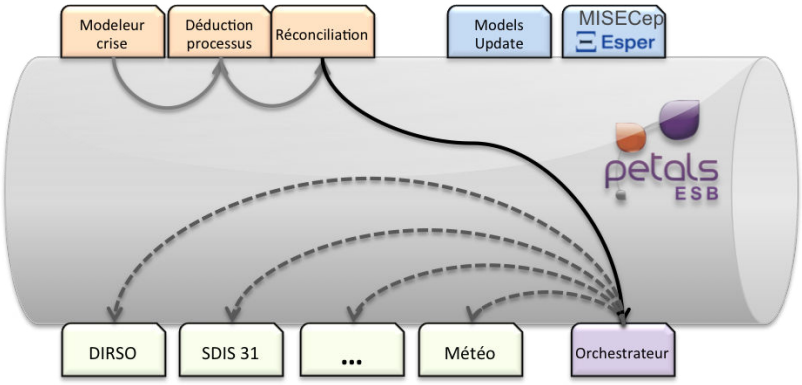


FIGURE VI.2 – Web Services de Design Time et de Runtime

De même que les services des partenaires appelés par l'orchestrateur lors de l'exécution des workflows), ces trois services servant à la conception des workflows sont hébergés sur le bus de services Petals ESB (Figure VI.2).

L'intérêt majeur d'une telle infrastructure réside dans la facilité avec laquelle il est possible de rappeler les services de Design Time, même si l'on se trouve en phase d'exécution des workflows (i.e. en phase de Runtime). Cette caractéristique est particulièrement appréciée lorsqu'on se place dans un contexte d'adaptation des workflows, où la capacité de rebouclage vers les outils de conception est cruciale.

VI.1.4 Architecture technique du Service d'Agilité

Comme nous avons pu le voir dans les précédents chapitres, l'architecture du Service d'Agilité [3] et son intégration dans la chaîne collaborative s'appuie à la fois sur une architecture orientée services (SOA) et une architecture dirigée par les événements (EDA).

L'architecture du Service d'Agilité est schématisée dans la Figure VI.3.

Elle suit également une logique architecturale 5-tiers composée d'un navigateur Internet (client léger), de JSP/Servlets (logique de représentation) [4], d'un serveur Apache Tomcat (serveur d'applications), d'un bus de services (autre serveur d'applications) et d'un serveur stockant les données (sous forme de fichiers XML actuellement). Les échanges se font via le protocole HTTP.

3. Le Service d'Agilité comprend les outils développés pour réaliser la collecte des données, la mise à jour des modèles, leur comparaison, et la recommandation d'adaptation.

4. Affichage de la réponse (résultat d'une requête) dans une interface graphique.

FIGURE VI.3 – Architecture technique de l'implémentation du Service d'Agilité

Notre Service d'Agilité peut être scindé en deux parties logicielles :

- La première partie s'occupe de collecter les événements émis par le terrain et la supervision des workflows et de les analyser grâce au Web Service MISECep qui encapsule Esper. Puis, sur la base des événements traités et émis par le CEP, le Web Service ModelsUpdate va réaliser la mise à jour des modèles *terrain* et *attendu* (qui sont stockés dans un serveur de fichier) Les Web Services MISECep et ModelsUpdate sont hébergés sur le bus de services Petals ESB,
- La seconde partie s'attache à exploiter les modèles mis à jour et à proposer à l'utilisateur une visualisation des résultats obtenus, via une interface d'application web accessible grâce à un navigateur Internet. Les logiques métier et de représentation sont portées par l'application web *AgilityService.war* qui est déployée sur le serveur d'applications Apache Tomcat. *AgilityService.war* est composée d'un ensemble de librairies que nous avons développées et qui assurent les fonction de détection (*XMLCompare.jar*) et de recommandation d'adaptation (*AdaptationDeductionCrisis*), ainsi que d'une servlet qui gère les échanges (requête/réponse) avec le client léger. Cette servlet va communiquer avec la JSP *AgilityService* et les Java Beans [5].

L'utilisateur accède au Service d'Agilité via une interface web qui lui propose :

- La visualisation des représentations graphiques des modèles *terrain* et *attendu*,
- La présentation du résultat de la détection d'une divergence entre les modèles et du détail de cette divergence,
- La visualisation des recommandations d'adaptation déduites sur la base de la divergence calculée,
- La possibilité de choisir une recommandation parmi celles proposées (suivant le choix effectué, l'utilisateur sera redirigé vers l'un des trois services de conception du SIM : modeleur de la crise, déduction des processus ou réconciliation syntaxo-sémantique).

Ces deux librairies s'appuient sur les données contenues dans les modèles *terrain* et *attendu*, ainsi que dans les paramètres (*params*). A l'heure actuelle, ces données sont stockées sous forme de fichier XML [6]. Remarquons que ces données sont mises à jour en temps réel par le Web Service *ModelsUpdate*. Nous allons revenir dans la suite de cette section sur la provenance des informations utilisées par ce service pour mettre à jour ces modèles.

L'utilisateur sollicite d'abord une détection de la divergence entre les modèles *terrain* et *attendu* (appel aux fonctionnalités de XMLCompare). Si une divergence (supérieure ou égale au seuil défini dans le fichier *params.xml*) est détectée, alors l'utilisateur peut solliciter un conseil d'adaptation des processus collaboratifs en appelant *AdaptationDe-*

5. Un Java Bean est un composant Java réutilisable qui permet essentiellement de persister les données et de communiquer au sein d'une application

6. On pourrait envisager un stockage des flux XML dans une base de données dans une deuxième version du prototype.

ductionCrisis.

Sur le bus de services Petals ESB, on retrouve différents types de Web Services (en plus de ceux qui appartiennent au Service d'Agilité) :

- Ceux dédiés à la conception du SIM : Modeleur crise, Déduction processus, Réconciliation,
- Celui dédié à l'exécution du SIM : Orchestrateur,
- Ceux simulant les SI des partenaires de la cellule de crise (dont les services sont sollicités par le SIM) : SDIS 81, SDIS 31, SAMU 31, DIRSO, Gendarmerie,
- Ceux simulant les capteurs disséminés sur le théâtre de la crise (bâtiments, stations météo, stations Teleray [7], capteurs embarqués, etc.) : Radiation, Météo, Température. Pour ces derniers, nous n'avons représenté qu'un Web Service de chaque type sur le schéma d'architecture. Dans le prototype nous en avons implémentés plusieurs afin de simuler l'existence de plusieurs capteurs de radiation, de météo et de température.

Hormis les Web Services dédiés à la conception du SIM, tous implémentent la norme WS-Notification (toujours grâce à la librairie EasyWSDL développée par Petals/Linagora) de façon à pouvoir émettre des événements sur le réseau et à s'abonner à des *sujets* d'événements.

Les Web Services émettent leurs événements sur le réseau. Le Web Service *MISECep* va s'abonner à tous les sujets d'événements potentiellement intéressants. Les choix d'abonnement sont des actions de paramétrage qui sont effectuées manuellement par l'utilisateur (pour l'instant). Il serait possible de créer des tables de correspondance entre des choix d'abonnements et des domaines particuliers (crise nucléaire, crise routière, sous-traitance aéronautique, etc.) qui seront utilisées avec des outils de correspondance syntaxique/sémantique afin d'automatiser cette étape. Grâce à ces abonnements, MISECep collecte des flux d'événements provenant tant de la supervision de l'exécution des processus (événements émis par les activités des processus exécutés par l'Orchestrateur) que ceux provenant du théâtre de la collaboration (événements émis par les capteurs, etc.). MISECep met ses flux à disposition du moteur de CEP qui va filtrer et/ou déduire des événements en fonctions des règles métier qui ont été paramétrées par la cellule de crise.

MISECep va ainsi créer de nouveaux événements qui retranscrivent l'état de la situation collaborative (émis sur le sujet *eventField*) et l'état d'avancement des processus collaboratifs (émis sur le sujet *eventExpected*). Le Web Service ModelsUpdate va s'abonner aux sujets *eventField* et *eventExpected* afin de collecter les informations nécessaires à la mise à jour des modèles *terrain* et *attendu*.

Cette mise à jour se produit en temps réel : tout nouvel événement généré par MISE-

7. Teleray est le réseau de capteurs de taux de radiation dans l'air ambiant géré par l'IRSN.

Cep est pris en compte par *ModelsUpdate* qui enrichit le modèle concerné. Il n'y a donc pas besoin d'une intervention utilisateur pour que cette dernière se déclenche, contrairement aux actions de détection et d'adaptation.

VI.1.5 Licence logicielle

Dans nos diverses briques logicielles, nous utilisons des librairies et des moteurs tiers. La licence de notre prototype est donc conditionnée par la licence la plus contaminante et la plus restrictive des librairies utilisées. Le tableau suivant liste les différentes librairies utilisées en dépendance et leur licence.

Élément logiciel	**Licence**	**Mix code propriétaire**	**Contamination**
XMLUnit 1.4	BSD	Oui	Non
Esper 4.6.0	GPL v2	Non	Oui
Petals ESB	LGPL	Non	Non
EasyWSDL	BSD	Oui	Non
XMLCompare	LGPL	Oui	Non
AdaptationDeductionCrisis	LGPL	Oui	Non

TABLE VI.1 – Licences des composants tiers utilisés dans l'implémentation du Service d'Agilité

Nous pouvons constater que nous avons affaire à trois types de licence : BSD (Berkeley Software Distribution), GPL v2 (General Public License v2) et LGPL (Lesser General Public License) (cf. Table VI.1).

Un code source publié sous *licence BSD* peut être incorporé dans tout type de licence, y compris dans des solutions propriétaires (les modifications apportées à un code BSD peuvent être propriétaires). La licence BSD est extrêmement permissive : il est possible de reproduire, utiliser, modifier et redistribuer le code. Cependant, elle permet à l'auteur du code source d'en conserver la paternité. Elle n'est pas contaminante et demande simplement à joindre une reproduction de la licence BSD dans le code source et l'exécutable.

La *licence GPL* autorise l'utilisation, la copie et la reproduction du code source ainsi que sa modification sous conditions. Elle permet de conserver la paternité du code dans la mesure où toute modification doit être tracée. Cette licence est contaminante et le code source ne peut être réutilisé que dans une solution également sous licence GPL.

Enfin, la *licence LGPL* est une version allégée de la licence GPL. Sa caractéristique essentielle est la suppression de la contamination induite par la licence GPL.

La licence la plus restrictive et contaminante est donc la GPL, provenant d'Esper. Notre prototype de Service d'Agilité est donc soumis à la *licence GPL*. Ceci ne pose pas de problème particulier pour la redistribution de XMLUnit, car un code publié sous licence BSD peut être redistribué sous licence GPL.

VI.2 Présentation du cas d'étude

Le cas d'étude présenté ici est un extrait simplifié du cas d'étude réaliste créé pour les projets de recherche PLAY et SocEDA (voir Chapitre I). Pour les besoins de l'illustration dans ce chapitre, certaines libertés ont été prises par rapport au cas d'étude initial. Nous allons illustrer l'utilisation de notre Service d'Agilité avec un exemple fictif de réponse à une crise nucléaire.

En fin de matinée, une fuite est détectée dans la centrale nucléaire « Les Coquilloux ». La paroi permettant les échanges de chaleur entre les liquides des circuits primaire et secondaire (voir Section I.4.2.2) présente une faiblesse structurelle (due à une usure normale) qui entraîne une fuite. Le liquide du circuit primaire se déverse dans le circuit secondaire, le contaminant. La vanne permettant de réguler la pression de vapeur dans le circuit secondaire s'ouvre pour compenser la pression qui augmente dans le circuit secondaire (à cause de la présence de liquide du circuit primaire). Les opérateurs tentent de la fermer pour éviter que cette vapeur désormais contaminée ne se répande dans et à l'extérieur de la centrale. Mais le système de fermeture de la vanne tombe en panne, empêchant la retenue de la vapeur et la laissant s'échapper dans l'atmosphère. Rapidement, la cellule de crise est mise en place (il est 11h40) et s'occupe de caractériser la situation de crise à l'aide du modeleur de crise (Figure VI.4).

Nous pouvons constater que les acteurs de la cellule de crise (SDIS 81, DIRSO, SAMU 81 et Gendarmerie) et leurs services[8], ainsi que le système d'étude (la crise et son environnement) sont représentés dans ce modèle de la situation de crise à t_0.

Une école primaire et une zone résidentielle se trouvent à proximité de la centrale nucléaire, en direction du Nord-Nord-Est. La route départementale qui relie le village le plus proche à la zone résidentielle passe devant la centrale nucléaire. Et en toute logique, une rivière est située dans le voisinage immédiat de la centrale nucléaire afin d'alimenter le circuit de refroidissement du circuit secondaire.
Le personnel de la centrale est soumis à un risque de « contamination lourde », compte tenu de leur présence dans l'enceinte de la centrale. La population constituée d'élèves, de personnel éducatif et d'habitants (situés dans l'école primaire « Louise Michel » et le lotissement « 14 juillet 1789 ») risque quant à elle une « contamination légère » et d'être en état de « panique ».

8. Ce cas d'étude n'a qu'une vocation d'illustration du Service d'Agilité et a été volontairement simplifié par rapport au cas réel. Nous n'y retrouverons qu'un nombre réduit d'acteurs, de services, etc.

Midi approchant, le trafic routier sur la « D12345 » va aller croissant, les salariés du village voisin rentrant déjeuner chez eux ou allant chercher leurs enfants à l'école. Il y a donc un risque « d'encombrement routier ».
Les biens matériels (l'école, la route, les habitations) et l'environnement (l'Agout) sont eux aussi soumis à un risque de « contamination » par la matière radioactive transportée par la vapeur contaminée.

Tous ces éléments analytiques sont formalisés dans le modèle présenté dans la Figure VI.4.

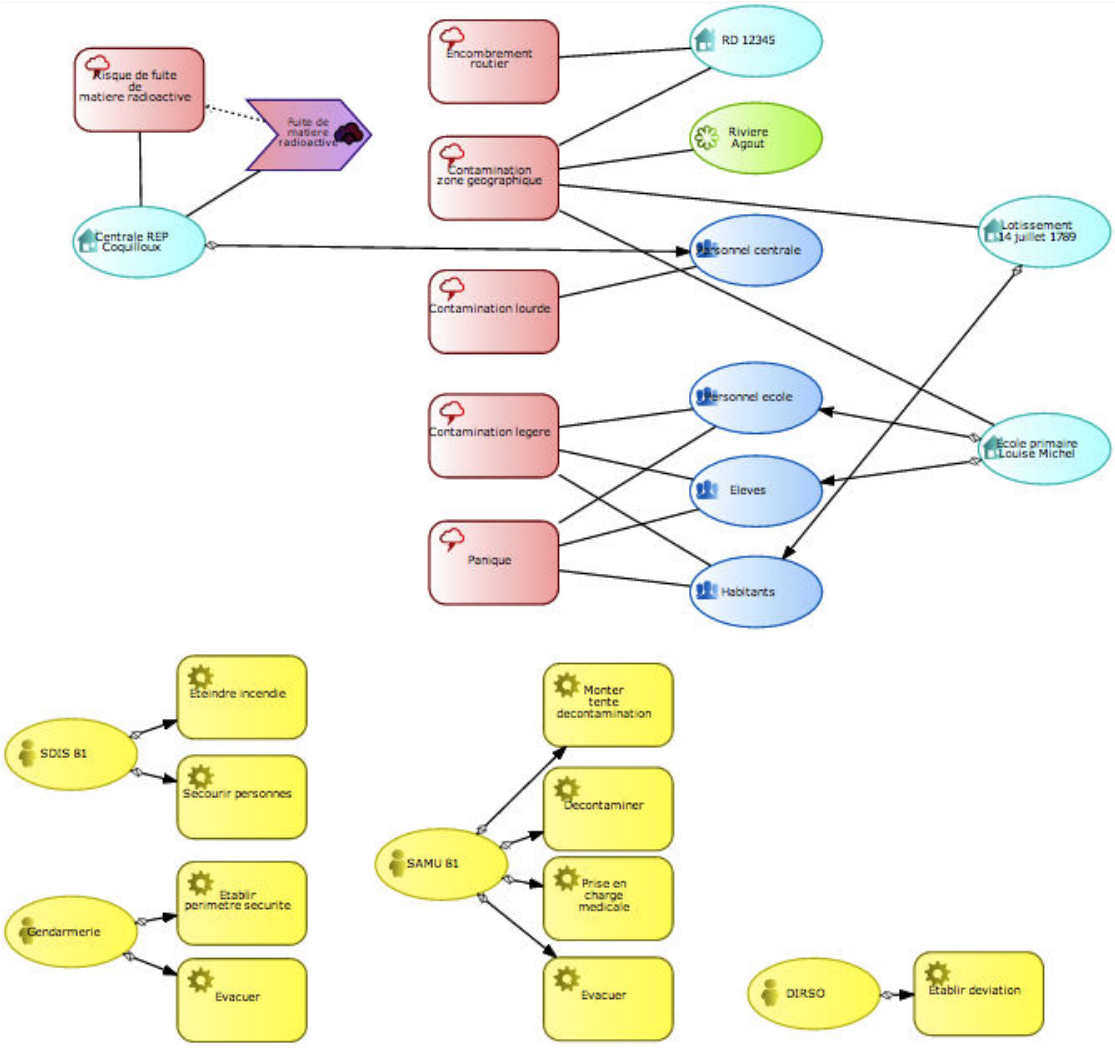


FIGURE VI.4 – Modèle de la situation de crise à t_0

Les acteurs mettent en place plusieurs mesures à travers des processus décisionnels (choix d'un périmètre de sécurité, choix de la mise en place d'une déviation pour court-circuiter la RD12345) et opérationnels (établir le périmètre de sécurité, établir la

déviation, secourir les personnes). La Figure VI.5 présente un processus opérationnel réalisé par la DIRSO, afin de mettre en place la déviation sur la RD 12345. Dès lors que la DIRSO reçoit la décision préfectorale d'établir une déviation, elle se met en attente du plan de déviation. Dès qu'elle le reçoit, elle l'analyse afin de réquisitionner le bon nombre et le bon type de barrières et de panneaux de signalisation. Dès que la DIRSO reçoit les ressources nécessaires, elle met en place le plan de déviation, et émet un événement collecté par l'Event Cloud qui fait état du déploiement du plan de déviation.

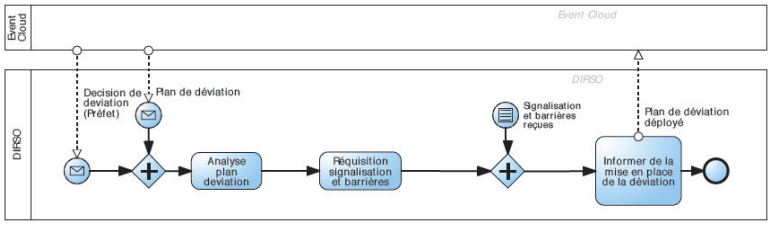


FIGURE VI.5 – Mise en place de la déviation par la DIRSO

Les autres processus sont présentés en Annexe M : ils émettent plusieurs événements dont ceux de type *Activité* qui nous intéressent plus particulièrement pour la supervision des processus . Ils nous permettent de connaître l'état des activités des processus (wait, wip, finish, ko) (cf. Chapitre III).

Pour l'instant, aucune décision d'évacuation de la population n'a été prise : l'analyse, réalisée par le moteur de CEP, des données envoyées par les capteurs de radiation et de météo (vitesse du vent et direction du vent) montre que le taux de radiation moyen n'augmente pas et que le nuage de particules contaminées ne se déplace pas (vitesse du vent extrêmement faible et dans la direction opposée aux zones habitées).

VI.3 La mise en place de l'agilité

Avant de dérouler le reste de notre cas d'étude (réponse simplifiée à un accident nucléaire civil), nous allons nous pencher sur le paramétrage du Service d'Agilité dans ce cadre de cet exemple.

VI.3.1 L'abonnement aux événements

Dans notre cas d'étude, nous souhaitons collecter les données provenant du terrain pour suivre l'évolution de la crise (risques, populations impactées, etc.), et celles provenant des processus (état de l'activité, état des décisions, etc.).

Précisons que lorsque nous disons que tel élément logiciel va s'abonner à un sujet, il faut comprendre que l'abonnement est décidé et paramétré manuellement par l'utilisateur. Comme nous l'avons vu en Section II.2.1.4, des pistes d'automatisation des abonnements sont à l'étude.

Le moteur de CEP va s'abonner aux sujets concernant la surveillance des taux de radioactivités, de l'activité éolienne, de la température dans l'air ambiant, des états des activités des processus, etc. Les EPA du moteur de CEP (cf. Chapitre II, Section II.2.1.4) vont traiter les événements par filtrage, transformation ou application de motifs. Le détail des règles de traitement des événements est donné dans la section suivante. Les événements filtrés et/ou dérivés par les EPA vont être émis par le CEP sur deux sujets distincts : « eventField » et « eventExpected ».

L'outil de mise à jour des modèles *terrain* et *attendu* va quant à lui s'abonner aux événements dont les sujets sont « eventField » et « eventExpected ». Les événements issus de « eventField » (respectivement « eventExpected ») vont être utilisés par l'outil afin de mettre à jour le modèle *terrain* (respectivement *attendu*) de la situation collaborative.

Enfin, les activités des processus vont s'abonner aux sujets d'événements qui les intéressent : décisions du préfet, conseils de l'IRSN, etc. Les événements concernés par ces sujets vont marquer le déclenchement ou l'arrêt de certaines activités des processus.

D'autre part, le moteur de CEP ne peut traiter les types d'événement dont il ne connaît pas la structure. Ainsi, un des aspects essentiels du paramétrage d'Esper est de lui communiquer le format des événements qu'il aura à analyser (en entrée) et à générer (en sortie). Pour cela, chaque type d'événement est décrit dans le fichier *esper.mise.cfg.xml* (disponible en Annexe L). Ce fichier réalise une correspondance entre les composants du type de l'événement (éléments, attributs) —via des expressions XPath— et les alias qui seront utilisés dans les règles CEP.

```
1  <xpath-property property-name="alertGravity"
2        xpath="ns18:NotificationMessage/ns18:Message/ns2:alertEvent/ns2:gravity"
3        type="string" />
```

Code VI.1 – Exemple de correspondance entre composant du type d'événement et alias

Le code VI.1 présente un extrait de *esper.mise.cfg.xml* et montre la correspondance entre l'élément *gravity*, fils de l'élément *AlertEvent*[9] (décrivant le type d'événement *Alert*) avec l'alias *alertGravity*. Cet alias va récupérer la valeur de l'élément dans l'événement et sera utilisé dans les règles portées par les EPA, permettant ainsi à Esper d'analyser l'événement reçu.

VI.3.2 Les règles métier de traitement des événements

Seule la règle métier *Détection d'un incendie* est détaillée dans cette section : c'est la seule concernée par les « zooms » effectués à différents moments de la situation collaborative ($t_0 + 20min$, $t_0 + 40min$ et $t_0 + 70min$).

9. D'après l'expression XPath, on constate que *AlertEvent* est contenu dans *Message*, lui-même fils de *NotificationMessage* (qui est la notification de l'événement).

Détection d'un incendie La détection d'un incendie peut se faire en corrélant les mesures de températures relevées par plusieurs capteurs situés à une distance raisonnable les uns des autres. La règle ci-dessous indique au service MISECep (qui encapsule le moteur de CEP Esper) qu'en cas de réception d'événements de type *Mesure* faisant état d'une température supérieure à 100°C et émis par deux capteurs différents (donc d'identifiants uniques différents), ces capteurs étant situés à proximité les uns des autres, dans un laps de temps de 10 minutes, alors MISECep doit créer et émettre un nouvel événement de type *Alerte*, indiquant une conséquence [10] nommée incendie et localisée dans le même secteur que les capteurs. A et B représentent deux événements de type *Mesure* reçus par MISECep.

SI A.id $\neq$ B.id
ET A.localisation $\sim$ B.localisation
ET A.unit = °C
ET B.unit = °C
ET A.value $\geqslant$ 100
ET B.value $\geqslant$ 100
TOUTES LES 10 minutes
==>Alerte{Consequence ; Incendie_secteur_A.localisation ; A.localisation ; gravite haute}

Les autres règles mises en place sont détaillées en Annexe N. La règle de détection d'un incendie est injectée dans Esper, sous la forme suivante (Figure VI.6). La règle

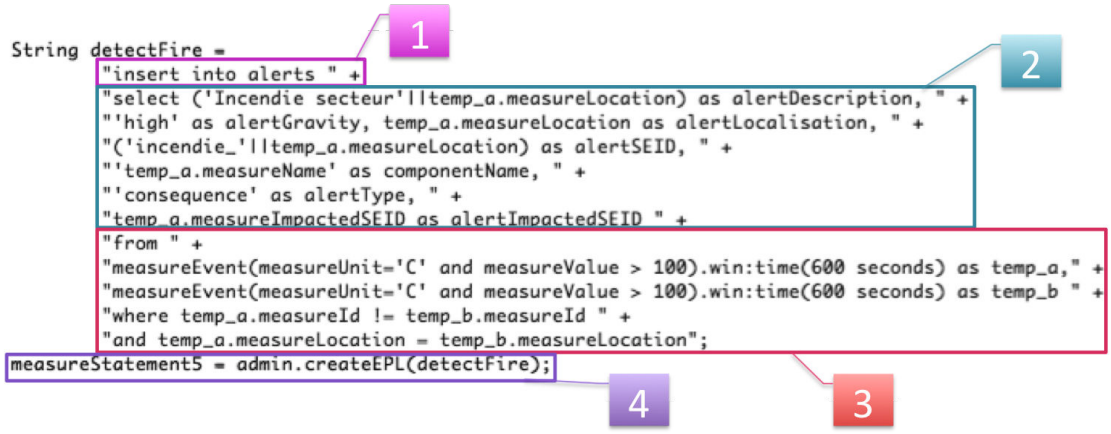


FIGURE VI.6 – Règle de détection d'un incendie écrite en EQL

écrite en EQL créé (partie **(1)**) un événement de type *Alerte*, que l'on peuple (partie **(2)**) à l'aide de données statiques (comme le *type*) et dynamiques qui proviennent des événements *Mesure* (partie **(3)**) qui ont satisfait les conditions de la règle. La règle est

10. Pour pouvoir ensuite faire le lien avec le modèle de la crise et mettre à jour le modèle *terrain*

ensuite injectée dans le moteur de traitement des événements complexes (partie **(4)**). Nous pouvons noter que le langage EQL est extrêmement proche du langage SQL.

VI.3.3 Les matrices de coût et d'importance, le seuil

Nous avons vu précédemment que nous définissons la divergence globale entre les modèles *terrain* et *attendu* comme étant la somme de toutes les différences δ_i détectées entre les points des deux modèles. Chaque δ_i est le produit du *coût de la différence* (la suppression d'un partenaire, l'ajout d'un risque, la modification d'une activité, etc.) et de son *importance* (entre un risque d'incendie nucléaire et un risque de feu de poubelle, le premier est plus important que le second).

Dans le cas de notre cas d'étude, nous avons fixé l'importance de toutes les instances à 1 étant donné le peu d'instances mises en jeu dans les modèles. Les coûts de chaque différence est défini dans le tableau ci-après (Table VI.2).

	Opération		
Concept	Insertion	Suppression	Modification
Actor	1	4	3
Service	1	3	1
Risk	2	1	2
Consequence	3	1	2
Gravity Factor	2	1	2
Complexity Factor	2	1	1
Goods	1	1	2
Civilian Society	1	1	2
People	2	1	2
Natural Site	1	1	1

TABLE VI.2 – Matrice des coûts {opération × concept}

D'autre part, les acteurs de la cellule de crise ont fixé δ_{seuil} égal à 2.

Notons que le contenu de la matrice des coûts a été décidé de façon totalement arbitraire. Il est donc possible (et même souhaitable) de faire appel à des méthodes permettant de justifier ces choix (algorithmes génétiques par exemple).

VI.4 Déroulement du cas d'étude

Dans cette section, nous allons nous intéresser aux instants significatifs, i.e. aux instants où une divergence a été détectée entre les modèles *terrain* et *attendu*. Nous étudierons le δ calculé, et les recommandations d'adaptation suggérées par le Service d'Agilité.

VI.4.1 À $t_0 + 20min$

À 12h00, MISECep génère un événement *Alerte* faisant état d'un incendie dans le secteur nord du site de la centrale nucléaire. En effet, les événements *Mesure* émis par les capteurs de température disséminés sur le site et alentours (et consommés par Esper) font état d'une augmentation de la température au-delà du seuil de 100°C durant les 10 dernières minutes dans le bâtiment situé le plus au nord du site de la centrale nucléaire.

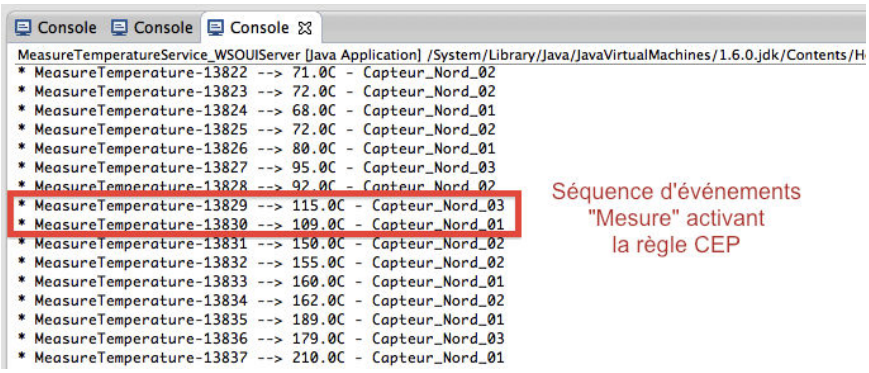


FIGURE VI.7 – Evénements *Mesure* émis par les capteurs

La capture d'écran du suivi (en mode console) de l'émission des événements *Mesure* (Figure VI.7) montre l'augmentation de la température en divers points du bâtiment et le dépassement du seuil de 100°C.

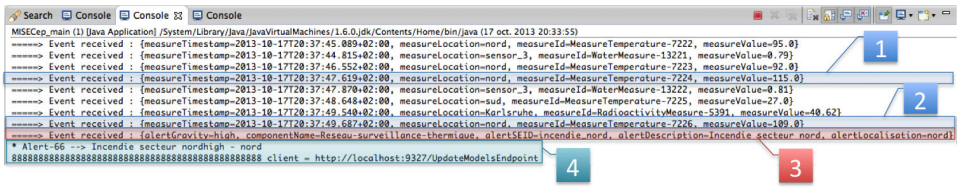


FIGURE VI.8 – Réception d'événements *Mesure* émis et déclenchant la règle d'émission d'une *Alerte* « incendie » par MISECep

Cela va provoquer le déclenchement de la règle *Détection d'incendie* qui génère un événement *Alerte* comme le montre la capture d'écran de la console d'exécution de MISECep (Figure VI.8). Les points **(1)** et **(2)** représentent deux événements *Mesure* émis par deux capteurs différents mais situés dans la même zone géographique (nord). Les températures qu'ils ont relevées sont supérieurs à 100°. De plus les événements se sont produits dans l'intervalle de temps défini dans la règle (10 minutes). Un événement *Alerte* a été émis dans le flux d'événements qu'Esper « écoute » (point **(3)**) : ceci explique que ce nouvel événement généré par Esper soit affiché comme un événement provenant d'une autre source. Le point **(4)** représente des traces (la seconde trace étant due à la récupération de l'événement par *ModelsUpdate* sur le point final d'adresse `http://localhost:9327/UpdateModelsEndpoint`).

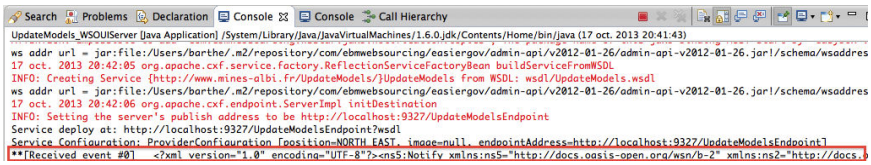


FIGURE VI.9 – Réception par ModelsUpdate de l'événement *Alerte* émis par MISECep

La collecte de l'événement par ModelsUpdate est visible dans la Figure VI.9 : l'enveloppe complète de la notification est affichée dans la console à la réception de la notification d'événement. Après traitement de la notification « brute », son contenu est extrait et les données de l'événement seront exploitées afin de mettre les modèles à jours.

VI.4.1.1 La mise à jour des modèles

L'événement *Alerte* est collecté par ModelsUpdate qui se sert de ses données pour mettre à jour le modèle *terrain* (Figure VI.10) : il s'agit d'un risque concrétisé (donc d'une conséquence au sens du méta-modèle de la crise), qui prendra comme caractéristique « Incendie_Nord » (pour l'identifiant), « Incendie secteur nord » pour le nom, « centrale » (pour le composant impacté), « secteur nord » pour la localisation.

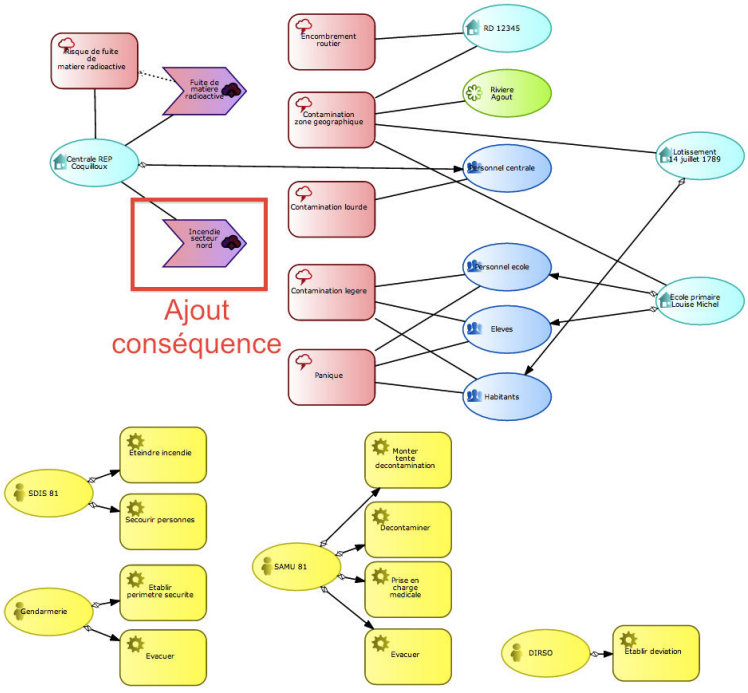


FIGURE VI.10 – Modèle de la situation de crise à $t_0 + 20min$

VI.4.1.2 Le calcul du delta

La détection est réalisée en continu de façon automatique. Un pas de temps peut être fixé afin de s'adapter aux ressources matérielles et au réseau sur lesquels le Service d'Agilité est déployé (dans notre cas, nous avons fixé ce pas à 3 minutes). Il faut souligner le fait que l'utilisateur dispose d'un moyen de déclencher lui-même l'adaptation (pour outrepasser le délai imposé par le pas de temps, par exemple) : il conserve le contrôle de l'exécution de la détection. Le Service d'Agilité informe l'utilisateur qu'une divergence a été relevée entre les deux modèles et lui en présente les détails (Figure VI.11). Le δ est égal à 4. On trouve bien une *Consequence* ajoutée (ce qui signifie qu'elle existe dans le modèle *terrain* mais pas dans le modèle *attendu* qui sert de référence) et le lien qui l'a lie à un des composants du système étudié (la centrale ici).

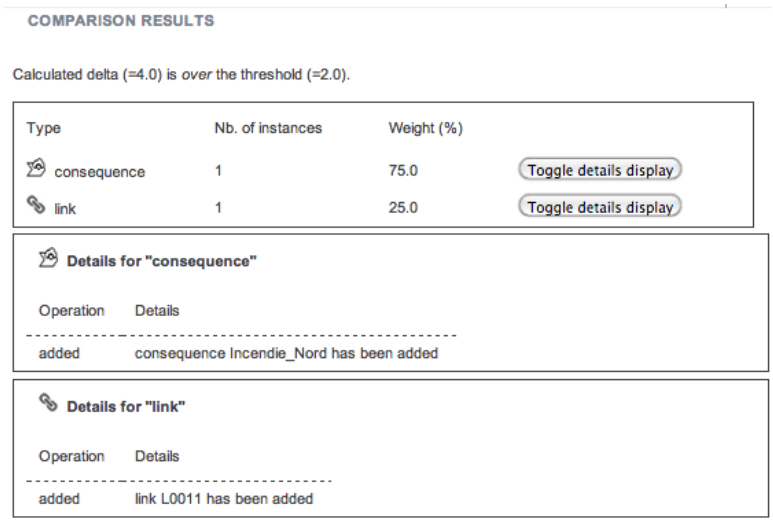


FIGURE VI.11 – Résultat de la détection à $t_0 + 20min$

VI.4.1.3 Le choix d'adaptation

Le δ calculé étant supérieur au δ_{seuil}, la fonction de proposition de choix d'adaptation est proposée à l'utilisateur. La capture d'écran de l'interface du Service d'Agilité (Figure VI.12) montre le résultat de l'exécution de l'algorithme du conseil en adaptation. Notons

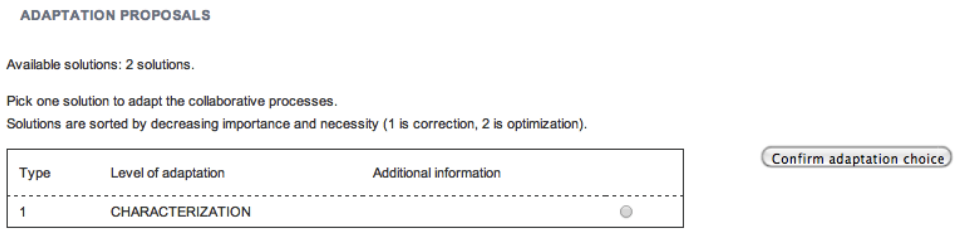


FIGURE VI.12 – Proposition d'adaptation à $t_0 + 20min$

que tant que l'utilisateur n'a pas sélectionné l'action d'adaptation à mener, le bouton de validation reste inactif. Ici, compte tenu des différences constatées, le seul choix possible est **Caractérisation** (Characterization).

Il s'agit de caractériser à nouveau la situation collaborative. Celle-ci ayant changé et les processus métiers mis en jeu ne répondant pas à cette évolution, il est nécessaire de redéfinir ces derniers sur la base d'une situation mieux renseignée et actualisée.

L'utilisateur a alors accès au Modeleur de la situation de crise (Figure VI.13). Il ne part pas d'une feuille blanche : le modèle *terrain* —le plus représentatif de la situation réelle— est utilisé comme base de travail pour y apporter les compléments nécessaires (s'il y a lieu).

Après un éventuel enrichissement du modèle, l'utilisateur clique sur le bouton « Terminer » (Finish), ce qui a pour effet de (i) réinitialiser les modèles *terrain* et *attendu* par duplication du modèle modifié par l'utilisateur, (ii) de relancer la chaîne de conception et déploiement des processus collaboratifs exécutables.

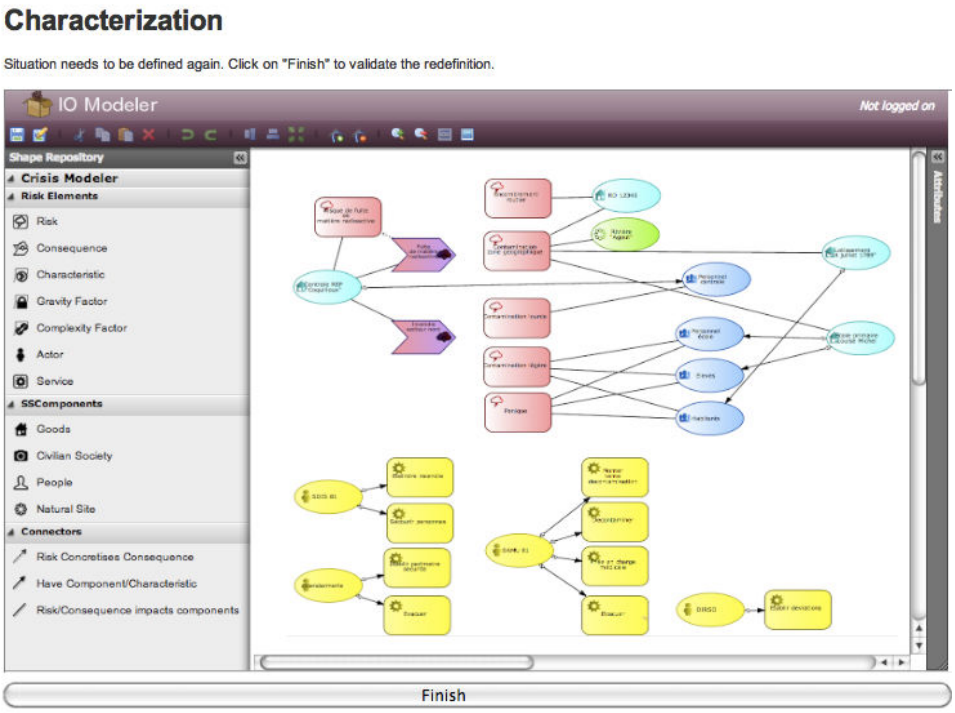


FIGURE VI.13 – Choix de retour à la caractérisation de la situation à $t_0 + 20min$

L'adaptation a eu pour effet la mise en place d'un processus de niveau décisionnel et d'un processus opérationnel, exécuté respectivement par le responsable du SDIS 81 et par les pompiers d'une unité d'intervention incendie du SDIS 81, afin d'éteindre le feu (Figure VI.14). Le responsable du SDIS 81 reçoit un événement de type *Consequence*

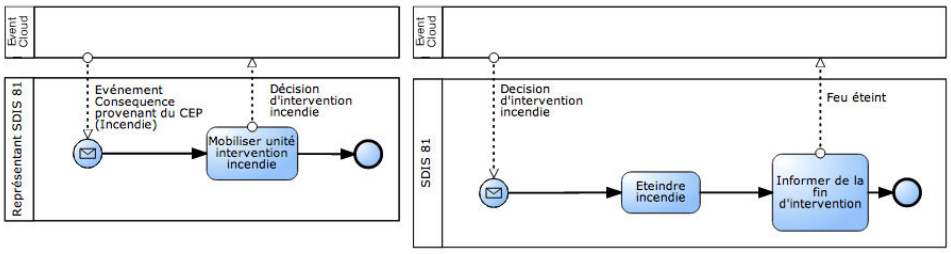


FIGURE VI.14 – Résultat de l'adaptation à $t_0 + 20min$

l'informant d'un incendie : il va mobiliser une unité d'intervention incendie pour répondre à cette nouvelle menace. Sa décision est envoyée sous forme d'événement à l'Event Cloud. Quand le SDIS 81 reçoit un événement l'informant d'une mobilisation pour intervention, l'unité se dépêche sur les lieux de l'incendie. Une fois le feu éteint, le SDIS 81 émet un événement mentionnant le résultat de leur service *Eteindre Incendie* dans l'Event Cloud

VI.4.2 À $t_0 + 40min$

À $t_0 + 40min$, la déviation est mise en place : l'opération menée par la DIRSO dans le processus opérationnel correspondant a généré l'événement *finish* pour indiquer au SIM (via MISECep) la bonne fin d'exécution de l'activité. D'autre part, les observations provenant du *terrain* indiquent qu'aucun véhicule civil n'est présent sur la portion de D12345 court-circuitée et qu'il n'y a donc pas d'encombrement.

VI.4.2.1 La mise à jour des modèles

MISECep reçoit ces événements : il filtre le premier (de type *Activité*), qui est reémis sans modification sur le sujet *eventExpected* et qui concerne l'état du service *Etablir déviation*. Quant au second, il permet de statuer sur la suppression du risque d'encombrement : MISECep émet un événement *Alerte* portant sur le composant « D12345 » et qui transmet l'information suivant laquelle le risque est désormais inexistant. ModelsUpdate va collecter les événements filtrés/déduits et réémis par MISECep. A partir de l'événement *Activité*, MISECep va récupérer l'identifiant du service concerné ainsi que son état. De là, ModelsUpdate va rechercher dans la correspondance entre problèmes et solutions (qui provient de l'étape de réconciliation sémantique) les risques et conséquences liées à ce service, ainsi que tous les autres services liés à ces risques et conséquences identifiés (conséquence d'une réconciliation de type n-m entre solutions et problèmes) . Il ne trouve qu'un risque « Encombrement routier » lié à ce service, et aucun autre service lié à ce risque. L'état de *Etablir déviation* étant à *finish*, ModelsUpdate en déduit que le risque « Encombrement routier » est supprimé.

A l'issue d'une communication radio entre l'unité du SDIS 81 déployée sur le terrain et un opérateur du SDIS 81 (au poste de commandement), ce dernier clôture par erreur l'exécution du service *Eteindre incendie* (état valué à *finish*) au niveau de la supervision

des workflows. Ce changement d'état d'exécution du service génère un événement de type *Activité*, statuant la fin et la bonne exécution du service *Eteindre incendie*. MISECep reçoit donc simultanément :

- Cet événement de type *Activité*, que MISECep filtre vers le sujet *eventExpected* (car il concerne la mise à jour du modèle *attendu*),
- Une série d'événement de type *Mesure* comportant des relevés de températures est toujours supérieur à 100°C dans la zone nord de la centrale (les températures avoisinent désormais les 600°C), indiquant que l'incendie n'est pas éteint (d'après les règles métier appliquées par Esper). MISECep ne va donc pas envoyer d'événement pour informer de la disparition de la *Conséquence* d'identifiant « Incendie_Nord ».

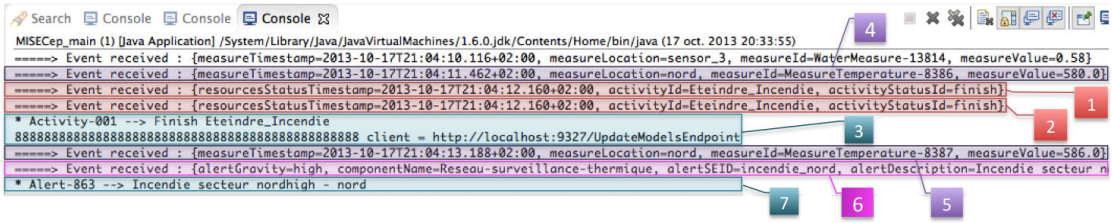


FIGURE VI.15 – Réception d'événements *Activité* émis par la supervision des workflows et déclenchant la règle de filtrage des *Activité*par MISECep

La Figure VI.15 présente les événements *Mesure* et *Activité* concernant l'incendie. Comme nous pouvons le voir dans les point **(1)** et point **(2)**, un événement *Activité* a été émis par la supervision des worklows (point **(1)**) puis filtré et émis sans modification par Esper (point **(2)**). Ces événement indique la fin d'exécution (sans erreur) du service *Eteindre_Incendie*. Simultanément, des événements de type *Mesure* sont consommés par Esper (point **((4)** et point **(5)**) qui génèrent à nouveau une *Alerte* incendie (point **(6)**. Les points **(3)** et **(7)** sont encore des traces.

ModelsUpdate ne va donc pas mettre le modèle *terrain* à jour. En revanche, en ce qui concerne le modèle *attendu*, ModelsUpdate va supprimer la *Conséquence* d'identifiant « Incendie_Nord ». ModelsUpdate va également mettre à jour le statut du service *Eteindre_Incendie* (valeur égale à « finish »). ModelsUpdate ne va pas apporter de modification au modèle *terrain* en ce qui concerne l'entité *Consequence* d'identifiant « Incendie_Nord ». Ces changements sont visibles dans la représentation graphique des modèles (Figure VI.16).

VI.4.2.2 Le calcul du delta

L'utilisateur déclenche à nouveau une détection de la divergence en cliquant sur le bouton *Détection*. Dans les deux modèles, le risque « Encombrement routier » a été supprimé : aucune différence n'est donc constatée à ce niveau là. Par contre, la suppression

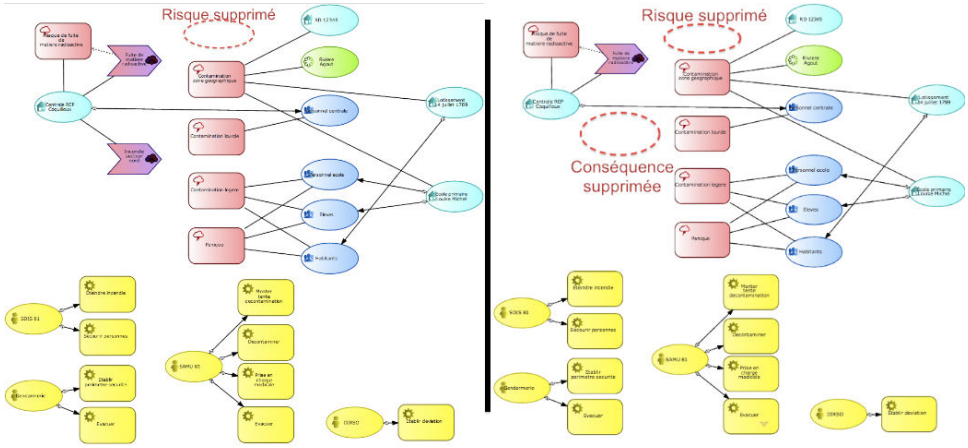


FIGURE VI.16 – Modèle terrain (à gauche) et modèle attendu (à droite) de la situation de crise à $t_0 + 40min$

de la conséquence *Incendie secteur nord* dans le modèle *attendu* et son maintien dans le modèle *terrain* entraîne la détection d'une divergence (cf. Figure VI.17).

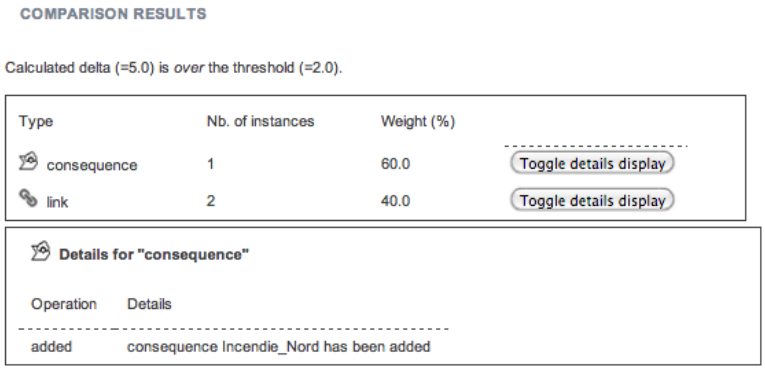


FIGURE VI.17 – Résultat de la détection à $t_0 + 40min$

VI.4.2.3 Le choix d'adaptation

Le δ calculé est égal à 5 et est donc supérieur au seuil. Nous pouvons constater que si le détail du δ est identique à celui obtenu précédemment, le choix d'adaptation proposé par le Service d'Adaptation est différent. En effet, il propose désormais (Figure VI.18) :

- Soit de caractériser à nouveau la situation, considérant que la *Conséquence* d'identifiant « Incendie_Nord » est apparue dans le modèle *terrain*,
- Soit —et là réside la nouveauté— de re-exécuter le service *Eteindre incendie* du SDIS 81 qui n'a pas été efficace pour prendre en charge la conséquence. Ce service n'est manifestement pas en panne (sinon l'état du service transmis par la supervision des processus n'aurait pas été égale à « finish » mais à « failed »).

L'utilisateur choisit d'exécuter à nouveau le service *Eteindre incendie* du SDIS 81. Une fenêtre s'ouvre et lui permet d'interagir directement sur l'exécution du processus

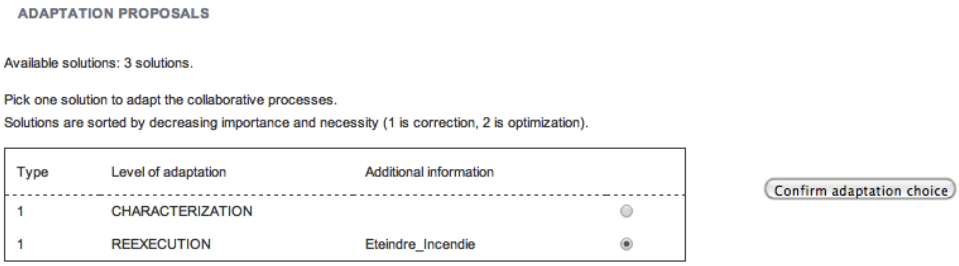


FIGURE VI.18 – Proposition d'adaptation à $t_0 + 40min$

en demandant à l'orchestrateur d'appeler à nouveau l'exécution du service *Eteindre incendie* du SDIS 81. Simultanément, l'état du service *Eteindre incendie* est réinitialisé à « waiting » dans le modèle *terrain*, afin de pouvoir refléter les changements d'état du service lors de cette re-exécution. Le modèle *terrain* sert ensuite à nouveau de référentiel pour l'étude de l'évolution des processus collaboratifs vis-à-vis de la dynamique du théâtre de la crise (le modèle *terrain* et le modèle *attendu* sont donc identiques au moment où l'on lance la re-exécution du service).

VI.4.3 À $t_0 + 70min$

VI.4.3.1 La mise à jour des modèles

Pendant la ré-exécution de l'activité *Eteindre incendie* du SDIS 81, un événement de type *OfferResource* est émis par SDIS 31 et reçu par MISECep. Il indique que le SDIS 31 vient se joindre à la cellule de crise. SDIS 31 ne dispose (dans le cas de cette collaboration) que d'un service *Eteindre feu*.

MISECep re-émet cet événement sur le sujet *eventField*, permettant à ModelsUpdate de reporter l'arrivée du partenaire SDIS 31 par l'ajout d'un *Partner* d'identifiant « SDIS31 » et de son *Service Eteindre feu* dans le modèle *terrain*.

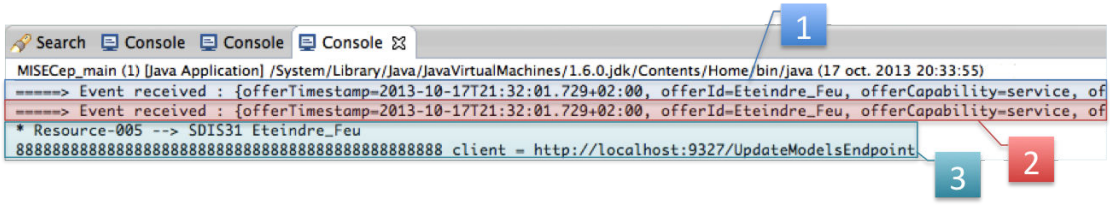


FIGURE VI.19 – Réception d'événements *OfferResource* émis et déclenchant la règle de filtrage des *OffreRessource*par MISECep

La Figure VI.19 présente l'événements *OfferResource* reçue par Esper. Comme nous pouvons le voir dans les point **(1)** et point **(2)**, un événement *OfferResource* a été émis par le terrain (point **(1)**) puis filtré et émis sans modification par Esper (point **(2)**). Ces événement indique l'ajout d'un partenaire et de son service dans la collaboration. Les

points point **(3)** représente les traces laissées par l'abonnement de *ModelsUpdate* sur ce sujet.

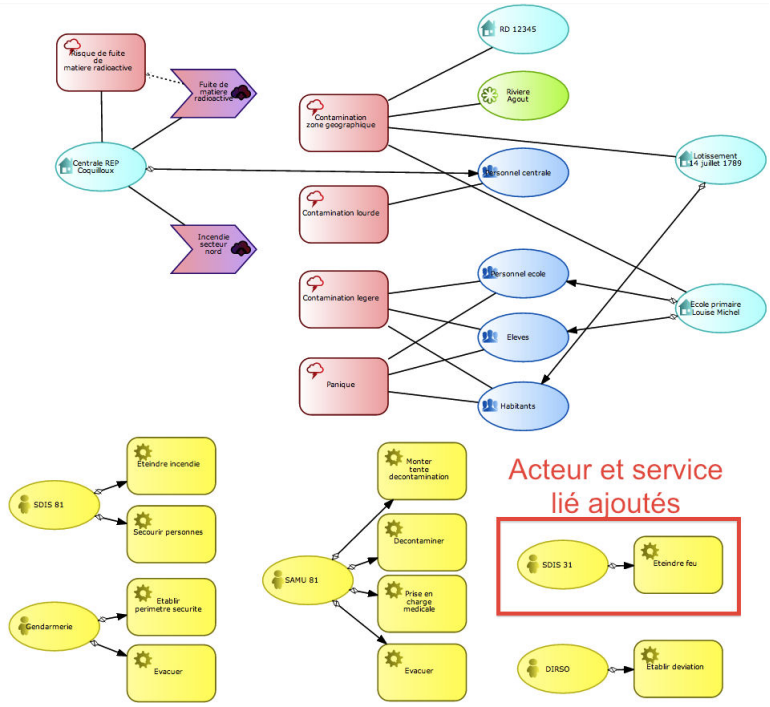


FIGURE VI.20 – Modèle terrain de la situation de crise à $t_0 + 70min$

Ces changements sont visibles dans l'interface graphique représentant le modèle *terrain* (Figure VI.20), où l'on constate bien l'ajout de l'acteur SDIS 31 et de son service.

VI.4.3.2 Le calcul du delta

La Figure VI.21 montre que l'ajout de ce partenaire à la collaboration se retranscrit par un δ calculé égal à 2 (i.e. au seuil fixé par la cellule de crise).

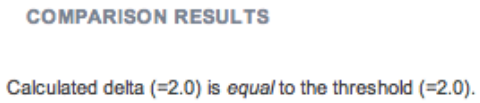


FIGURE VI.21 – Résultat de la détection à $t_0 + 70min$

VI.4.3.3 Le choix d'adaptation

Le Service d'Adaptation propose alors une adaptation du réseau (Figure VI.22), c'est-à-dire une redéfinition des liens entre les problèmes identifiés et les solutions apportées. En effet, le service fourni par le SDIS 31 peut éventuellement être plus couvrant ou pertinent pour traiter certains des risques et conséquences de la situation de crise que

ceux sélectionnés actuellement (et correspondant à l'offre disponible avant l'arrivée de ce nouveau partenaire).

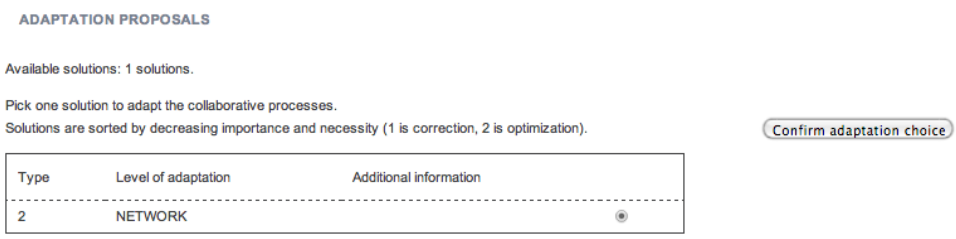


FIGURE VI.22 – Proposition d'adaptation à $t_0 + 70min$

Nous pouvons remarquer que cette proposition, contrairement aux précédentes, est typée *Optimisation*. La nuance [11] retenue par le Service d'Agilité signifie qu'elle n'est pas nécessaire à la poursuite des objectifs de la collaboration, contrairement aux adaptations typées *Correction*, qui permettent de corriger la trajectoire empruntée par la collaboration. Son seul but est d'optimiser le fonctionnement courant.

Si l'utilisateur choisit de conduire cette adaptation, alors le Web Service de réconciliation sémantique (service du Design Time du SIM) va s'exécuter afin de déterminer si le choix des services est à modifier ou non, compte tenu des caractéristiques du nouveau service *Eteindre feu* qui est mis à disposition de la collaboration. Dans le même temps, le modèle *terrain* sert à nouveau de référence (le modèle *attendu* devient identique au modèle *terrain*).
Si le choix des services est modifié, de nouveaux workflows sont générés pour intégrer ce changement et déployés sur le SIM.

Notons que chacun des niveaux d'adaptation proposé (caractérisation, réseau, re-exécution) peut être typé *Correction* ou *Optimisation*, suivant le contexte dans lequel se trouve la situation collaborative à l'instant où la détection est effectuée.

VI.5 Conclusion

Ce chapitre présente dans un premier temps l'architecture technique du prototype utilisant le Service d'Agilité, qui s'appuie sur les paradigmes EDA et SOA pour sa réalisation. Les deux composantes majeures du Service d'Agilité sont (i) la mise à jour des modèles dirigée par les événements (MISECep encapsulant le moteur de CEP Esper, et ModelsUpdate) et (ii) la détection et le choix d'adaptation (AgilityService.war et son interface utilisateur). Ce choix architectural permet de réutiliser la composant de détection/adaptation avec un autre service de mise à jour des modèles (utilisant un autre moteur de CEP par exemple). L'implémentation des mécanismes d'agilité est soumise à la licence GPL, due à la contamination de la licence GPL provenant d'Esper.

11. Telle que définie dans le Chapitre V, en Section V.4.2

D'autre part, le cas d'étude déroulé, bien que simplifié à l'extrême, nous a permis d'illustrer l'utilisation du Service d'Agilité et de rendre palpable l'intérêt d'un tel outil dans le suivi de la pertinence des processus collaboratifs au cours du temps . Nous avons pu voir des exemples de situation amenant à adapter la logique collaborative au niveau de sa caractérisation même, de la définition de ses processus et enfin de l'exécution des services de ses processus collaboratifs. Les cas d'étude développés dans les projets PLAY et SocEDA sont beaucoup plus fournis et complexes, tant dans la volumétrie d'événements à traiter, que dans les règles CEP mises en place.

Nous pouvons noter que plusieurs améliorations peuvent être apportées, notamment en ce qui concerne le stockage des données et l'accès aux données. La lecture de fichiers XML pour accéder aux données des modèles de situation n'est pas la solution la plus optimale, surtout en cas d'accès fréquents. Il pourrait être envisageable de mettre en place une base de données.

Enfin, en ce qui concerne les événements analysés par le CEP, nous nous sommes concentrés sur les flux d'événements émis en temps réel. Il pourrait être judicieux d'accéder aux événements dits historiques : ces événements se sont produits *avant* que le CEP ne se mette à « écouter » l'environnement [Etzion, 2010]. Ils sont conservés dans une base de données. Notons qu'une autre définition d'un événement temporel (retenue dans le projet PLAY) veut qu'il s'agisse d'un événement que le CEP a déjà pu analyser et qui est stocké en base de données (en attendant une autre utilisation par le CEP). Esper permet cet accès simultané à des événements conservés en base de données et aux flux d'événements émis en temps réel. Il faudrait donc améliorer le paramétrage de l'encapsulation d'Esper dans notre architecture pour qu'il s'abonne aux événements provenant de ces deux types de sources. De plus, ce type d'évolution nécessite le développement de nouvelles règles métier à injecter dans le moteur de CEP.

La réalisation de tests de performance à ce stade de l'implémentation du prototype n'est pas très pertinente. En effet, l'ensemble des serveurs et application web est hébergé en local sur la même machine. Les performances en terme de vitesse de transmission des données (événements, fichiers des modèles *terrain* et *attendu*, etc.) et de charge ne sont pas représentatives de la réalité. Des tests de performance du prototype sont en cours, en hébergeant les parties serveur et client sur des machines physiques différentes.

En revanche, d'après les tests de performance réalisés par l'éditeur [12], Esper (dans son édition Open Source gratuite) permet de traiter plus de 500 000 événements par secondes sur la base d'un processeur double coeur Intel cadençant à 2GHz. Les tests de performance ont été réalisés en effectuant un test de type VWAP (Volume-Weighted Average Price) avec 1000 règles enregistrées dans le système. La latence moyenne du moteur est en dessous des 3 microsecondes en moyenne. La CPU est utilisée à 85%. Si les

12. Esper, Performance-Related Information : `http://esper.codehaus.org/esper/performance/performance.html`.

données techniques concernant les performances en volumétrie d'Esper sont rassurantes, il ne s'agit que d'informations issues d'une plaquette à but commercial. Il faudrait donc également tester les performances du moteur de CEP avec nos propres tests de benchmark (i.e. utilisant nos types d'événements) afin d'avoir une base de travail réaliste.

Le Service d'Agilité permet de décharger l'humain des tâches fastidieuses comme le filtrage et l'agrégation de données en automatisant les règles métier et les mécanismes de pensée habituellement exécutés par l'utilisateur. Cet apport prend tout son sens dans les problématiques de type Big Data. Notre environnement devient de plus en plus numérisé et connecté. Il génère des milliards de données qui, analysées ensemble, peuvent créer de l'information. Dans un tel contexte, il apparaît que la volumétrie des données générées devient un obstacle pour exécuter un traitement manuel. Le Service d'Agilité devient alors le support de l'analyse de milliards de données et permet à l'humain d'avoir une première analyse des données reçues en extrayant l'information essentielle.

Conclusion et perspectives

Il est désormais admis que la collaboration entre organisations (entreprises, institutions, administrations, ou autres) est un enjeu majeur en vue de leur survie dans un environnement instable. De plus, l'interconnexion entre les organisations, les personnes et les appareils qu'elles contiennent, devient de plus en plus forte. La proposition dans laquelle s'inscrivent ces travaux est de faire en sorte que leur collaboration s'appuie sur l'interopérabilité de leurs Systèmes d'Information (SI), réalisée au travers d'un SI de médiation (support de la collaboration). Ce dernier prend en charge l'établissement des processus collaboratifs entre les partenaires de la collaboration et le déploiement des workflows correspondants. Or la nature instable de la collaboration et de l'environnement dans lequel elle évolue peut compromettre la pertinence de ces processus et la bonne exécution de ces workflows. La prise en compte de l'apparition de nouvelles contraintes au cours de l'exécution des workflows collaboratifs nous a amené à étudier l'agilité des processus collaboratifs. Deux étapes sont nécessaires pour rendre les processus collaboratifs (et les workflows qui en découlent) agiles : (i) la détection du moment où une évolution apparaît et compromet la poursuite des objectifs de la collaboration par les workflows déployés ; (ii) la déduction dans un laps de temps très court du comportement à adopter pour surmonter les évolutions détectés et atteindre, quoi qu'il arrive, les objectifs.

Après avoir défini ce qu'était l'agilité des processus collaboratifs, ces travaux de thèse proposent la démarche suivante afin de mettre en œuvre cette agilité : (i) mise en place d'une sur-couche architecturale dirigée par les événements, afin de prendre en compte les événements émis par la situation collaborative (environnement) et la collaboration (processus collaboratifs, partenaires) et caractériser la situation collaborative à travers ces deux points de vue (modèles *terrain* et *attendu*), (ii) création d'un algorithme de mesure de distances entre les modèles, et enfin (iii) conception d'outil d'aide à la décision et à la re-conception complète/partielle des processus collaboratifs ou à la ré-exécution d'une partie des workflows collaboratifs sur la base de l'analyse de la divergence entre la situation attendue et la situation réelle.

L'étude bibliographique réalisée nous a amenés à définir la notion d'agilité en nous appuyant sur différents domaines tels que la gestion de chaines d'approvisionnement, la gestion de systèmes d'informations, etc. L'agilité des processus collaboratifs peut donc être vue comme la capacité d'un système à réaliser, dans le laps de temps le plus court possible (réactivité), la détection de son inadéquation à l'environnement dans lequel il

évolue et la mise en oeuvre de l'adaptation nécessaire, en vue d'atteindre les objectifs de la collaboration efficacité. Les contraintes dues au contexte particulier qu'est la réponse à la situation de crise nous ont tout naturellement amenées à positionner l'adaptation des processus collaboratifs par rapport à une approche de conception ad-hoc [Schonenberg et al., 2008]. Dans le second volet de cette étude, nous avons vu que la connaissance du contexte de la collaboration est essentielle pour suivre les évolutions du théâtre de la collaboration et de la collaboration elle-même. Il est crucial de pouvoir collecter toutes les données pertinentes qui peuvent traduire un changement dans la caractérisation de la situation collaborative. La mise en place d'une architecture dirigée par les événements (EDA) répond à ce besoin en fournissant également les moyens d'analyser et interpréter les données recueillies compte tenu de leur volumétrie.

Par la suite, nous nous sommes penchés sur la question de la mise à jour de la caractérisation de la situation du point de vue du terrain et du point de vue de la supervision des workflows. La sensibilité au contexte est l'élément clé sur lequel repose l'étape de détection. Pour cela, nous nous sommes appuyés sur le méta-modèle de la situation de crise afin de pouvoir caractériser la situation de crise à t_0. Puis, à partir de ce modèle initial, nous avons dérivé deux modèles : le modèle *terrain* qui photographie la situation collaborative réelle (via les données provenant du terrain), et le modèle attendu qui esquisse le portrait-robot de la situation collaborative telle qu'elle est perçue à travers les états des activités exécutées par les workflows. Pour réaliser la collecte des données, puis leur analyse, nous avons mis en place une architecture EDA. Dans cette dernière, nous avons intégré un moteur de traitement des événements complexes (CEP) qui permet de déduire de nouvelles informations par filtrage, dérivation ou corrélation des événements reçus du terrain et de la supervision des workflows. Chacun des événements complexes générés par le CEP va être utilisé par un outil de mise à jour des modèles *attendu* et *terrain*.

Nous avons ensuite étudié la façon dont nous pouvions effectuer la détection de la distance entre les deux modèles *attendu* et *terrain*. Celle-ci est réalisée en comparant chacun des nœuds des fichiers XML des modèles entre eux. Le calcul de la distance globale se fait en étudiant chaque distance δ_i relevée entre les deux modèles. Pour chaque différence, le coût w_i de l'opération associée (insertion, suppression ou modification) et l'importance m_i de l'instance concernée par la différence permet de calculer la distance entre les deux modèles pour cette différence donnée. La somme de ces distances permet d'obtenir une distance globale δ entre les deux modèles. Plus δ est élevé, plus les deux représentations de la situation collaborative divergent. Et plus ces modèles divergent, moins les processus collaboratifs (et les workflows en dérivant) sont en phase avec la réalité de la situation collaborative à un instant t. La nécessité de déclencher une action d'adaptation des processus collaboratifs est prise si le δ calculé dépasse un seuil δ_{seuil} défini en amont par les partenaires de la collaboration. Le δ devient un indicateur de performance de la collaboration, du point de vue des effets des processus collaboratifs.

Après avoir détaillé l'étape de détection, nous avons proposé un mécanisme d'aide à

la décision concernant l'adaptation des processus collaboratifs. Cette adaptation, si elle est nécessaire, est conduite à l'un des trois niveaux de conception du Système d'Information de Médiation. L'algorithme défini analyse les composantes δ_i du δ calculé. Nous avons ainsi listé des profils types de combinaisons entre les différences que l'étape de détection avait révélé : par exemple, l'ajout d'un *bien* (ou d'une *personne*, d'une *société civile* ou de l'*environnement*) associé à l'ajout d'un *risque* (ou d'une *conséquence*, d'un *facteur de complexité ou aggravant*) nécessite une adaptation corrective de type *caractérisation de la situation collaboratif*. Cette façon de procéder a drastiquement réduit le nombre de combinaisons possibles entre les différents cas de divergence pouvant se produire. Une liste de solutions d'adaptation (triée par ordre de priorité décroissant) est proposé à l'utilisateur qui en choisit une seule. La limite de ce mécanisme est sa forte dépendance au domaine : si le calcul de la différence entre les modèles peut s'appliquer au domaine de la crise nucléaire mais aussi de la crise en générale ou d'une collaboration B2B, ce n'est pas le cas des mécanismes d'aide à la décision. Ceux-ci se basent sur une analyse des composantes δ_i du δ calculé et donc sur les concepts (partenaire, risque, conséquence, biens, environnement, etc. dans le cas de la crise) développés dans la couche du méta-modèle de situation collaborative retenue. Les mécanismes de choix d'adaptation tels que décrits ici s'appliquent donc uniquement au cas de la crise. L'application de l'analyse des différences identifiées à un autre domaine métier (couvert par une autre couche du méta-modèle) demande à développer la logique de choix d'adaptation sur les concepts propres à cette couche du méta-modèle.

Les mécanismes du Service d'Agilité présentés dans ces travaux de thèse ont été implémentés dans un *prototype* constitué de plusieurs applications Java que nous avons créées *ModelsUpdate*, *XMLCompare* et *AdaptationDeductionCrisis*. Ces applications (distribuées sous forme de librairie) prennent respectivement en charge la mise à jour des modèles, la détection d'une divergence entre les modèles, et l'analyse de la divergence afin de déduire les choix d'adaptation possibles des workflows collaboratifs. Nous avons également utilisé un *moteur de CEP* (Esper) et un *bus de services* (Petals ESB) afin de supporter l'*architecture ED-SOA* que nous avons définie pour la prise en charge des événements (abonnement, émission, traitement) et sur laquelle s'appuient nos applications. Un *serveur d'applications Web* Tomcat a permis le déploiement de l'*interface utilisateur* du Service d'Agilité. La démarche a été illustrée par un cas d'étude fictif inspiré des cas d'études créés et implémentés dans les projets de recherche EUR FP7 PLAY et ANR SocEDA. Ce cas d'étude traite de la réponse à un incendie survenu de façon imprévue au cours de la réponse à une crise nucléaire (fuite dans une centrale).

Pour le moment, les étapes d'abonnement aux événements et d'envoi/réception des événements par l'orchestrateur (en charge de l'exécution des workflows collaboratifs) ne sont pas automatisées : ces étapes doivent être réalisées manuellement par l'utilisateur, qui doit choisir les sujets auxquels il souhaite que le système s'abonne. Puis il doit configurer manuellement ces étapes d'abonnement. D'autre part, l'orchestrateur utilisé ne sait pas nativement gérer l'envoi et la réception d'événements. Ces étapes ne sont pas

intégrées dans le processus de déduction et de déploiement des processus collaboratifs (partie Design Time du projet MISE). Nous avons donc dû ajouter les abonnements et les capacités d'envoi et de réception des événements manuellement au niveau des workflows. Cela peut poser problème au moment de l'adaptation des processus, suivant le niveau choisi (redéfinition partielle ou totale des processus) : tout le travail effectué sur les workflows pour ajouter la capacité « événementielle » est perdu. Il faut à nouveau effectuer les étapes de souscription pour chaque activité concernée. La première perspective d'évolution de ces travaux est donc la prise en compte de la dimension d'architecture dirigée par les événements dans la conception du SIM.

Une autre limite de nos travaux réside dans la nature même des composants du système. Toute notre infrastructure est basée sur les Systèmes d'Information des organisations et est donc dépendant du réseau (au sens matériel du terme). Si tout ou partie du réseau tombe en panne, l'ensemble du SIM échoue également. Des mesures de sécurité matérielles doivent être prises avant de protéger le réseau physique, en plus des mesures de sécurité logicielles pour protéger le réseau de données contre les actes de piratage ou l'utilisation de données non fiables. Au sujet de la sécurité, un autre point peut être soulevé : celui de l'accès aux événements. Comment s'assurer qu'un événement, qu'une source d'événement ou les types d'événements auxquels le système s'abonne soient dignes de confiance ? Dans le projet ANR SocEDA, un filtre social a été mis en place. S'appuyant sur un algorithme de type FOAF (Friend Of A Friend, pour Ami d'un Ami), il permet de calculer le degré de confiance accordé dans des événements, suivant le degré de confiance qu'entretiennent les événements entre eux. Et de même qu'il existe des outils de gouvernance pour les services dans les infrastructures SOA, il faut envisager la création d'outils de gouvernance pour les événements dans les infrastructures EDA.

Nous avons par ailleurs défini les *valeurs des matrices de coût et d'importance* utilisées dans le calcul de la divergence entre les modèles de façon assez empirique. Même si l'obtention rigoureuse de ces valeurs n'était pas un des enjeux de ces travaux de thèse, il ne faut pas le passer sous silence. Il semblerait opportun de définir une méthode de détermination de ces facteurs. L'utilisation d'une méthode d'analyse multi-critères de type AHP (Analytic Hierarchy Process) [Saaty, 1988] peut constituer une première piste d'étude pour formaliser le processus d'obtention des valeurs des *coûts* et des *importances*.

Sur la base des caractéristiques du Service d'Agilité, nous pouvons facilement imaginer d'autres usages à cet outil. Par exemple si l'on considère une zone prédéfinie (dans un contexte de gestion de crise [Barthe-Delanoë et al., 2012a] ou de logistique urbaine [Barthe-Delanoë et al., 2013b], il s'agit d'une zone géographique, mais cela peut représenter un domaine d'activité dans un contexte industriel), un tel service peut recueillir les événements provenants de cette zone (à partir de capteurs ou de toute personne ou composant connecté). Ces événements pourraient être utiliser pour créer et mettre à jour une image vivante de la zone observée (i.e. un modèle du système considéré). Les évolutions pertinentes de ce modèle pourraient être exploitées à des fins de

diagnostic si un changement soudain survient et servir de modèle de départ (modèle de la situation à t_0) pour déduire les processus collaboratifs. Cette fonction de surveillance pourrait permettre la détection de la survenue de la crise et le début de la réponse à la crise fondée sur de la connaissance mise à jour et automatiquement recueillie. On peut également imaginer un réseau virtuel de capteurs dont l'ampleur et les données surveillées évolue en fonction des événements reçus et exploités par le Service d'Agilité [Barthe-Delanoë et al., 2013a]. Ces travaux peuvent être utilisés afin de caractériser en continu une situation collaborative, dans le but de la superviser. Il serait ainsi envisageable de ne plus avoir à créer le modèle initial de la situation de crise par exemple : ce dernier serait alors créé automatiquement par les événements.

Ces travaux s'inscrivent dans la version 2 du projet MISE (Mediation Information System Engineering) mené dans l'axe Interopérabilité des Organisations du Centre Génie Industriel de Mines Albi. Le but de ce projet est de proposer une démarche complète de conception (du recueil de l'information caractérisant la collaboration [13] jusqu'à la phase d'exécution) du Système d'Information de Médiation (SIM) qui est le support dédié à cette collaboration. Ce SIM doit présenter des qualités d'agilité afin de pouvoir intégrer les changements du contexte dans lequel il évolue. Plusieurs travaux de thèse (sept passées, quatre en cours et une qui va débuter) ont alimenté la constructions des différentes itérations du projet MISE (voit Figure VI.23). Cette thèse porte sur les aspects du déploiement (mise en place de l'EDA), de la détection des évolutions et de l'agilité (via la détection et l'adaptation supportées par les infrastructures EDA et SOA).

Les travaux de thèse présentés ici ainsi que ceux de [Zribi et al., 2012] [14] concluent la deuxième itération du projet MISE. La troisième itération de MISE est supportée par les travaux suivants :

- Guillaume Macé-Ramète, qui s'intéresse à la performance de la dynamique collaborative et à l'introduction d'indicateurs de performances (KPI, pour Key Performance Indicators) dans la phase d'agilité du SIM [Macé-Ramète et al., 2012]. Il a de plus redéfini le méta-modèle de la collaboration afin de le rendre sensible au domaine métier considéré. Son cas d'étude porte sur la crise routière dans la région Grand Ouest de la France et ses travaux de thèse s'appuient sur le projet SIM-PeTra,
- Aurélie Montarnal, qui se penche vers la caractérisation continue de la dynamique collaborative dans le contexte d'une utilisation sur une plateforme ouverte et hautement distribuée [Montarnal et al., 2013]. Elle met en place une couche sociale afin de faire émerger des opportunités de collaboration entre des partenaires qui ne se connaissent pas. Ses travaux de thèse sont appuyés par le projet OpenPAAS,
- Loïc Bidoux, qui s'intéresse lui aussi à la déduction de la dynamique collaborative en y intégrant la prise en compte des ressources disponibles (hommes, véhicules, matériel, etc.). Pour cela, il se place dans le cadre de la gestion de crise, domaine

13. Partenaires, compétences desdits partenaires, objectifs poursuivis, etc.

14. Ses travaux de thèse s'intéressent à la réconciliation de services mais du point de vue des aspects non-fonctionnels, contrairement à [Boissel-Dallier, 2012] qui s'intéresse aux aspects fonctionnels.

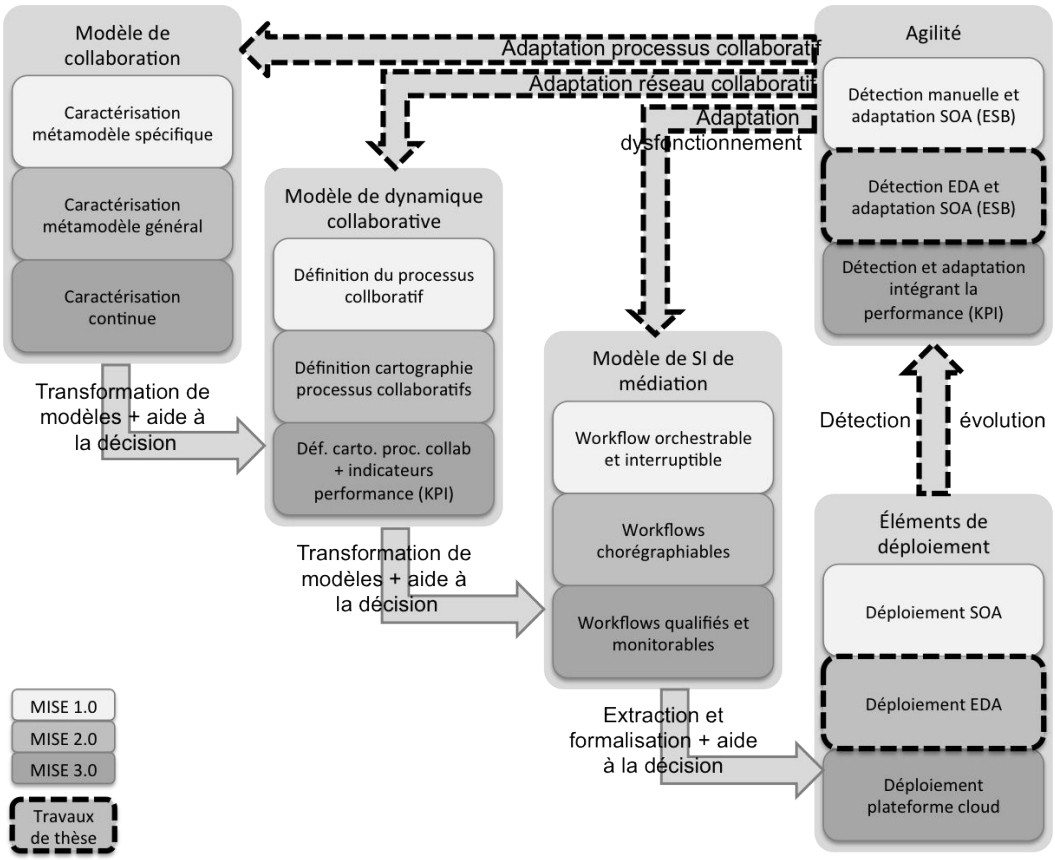


FIGURE VI.23 – Positionnement de la thèse par rapport aux travaux menés dans les trois itérations du projet MISE

où la gestion des ressources est capitale,

- Tiexin Wang, qui étudie l'amélioration conceptuelle et technique des différentes transformations de modèles mises en jeu dans MISE. Il vise ainsi la conception d'un atelier de définition automatique de transformation de modèles, ce qui permettra un développement rapide du SIM quelque soit le domaine métier ciblé,
- Alexandre Sirko, qui va s'intéresser aux notions de Big Data et Open-Data, en vue d'assurer une collecte « intelligente » de données afin de les transformer en information et connaissance exploitables. Les aspects de sécurité et de confiance dans les données vont faire partie intégrante de ses travaux.

Ces travaux de thèse s'inscrivent dans les problématiques émergentes de Big Data et fournissent une plateforme permettant d'exploiter cet espace numérique naissant.

Bibliographie

[Adams et al., 2006] Adams, M., ter Hofstede, A. H., Edmond, D., and van der Aalst, W. M. P. (2006). Worklets : A service-oriented implementation of dynamic flexibility in workflows. In Meersman, R. and Tari, Z., editors, *On the Move to Meaningful Internet Systems 2006 : CoopIS, DOA, GADA, and ODBASE*, number 4275 in Lecture Notes in Computer Science, pages 291–308. Springer Berlin Heidelberg. `http://link.springer.com/chapter/10.1007/11914853_18`.

[AFP, 2011] AFP (2011). L'accident de fukushima classé au même niveau que tchernobyl. *Le Monde (Asie-Pacifique)*. `http://www.lemonde.fr/japon/article/2011/04/12/tokyo-eleve-au-niveau-7-l-accident-nucleaire-dans-la-centrale-de-fukushima_1506185_1492975.html#ens_id=1493262`.

[Alberganti, 2011] Alberganti, M. (2011). Fukushima : La vague qui a frappé la centrale atteignait 15 mètres. `http://tinyurl.com/franceculture-fukushima-15me`.

[Alexander, 2002] Alexander, D. E. (2002). *Principles of Emergency Planning and Management*. Oxford University Press.

[Altay and Green III, 2006] Altay, N. and Green III, W. G. (2006). OR/MS research in disaster operations management. *European Journal of Operational Research*, 175(1) :475–493. `http://www.sciencedirect.com/science/article/B6VCT-4GWJ8D1-7/2/236a22a3d25cfc64a455b16ffab8d001`.

[Andonoff et al., 2007] Andonoff, E., Bouaziz, W., and Hanachi, C. (2007). Protocol management systems as a middleware for inter-organizational workflow coordination. *International Journal of Computer Science and Applications*, 4(2) :23–41.

[Andonoff et al., 2009] Andonoff, E., Bouaziz, W., and Hanachi, C. (2009). An agent-based model for autonomic coordination of inter-organizational business processes. *Informatica*, 20(3) :323–342. `http://iospress.metapress.com/index/CRR2151760727141.pdf`.

[Atzori et al., 2010] Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things : A survey. *Computer Networks*, 54(15) :2787–2805. `http://www.sciencedirect.com/science/article/pii/S1389128610001568`.

[Bacon and Martin, 2013] Bacon, T. and Martin, J. (2013). XMLUnit. `http://xmlunit.sourceforge.net/`.

[Badot, 1998] Badot, O. (1998). *Théorie de l'entreprise agile*. Editions L'Harmattan.

[Barthe et al., 2011a] Barthe, A.-M., Bénaben, F., Truptil, S., Lorré, J.-P., and Pingaud, H. (2011a). MDI for SOA management of a crisis. In Zelm, M., van Sinderen, M., Doumeingts, G., and Johnson, P., editors, *Enterprise Interoperability : IWEI 2011 Proceedings*, pages 123–132, Stockholm, Sweden. Wiley-ISTE. `http://eu.wiley.com/WileyCDA/WileyTitle/productCd-1848213174.html`.

[Barthe et al., 2011b] Barthe, A.-M., Lauras, M., and Benaben, F. (2011b). Improving the design of interoperable platform through a structured case study description approach : Application to a nuclear crisis. In *Proceedings of the 8th International ISCRAM Conference*, Lisbon, Portugal. `http://www.iscramlive.org/ISCRAM2011/proceedings/papers/146.pdf`.

[Barthe-Delanoë et al., 2012a] Barthe-Delanoë, A.-M., Bénaben, F., Carbonnel, S., and Pingaud, H. (2012a). Event-driven agility of crisis management collaborative processes. In Rothkrantz, L., Ristvej, J., and Franco, Z., editors, *Proceedings of the 9th International ISCRAM Conference*, Vancouver B.C., Canada. `http://www.iscramlive.org/ISCRAM2012/proceedings/124.pdf`.

[Barthe-Delanoë et al., 2013a] Barthe-Delanoë, A.-M., Bénaben, F., Truptil, S., and Pingaud, H. (2013a). A flexible network of sensors : case study. In Comes, T., Fiedrich, F., Fortier, S., Geldermann, J., and Yang, L., editors, *Proceedings of the 10th International ISCRAM Conference*, Baden-Baden, Germany.

[Barthe-Delanoë et al., 2013b] Barthe-Delanoë, A.-M., Lauras, M., Truptil, S., Bénaben, F., and Pingaud, H. (2013b). A platform for event-driven agility of processes : A delivery context use-case. In Camarinha-Matos, L. M. and Scherer, R. J., editors, *IFIP AICT 408*, pages 681—690, Dresden, Germany. Springer.

[Barthe-Delanoë et al., 2012b] Barthe-Delanoë, A.-M., Truptil, S., Stühmer, R., and Benaben, F. (2012b). Definition of a nuclear crisis use-case management to s(t)imulate an event management platform. In Stojanovic, N., Stojanovic, L., and Etzion, O., editors, *Proceedings of the 7th International Workshop on Semantic Business Process Management (SBPM 2012)*, volume 862, Heraklion (Greece). `http://ceur-ws.org/Vol-862/SBPMp2.pdf`.

[Baumont, 2011] Baumont, D. (2011). Séisme majeur au large de l'île d'Honshu (japon). Technical report, Institut de Radioprotection et de Sûreté Nucléaire. `http://www.irsn.fr/fr/actualites_presse/actualites/documents/irsn_seisme-japon_v1-11032011.pdf`.

[Beamon and Kotleba, 2006] Beamon, B. and Kotleba, S. (2006). Inventory modelling for complex emergencies in humanitarian relief operations. *International Journal of Logistics : Research and Applications*, 9(1) :1–18.

[Beck and Gamma, 2013] Beck, K. and Gamma, E. (2013). JUnit. `https://github.com/junit-team/junit`.

[Bénaben, 2012] Bénaben, F. (2012). *Conception de Système d'Information de Médiation pour la prise en charge de l'Interopérabilité dans les Collaborations d'Organisa-*

tions. Habilitation à diriger les recherches, Mines Albi - Institut National Polytechnique de Toulouse, Albi.

[Bénaben et al., 2007] Bénaben, F., Touzi, J., Rajsiri, V., Lorré, J.-P., and Pingaud, H. (2007). L'Interopérabilité des systèmes d'information comme moyen vers l'intégration de l'écosystème industriel. In *7e Congrès international de génie industriel*, Trois-Rivières, Québec (Canada).

[Bénaben et al., 2008] Bénaben, F., Touzi, J., Rajsiri, V., Truptil, S., Lorré, J.-p., and Pingaud, H. (2008). Mediation information system design in a collaborative SOA context through a MDD approach. In *Proceedings of MDISIS 2008*, Montpellier, France. `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.143.1355`.

[Berglund, 2006] Berglund, A. (2006). XSL transformations (XSLT) version 1.1. `http://www.w3.org/TR/xslt`.

[Boissel-Dallier, 2012] Boissel-Dallier, N. (2012). *Réconciliation sémantique des données et des services mis en œuvre au sein d'une situation collaborative*. PhD thesis, Mines Albi - Institut National Polytechnique de Toulouse. `http://ethesis.inp-toulouse.fr/archive/00002270/`.

[Bonitasoft, 2013] Bonitasoft (2013). Bonita BPM. `http://www.bonitasoft.com/products`.

[Borghoff et al., 1997] Borghoff, U. M., Bottoni, P., Mussio, P., and Pareschi, R. (1997). Reflective agents for adaptive workflows. In *Proc. 2nd Int. Conf. on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'97)*, page 405–420. `http://www.xrce.xerox.com/content/download/20598/147785/file/reflective.pdf`.

[Bourey et al., 2007] Bourey, J.-P., Grangel, R., Ducq, Y., Berre, A., Bertoni, M., D'Antonio, F., Daclin, N., Doumeingts, G., Grandin-Dubost, M., and Kalampoukas, K. (2007). Report on model driven interoperability.

[Casati et al., 1998] Casati, F., Ceri, S., Pernici, B., and Pozzi, G. (1998). Workflow evolution. *Data & Knowledge Engineering*, 24(3) :211–238. `http://www.sciencedirect.com/science/article/pii/S0169023X97000335`.

[Catarci et al., 2011] Catarci, T., de Leoni, M., Marrella, A., Mecella, M., Russo, A., Steinmann, R., and Bortenschlager, M. (2011). WORKPAD : process management and geo-collaboration help disaster response. *IJISCRAM*, 1(3) :32–49. `http://www.tue.nl/en/publication/ep/p/d/ep-uid/260732/`.

[Chandy and Schulte, 2009] Chandy, K. M. and Schulte, W. R. (2009). *Event Processing : Designing It Systems for Agile Companies*. McGraw-Hill Prof Med/Tech.

[Charles, 2010] Charles, A. (2010). *Improving the design and management of agile supply chains : feedback and application in the context of humanitarian aid*. INPT. `http://www.theses.fr/2010INPT0055`.

[Chinnici et al., 2007] Chinnici, R., Moreau, J.-J., Ryman, A., and Weerawarana, S. (2007). Web services description language (WSDL) 2.0. `http://www.w3.org/TR/wsdl20/`.

[Christensen et al., 2001] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S. (2001). Web services description language (WSDL) 1.1. `http://www.w3.org/TR/wsdl`.

[Christopher, 2010] Christopher, M. (2010). *Logistics and Supply Chain Management*. Pearson UK, 4th edition edition.

[Christopher and Towill, 2000] Christopher, M. and Towill, D. R. (2000). Supply chain migration from lean and functional to agile and customised. *Supply Chain Management : An International Journal*, 5(4) :206–213. `http://www.emeraldinsight.com/journals.htm?articleid=858210&show=abstract`.

[Clark, 1999] Clark, J. (1999). XML path language (XPath) version 1.0. `http://www.w3.org/TR/xslt`.

[CNRS, 2013] CNRS (2013). Définition du mot agile. `http://www.cnrtl.fr/definition/agile`.

[Cobéna et al., 2000] Cobéna, G., Abdessalem, T., and Hinnach, Y. (2000). A comparative study of XML diff tools. *Delta*, 10(100) :1000. `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.5366&rep=rep1&type=pdf`.

[Cobéna et al., 2004] Cobéna, G., Abiteboul, S., and Marian, A. (2004). XyDiff. `http://leo.saclay.inria.fr/software/XyDiff/cdrom/www/xydiff/index-eng.htm`.

[Council, 2010] Council, S. C. (2010). Supply chain operations reference (SCOR®) model - overview - version 10.0. `http://supply-chain.org/f/SCOR-Overview-Web.pdf`.

[Dadam et al., 2010] Dadam, P., Reichert, M., Rinderle-Ma, S., Lanz, A., Pryss, R., Predeschly, M., Kolb, J., Ly, L. T., Jurisch, M., and Kreher, U. (2010). From ADEPT to AristaFlow BPM suite : a research vision has become reality. In *Business process management workshops*, page 529–531. `http://link.springer.com/chapter/10.1007/978-3-642-12186-9_50`.

[Dellen et al., 1997] Dellen, B., Maurer, F., and Pews, G. (1997). Knowledge-based techniques to increase the flexibility of workflow management. *Data & Knowledge Engineering*, 23(3) :269–295. `http://www.sciencedirect.com/science/article/pii/S0169023X97000153`.

[DeltaXML, 2013] DeltaXML (2013). DeltaXML core. `http://www.deltaxml.com/products/core/`.

[Devlin, 2006] Devlin, E. S. (2006). *Crisis Management Planning and Execution*. CRC Press.

[Dey, 2001] Dey, A. K. (2001). Understanding and using context. *Personal and ubiquitous computing*, 5(1) :4–7. `http://dl.acm.org/citation.cfm?id=593572`.

[Dove, 2002] Dove, R. (2002). *Response Ability : The Language, Structure, and Culture of the Agile Enterprise*. John Wiley & Sons.

[EPTS, 2008] EPTS (2008). Event processing technical society. `http://www.ep-ts.com`.

[Etzion, 2007] Etzion, O. (2007). More on EDA is EDA and SOA is SOA. `http://epthinking.blogspot.fr/2007/09/more-on-eda-is-eda-and-soa-is-soa.html`.

[Etzion, 2008] Etzion, O. (2008). On the right event processing language. `epthinking.blogspot.com/2008/06/on-right-event-processing-language.html`.

[Etzion, 2010] Etzion, O. (2010). On events versus data. `http://epthinking.blogspot.fr/2010/03/on-events-versus-data.html`.

[Etzion and Niblett, 2010] Etzion, O. and Niblett, P. (2010). *Event Processing in Action*. Manning Publications Co., Greenwich, CT, USA, 1st edition.

[Faure et al., 2009] Faure, C., Andonoff, E., Hanachi, C., Sibertin-Blanc, C., and Salatge, N. (2009). Flexibilité de processus de gestion de crise par intégration de protocoles d'interaction. In *INFORSID*, page 77–92. `https://liris.cnrs.fr/inforsid/sites/default/files/77_inforsid2009.pdf`.

[Faustmann, 1999] Faustmann, G. (1999). Enforcement vs. freedom of action an integrated approach to flexible workflow enactment. *SIGGROUP Bull.*, 20(3) :5–6. `http://doi.acm.org/10.1145/605610.605613`.

[Fülöp et al., 2012] Fülöp, L. J., Beszédes, Á., Tóth, G., Demeter, H., Vidács, L., and Farkas, L. (2012). Predictive complex event processing : a conceptual framework for combining complex event processing and predictive analytics. In *Proceedings of the Fifth Balkan Conference in Informatics*, BCI '12, page 26–31, New York, NY, USA. ACM. `http://doi.acm.org/10.1145/2371316.2371323`.

[Fülöp et al., 2010] Fülöp, L. J., Tóth, G., Rácz, R., Pánczél, J., Gergely, T., Beszédes, Á., and Farkas, L. (2010). Survey on complex event processing and predictive analytics. In *Proceedings of the Fifth Balkan Conference in Informatics*, BCI '12, page 26–31. `http://doi.acm.org/10.1145/2371316.2371323`.

[Gedik et al., 2008] Gedik, B., Andrade, H., Wu, K.-L., Yu, P. S., and Doo, M. (2008). SPADE : the system s declarative stream processing engine. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, page 1123–1134, New York, NY, USA. ACM. `http://doi.acm.org/10.1145/1376616.1376729`.

[Glover, 2006] Glover, A. (2006). In pursuit of code quality : Discover XMLUnit. `http://www.ibm.com/developerworks/java/library/j-cq121906/index.html`.

[Gudgin et al., 2007] Gudgin, M., Hadley, M., Mendehlsohn, N., Moreau, J.-J., Nielsen, H. F., Karmarkar, A., and Lafon, Y. (2007). *Simple Object Access Protocol (SOAP) 1.2*. World Wide Web Consortium (W3C).

[Guenther et al., 2008] Guenther, C. W., Reichert, M., and van der Aalst, W. M. (2008). Supporting flexible processes with adaptive work ? ow and case handling. In *Workshop on Enabling Technologies : Infrastructure for Collaborative Enterprises, 2008. WETICE'08. IEEE 17th*, page 229–234. `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4806924`.

[Halliday et al., 2001] Halliday, J. J., Shrivastava, S. K., and Wheater, S. M. (2001). Flexible workflow management in the OPENflow system. In *Proceedings of the 5th IEEE International Conference on Enterprise Distributed Object Computing*, EDOC '01, page 82–, Washington, DC, USA. IEEE Computer Society. `http://dl.acm.org/citation.cfm?id=645344.650362`.

[Haque, 2004] Haque, U. (2004). Pachube project. `http://www.haque.co.uk/pachube.php`.

[Highsmith, 2002] Highsmith, J. (2002). *Agile Software Development Ecosystems*. Addison-Wesley Professional. Part of the Agile Software Development Series series., 1st edition. `http://www.informit.com/store/agile-software-development-ecosystems-9780201760439?w_ptgrevartcl=Agility+in+Software+Development_25980`.

[Hollingsworth, 1995] Hollingsworth, D. (1995). The workflow management coalition specification : The workflow reference model. `http://www.wfmc.org/standards/docs/tc003v11.pdf`.

[Hopkin, 2010] Hopkin, P. (2010). *Fundamentals of risk management : understanding, evaluating and implementing effective risk management*. KoganPage [u.a.], London [u.a.

[Hunt and McIllroy, 1976] Hunt, J. W. and McIllroy, M. D. (1976). An algorithm for differential file comparison. *Computing Science Technical Report*, 1(41) :9.

[IRSN, 2013] IRSN (2013). L'échelle INES : incidents et accidents. `http://www.irsn.fr/FR/connaissances/Installations_nucleaires/La_surete_Nucleaire/echelle-ines/Pages/2-Incidents-accidents.aspx?dId=8a15297f-e5f9-42cd-9765-ed2049203773&dwId=a1de7c68-6d78-4537-9e6a-e2faebed3900`.

[ISO, 2000] ISO (2000). Norme européenne NF EN ISO 9001 version 2000, système de management de la qualité - exigences. `http://www.boutique.afnor.org/norme/nf-en-iso-9001/systemes-de-management-de-la-qualite-exigences/article/702508/fa145966`.

[ISO, 2005a] ISO (2005a). NF EN ISO 9000 systèmes de management de la qualité - principes essentiels et vocabulaire.

[ISO, 2005b] ISO (2005b). NF EN ISO 9000 x50-130 systèmes de management de la qualité — principes essentiels et vocabulaire.

[ISO, 2009] ISO (2009). International standard ISO 31000 :2009, risk management, principles and guidelines. `http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43170`.

[Josuttis, 2007] Josuttis, N. (2007). *SOA in Practice : The Art of Distributed System Design (Theory in Practice)*. O'Reilly Media.

[Kidd, 1994] Kidd, P. T. (1994). *Agile Manufacturing : Forging New Frontiers*. Addison-Wesley, addison-wesley edition.

[Kokkai, 2012] Kokkai (2012). The official report of the fukushima nuclear accident independent investigation commission. Technical report, Kokkai - The National Diet of Japon. http://www.nirs.org/fukushima/naiic_report.pdf.

[Konstantas et al., 2006] Konstantas, D., Bourrieres, J.-P., Leonard, M., and Boudjlida, N. (2006). *Interoperability of Enterprise Software and Applications*. Engineering. Springer London, Geneva. http://www.springer.com/computer/swe/book/978-1-84628-151-8.

[Kopecký et al., 2007] Kopecký, J., Vitvar, T., Bournez, C., and Farrell, J. (2007). SAWSDL : semantic annotations for WSDL and XML schema. *IEEE Internet Computing*, 11(6) :60–67. http://dret.net/biblio/reference/kop07.

[Kuntoff, 2005] Kuntoff, S. (2005). Nuclear plant principles. http://www.hellfirez.de/web/referate/inhalte/Physik_Energie.htm.

[Lagadec, 1992] Lagadec, P. (1992). *La gestion des crises*. Ediscience International.

[Larousse (dictionnaire), 2013] Larousse (dictionnaire) (2013). Événement (définition). http://www.larousse.fr/dictionnaires/francais/%C3%A9v%C3%A9nement/31839.

[Lauras, 2012] Lauras, M. (2012). *Systèmes d'Aide à la Décision pour des Organisations Collaboratives Efficientes et Agiles : Application à la Gestion de Chaînes Logistiques et à la Gestion de Crises*. Habilitation à diriger les recherches, Mines Albi - Institut National Polytechnique de Toulouse, Albi.

[Le Hors et al., 2004] Le Hors, A., Le Hégaret, P., Wood, L., Nicol, G., Robie, J., Champion, M., and Byrne, S. (2004). Document object model (DOM) level 3 core specification. http://www.w3.org/TR/DOM-Level-3-Core/.

[Levenshtein, 1966] Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8) :707–710.

[Lindberg, 1990] Lindberg, P. (1990). Strategic manufacturing management : A proactive approach. *International Journal of Operations & Production Management*, 10(2) :94–106. http://www.emeraldinsight.com/journals.htm?articleid=848511.

[Luckham and Schulte, 2008] Luckham, D. and Schulte, W. R. (2008). Event processing glossary - version 1.1. http://complexevents.com/wp-content/uploads/2008/08/epts-glossary-v11.pdf.

[Luzeaux and Ruault, 2008] Luzeaux, D. and Ruault, J.-R. (2008). *Systèmes de systèmes : concepts et illustrations pratiques*. Traité IC2, série Informatique et Systèmes d'Information. Hermes Science.

[Macé-Ramète et al., 2012] Macé-Ramète, G., Lamothe, J., Lauras, M., and Benaben, F. (2012). A road crisis management metamodel for an information decision support system. In *2012 6th IEEE International Conference on Digital Ecosystems Technologies (DEST)*, pages 1–5.

[Mangan and Sadiq, 2003] Mangan, P. J. and Sadiq, S. (2003). A constraint specification approach to building flexible workflows. *Journal of Research and Practice in Information Technology*, 35(1) :21–39. https://test.acs.org.au/__data/assets/pdf_file/0017/15740/JRPIT35.1.21.pdf.

[Maréchaux, 2006] Maréchaux, J.-L. (2006). Combining service-oriented architecture and event-driven architecture using an enterprise service bus. http://www.ibm.com/developerworks/library/ws-soa-eda-esb/.

[Marterer et al., 2012] Marterer, R., Moi, M., and Koch, R. (2012). An architecture for distributed, event-driven systems to collect and analyze data in emergency operations and training exercises. *Proceedings of ISCRAM*. http://www.iscramlive.org/ISCRAM2012/proceedings/215.pdf.

[McCullen et al., 2006] McCullen, P., Christopher, R. S., and Martin (2006). The f1 supply chain : Adapting the car to the circuit-the supply chain to the market. *Supply Chain Forum*, 7-1(12). www.supplychain-forum.com?num=12&art=105.

[McGuinness and Van Harmelen, 2004] McGuinness, D. L. and Van Harmelen, F. (2004). OWL web ontology language overview. *W3C recommendation*, 10(2004-03) :10. http://cies.hhu.edu.cn/pweb/~zhuoming/teachings/MOD/N4/Readings/5.3-B1.pdf.

[Michelson, 2006] Michelson, B. M. (2006). Event-driven architecture overview : Event-driven SOA is just part of the EDA story. http://dx.doi.org/10.1571/bda2-2-06cc.

[Montarnal et al., 2013] Montarnal, A., Mu, W., Bénaben, F., Barthe-Delanoë, A.-M., and Lamothe, J. (2013). Social vision of collaboration of organizations on a cloud platform. In Ferreira Pires, L. and Johnson, P., editors, *Enterprise Interoperability : IWEI 2013 Proceedings*, pages 123–132, Enschede, The Netherlands. Wiley-ISTE. http://eu.wiley.com/WileyCDA/WileyTitle/productCd-1848213174.html.

[Moore et al., 2000] Moore, J. P., Inder, R., Chung, P. W., Macintosh, A., and Stader, J. (2000). Who does what ? matching agents to tasks in adaptive workflow. In *ICEIS*, page 181–185. http://www.aiai.ed.ac.uk/project/pub/documents/2000/00-iceis-who-does-what.pdf.

[Morley et al., 2011] Morley, C., Bia-Figueiredo, M., and Gillette, Y. (2011). *Processus métiers et systèmes d'information : gouvernance, management, modélisation.* Dunod, Paris, 3ème edition. http://www.decitre.fr/livres/processus-metiers-et-si-9782100557059.html.

[Morley et al., 2007] Morley, C., Hugues, J., Leblanc, B., and Hugues, O. (2007). *Processus métiers et systèmes d'information : évaluation, modélisation, mise en oeuvre.* Dunod, Paris, 2ème edition.

[Mu, 2012] Mu, W. (2012). *Caractérisation et logique d'une situation collaborative.* PhD thesis, Mines Albi - Institut National Polytechnique de Toulouse. http://www.theses.fr/2012INPT0130.

[Nurcan and Hicheur, 2005] Nurcan, S. and Hicheur, A. (2005). A comparative state-of-the-art for flexible workflow modeling. In *Proceedings of Information Resources Management Association*, page 1, États-Unis. http://hal.archives-ouvertes.fr/hal-00706418.

[OASIS, 2006a] OASIS (2006a). Web services base notification 1.3 OASIS standard. http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf.

[OASIS, 2006b] OASIS (2006b). Web services brokered notification 1.3 OASIS standard. http://docs.oasis-open.org/wsn/wsn-ws_brokered_notification-1.3-spec-os.pdf.

[OASIS, 2006c] OASIS (2006c). Web services topics 1.3 OASIS standard. http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-os.pdf.

[OMG, 2011] OMG (2011). Business process model and notation.

[Overby et al., 2006] Overby, E., Bharadwaj, A., and Sambamurthy, V. (2006). Enterprise agility and the enabling role of information technology. *European Journal of Information Systems*, 15(2) :120–131. http://www.palgrave-journals.com/doifinder/10.1057/palgrave.ejis.3000600.

[Peters, 2005] Peters, L. (2005). Change detection in XML trees : a survey. In *3rd Twente Student Conference on IT*. http://www-poleia.lip6.fr/~gancarsk/grbd09/2005_03_B_Peters,L.J.-Change_detection_in_XML_trees_a_survey.pdf.

[Pingaud, 2009a] Pingaud, H. (2009a). Prospective de recherches en interopérabilité : vers un art de la médiation ? In *Actes du 8ème Congrès International de Génie Industriel*, Tarbes, France.

[Pingaud, 2009b] Pingaud, H. (2009b). Rationalité du développement de l'interopérabilité dans les organisations. In Riccio, P.-M. and Bonnet, D., editors, *Management des technologies organisationnelles*, pages 19–30, Montpellier, France. Transvalor - Presses des Mines.

[PLAY, 2010] PLAY (2010). PLAY project fact sheet. http://www.play-project.eu/images/pdf/ProjectFactsheet.pdf.

[Reichert et al., 1999] Reichert, M., Bauer, T., and Dadam, P. (1999). Enterprise-wide and cross-enterprise workflow management : Challenges and research issues for adaptive workflows. In *Enterprise-wide and Cross-enterprise Workflow Management*, pages 56–64. http://dbis.eprints.uni-ulm.de/243/.

[Reichert and Dadam, 1998] Reichert, M. and Dadam, P. (1998). ADEPTflex—Supporting dynamic changes of workflows without losing control. *Journal of Intelligent Information Systems*, 10(2) :93–129. http://link.springer.com/article/10.1023/A:1008604709862.

[Rolland et al., 1999] Rolland, C., Prakash, N., and Benjamen, A. (1999). A multi-model view of process modelling. *Requirements Engineering*, 4(4) :169–187. http://link.springer.com/article/10.1007/s007660050018.

[Rooney, 2011] Rooney, C. (2011). Crisis project website. `http://idc.mdx.ac.uk/projects/crisis/`.

[Rüppel and Wagenknecht, 2007] Rüppel, U. and Wagenknecht, A. (2007). Improving emergency management by formal dynamic process-modelling. In *24th conference on information technology in construction*. `http://itc.scix.net/data/works/att/w78-2007-085-040-Rueppel.pdf`.

[Saaty, 1988] Saaty, T. L. (1988). What is the analytic hierarchy process ? In Mitra, G., Greenberg, H. J., Lootsma, F. A., Rijkaert, M. J., and Zimmermann, H. J., editors, *Mathematical Models for Decision Support*, number 48 in NATO ASI Series, pages 109–121. Springer Berlin Heidelberg. `http://link.springer.com/chapter/10.1007/978-3-642-83555-1_5`.

[Sadiq and Orlowska, 2000] Sadiq, S. W. and Orlowska, M. E. (2000). On capturing exceptions in workflow process models. In *BIS 2000*, School of Information Technology and Electrical Engineering Publications, pages 3–19. Springer Verlag.

[Scheer and Nüttgens, 2000] Scheer, A.-W. and Nüttgens, M. (2000). ARIS architecture and reference models for business process management. In Aalst, W. v. d., Desel, J., and Oberweis, A., editors, *Business Process Management*, number 1806 in Lecture Notes in Computer Science, pages 376–389. Springer Berlin Heidelberg. `http://link.springer.com/chapter/10.1007/3-540-45594-9_24`.

[Schonenberg et al., 2008] Schonenberg, H., Mans, R., Russell, N., Mulyar, N., and Aalst, W. v. d. (2008). Process flexibility : A survey of contemporary approaches. In Dietz, J. L. G., Albani, A., and Barjis, J., editors, *Advances in Enterprise Engineering I*, Lecture Notes in Business Information Processing, pages 16–30. Springer Berlin Heidelberg. `http://link.springer.com/chapter/10.1007/978-3-540-68644-6_2`.

[Schulte, 2003] Schulte, R. W. (2003). The growing role of events in enterprise applications. *Gartner Research*, 7. `http://www.immagic.com/eLibrary/ARCHIVES/GENERAL/GARTNER/G030709S.pdf`.

[Selkow, 1977] Selkow, S. (1977). The tree-to-tree editing problem. *Inform. Processing Letters*, 6(6) :184–186.

[Sharifi and Zhang, 1999] Sharifi, H. and Zhang, Z. (1999). A methodology for achieving agility in manufacturing organisations : An introduction. *International Journal of Production Economics*, 62(1) :7–22. `http://www.sciencedirect.com/science/article/pii/S0925527398002175`.

[Sheffi, 2004] Sheffi, Y. (2004). Demand variability and supply chain flexibility. In *Entwicklungspfade und Meilensteine moderner Logistik*, pages 85–117. Gabler Verlag. `http://link.springer.com/chapter/10.1007/978-3-322-89044-3_5`.

[Sienou, 2009] Sienou, A. (2009). *Proposition d'un cadre méthodologique pour le management intégré des risques et des processus d'entreprise*. PhD thesis, Université de Toulouse, Mines Albi. `http://oatao.univ-toulouse.fr/7835/`.

[Slack, 2005] Slack, N. (2005). The flexibility of manufacturing systems. *International Journal of Operations & Production Management*, 25(12) :1190–1200. http://www.emeraldinsight.com/journals.htm?articleid=1529548.

[Smith, 2007] Smith, A. (2007). New risk-management standard for delivering superior environmental and safety performance. http://www.onepetro.org/mslib/servlet/onepetropreview?id=SPE-105503-MS.

[SocEDA, 2010] SocEDA (2010). SocEDA project homepage. https://research.petalslink.org/display/soceda/SocEDA+Overview.

[Swafford et al., 2006] Swafford, P. M., Ghosh, S., and Murthy, N. (2006). The antecedents of supply chain agility of a firm : Scale development and model testing. *Journal of Operations Management*, 24(2) :170–188. http://www.sciencedirect.com/science/article/pii/S0272696305000975.

[Tai, 1979] Tai, K.-C. (1979). The tree-to-tree correction problem. *J. ACM*, 26(3) :422–433. http://doi.acm.org/10.1145/322139.322143.

[TIBCO, 2011] TIBCO (2011). Event processing patterns decisions reactions. http://tibcoblogs.com/cep/wp-content/uploads/2010/02/ep-patterns-decisions-reactions.png.

[TIBCO, 2013] TIBCO (2013). TIBCO business studio. http://developer.tibco.com/business_studio/default.jsp.

[Tokyo Electric Power Company, 2012] Tokyo Electric Power Company, I. (2012). Fukushima nuclear accident analysis report. Technical report, Tokyo Electric Power Company, Inc., Japan. http://www.tepco.co.jp/en/press/corp-com/release/betu12_e/images/120620e0104.pdf.

[Truptil, 2011] Truptil, S. (2011). *Etude de l'approche de l'interopérabilité par médiation dans le cadre d'une dynamique de collaboration appliquée à la gestion de crise.* PhD thesis, Université de Toulouse, Mines Albi. http://ethesis.inp-toulouse.fr/archive/00001482/.

[van der Aalst and ter Hofstede, 2005] van der Aalst, W. M. and ter Hofstede, A. H. (2005). YAWL : yet another workflow language. *Information systems*, 30(4) :245–275. http://www.sciencedirect.com/science/article/pii/S0306437904000304.

[van der Aalst et al., 1998] van der Aalst, W. M. P., Basten, T., Verbeek, H. M. W., Verkoulen, P. A. C., and Voorhoeve, M. (1998). Adaptive workflow : On the interplay between flexibility and support. In *Proceedings of the first International Conference on Enterprise Information Systems*, volume 2, pages 353–360, Setúbal, Portugal.

[van der Aalst and Jablonski, 2000] van der Aalst, W. M. P. and Jablonski, S. (2000). Dealing with workflow change : identification of issues and solutions. *Computer systems science and engineering*, 15(5) :267–276. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.120.4413&rep=rep1&type=pdf.

[van der Aalst et al., 2009] van der Aalst, W. M. P., Pesic, M., and Schonenberg, H. (2009). Declarative workflows : Balancing between flexibility and support. *Computer*

Science - Research and Development, 23(2) :99–113. http://link.springer.com/10.1007/s00450-009-0057-9.

[van der Aalst et al., 2005] van der Aalst, W. M. P., Weske, M., and Grünbauer, D. (2005). Case handling : a new paradigm for business process support. *Data Knowl. Eng.*, 53(2) :129–162. http://dx.doi.org/10.1016/j.datak.2004.07.003.

[Verbeek et al., 2002] Verbeek, H. M. W., Hirnschall, A., and Aalst, W. M. P. v. d. (2002). XRL/Flower : supporting inter-organizational workflows using XML/Petri-net technology. In *Web Services, E-Business, and the Semantic Web, CAiSE 2002 International Workshop, WES 2002*, page 93–109. LNCS Springer.

[Verginadis et al., 2012] Verginadis, Y., Patiniotakis, I., Papageorgiou, N., Apostolou, D., Mentzas, G., and Stojanovic, N. (2012). Context management in event marketplaces. In *Joint Proceedings of FEOSW 2012, RED 2012 and SBPM 2012*, volume Vol-862 of *CEUR Workshop Proceedings*, page 138–145. CEUR Workshop Proceedings. http://ceur-ws.org/Vol-862/SBPMp3.pdf.

[Vernon, 2008] Vernon, F. (2008). Supply chain visibility : lost in translation ? *Supply Chain Management : An International Journal*, 13(3) :180–184. http://www.emeraldinsight.com/journals.htm?articleid=1724145&show=abstract.

[Vianu, 1997] Vianu, V. (1997). Rule-based languages. *Annals of Mathematics and Artificial Intelligence*, 19(1-2) :215–259. http://link.springer.com/article/10.1023/A%3A1018907806177.

[W3C, 2008] W3C (2008). Extensible markup language (XML) 1.0 (fifth edition). http://www.w3.org/TR/REC-xml/.

[Weske, 2001] Weske, M. (2001). Formal foundation and conceptual design of dynamic adaptations in a workflow management system. In *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, page 10–pp. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=927082.

[Wiederhold, 1992] Wiederhold, G. (1992). Mediators in the architecture of future information systems. *Computer*, 25(3) :38–49. http://dx.doi.org/10.1109/2.121508.

[Zhang and Shasha, 1989] Zhang, K. and Shasha, D. (1989). Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 18(6) :1245–1262. http://epubs.siam.org/doi/abs/10.1137/0218082.

[Zhang et al., 2003] Zhang, Q., Vonderembse, M. A., and Lim, J.-S. (2003). Manufacturing flexibility : defining and analyzing relationships among competence, capability, and customer satisfaction. *Journal of Operations Management*, 21(2) :173–191. http://www.sciencedirect.com/science/article/pii/S0272696302000670.

[Zribi et al., 2012] Zribi, S., Bénaben, F., Hamida, A. B., and Lorré, J.-P. (2012). Towards a service and choreography governance framework for future internet. In *I-ESA*, pages 281–291.

Annexe A

Détails et chronologie du cas d'étude

Ce cas d'étude constitue une synthèse des cas d'étude créés dans le cadre des projets PLAY et SocEDA. Ces cas d'études décrivent 63 processus, tous niveaux (Décisionnel, Opérationnel et Support) confondus.

Le cas d'étude de crise nucléaire est basé sur la législation française, dont les recommandations imposent certains seuils. Il est à noter que le mSv (milli Siviert) par heure est l'unité de mesure du système international pour mesure l'impact des radiations sur les êtres humains :

- Surveillance du site si le taux de dose dépasse 0,025 mSv/h,
- Confinement et distribution de pastilles d'iode si le taux de dose excède les 2 mSv/h,
- Evacuation si le taux de dose cumulé dépasse les 50 mSv/h,
- Ingestion des pastilles d'iode si le taux de dose cumulé excède les 50mSv/h.

Le scénario du cas d'étude se compose de trois phases majeures.

A.1 Première phase du scénario : t_0 à t_0+20 minutes

Dans la première phase du scénario, nous considérerons que la crise nucléaire a déjà commencé et que les acteurs de la gestion de crise sont déjà présents dans le poste de commandement de la cellule de crise et sur le théâtre de la crise (tel que résumé dans la Figure A.1). Dans la cellule de crise, le représentant de l'autorité au niveau national (à savoir le Préfet) gère la situation suivant le workflow de gestion de la situation (cf. Figure A.1).

Au début du scénario (déterminé à t_0), un périmètre de surveillance de 5 km a été établi autour de la centrale nucléaire. A l'intérieur de ce périmètre, 5 capteurs de radioactivité envoient une mesure de la radioactivité toutes les 30 secondes, ce qui correspond à une émission de 10 événements par minute (cf. Figure A.2). Ces événements incluent

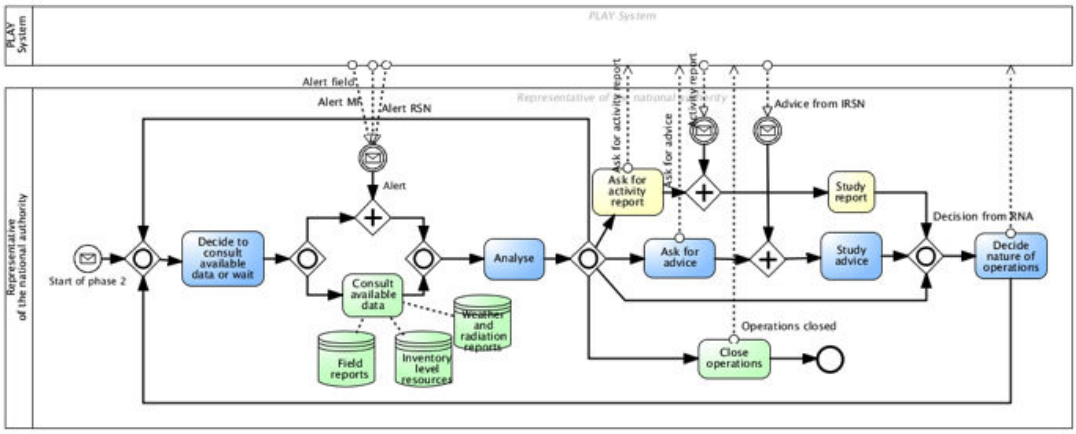


FIGURE A.1 – Le workflow de gestion de la situation

des informations sur l'identité du capteur, sa position géographique, la date de la mesure et la valeur de la mesure.

En parallèle, Météo France envoie des informations concernant la situation éolienne (voir Figure A.2). Ces mesures sont basées sur 5 capteurs localisés dans le périmètre de surveillance de 5 km autour de la centrale nucléaire. A chacune des mesures météorologiques, deux événements sont associés. Le premier inclut l'information sur l"identité du capteur, sa position géographique, la date de la mesure et la valeur de la mesure de la vitesse du vent. Le second embarque l'information à propos de l"identité du capteur, sa position géographique, la date de la mesure et la valeur de la direction du vent. Chaque capteur envoie ces informations toutes les 30 secondes, ce qui correspond à une émission de 20 événements par minute.

Durant les 3 premières minutes du scénario, toutes les valeurs des mesures émises par les capteurs ne déclenchent rien (la radioactivité étant inférieure à 1 mSv et il y a absence de vent).

Après 3 minutes, les mesures du taux de dose de radioactivité augmentent progressivement (tendance à 0,3) jusqu'à 1,8 mSv à t_0`+7 minutes`.

Considérant la règle métier suivante, le SIM envoie une alerte au Préfet :

SI mesure radiation $> V_+$
OU (mesure radiation $> V_-$ **ET** δRM/δt $>$ s)
==>Alerte{Risque ; Contamination ; Capteurs}

Si la mesure de radiation excède V_+ or si la mesure de radiation excède V_- et augmente trop rapidement, alors une alerte concernant le réseau de surveillance de radioactivité est envoyée (avec $V_+ > V_-$, V_+=2 mSv, V_-= 1 mSv, s=0,2).

Les valeurs des mesures des différents capteurs restent globalement stable entre t_0+7 minutes et t_0+9 minutes. À t_0+9 minutes, les informations météorologiques indiquent

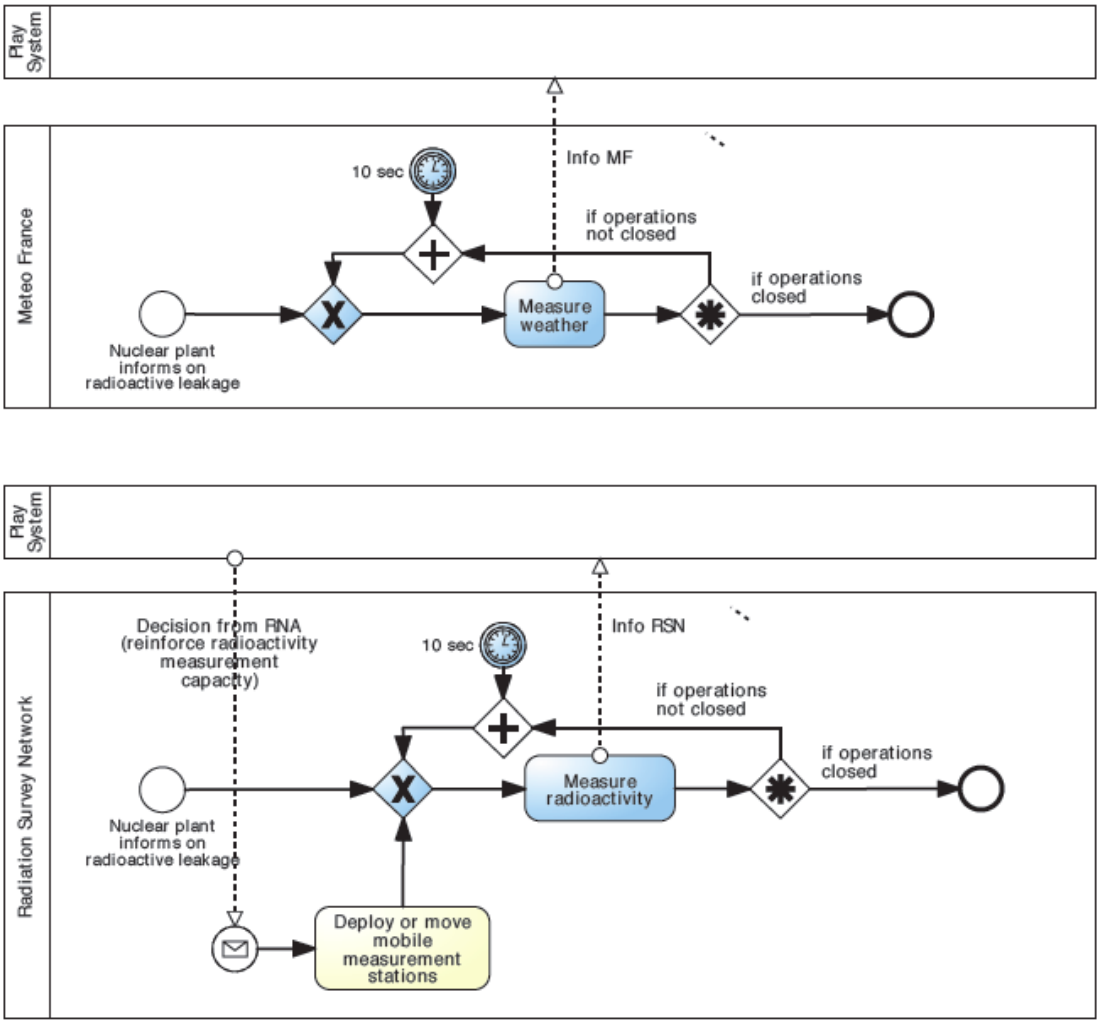


FIGURE A.2 – Le workflow de gestion de la situation

que la vitesse du vent atteint 40 km/h et souffle dans une direction donnée (Sud Est). Eût égard à la règle métier suivante, le SIM doit envoyer une alerte :

SI δDirectionVent/δt > δDirectionVent
OU δIntensitéVent/δt > δIntensitéVent
==>Alerte{Risque ; Contamination ; Vent}

Si le vent change trop brutalement de direction ou d'intensité, alors une alerte concernant Météo France doit être envoyée (δ`IntensitéVent = 30 km/h`).

Sur la base de cette alerte, le Préfet décide d'étendre le périmètre de surveillance à 30 km en direction du Sud Est. Par conséquent, 20 capteurs de radioactivité supplémentaires sont activées. A cet instant là, le nombre d'événements atteint les 70 événements par minute.

En parallèle (i.e. à t_0+9 `minutes`), le Préfet demande un avis scientifique à l'IRSN (Institut de Radioprotection et de Sûreté Nucléaire). 5 minutes plus tard, (i.e. à t_0+14 `minutes`), le rapport (écrit) de l'IRSN est envoyé via le SIM. Sur la base de cette information, le Préfet décide d'étendre le périmètre de surveillance à l'ensemble du territoire régional (toujours du point de vue de la surveillance de la radioactivité). 5 capteurs météorologiques supplémentaires sont activés en complément des 320 capteurs de radioactivité. La fréquence d'événements émis atteint les 660 événements par minute.

La première phase du scénario prend fin à t_0+20 `minutes`. Il est à noter que tout au long du scénario, la règle métier suivante est gérée par le SIM de façon à aider la prise de décision :

TOUTES les 5 minutes
==>**FAIRE** DessinerGraphe (mesures IRSN des 5 dernières minutes) **ET** GraphеIRSN

Toutes les 5 minutes, sur la base des mesures de radioactivité recueillies dans les 5 dernières minutes, dessiner le graphe des mesures et l'envoyer.

A.2 Deuxième phase du scénario : t_0+20 minutes à t_0+30 minutes

Durant les 5 dernières minutes, 3 capteurs de radioactivité envoient des mesures supérieures à V_+. La première règle métier s'applique à nouveau. Par conséquent, le Préfet décide à t_0+20 `minutes` de confiner la population dans un périmètre de 5 km autour de la centrale nucléaire. La règle métier suivante est appliquée à travers le SIM en suivant le workflow de protection de la population (voir Figure A.3).

SI une décision de confinement est prise
==>Demand{Plan confinement ; Représentant Police}

Entre t_0+20 `minutes` à t_0+25 `minutes`, le représentant de la Police définit le plan de confinement. Ensuite, le plan est transmis aux différents acteurs abonnés au type d'événement lié à ce plan. Ce plan inclue les informations sur les limites précises du confinement et liste les différentes actions à mener. Entretemps, la DIR organise un périmètre de sécurité en fermant certains axes routiers et en créant des déviations. Comme montré dans la Figure A.3, la règle suivante est activée par le SIM :

SI PlanConfinement validé
==>Alert{Consequence ; Plan confinement ; Représentant DIR}

Si le plan de confinement est validé, alors il faut le transmettre au représentant de la DIR. Il/Elle devra développer un plan de circulation dédié pour éviter des entrées intruses dans le périmètre de confinement.

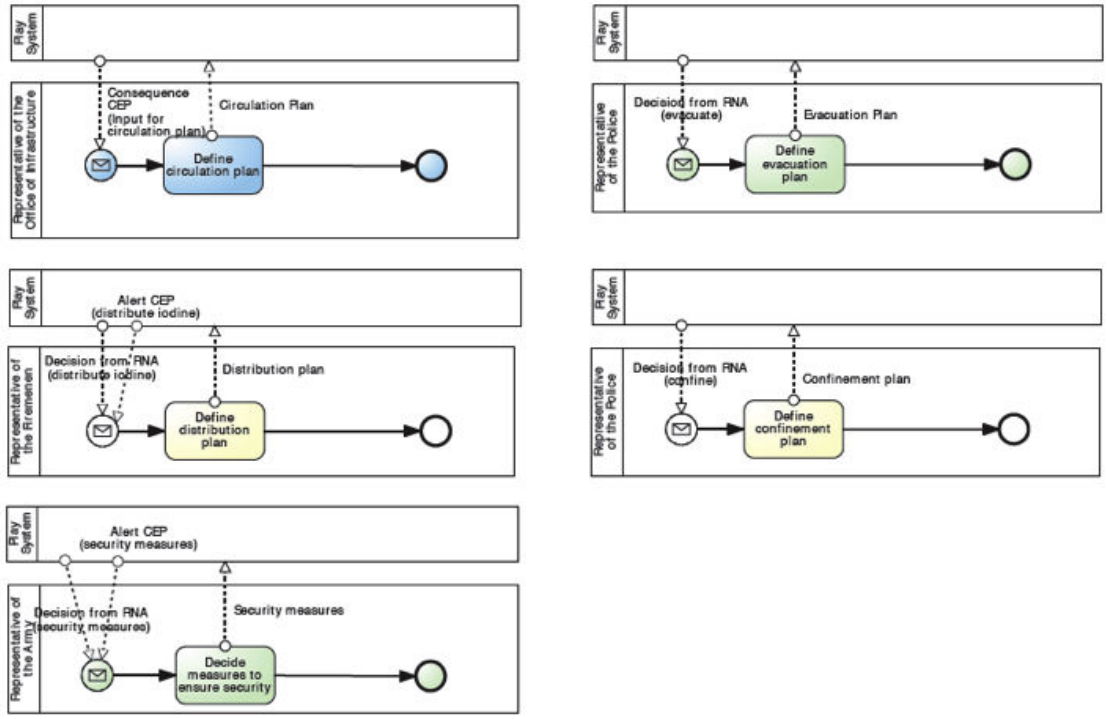


FIGURE A.3 – Le workflow de protection de la population

Dans les scénarios des cas d'études PLAY et SocEDA, le plan de confinement doit aussi activer d'autres décisions comme sécuriser le périmètre (Armée et Police), distribuer des pastilles d'iode supplémentaires (Pompiers et Municipalités). Le scénario du présent cas d'étude (issu des cas d'étude PLAY et SocEDA) se focalise uniquement sur la DIR. A t_0+30 `minutes`, le représentant de la DIR a conçu le plan de circulation (au niveau opérationnel) qui doit être déployé sur le théâtre des opérations. Ce plan indique que 8 routes doivent être fermées et que 12 itinéraires de déviation doivent être mis en place dans un périmètre de 5 km autour de la centrale nucléaire. Ce plan de circulation est envoyé aux différents souscripteurs de la cellule de crise (Préfet, représentants de la Police, de l'Armée et des Pompiers) et sur le terrain (équipes de la DIR). Cette deuxième phase du scénario se clôture à t_0+30 `minutes`.

A.3 Troisième phase du scénario : t_0+30 minutes à t_0+105 minutes

Cette dernière phase du scénario est définie entre t_0+30 `minutes` à t_0+105 `minutes`. L'accent est mis sur deux ensembles de workflows, qui sont Implémenter le plan de circulation (Figure A.4) et Gérer les ressources (Figure A.5), du point de vue des ressources de la DIR.

Durant cette période, le responsable de l'équipe DIR sur le terrain consulte l'inven-

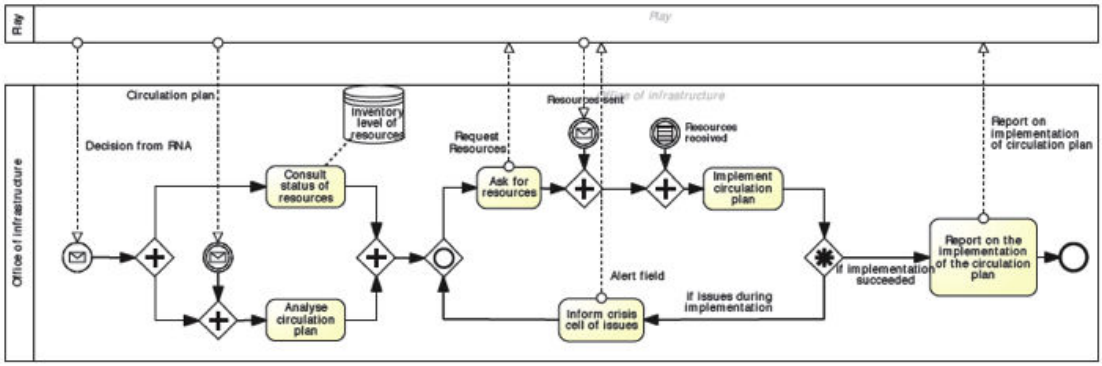


FIGURE A.4 – Le workflow de d'implémentation du plan de circulation

taire des ressources via le SIM et analyse le plan de circulation envoyé par le Préfet via le SIM également. Les tâches sont exécutées entre t_0+30 `minutes` et t_0+35 `minutes`. Sur la base des informations recueillies (4 véhicules sont disponibles et 3 zones géographiques différentes ont été définies dans le plan de circulation), l'équipe demande à t_0+35 `minutes` que 3 véhicules équipés de panneaux routiers génériques afin de déployer le plan de circulation. À t_0+35 `minutes`, la personne en charge de la gestion de l'inventaire de la DIC reçoit la requête à travers le SIM. La demande est validée et la mise à disposition pour t_0+40 `minutes` est confirmée. Le déploiement effectif démarre à t_0+40 `minutes` avec l'arrivée des 3 véhicules. La fin du déploiement est programmée pour t_0+70 `minutes`.

Mais à t_0+52 `minutes`, une alerte provenant du terrain est envoyée au SIM : un des véhicules a eu un accident. Grâce à la règle métier suivante, le composant d'agilité vérifie toutes les 10 minutes l'existence potentielle d'écart entre le modèle attendu (3 véhicules travaillent sur le terrain) et le modèle terrain (seuls 2 véhicules travaillent effectivement sur le terrain).

SI $\text{Ressources}_{allouées}$ <> $\text{Ressources}_{requises}$
==>Demand{Ressource} **OU** Resource{Repartition charge}

S'il y a une différence entre les besoins en ressources et les ressources mises à disposition, alors une alerte est envoyée à la personne en charge du workflow pour l'informer du problème et lui proposer des solutions alternatives.

Par conséquent, à t_0+60 `minutes`, le système détecte le problème et propose à l'équipe de la DIR de demander un nouveau véhicule ou de répartir l'activité restante sur les véhicules restants. La seconde option est retenue et l'information est envoyée aux deux véhicules qui sont encore sur le terrain.

À t_0+80 `minutes`, le déploiement du plan de circulation est toujours en cours. Aucune information n'a été fournie par les équipes sur le terrain. Par conséquent, à ce

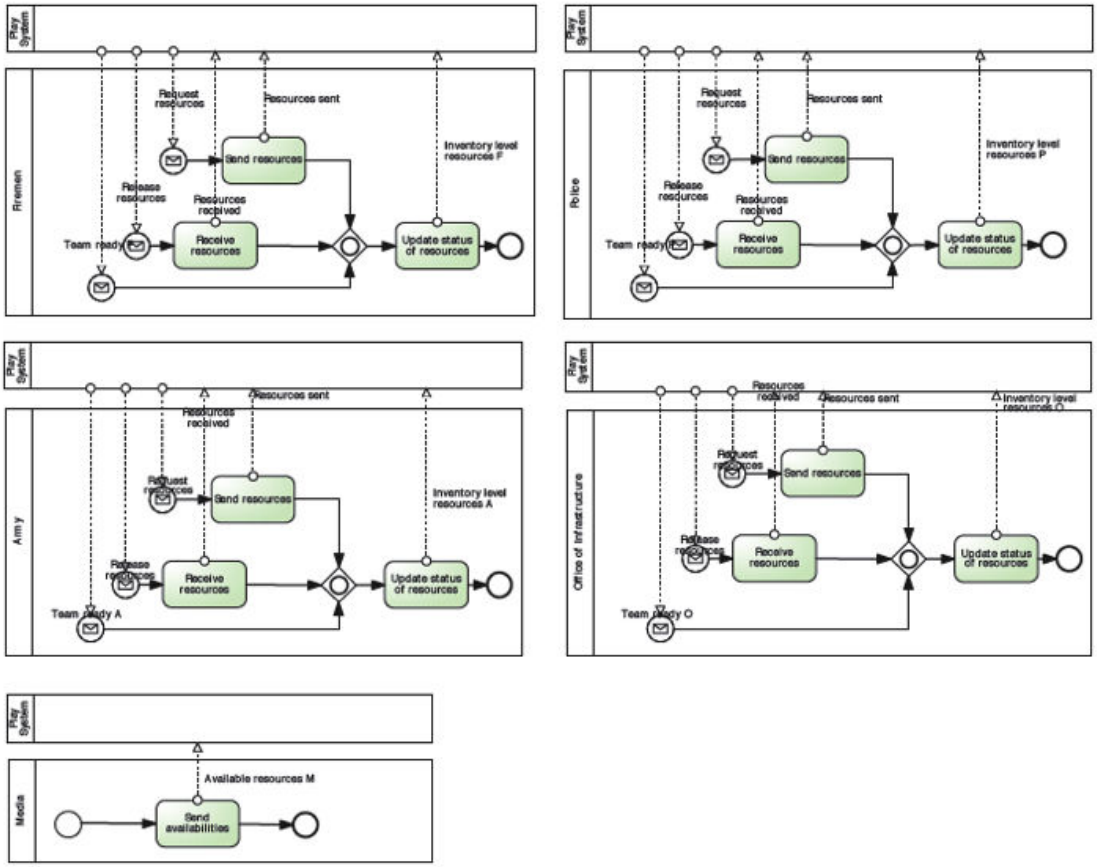


FIGURE A.5 – Le workflow de gestion des ressources de la DIR

moment là pour les membres de la cellule de crise estiment que le déploiement du plan de circulation est terminée (fin théorique à t_0+70 `minutes`). Mais le composant d'agilité vérifie toutes les 10 minutes l'évolution des activités en cours. Pour rappel, toutes les activités connectées au SIM sont suivies à travers l'évolution de leur statut : (i) en attente (waiting), (ii) en cours (ongoing), (iii) terminé (finished). Par conséquent, à t_0+80 `minutes`, la règle suivante est appliquée :

SI $\text{Statut}_{courant}$ Activité <> $\text{Statut}_{attendu}$ Activité
==>Proposer différentes alternatives Rapport Immédiat Requis **OU** Envoyer Acteur Sur Terrain **OU** Attendre

Si une différence est détectée entre le statut actuel de l'activité et le statut attendu de l'activité, alors une alerte est envoyée à la personne en charge du workflow pour l'informer du problème et lui proposer des solutions alternatives.

Sur la base de ces propositions, le représentant de la DIR décide de demander un rapport immédiat de la situation. Les équipes sur le terrain envoient un court rapport à t_0+83 `minutes` pour expliquer qu'ils vont bientôt terminer leur activité. À t_0+88

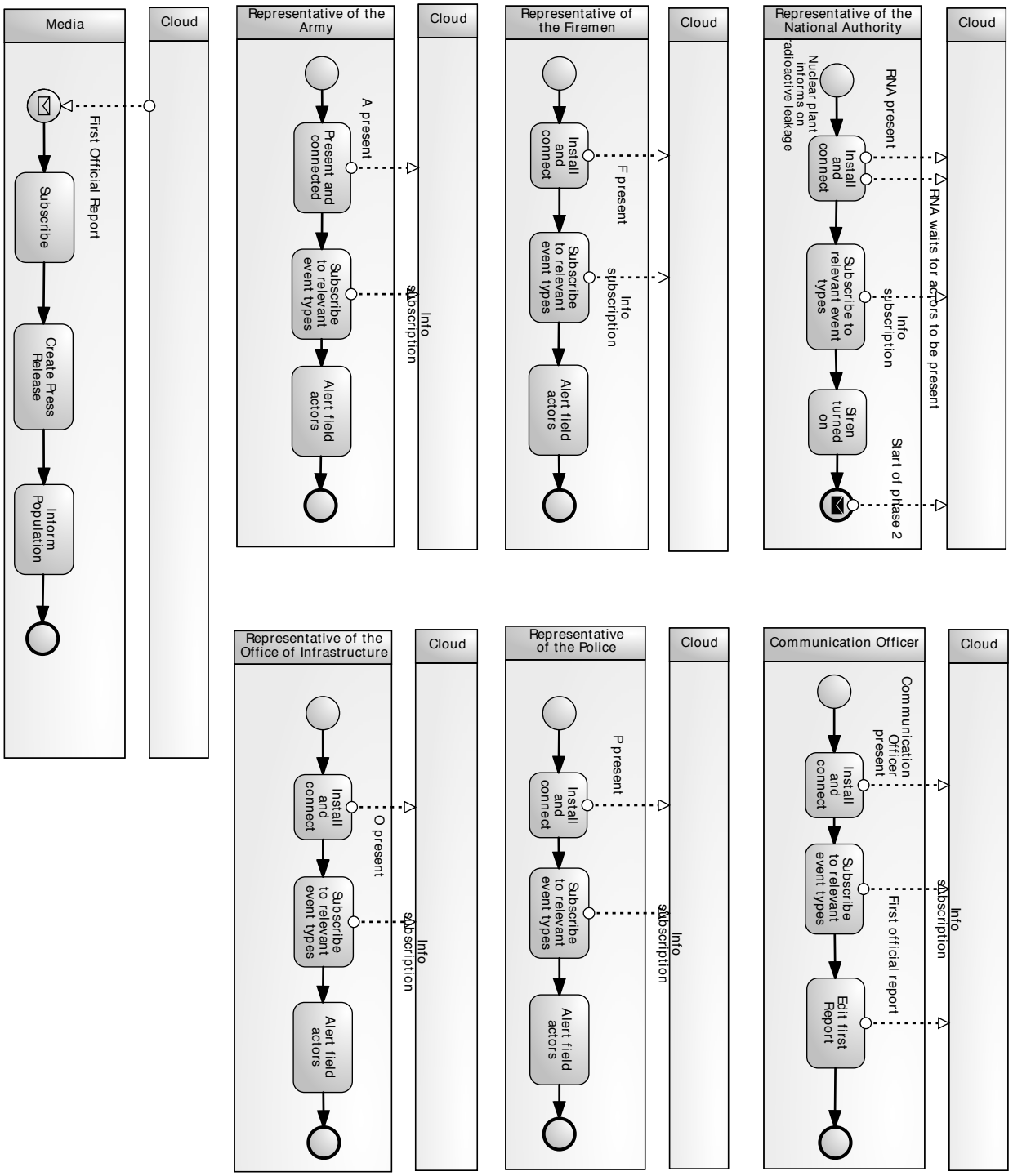
`minutes`, le déploiement du plan de circulation est terminée et un rapport final est envoyé aux abonnés. Les véhicules sont libérés à t_0`+105 minutes` et les inventaires sont mis à jour.

Annexe B

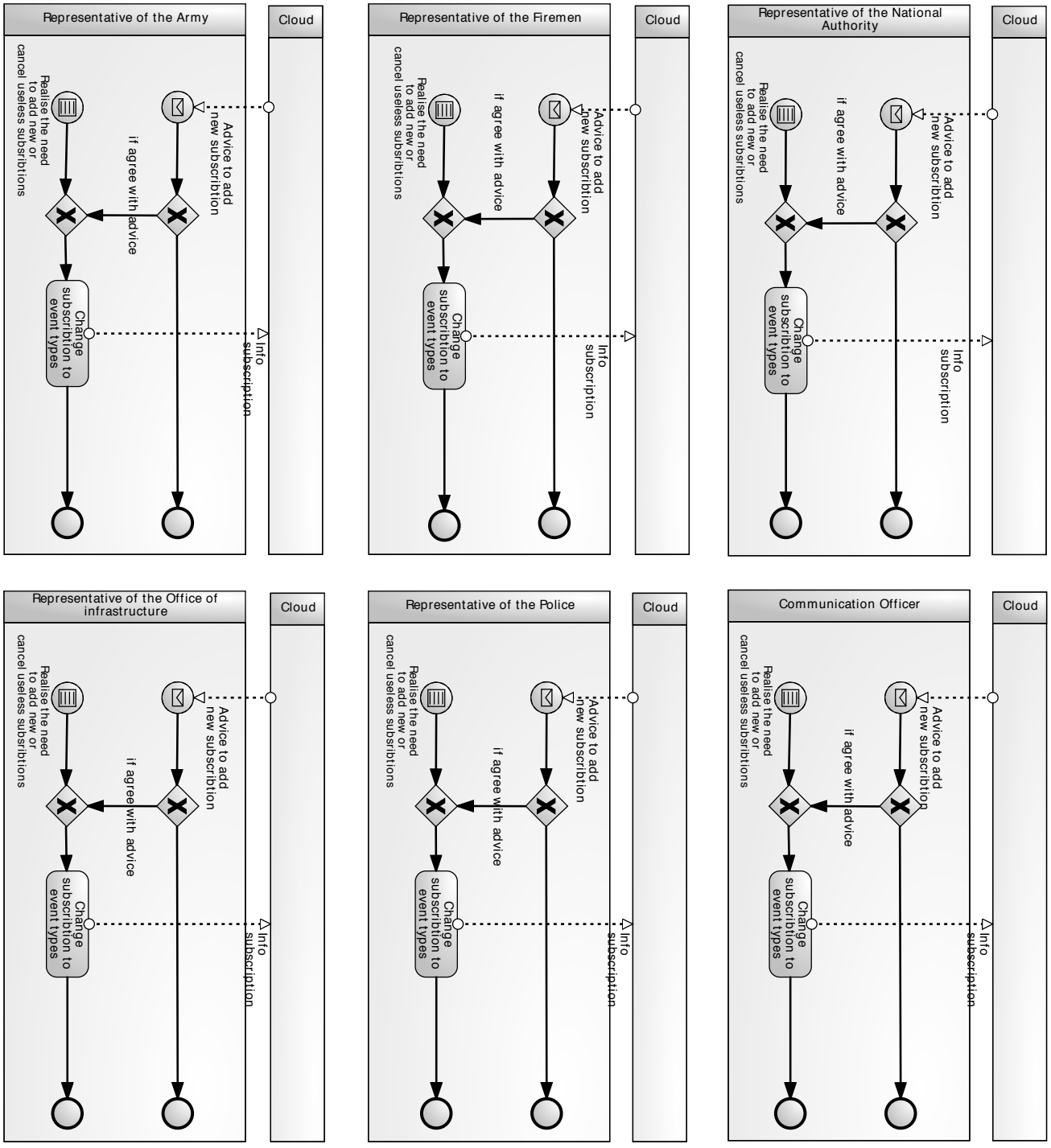
Les processus collaboratifs BPMN du cas d'étude

Les processus BPMN du cas d'étude ont été définis suivant les niveaux décisionnel, opérationnel ou support. L'inclusion des fichiers les représentant commence à la page suivante.

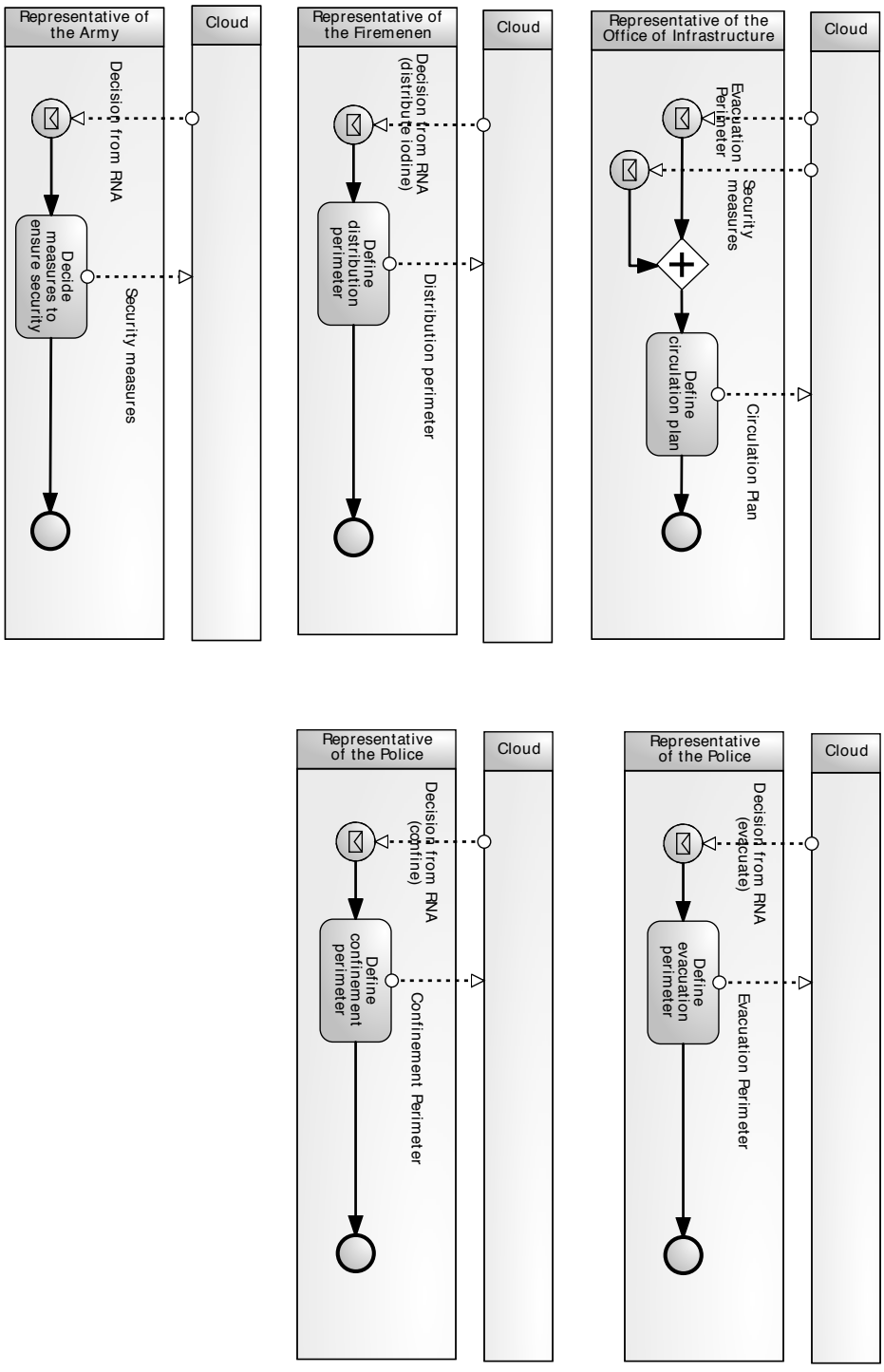
Decision_sc1_Alert_subscribe



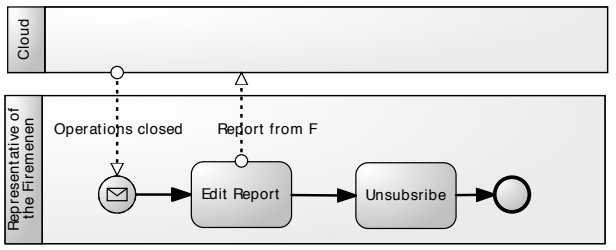
Decision_sc1_Dynamic_subscribe

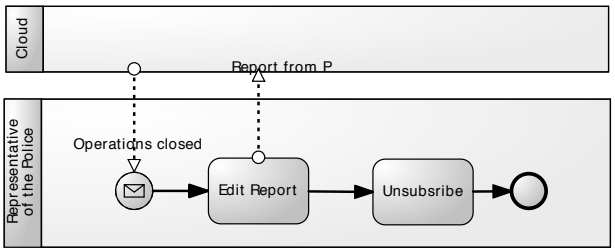


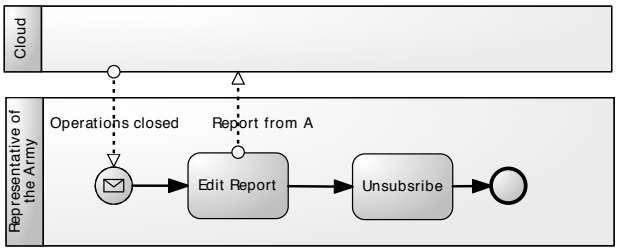
Decision_sc1_Protect

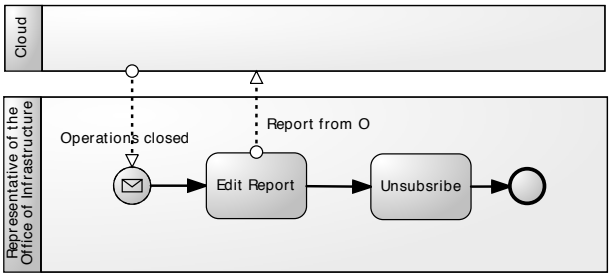


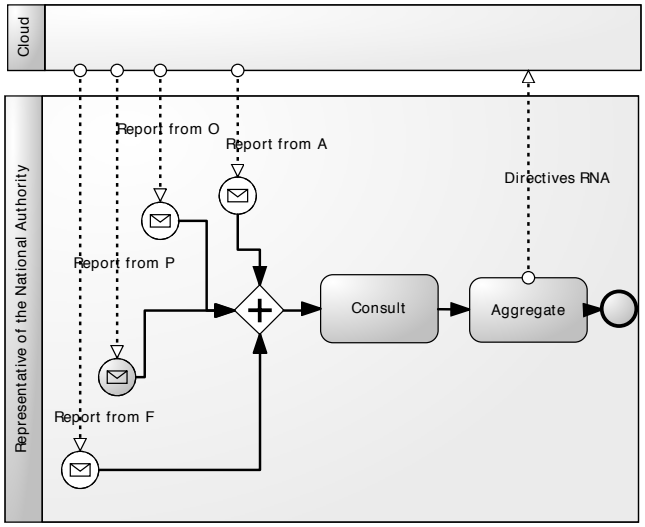
Decision_sc1_Recover

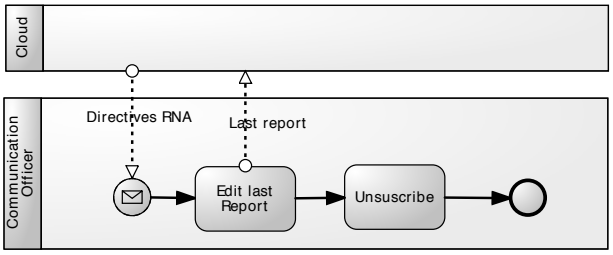


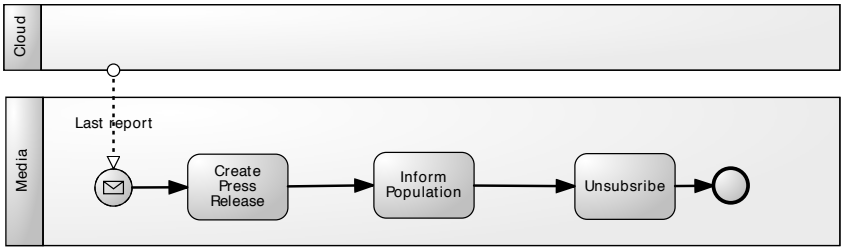




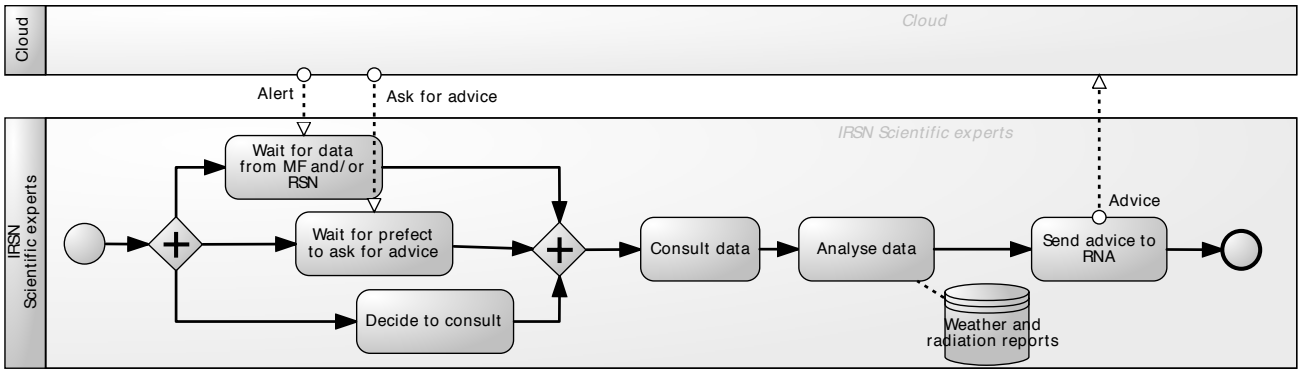




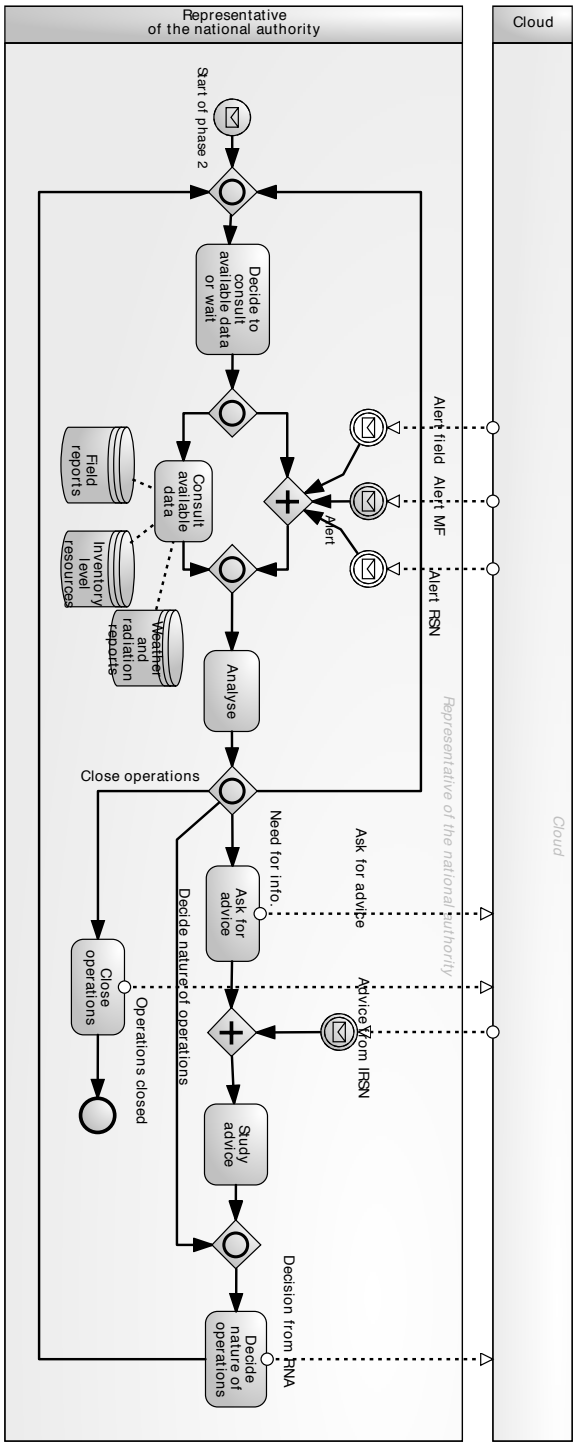




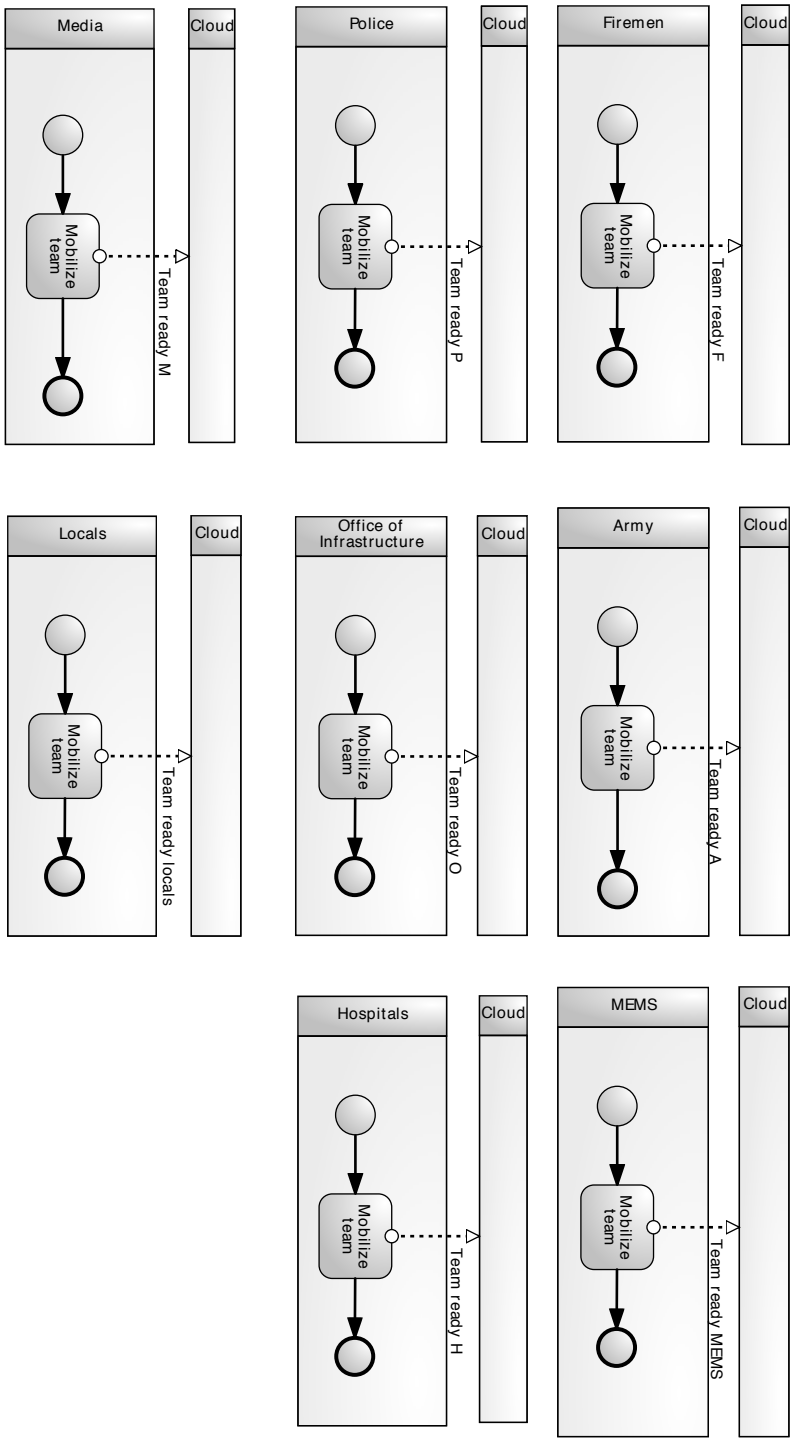
Decision_sc1_Scientific_support



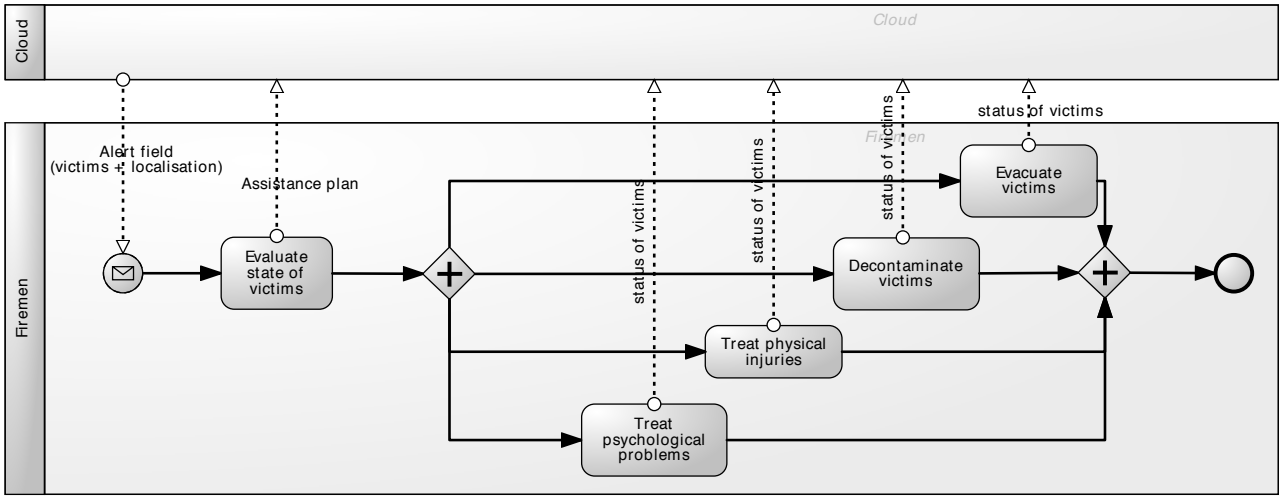
Decision_sc1_Situation_management

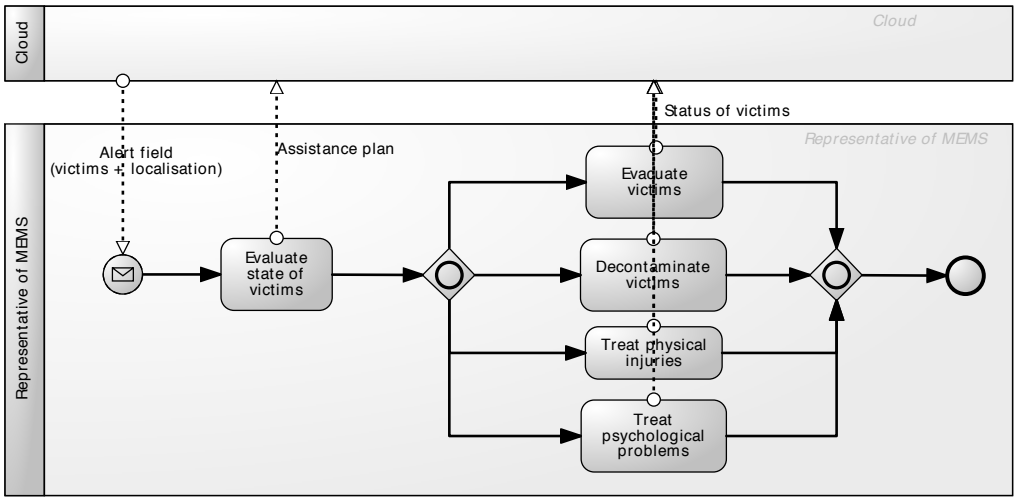


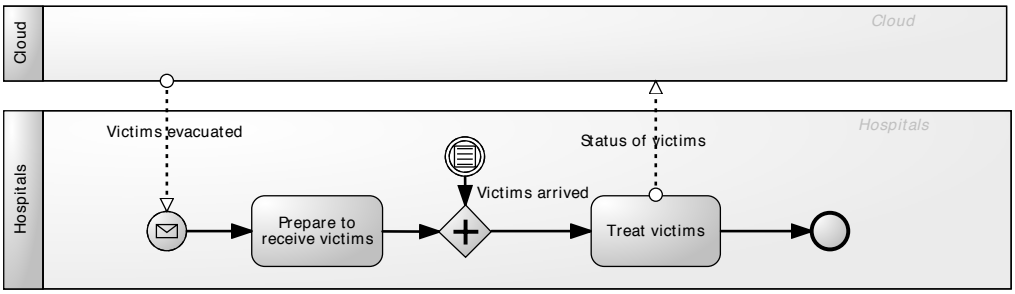
Operation_sc1_Alert



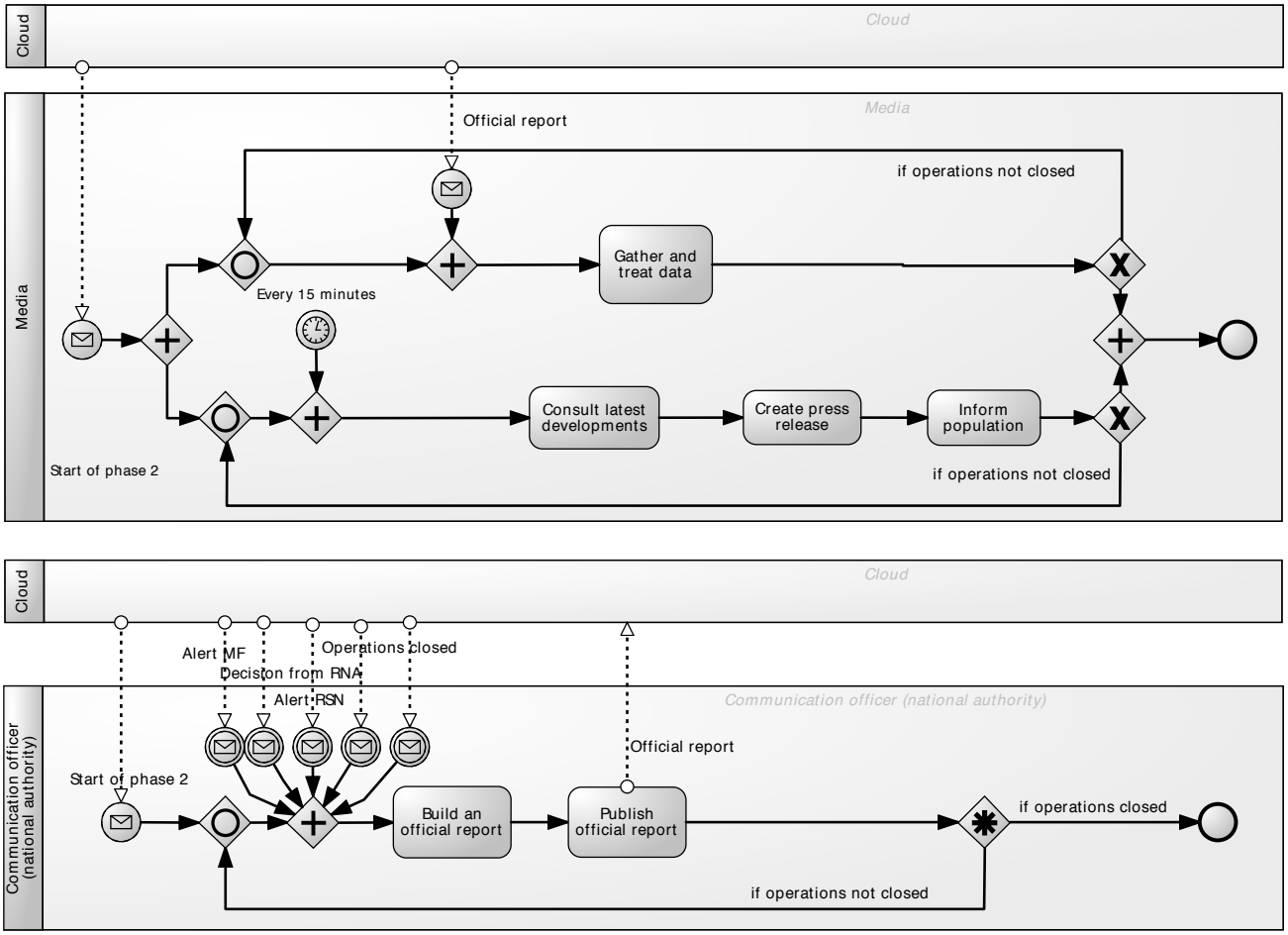
Operation_sc1_Assist



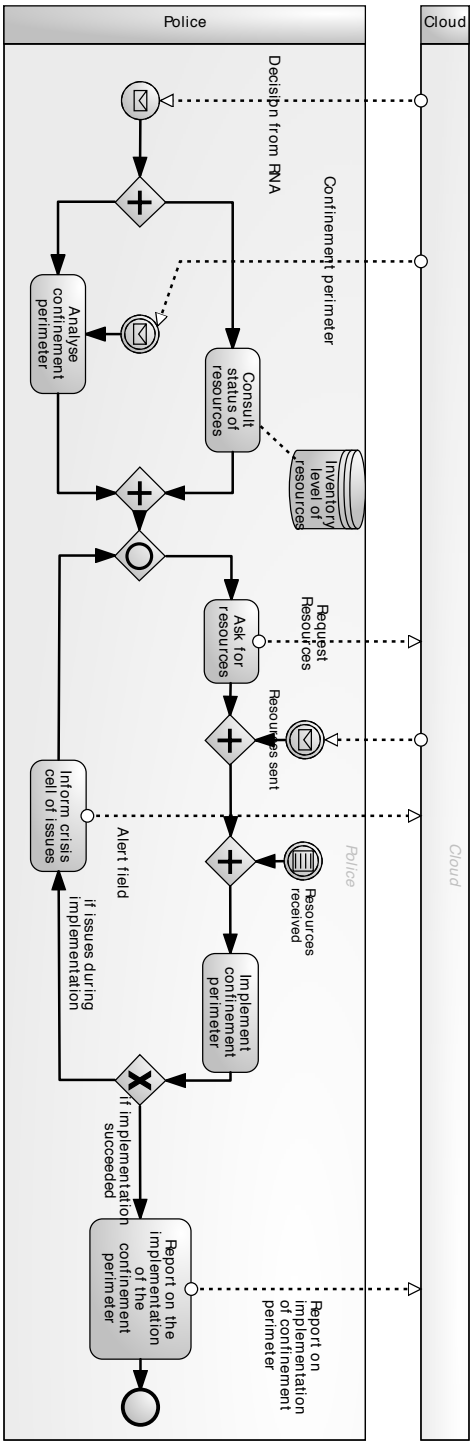




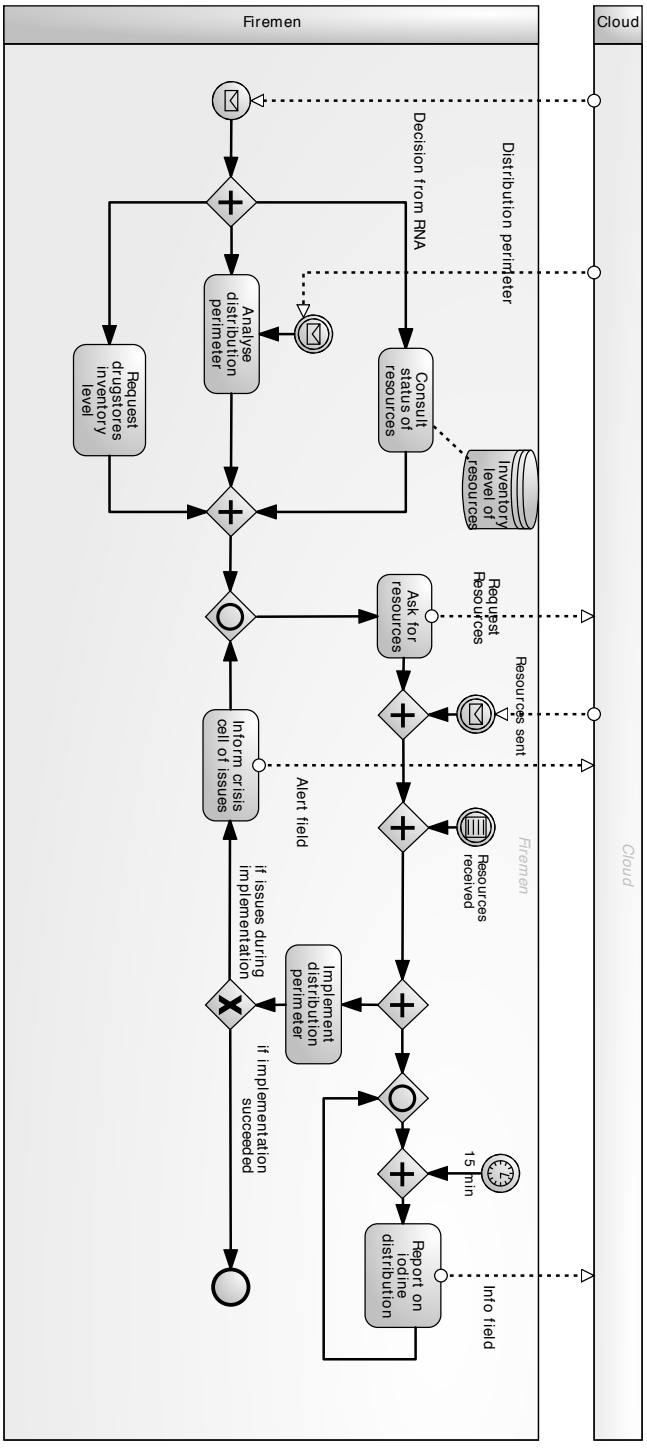
Operation_sc1_Communicate



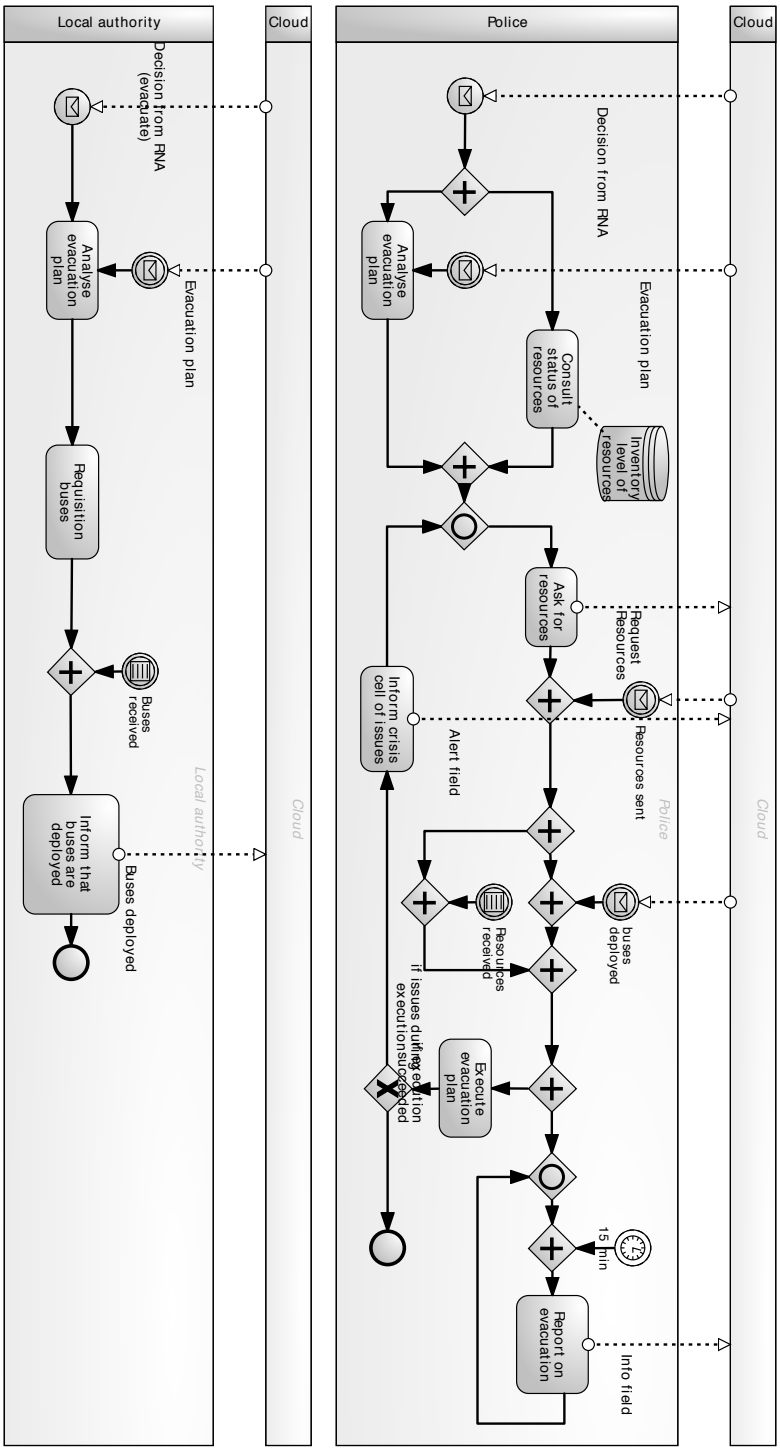
Operation_sc1_Confine_population



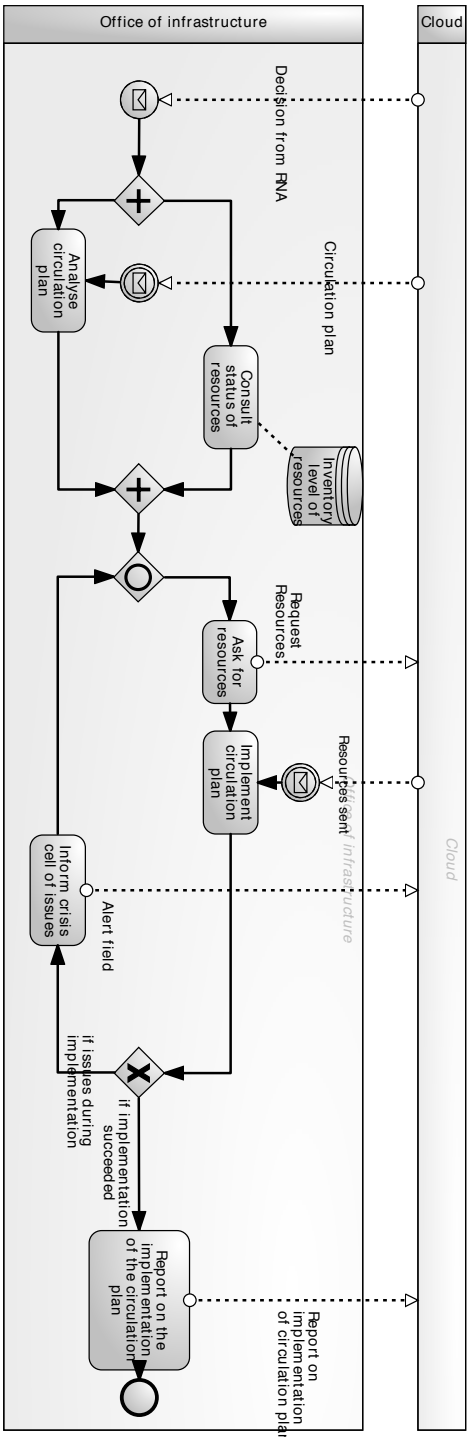
Operation_sc1_Distribute_iodine



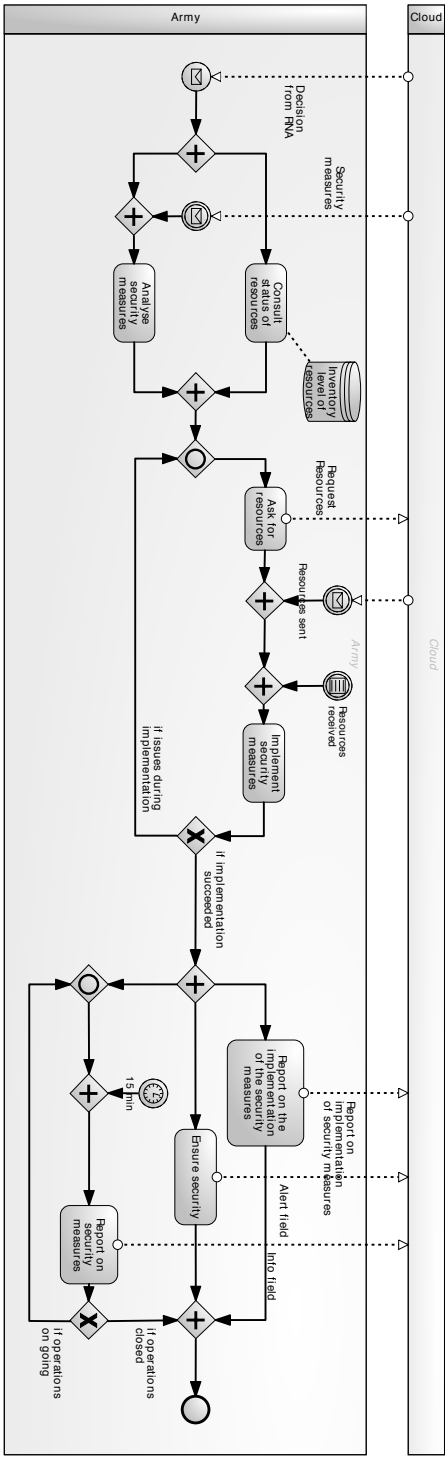
Operation_sc1_Evacuate



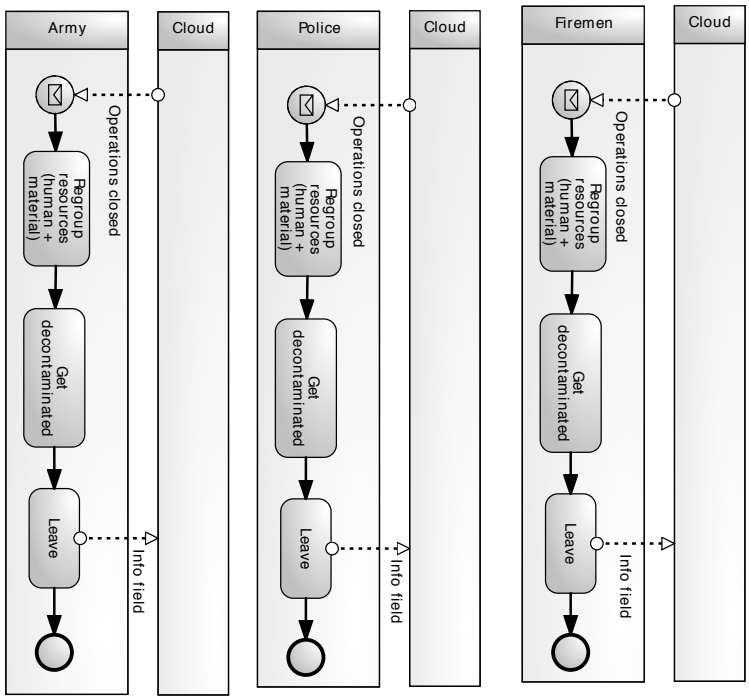
Operation_sc1_Implement_circulation_plan

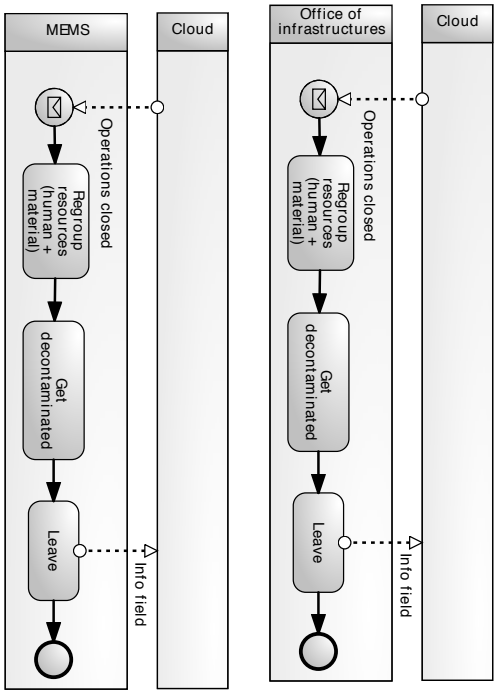


Operation_sc1_Implement_security_measures

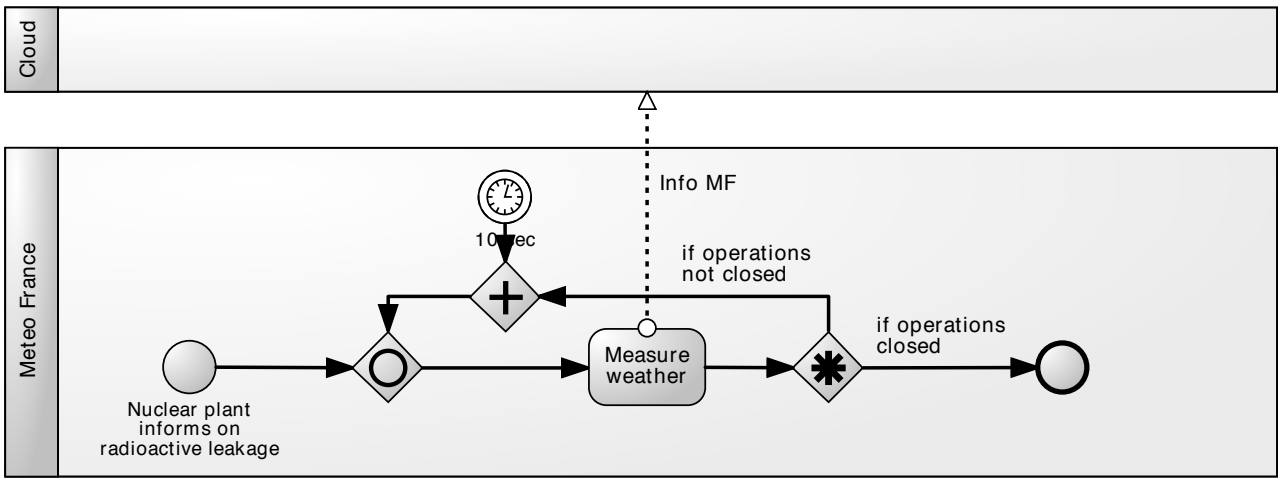


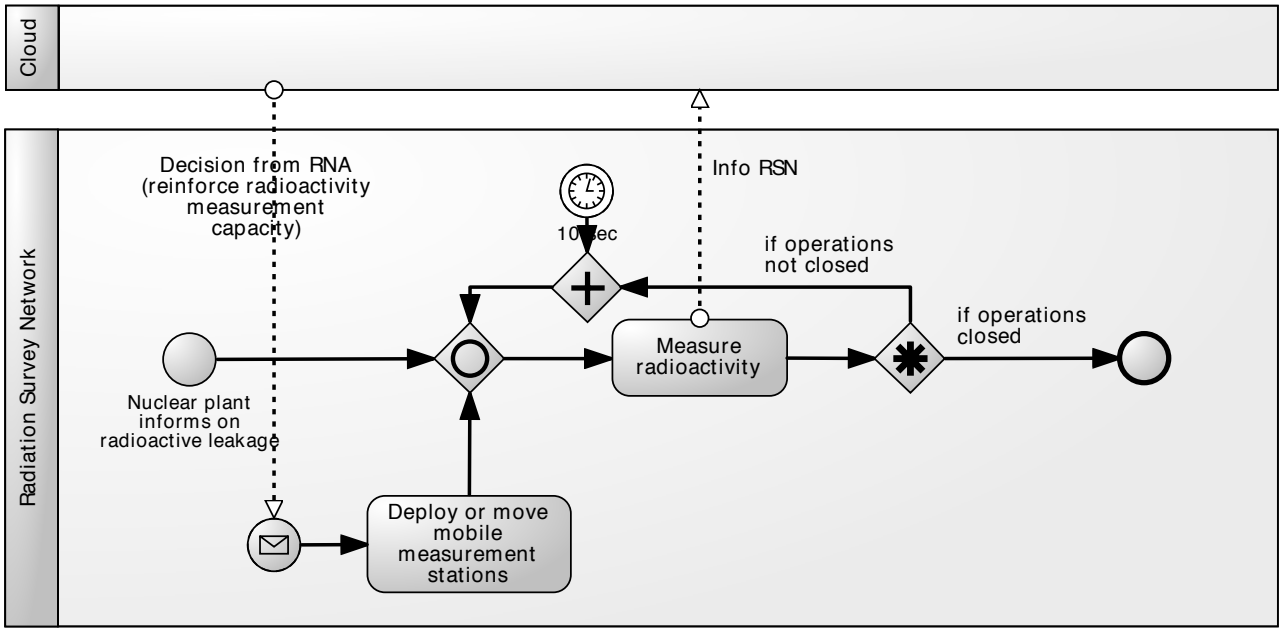
Operation_sc1_Withdraw_back



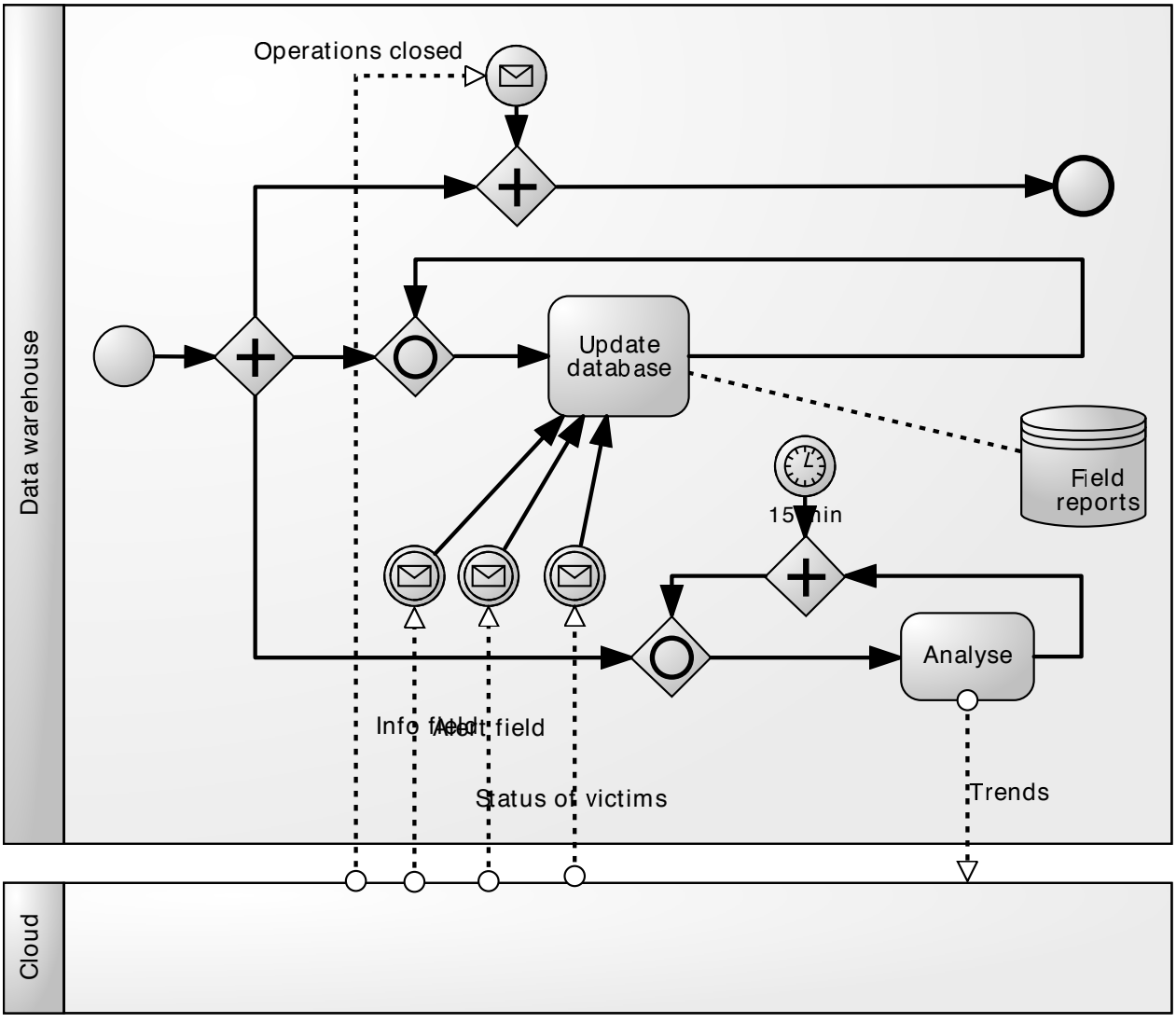


Support_sc1_Assess_situation

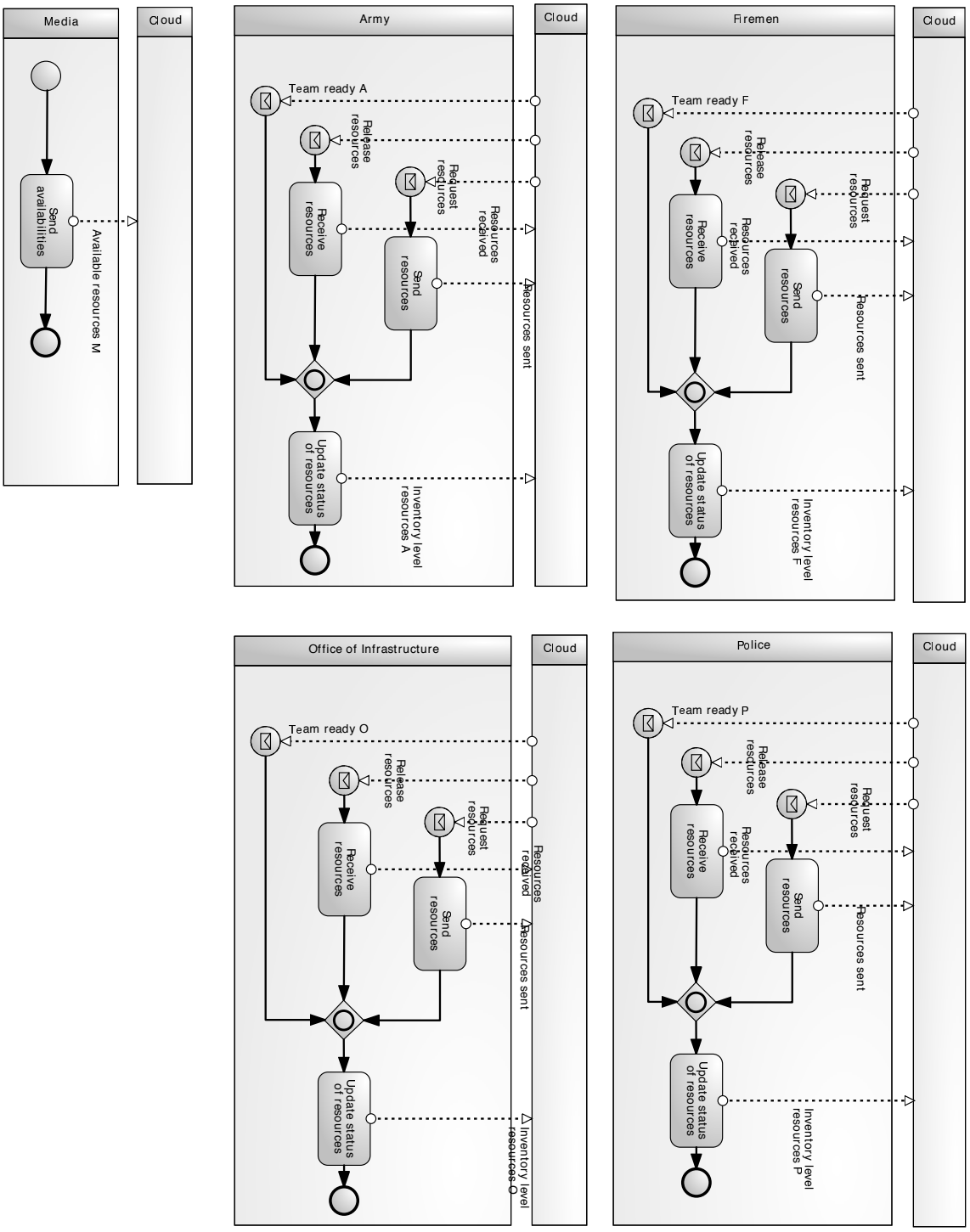




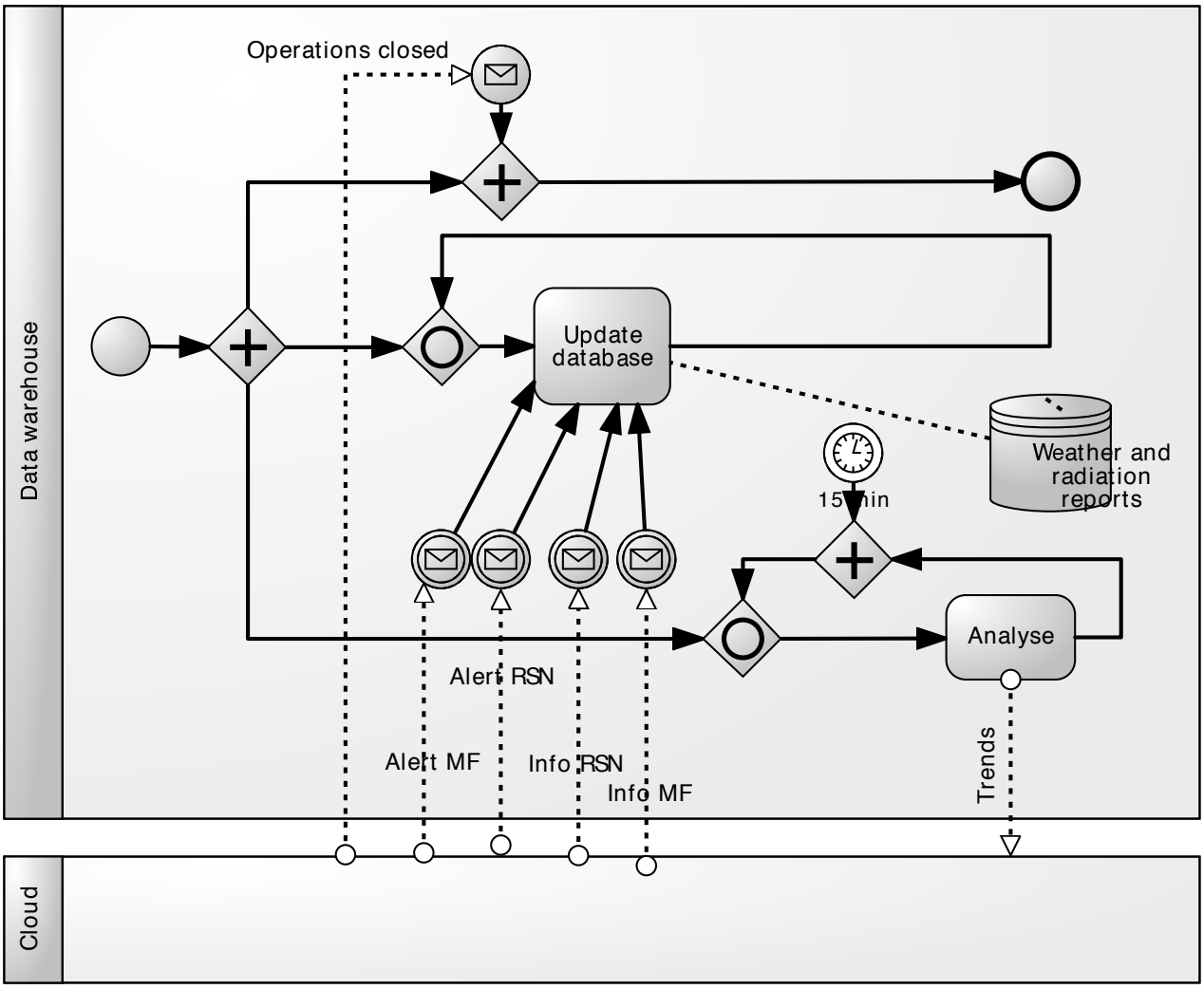
Support_sc1_Gather_reports



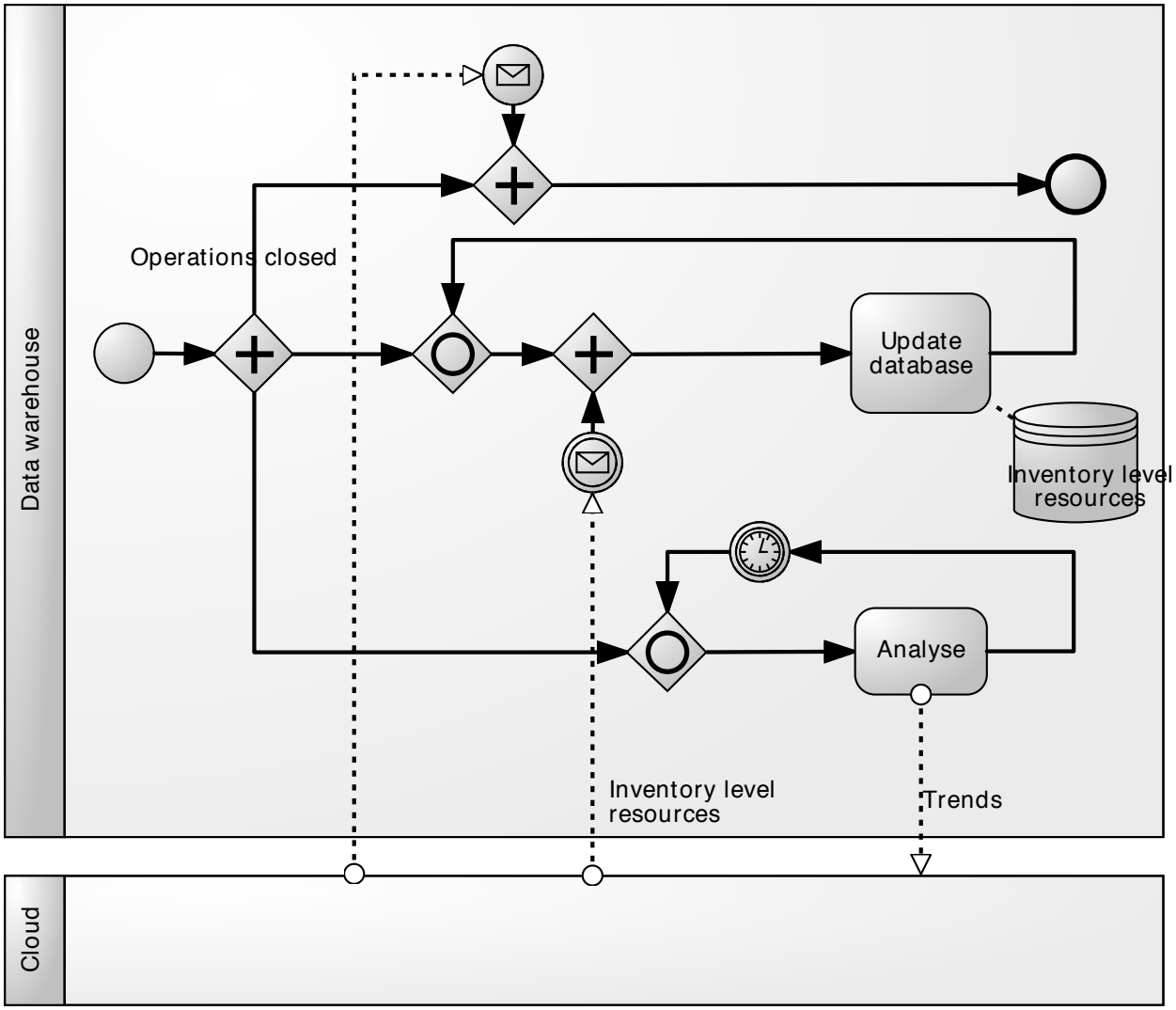
Support_sc1_Manage_resources



Support_sc1_Store_measures



Support_sc1_Store_status_resources



Annexe C

Etude comparative des différents EPL

Cette étude comparative a été menée près d'un an après le début de la thèse, en juin 2011. Elle n'a volontairement pas été mise à jour afin de préserver les informations qui nous ont menés à choisir tel EPL plutôt qu'un autre. Les principaux EPL commerciaux sont :

1. Progress Apama Event Processing Language (Java) (`http://www.progress.com/apama`),
2. TIBCO BusinessEvents Processing Language (orienté objet / Java like) (`http://www.tibco.com/products/business-optimization/complex-event-processing/businessevents/default.jsp`),
3. StreamBase StreamSQL (SQL like) (`http://www.streambase.com`),
4. Coral8 CCL (CCL = Continuous Computational Language / SQL based) (`http://www.coral8.`),
5. Aleri SQL et Aleri ML (SQL like / étend XML) (`http://www.aleri.com`),

Les EPL issus de la recherche et open sources sont :

1. Rapide (`http://complexevents.com/stanford/rapide/`),
2. FLEA,
3. Esper EQL (SQL like) (`http://esper.codehaus.org/`) (SQL like, mais utilise plutôt les vues que les tables),
4. ruleCore Markup Language (rCML) (Seiriö M. et al., 2005),
5. DEAL,
6. STREAM CQL.

Dans les sections suivantes, nous allons détailler chaque solution, suivant son appartenance au domaine commercial ou académique. Nous préciserons a minima pour chaque solution de CEP son éditeur, l'état de développement, la licence d'exploitation, l'EPL utilisé avec ce CEP.

C.1 Les solutions commerciales

C.1.1 Progress Software - Progress Apama

Apama Event Processing Platform est une suite d'outils complète basée sur le CEP. Il dispose d'un modeleur d'événements graphique. En outre, il possède un module de monitoring des opérations dirigées par les événements.

1. Editeur : Progress Software,
2. Etat de développement : en vie, et intégralement maintenu (2004 - maintenant),
3. License : license commerciale,
4. EPL : Apama event processing language,
5. Page web : Progress Apama, `http://www.progress.com/apama/index.ssp`

C.1.2 Tibco - TIBCO BusinessEvents

En ce qui concerne les règles de gestion des événements, cette solution embarque un éditeur de règles afin de déterminer la façon dont les événements se corrèlent et s'agrègent, définir les filtres. L'EPL utilisé est de type règle de production.

1. Editeur : Tibco,
2. Etat de développement : en vie, et intégralement maintenu (années 1990 - maintenant),
3. License : license commerciale,
4. EPL : éditeur de règles pour la gestion des événements,
5. Page web : TIBCO BusinessEvents, `http://www.tibco.com/`

C.1.3 Aleri - Aleri CEP 3.0

Le CEP Aleri permet de filtrer, combiner et normaliser les données entrantes et peut être utilisé pour détecter des motifs d'événements, des changements de conditions. L'EPL utilisé est une extension de XML (AleriML).

1. Editeur : Aleri
2. Etat de développement : en vie, et intégralement maintenu (2006 - maintenant),
3. License : license commerciale,
4. EPL : AleriML pour les modèles de données, le langage de script SPLASH pour le contrôle du CEP,
5. Autres : Aleri a été fusionné avec Coral8,
6. Page web : Aleri CEP 3.0, `http://www.aleri.com/`

C.1.4 Coral8 Inc. - Coral8

Le PEL de Coral8, appelé Continuous Computation Language (CCL), est basé sur SQL (CCL est donc un langage StreamingSQL).

1. Editeur : Coral8 Inc.,
2. Etat de développement : en vie, et intégralement maintenu (2005 - maintenant),
3. License : license commerciale,
4. EPL : Continuous Computation Language,
5. Autres : Coral8 a été fusionné avec Aleri,
6. Page web : `Coral8,http://www.coral8.com/`

C.1.5 Realtime Monitoring GMBH - RealTime Monitoring

RealTime Monitoring est une application CEP Java. Le moteur est le CEP RTM Analyszer, qui peut filtrer, corréler et agréger les événements entrands à travers des requêtes de type SQL.

1. Editeur : Realtime Monitoring GMBH.
2. Etat de développement : en vie, et intégralement maintenu (? - maintenant),
3. License : license commerciale,
4. EPL : Java, .Net,
5. Page web : RealTime Monitoring, `http://www.realtime-monitoring.com/`

C.1.6 IBM - IBMWebSphere Business Events

IBM's WebSphere Business Events est un outil de Business Event Processing (BEP). Il se présente sous forme d'interface graphique pour gérer le système sans avoir à coder quoi que ce soit. Il permet de définir de corrélations entre événements.

1. Editeur : IBM,
2. Etat de développement : en vie, et intégralement maintenu (? - maintenant),
3. License : license commerciale,
4. EPL : inconnu en tant que tel, tout se fait via interface graphique,
5. Page web : IBM WebSphere Business Events, `http://www-01.ibm.com/software/integration/wbe/`

C.1.7 IBM - IBM Active Middleware Technology (Amit)

IBM Active Middleware Technology est un moteur de CEP développé en Java. Il utilise la technologie de brocker de WebSpheres.

1. Editeur : IBM,
2. Etat de développement : en vie, et intégralement maintenu (? - maintenant),

3. License : license commerciale,
4. EPL : Java,
5. Page web : IBM Active Middleware Technology (Amit), `http://www.haifa.il.ibm.com/dept/services/somsebs.html`

C.1.8 Oracle - Oracle Complex Event Processing 10g

Oracle Complex Event Processing est un outil de CEP développé en Java. Oracle Complex Event Processing fournit un EPL déclaratif pour créer des ensembles de requêtes pour traiter les événements entrants et générer des ensembles de POJO pour un traitement ultérieur par des applications Java tierces.

1. Editeur : Oracle,
2. Etat de développement : en vie, et intégralement maintenu (? - maintenant),
3. License : license commerciale,
4. EPL : Java,
5. Page web : Oracle Complex Event Processing 10g, `http://www.oracle.com/technology/products/event-driven-architecture/complex-event-processing.html`

C.1.9 BEA - BEAWebLogic

BEA WebLogic est un outil Java basé sur Esper.

1. Editeur : BEA,
2. Etat de développement : les solutions BEA ont été intégrées dans les applications Oracle, il n'y a pas eu de sortie ultérieure à cela (2007 - 2008),
3. License : license commerciale,
4. EPL : langage de développement de requêtes basé sur Java,
5. Autres : fusionné avec Oracle, la base de sa technologie provient d'Esper,
6. Page web : BEA WebLogic, `http://edocs.bea.com/`

C.1.10 Agent Logic - RulePoint

Rulepoint est un serveur basé sur une plateforme de traitement des événements. Il permet la détection des événements et la réponse à ces événements, comme l'exécution de programmes ou l'alerte automatique d'utilisateurs.

1. Editeur : Agent Logic,
2. Etat de développement : en vie et maintenu jusqu'en 2009 (1999 - 2009),
3. License : Commercial license commerciale,
4. EPL : langage de programmation, assistant de création (« wizard »),
5. Page web : RulePoint, `http://www.agentlogic.com/products/rulepoint.html`

C.1.11 StreamBase Systems - StreamBase

StreamBase est une plateforme d'ESP, qui supporte un langage graphique de représentation de flux d'événements et le langage StreamSQL.

1. Editeur : StreamBase Systems,
2. Etat de développement : en vie et intégralement maintenu (1999 - maintenant),
3. License : license commerciale,
4. EPL : langage graphique et StreamSQL,
5. Page web : StreamBase. `http://www.streambase.com/`

C.1.12 Truviso Inc. - Truviso

Truviso est un outil basé sur l'approche académique TelegraphCQ, réalisée par des professeurs et des étudiants en thèse de l'Université de Berkeley en 2006. La principale approche de cet outil est de pouvoir traiter des requêtes à la volée. Le moteur TruCQ traite le langage de requête SQL (basé sur PostgreSQL) et supporte les environnements de communication publish/subscribe. Il peut recevoir et envoyer des flux de données vers d'autres instances de TruCQ et peut s'interconnecter avec d'autres applications via ses connecteurs ODBC/JDBC et ses interfaces publish/subscribe.

1. Editeur : Truviso Inc.,
2. Etat de développement : en vie et intégralement maintenu (2006 - maintenant),
3. License : license commerciale,
4. EPL : SQL-Based Continuous Query language,
5. Autres : il s'agit d'un résultat du projet Telegraph de Berkeley,
6. Page web : Truviso, `http://truviso.com/`

C.1.13 Rulecore - ruleCore CEP Server

RuleCore CEP Server est un moteur de CEP qui peut s'interconnecter avec des applications de traitement des événements tierces. Les requêtes sont exprimées via un langage déclaratif de type XML.

1. Editeur : Rulecore,
2. Etat de développement : en vie et intégralement supporté (2002 - 2009) (ruleCore est sorti en 2008),
3. License : license commerciale,
4. EPL : ruleCore Markup Language (rCML),
5. Page web : RuleCore CEP Server. `http://www.rulecore.com/`

C.1.14 Senactive - Senactive InTime

InTime est une application CEP qui analyse les événements, détecte les motifs et réagi à certains d'entre eux avec des actions automatiques ou des alertes.

1. Editeur : Senactive,
2. Etat de développement : en vie et intégralement maintenu (? - maintenant),
3. License : license commerciale,
4. EPL : outil de modélisation graphique des règles d'événements,
5. Page web : Senactive InTime, `http://www.senactive.com/index.php?id=113$&$L=1`

C.1.15 Event Zero - Event Zero

Event Zero est une plateform d'Event Processing Network (EPN). Comme les autres outils, elle est capable de capturer des événements, de les analyser, de réagir en conséquence ou d'alerter les utilisateurs. Une console permet aux utilisateurs de visualiser le flux de données, l'état du système et de visualiser les données sortantes.

1. Editeur : Event Zero,
2. Etat de développement : en vie et intégralement maintenu (2005 - maintenant),
3. License : license commerciale,
4. EPL : non identifié (construction des requêtes et autres graphiquement),
5. Page web : Event Zero, `http://www.event-zero.com/`

C.2 Les moteurs issus de la recherche académique

C.2.1 EsperTech Inc. - Esper/NEsper

EsperTech est un moteur de CEP et un moteur de ESP open-source et gratuit.

1. Editeur : EsperTech Inc.
2. Etat de développement : en vie et intégralement supporté (2008 – maintenant),
3. License : gratuit,
4. EPL : Esper EPL,
5. Page web : Esper/NEsper, `http://www.espertech.com/`

C.2.2 Middleware Systems Research Group (MSRG)/University of Toronto - PADRES

PADRES (Publish/Subscribe Applied to Distributed Resource Scheduling) est une infrastructure de management des événements conçue pour des applications de management des événements à grande échelle.

1. Editeurs : Middleware Systems Research Group (MSRG)/University of Toronto,

2. Etat de développement : en vie et intégralement supporté (2008 – maintenant),
3. License : gratuit,
4. EPL : PADRES subscription language,
5. Page web : PADRES, `http://research.msrg.utoronto.ca/Padres/`

C.2.3 CollabNet, Inc. - Intelligent Event Processor (IEP)

Intelligent Event Processor (IEP) est un moteur CEP et un moteur ESP open-source. IEP est un Service Engine JBI et fait parti de l'Open ESB community. OpenESB est un projet open-source.

1. Editeur : CollabNet, Inc.,
2. Etat de développement : en vie et intégralement supporté (2007 – maintenant),
3. License : gratuit,
4. EPL : Continuous Query Language (CQL),
5. Page web : Intelligent Event Processor (IEP), `https://open-esb.dev.java.net/IEPSE.html`

C.2.4 UC Berkeley/University of Massachusetts Amherst - Stream-based And Shared Event Processing (SASE)

Le but du projet de recherche SASE est de concevoir et développer un système de traitement RFDI robuste et efficient.

1. Editeurs : UC Berkeley/University of Massachusetts Amherst,
2. Etat de développement : il est en cours de développement depuis 2008, mais à l'heure actuelle aucune version n'est sortie,
3. EPL : SQL-like,
4. Page web : Stream-based And Shared Event Processing (SASE), `http://sase.cs.umass.edu/`

C.2.5 Cornell University - Cayuga

1. Editeur : Cornell University,
2. Etat de développement : terminé (2006 - 2008),
3. License : Free BSD,
4. EPL : Cayuga Event Language (CEL),
5. Page web : Cayuga, `http://www.cs.cornell.edu/database/cayuga`

C.2.6 Brandeis University, Brown University, and MIT. - Aurora

Ce projet a été remplacé par le projet Borealis.

1. Editeurs : Brandeis University, Brown University, and MIT,
2. Etat de développement : le projet Aurora a été remplacé par le projet Borealis (terminé ; 2003 – 2006)
3. License : gratuit,
4. EPL : Aurora Stream QUery ALgebra (SQUAL)
5. Page web : Aurora, `http://www.cs.brown.edu/research/aurora/`

C.2.7 Brandeis University, Brown University, and MIT - Borealis

Borealis est un moteur de traitement de flux distribué. Il prend des flux d'événements en entrée et les corrèle/agrège/filtre en continu de façon à produire des sorties intéressantes pour les applications.

1. Editeurs : Brandeis University, Brown University, and MIT,
2. Etat de développement : terminé (2006 - Eté 2008),
3. License : gratuit,
4. EPL : ?,
5. Page web : Borealis. `http://www.cs.brown.edu/research/borealis/public`

C.2.8 Stanford University - STREAM

1. Editeur : Stanford University,
2. Etat de développement : le projet STREAM a été officiellement clos (sorties de 2003 à 2006),
3. License : Free BSD,
4. EPL : Continuous query language (CQL),
5. Page web : STREAM, `http://www-db.stanford.edu/stream`

C.2.9 University of Marburg - PIPES

PIPES est une infrastructure permettant d'implémenter un système de gestion des flux de données (Data Stream Management System – DSMS).

1. Editeur : University of Marburg,
2. Etat de développement : première et dernière sortie publique en 2004,
3. License : LGPL,
4. EPL : Continuous Query Language (CQL),
5. Page web : PIPES, `http://dbs.mathematik.uni-marburg.de/Home/Research/Projects/PIPES/`

Annexe D

Prioriser les risques en gestion de projet

Attendu qu'il est impossible de traiter tous les risques, il faut leur donner une priorité. Pour cela, le concept de criticité est utilisé pour qualifier chacun des risques identifiés. Rappelons la formule de la criticité :

Formule 5 $C = P \times G$

Où C représente la criticité, P la probabilité et G la gravité. Des grilles d'évaluation des critères P et G permettent de calculer C. Plus C est grand, plus le mode de défaillance est critique. A titre d'exemple, voici une grille de cotation pour P et G (Table D.1). Seuls 3 niveaux sont représentés (1, 5 et 10) mais habituellement les échelles contiennent 10 niveaux (démarrant à 1, car on ne cote jamais à 0).

Note P	**Probabilité d'apparition**	**G**	**Gravité**
10	Permanent	10	Mort d'homme
5	Fréquent	5	Conséquences financières et/ou matérielles
1	Jamais	1	Pas grave

Table D.1 – Grille de cotation des risques

Il est également possible d'évaluer la criticité à partir d'une matrice de criticité (Figure D.1).

		Niveau de Gravité			
		Insignifiant	**Marginal**	**Critique**	**Catastrophique**
Fréquence	**Fréquent**	Indésirable	Inacceptable	Inacceptable	Inacceptable
	Probable	Acceptable	Indésirable	Inacceptable	Inacceptable
	Occasionnel	Acceptable	Indésirable	Indésirable	Inacceptable
	Rare	Négligeable	Acceptable	Indésirable	Indésirable
	Improbable	Négligeable	Négligeable	Acceptable	Acceptable
	Invraisemblable	Négligeable	Négligeable	Négligeable	Négligeable

FIGURE D.1 – Exemple de matrice de criticité

Annexe E

Format du fichier ServiceChoose.xml

Le fichier ServiceChoose.xml qui sert à faire le lien entre risque, activité et partenaire de la collaboration suit le formalisme décrit dans le fichier ServiceChoose.xsd suivant :

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
3    xmlns="http://www.enstimac.fr/mise/demio/crisis/servicechoose"
4    targetNamespace="http://www.enstimac.fr/mise/demio/crisis/servicechoose"
5    elementFormDefault="qualified">
6
7    <xsd:element name="solution">
8      <xsd:complexType>
9        <xsd:attribute name="id" type="xsd:ID" />
10       <xsd:attribute name="name" type="xsd:string" />
11       <xsd:attribute name="state" type="xsd:string" />
12       <xsd:attribute name="actor" type="xsd:string" />
13       <xsd:attribute name="tx_acc" type="xsd:string" />
14       <xsd:attribute name="wip" type="xsd:string" />
15     </xsd:complexType>
16   </xsd:element>
17
18   <xsd:element name="probleme">
19     <xsd:complexType>
20       <xsd:attribute name="id" type="xsd:ID" />
21       <xsd:attribute name="name" type="xsd:string" />
22       <xsd:attribute name="type" type="xsd:string" />
23       <xsd:attribute name="state" type="xsd:string" />
24       <xsd:attribute name="priority" type="xsd:string" />
25     </xsd:complexType>
26   </xsd:element>
27
28   <xsd:element name="problemes">
29     <xsd:complexType>
30       <xsd:sequence>
31         <xsd:element ref="probleme" minOccurs="0" maxOccurs="unbounded" />
```

```
32        </xsd:sequence>
33      </xsd:complexType>
34    </xsd:element>
35
36    <xsd:element name="solutions">
37      <xsd:complexType>
38        <xsd:sequence>
39          <xsd:element ref="solution" minOccurs="0" maxOccurs="unbounded" />
40        </xsd:sequence>
41      </xsd:complexType>
42    </xsd:element>
43
44    <xsd:element name="linkPBSol">
45      <xsd:complexType>
46        <xsd:attribute name="id" type="xsd:ID" />
47        <xsd:attribute name="service" type="xsd:IDREF" />
48        <xsd:attribute name="probleme" type="xsd:IDREF" />
49        <xsd:attribute name="confiance" type="xsd:string" />
50        <xsd:attribute name="state" type="xsd:string" />
51      </xsd:complexType>
52    </xsd:element>
53
54    <xsd:element name="linkPBSols">
55      <xsd:complexType>
56        <xsd:sequence>
57          <xsd:element ref="linkPBSol" minOccurs="0" maxOccurs="unbounded" />
58        </xsd:sequence>
59      </xsd:complexType>
60    </xsd:element>
61
62    <xsd:element name="linkSolSol">
63      <xsd:complexType>
64        <xsd:attribute name="id" type="xsd:ID" />
65        <xsd:attribute name="from" type="xsd:IDREF" />
66        <xsd:attribute name="to" type="xsd:IDREF" />
67        <xsd:attribute name="type" type="xsd:string" />
68      </xsd:complexType>
69    </xsd:element>
70
71    <xsd:element name="linkSolSols">
72      <xsd:complexType>
73        <xsd:sequence>
74          <xsd:element ref="linkSolSol" minOccurs="0" maxOccurs="unbounded" />
75        </xsd:sequence>
76      </xsd:complexType>
77    </xsd:element>
78
79    <xsd:element name="linkSolCond">
80      <xsd:complexType>
81        <xsd:attribute name="id" type="xsd:ID" />
82        <xsd:attribute name="service" type="xsd:IDREF" />
83        <xsd:attribute name="condition" type="xsd:string" />
```

```
84        <xsd:attribute name="type" type="xsd:string" />
85      </xsd:complexType>
86    </xsd:element>
87
88    <xsd:element name="linkSolConds">
89      <xsd:complexType>
90        <xsd:sequence>
91          <xsd:element ref="linkSolCond" minOccurs="0" maxOccurs="unbounded" />
92        </xsd:sequence>
93      </xsd:complexType>
94    </xsd:element>
95
96    <xsd:element name="serviceChoose">
97      <xsd:complexType>
98        <xsd:sequence>
99          <xsd:element ref="problemes" minOccurs="1" maxOccurs="1" />
100         <xsd:element ref="solutions" minOccurs="1" maxOccurs="1" />
101         <xsd:element ref="linkPBSols" minOccurs="1" maxOccurs="1" />
102         <xsd:element ref="linkSolSols" minOccurs="1" maxOccurs="1" />
103         <xsd:element ref="linkSolConds" minOccurs="1" maxOccurs="1" />
104       </xsd:sequence>
105       <xsd:attribute name="id" type="xsd:ID" />
106       <xsd:attribute name="name" type="xsd:string" />
107     </xsd:complexType>
108   </xsd:element>
109 </xsd:schema>
```

Code E.1 – Fichier ServiceChoose.xsd

Annexe F

Exemple de fichier WSDL : contrat d'un Web Service permettant l'envoi d'email

```
1  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2  <wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
3    xmlns:tns="http://www.enstimac.fr/mise/" xmlns:wsdl="http://schemas.xmlsoap.org/
         wsdl/"
4    xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="WebServiceSendMail"
5    xmlns:sawsdl10="http://www.w3.org/ns/sawsdl" xmlns:wsn="http://www.petalslink.com/
         wsn/service/WsnProducer"
6    xmlns:wsnc="http://www.petalslink.com/wsn/service/WsnConsumer"
7    targetNamespace="http://www.enstimac.fr/mise/">
8
9   <wsdl:types>
10    <xsd:schema targetNamespace="http://www.enstimac.fr/mise/"
11      elementFormDefault="qualified">
12      <xsd:element name="send">
13        <xsd:complexType>
14          <xsd:sequence>
15            <xsd:element name="mailTo" type="xsd:string" />
16            <xsd:element name="subject" type="xsd:string" />
17            <xsd:element name="content" type="xsd:string" />
18          </xsd:sequence>
19        </xsd:complexType>
20      </xsd:element>
21      <xsd:element name="sendResponse">
22        <xsd:complexType>
23          <xsd:sequence>
24            <xsd:element name="out" type="xsd:string" />
25          </xsd:sequence>
26        </xsd:complexType>
27      </xsd:element>
28      <xsd:element name="stop">
29        <xsd:complexType>
```

```
30            <xsd:sequence>
31              <xsd:element name="in" type="xsd:string"></xsd:element>
32            </xsd:sequence>
33          </xsd:complexType>
34        </xsd:element>
35        <xsd:element name="stopResponse">
36          <xsd:complexType>
37            <xsd:sequence>
38              <xsd:element name="out" type="xsd:string"></xsd:element>
39            </xsd:sequence>
40          </xsd:complexType>
41        </xsd:element>
42      </xsd:schema>
43    </wsdl:types>
44
45    <wsdl:message name="sendRequest">
46      <wsdl:part element="tns:send" name="parameters" />
47    </wsdl:message>
48    <wsdl:message name="sendResponse">
49      <wsdl:part element="tns:sendResponse"
50        name="parameters" />
51    </wsdl:message>
52    <wsdl:message name="stopRequest">
53      <wsdl:part name="parameters" element="tns:stop"></wsdl:part>
54    </wsdl:message>
55    <wsdl:message name="stopResponse">
56      <wsdl:part name="parameters" element="tns:stopResponse"></wsdl:part>
57    </wsdl:message>
58
59    <wsdl:portType name="WebServiceSendMail">
60      <wsdl:operation name="send">
61        <wsdl:input message="tns:sendRequest" />
62        <wsdl:output message="tns:sendResponse" />
63      </wsdl:operation>
64      <wsdl:operation name="stop">
65        <wsdl:input message="tns:stopRequest"></wsdl:input>
66        <wsdl:output message="tns:stopResponse"></wsdl:output>
67      </wsdl:operation>
68    </wsdl:portType>
69
70    <wsdl:binding name="WebServiceSendMailSOAP" type="tns:WebServiceSendMail">
71      <soap:binding style="document"
72        transport="http://schemas.xmlsoap.org/soap/http" />
73      <wsdl:operation name="send">
74        <soap:operation soapAction="http://www.enstimac.fr/mise/send" />
75        <wsdl:input><soap:body use="literal" /></wsdl:input>
76        <wsdl:output><soap:body use="literal" /></wsdl:output>
77      </wsdl:operation>
78      <wsdl:operation name="stop">
79        <soap:operation soapAction="http://www.enstimac.fr/mise/stop" />
80        <wsdl:input><soap:body use="literal" /></wsdl:input>
81        <wsdl:output><soap:body use="literal" /></wsdl:output>
```

```
82      </wsdl:operation>
83    </wsdl:binding>
84
85    <wsdl:service name="WebServiceSendMailService">
86      <wsdl:port binding="tns:WebServiceSendMailSOAP" name="WebServiceSendMailPort">
87        <soap:address location="http://localhost:7520/WebServiceSendMailPort"/>
88      </wsdl:port>
89    </wsdl:service>
90
91  </wsdl:definitions>
```

Code F.1 – Exemple de fichier WSDL d'un Service Web permettant l'envoi d'email

Annexe G

Schema XML des types d'événements en contexte de situation de crise CrisisEvents.xsd

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
3    xmlns="http://www.mines-albi.fr/nuclearcrisisevent" targetNamespace="http://www.
         mines-albi.fr/nuclearcrisisevent"
4    elementFormDefault="qualified">
5
6    <!-- Simple type elements -->
7    <xsd:element name="uid" type="xsd:anyURI" />
8    <xsd:element name="timestamp" type="xsd:dateTime" />
9    <xsd:element name="componentSEID" type="xsd:string" />
10   <xsd:element name="componentName" type="xsd:string" />
11   <xsd:element name="unit" type="xsd:string" />
12   <xsd:element name="frequency" type="xsd:string" />
13   <xsd:element name="value" type="xsd:string" />
14   <xsd:element name="origin" type="xsd:string" />
15   <xsd:element name="localisation" type="xsd:string" />
16   <xsd:element name="description" type="xsd:string" />
17   <xsd:element name="gravity" type="xsd:string" />
18   <xsd:element name="type" type="xsd:string" />
19   <xsd:element name="quantity" type="xsd:double" />
20   <xsd:element name="capability" type="xsd:string" />
21   <xsd:element name="state" type="xsd:string" />
22   <xsd:element name="provider" type="xsd:string" />
23   <xsd:element name="status" type="xsd:string" />
24   <xsd:element name="activityID" type="xsd:string" />
25   <xsd:element name="instructionMessage" type="xsd:string" />
26   <xsd:element name="resultMessage" type="xsd:string" />
27
28   <!-- Complex type elements -->
29   <xsd:element name="uncertainty">
30     <xsd:complexType>
```

```
31        <xsd:sequence>
32          <xsd:element ref="unit" minOccurs="1" maxOccurs="1" />
33          <xsd:element ref="value" minOccurs="1" maxOccurs="1" />
34          <xsd:element ref="origin" minOccurs="1" maxOccurs="1" />
35        </xsd:sequence>
36      </xsd:complexType>
37    </xsd:element>
38    <xsd:element name="situation">
39      <xsd:complexType>
40        <xsd:sequence>
41          <xsd:element ref="localisation" minOccurs="1" maxOccurs="1" />
42          <xsd:element ref="componentSEID" minOccurs="1"
43            maxOccurs="1" />
44          <xsd:element ref="componentName" minOccurs="1"
45            maxOccurs="1" />
46        </xsd:sequence>
47      </xsd:complexType>
48    </xsd:element>
49    <xsd:element name="activity">
50      <xsd:complexType>
51        <xsd:sequence>
52          <xsd:element ref="activityID" minOccurs="1" maxOccurs="1" />
53        </xsd:sequence>
54      </xsd:complexType>
55    </xsd:element>
56    <xsd:element name="resources">
57      <xsd:complexType>
58        <xsd:sequence>
59          <xsd:element ref="type" minOccurs="1" maxOccurs="1" />
60          <xsd:element ref="localisation" minOccurs="1" maxOccurs="1" />
61          <xsd:element ref="quantity" minOccurs="1" maxOccurs="1" />
62        </xsd:sequence>
63      </xsd:complexType>
64    </xsd:element>
65    <xsd:element name="consequence">
66      <xsd:complexType>
67        <xsd:sequence>
68          <xsd:element ref="activityID" minOccurs="1" maxOccurs="1" />
69        </xsd:sequence>
70      </xsd:complexType>
71    </xsd:element>
72
73    <xsd:element name="measureEvent">
74      <xsd:complexType>
75        <xsd:sequence>
76          <xsd:element ref="uid" minOccurs="1" maxOccurs="1" />
77          <xsd:element ref="timestamp" minOccurs="1" maxOccurs="1" />
78          <xsd:element ref="unit" minOccurs="1" maxOccurs="1" />
79          <xsd:element ref="frequency" minOccurs="1" maxOccurs="1" />
80          <xsd:element ref="value" minOccurs="1" maxOccurs="1" />
81          <xsd:element ref="uncertainty" minOccurs="0" maxOccurs="1" />
82          <xsd:element ref="situation" minOccurs="1" maxOccurs="1" />
```

```
83        </xsd:sequence>
84      </xsd:complexType>
85    </xsd:element>
86
87    <xsd:element name="alertEvent">
88      <xsd:complexType>
89        <xsd:sequence>
90          <xsd:element ref="uid" minOccurs="1" maxOccurs="1" />
91          <xsd:element ref="timestamp" minOccurs="1" maxOccurs="1" />
92          <xsd:element ref="description" minOccurs="1" maxOccurs="1" />
93          <xsd:element ref="gravity" minOccurs="1" maxOccurs="1" />
94          <xsd:element ref="uncertainty" minOccurs="0" maxOccurs="1" />
95          <xsd:element ref="situation" minOccurs="1" maxOccurs="1" />
96        </xsd:sequence>
97      </xsd:complexType>
98    </xsd:element>
99
100   <xsd:element name="demandEvent">
101     <xsd:complexType>
102       <xsd:sequence>
103         <xsd:element ref="uid" minOccurs="1" maxOccurs="1" />
104         <xsd:element ref="timestamp" minOccurs="1" maxOccurs="1" />
105         <xsd:element ref="capability" minOccurs="1" maxOccurs="1" />
106         <xsd:element ref="state" minOccurs="1" maxOccurs="1" />
107         <xsd:element ref="uncertainty" minOccurs="0" maxOccurs="1" />
108         <xsd:element ref="resources" minOccurs="1" maxOccurs="1" />
109       </xsd:sequence>
110     </xsd:complexType>
111   </xsd:element>
112
113   <xsd:element name="offerEvent">
114     <xsd:complexType>
115       <xsd:sequence>
116         <xsd:element ref="uid" minOccurs="1" maxOccurs="1" />
117         <xsd:element ref="timestamp" minOccurs="1" maxOccurs="1" />
118         <xsd:element ref="capability" minOccurs="1" maxOccurs="1" />
119         <xsd:element ref="provider" minOccurs="1" maxOccurs="1" />
120         <xsd:element ref="uncertainty" minOccurs="0" maxOccurs="1" />
121         <xsd:element ref="resources" minOccurs="1" maxOccurs="1" />
122       </xsd:sequence>
123     </xsd:complexType>
124   </xsd:element>
125
126   <xsd:element name="resourcesStatusEvent">
127     <xsd:complexType>
128       <xsd:sequence>
129         <xsd:element ref="uid" minOccurs="1" maxOccurs="1" />
130         <xsd:element ref="timestamp" minOccurs="1" maxOccurs="1" />
131         <xsd:element ref="state" minOccurs="1" maxOccurs="1" />
132         <xsd:element ref="provider" minOccurs="1" maxOccurs="1" />
133         <xsd:element ref="uncertainty" minOccurs="0" maxOccurs="1" />
134         <xsd:element ref="resources" minOccurs="1" maxOccurs="1" />
```

```
135       </xsd:sequence>
136     </xsd:complexType>
137   </xsd:element>
138
139   <xsd:element name="activityStatusEvent">
140     <xsd:complexType>
141       <xsd:sequence>
142         <xsd:element ref="uid" minOccurs="1" maxOccurs="1" />
143         <xsd:element ref="timestamp" minOccurs="1" maxOccurs="1" />
144         <xsd:element ref="status" minOccurs="1" maxOccurs="1" />
145         <xsd:element ref="uncertainty" minOccurs="0" maxOccurs="1" />
146         <xsd:element ref="activity" minOccurs="1" maxOccurs="1" />
147       </xsd:sequence>
148     </xsd:complexType>
149   </xsd:element>
150
151   <xsd:element name="instructionEvent">
152     <xsd:complexType>
153       <xsd:sequence>
154         <xsd:element ref="uid" minOccurs="1" maxOccurs="1" />
155         <xsd:element ref="timestamp" minOccurs="1" maxOccurs="1" />
156         <xsd:element ref="instructionMessage" minOccurs="1"
157           maxOccurs="1" />
158         <xsd:element ref="uncertainty" minOccurs="0" maxOccurs="1" />
159         <xsd:element ref="consequence" minOccurs="1" maxOccurs="1" />
160       </xsd:sequence>
161     </xsd:complexType>
162   </xsd:element>
163
164   <xsd:element name="reportEvent">
165     <xsd:complexType>
166       <xsd:sequence>
167         <xsd:element ref="uid" minOccurs="1" maxOccurs="1" />
168         <xsd:element ref="timestamp" minOccurs="1" maxOccurs="1" />
169         <xsd:element ref="resultMessage" minOccurs="1"
170           maxOccurs="1" />
171         <xsd:element ref="uncertainty" minOccurs="0" maxOccurs="1" />
172         <xsd:element ref="consequence" minOccurs="1" maxOccurs="1" />
173       </xsd:sequence>
174     </xsd:complexType>
175   </xsd:element>
176
177 </xsd:schema>
```

Code G.1 – Schema XML des types d'événements en contexte de situation de crise

Annexe H

Schema XML du modèle de la situation allégé CrisisModel.xsd

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
3    xmlns="http://www.enstimac.fr/mise/demio/crisis/crisismodel"
4    targetNamespace="http://www.enstimac.fr/mise/demio/crisis/crisismodel"
5    elementFormDefault="qualified">
6
7    <xsd:element name="civilianSociety">
8      <xsd:complexType>
9        <xsd:attribute name="id" type="xsd:ID" />
10       <xsd:attribute name="name" type="xsd:string" />
11     </xsd:complexType>
12   </xsd:element>
13
14   <xsd:element name="naturalSite">
15     <xsd:complexType>
16       <xsd:attribute name="id" type="xsd:ID" />
17       <xsd:attribute name="name" type="xsd:string" />
18     </xsd:complexType>
19   </xsd:element>
20
21   <xsd:element name="goods">
22     <xsd:complexType>
23       <xsd:attribute name="id" type="xsd:ID" />
24       <xsd:attribute name="name" type="xsd:string" />
25     </xsd:complexType>
26   </xsd:element>
27   <xsd:element name="people">
28     <xsd:complexType>
29       <xsd:attribute name="id" type="xsd:ID" />
30       <xsd:attribute name="name" type="xsd:string" />
31     </xsd:complexType>
32   </xsd:element>
33
34   <xsd:element name="risk">
```

```
35      <xsd:complexType>
36        <xsd:attribute name="id" type="xsd:ID" />
37        <xsd:attribute name="name" type="xsd:string" />
38      </xsd:complexType>
39    </xsd:element>
40
41    <xsd:element name="consequence">
42      <xsd:complexType>
43        <xsd:attribute name="id" type="xsd:ID" />
44        <xsd:attribute name="name" type="xsd:string" />
45      </xsd:complexType>
46    </xsd:element>
47
48
49    <xsd:element name="characteristic">
50      <xsd:complexType>
51        <xsd:attribute name="id" type="xsd:ID" />
52        <xsd:attribute name="name" type="xsd:string" />
53      </xsd:complexType>
54    </xsd:element>
55
56    <xsd:element name="gravityFactor">
57      <xsd:complexType>
58        <xsd:attribute name="id" type="xsd:ID" />
59        <xsd:attribute name="name" type="xsd:string" />
60      </xsd:complexType>
61    </xsd:element>
62
63    <xsd:element name="complexityFactor">
64      <xsd:complexType>
65        <xsd:attribute name="id" type="xsd:ID" />
66        <xsd:attribute name="name" type="xsd:string" />
67      </xsd:complexType>
68    </xsd:element>
69
70    <xsd:element name="link">
71      <xsd:complexType>
72        <xsd:attribute name="id" type="xsd:ID" />
73        <xsd:attribute name="type" type="xsd:string" />
74        <xsd:attribute name="from" type="xsd:IDREF" />
75        <xsd:attribute name="to" type="xsd:IDREF" />
76      </xsd:complexType>
77    </xsd:element>
78
79    <xsd:element name="links">
80      <xsd:complexType>
81        <xsd:sequence>
82          <xsd:element ref="link" minOccurs="1" maxOccurs="unbounded" />
83        </xsd:sequence>
84      </xsd:complexType>
85    </xsd:element>
86
```

```
87    <xsd:element name="crisisModel">
88      <xsd:complexType>
89        <xsd:sequence>
90          <xsd:element ref="civilianSociety" minOccurs="0" maxOccurs="unbounded" />
91          <xsd:element ref="naturalSite" minOccurs="0" maxOccurs="unbounded" />
92          <xsd:element ref="people" minOccurs="0" maxOccurs="unbounded" />
93          <xsd:element ref="goods" minOccurs="0" maxOccurs="unbounded" />
94          <xsd:element ref="risk" minOccurs="0" maxOccurs="unbounded" />
95          <xsd:element ref="consequence" minOccurs="0" maxOccurs="unbounded" />
96          <xsd:element ref="characteristic" minOccurs="0" maxOccurs="unbounded" />
97          <xsd:element ref="gravityFactor" minOccurs="0" maxOccurs="unbounded" />
98          <xsd:element ref="complexityFactor" minOccurs="0" maxOccurs="unbounded" />
99          <xsd:element ref="links" minOccurs="1" maxOccurs="1" />
100       </xsd:sequence>
101       <xsd:attribute name="id" type="xsd:ID" />
102       <xsd:attribute name="name" type="xsd:string" />
103     </xsd:complexType>
104   </xsd:element>
105
106 </xsd:schema>
```

Code H.1 – CrisisModel.xsd

Annexe I

La norme WS-Notification

WS-Notification (WS-N) est un ensemble de spécifications liées au framework WS-Resource qui autorise les interactions dirigées par les événements (notifications) entre les web services. WS-N est une recommandation développée sous le contrôle d'OASIS (Organization for the Advancement of Structured Information Standards) et est reconnue par la plupart des industriels et académiques.

WS-N est décrite dans une famille de trois normes :

- WS-BaseNotification (OASIS, 2006a), qui définit les rôles de base des applications émettrice/destinataire, et le filtrage du contenu des messages via une expression de sélecteur,
- WS-BrokeredNotification (OASIS, 2006b), qui étend la notification de base afin de définir un rôle de courtier,
- WS-Topic (OASIS, 2006c), qui définit les syntaxes de sujet pouvant être utilisées par les implémenteurs de la notification de base ou de la notification par courtier.

I.1 WS-BaseNotification

WS-BaseNotification définit les interfaces WS-Notification que les clients (consommateurs) et les serveurs (producteurs) doivent exposer.
Techniquement, la spécification de notification de base des services Web définit *des types de port WSDL* pour les applications qui souhaitent jouer le rôle d'expéditeur de notifications (NotificationProducer) ou de destinataire de notification (NotificationConsumer). Un producteur de notification est une application qui envoie des notifications d'événement dans le système, alors qu'une application consommatrice de notifications reçoit les notifications d'événement publiées par une application différente (généralement, un producteur).
Les applications qui veulent consommer les notifications d'événements de façon asynchrone doivent exposer un noeud final (endpoint) de Service Web. Celui-ci doit implémenter un port de type NotificationConsumer. La consommation synchrone des notifications d'événements a lieu quant à elle à l'aide d'un point d'extraction et ne nécessite

pas l'exposition d'un noeud final par un Service Web. Les applications localisent ensuite le NotificationProducer [1] qui produit les notifications d'événements qu'elles souhaitent recevoir et appellent l'opération d'abonnement sur cette application NotificationProducer.
L'opération d'abonnement a plusieurs paramètres qui permettent à l'application consommatrice d'indiquer le type de notifications qui l'intéressent : par exemple, en utilisant le sujet de la notification. Le ConsumerReference est un paramètre obligatoire de l'opération d'abonnement. Il permet à l'application de réception d'indiquer le noeud final (endpoint) sur lequel l'opération de notification peut être appelée lorsque des notifications d'événements correspondantes sont générées par l'application NotificationProducer.
L'application NotificationProducer est chargée des tâches liées à l'expédition de notifications d'événements :

- *Accepter l'opération d'abonnement* pour permettre aux applications de type NotificationConsumer de souscrire un abonnement au sujet qui les intéresse,
- *Conserver la liste des abonnements* en cours qu'elle a acceptés,
- *Génèrer les messages* de notification d'événements,
- *Mettre en correspondance* les notifications d'événements générées et les abonnements en cours,
- *Distribuer les notifications* d'événements pertinentes (i.e. liées aux abonnements) aux applications NotificationConsumer.

Une application NotificationProducer fonctionne avec un service SubscriptionManager pour gérer la durée de vie d'un abonnement, ce qui permet la destruction planifiée (compte tenu d'une heure de fin donnée) ou immédiate, et la suppression d'un abonnement.

I.1.1 Abonnement

La méthode *subscribe* (exposée à travers l'interface *NotificationProducer*) permet aux consommateurs d'enregistrer leurs préférences concernant les sujets. A l'aide de cette méthode, il est également possible de configurer une date d'expiration de souscription et un filtre booléen personnalisé. Un abonnement à un sujet implique un abonnement à tous les enfants de ce sujet dans l'espace de sujets.

I.1.2 Suspendre/recommencer l'abonnement

Les méthodes *PauseSubscription* et *ResumeSubscription* (exposées à travers l'interface *SubscriptionManager*) autorisent les consommateurs à suspendre temporairement leur abonnement et permettent la reprise des notifications. Les abonnements peuvent être repris avec l'une des stratégies suivantes concernant les notifications manquées :

- Envoyer une notification au prochain changement : l'effet est similaire à un désabonnement suivi d'un réabonnement,

1. Ou le NotificationBroker dans le cas où la communication n'est pas directe, cf. section I.2.

- Envoyer la dernière notification que le consommateur aurait reçu si l'abonnement n'avait pas été suspendu,
- Envoyer toutes les notifications que le consommateur a manqué depuis que l'abonnement a été suspendu.

I.1.3 Désabonnement

Comme son nom l'indique, la méthode *unsubscribe* permet au consommateur d'annuler un abonnement existant.

I.1.4 Notification

La méthode *notify* du consommateur (exposée à travers l'interface NotificationConsumer) est invoquée à chaque fois qu'une notification est disponible pour un topic auquel il s'est abonné (et que la notification a passé le filtre avec succès).

I.2 WS-BrokeredNotification

WS-BrokeredNotification définit l'interface de web services pour le NotificationBroker (courtier de notifications). Un courtier de notifications est un intermédiaire qui permet, entre autre, la publication de messages provenant d'entités qui ne sont pas elles-mêmes fournisseurs de services. Les principes généraux de fonctionnement de la publication et de l'abonnement via courtier sont présentés dans la Figure I.1.

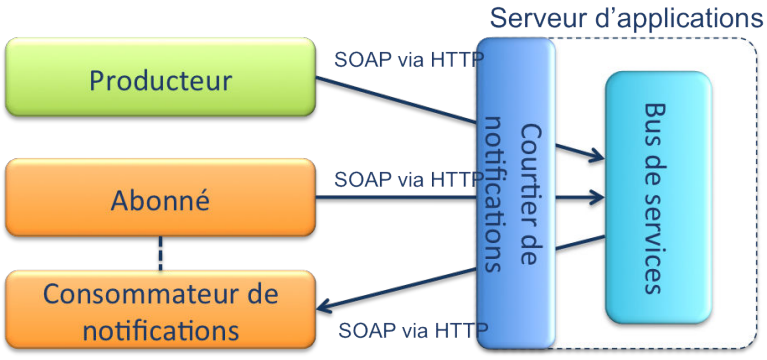


FIGURE I.1 – Principes de fonctionnement du NotificationBroker

I.2.1 Notification

Le NotificationBroker doit supporter la méthode *notify* telle qu'exposée par l'interface NotificationConsumer.

I.2.2 Abonnement

La méthode *subscribe* exposée à travers l'interface du NotificationBroker permet de router ou de produire une séquence de 0 à N notifications.

I.3 WS-Topic

WS-Topic est une spécification qui fournit une méthode pour :

- Définir et adresser les topics (ou sujets des notifications d'événements) auxquels les web services consommateurs peuvent souscrire,
- Organiser les sujets en structures complexes permettant la constitution de hiérarchies (topic trees) et de synonymes (topic aliasing).

Ces sujets sont utilisés avec les mécanismes de notification définis dans WS-BaseNotification. Dans une hiérarchie de sujets (topic tree), l'abonnement à un nœud implique l'abonnement à tous les nœuds-enfants.

I.3.1 Adressage des sujets

La spécification WS-Topics définit trois dialectes d'expression du sujet, utilisables comme expressions d'abonnement dans les messages de demande d'abonnement et dans d'autres parties du système WS-Notification. Les sujets peuvent être adressés des façons suivantes (chacune d'entre elles est un sous-ensemble de la suivante) :

- *SimpleTopic* : un QName [2] décrit le sujet (cette méthode ne peut être utilisée que pour adresser les sujets racines),
- *ConcreteTopicPath* : une liste de QNames qui décrit un chemin pour atteindre le sujet,
- *FullTopicPath* : une expression XPath [3] qui adresse les sujets, en utilisant des jokers [4].

On trouve également l'action d'« aliaser » les sujets (topic aliasing). L'aliasage de sujets fournit une façon formelle pour déclarer qu'un sujet est identique à un autre sujet ou à un groupe de sujets trouvés quelque part dans l'espace des sujets.

I.3.2 Topics ad-hoc

L'espace des sujets ad-hoc permet une définition dynamique des arbres de sujets (topic trees) sans la surcharge de la définition d'un nouvel espace de sujets.

2. La notion de QName (Qualified Name) a été introduite par les espaces de nom (namespaces) XML. Un QName définit un identifiant valide pour des éléments et des attributs et prend la forme d'une URI. Il permet de les référencer des éléments et des attributs particuliers : le QName est aux noms des éléments et des attributs ce que le nom de famille est aux prénoms.

3. Le XPath est un langage (non XML) utilisé pour localiser une portion d'un document XML. Une expression XPath est un chemin de localisation, qui dans son essence ressemble au chemin d'un système de fichiers.

4. Un joker est un caractère spécial utilisé dans un programme pour tout inclure dans une sélection. En XPath, le joker est le caractère étoile *****.

Annexe J

Correspondance entre les événements et les éléments du modèle terrain

Type d'événement — **Alert**

		Modèle impacté : fieldModel.xml			
		Élements du modèle (et leurs caractéristiques) concernés			
		Risk	**Consequence**	**G/P/C/N**	**link**
Timestamp					
UniqueID					
Localisation	=	localisation	localisation		
ComponentSEID	=	id	id		from *ou* to
ComponentName	=	name	name		
Description	=	description	description		
Gravity	=	gravity	gravity		
Type	=	*nom_element*	*nom_element*		
ImpactedComponentSEID	=			id	from *ou* to

Remarques

- **G/P/C/N = Goods / People / Civilian Society / Natural Site**
- **Effet :** ajout, mise à jour ou suppression de consequence / risk
- La valeur de *Type* portée par l'événement *Alert* indique quel type d'élément *(Risk* ou *Consequence)* doit être créé dans le modèle (si l'élément n'existe pas déjà).
- Si l'élément *Risk* (ou *Consequence)* est créé, alors il faut également créer *le lien* qui le relie à un des éléments de type *People, Goods, Civilian Society* ou *Natural Site.*
- Ce lien prend pour valeur d'attributs *from et to* les valeurs des identifiants du *Risk (Consequence)* créé et de l'élément impacté.
- Si l'événement concerne un Risk/Consequence existant dans le modèle, et que la valeur de ImpactedComponentSEID est vide, alors cela signifie que le Risk (Consequence) est supprimé du modèle (et de facto tous les liens concernés par ce Risk/consequences sont aussi supprimés).

Type d'événement — **Alert**

		Modèle impacté : fieldModel.xml			
		Élements du modèle (et leurs caractéristiques) concernés			
		GravityFactor	**ComplexityFactor**	**R/C/G/P/C/N**	**link**
Timestamp					
UniqueID					
Localisation					
ComponentSEID	=	id	id		from *ou* to
ComponentName	=	name	name		
Description	=	description	description		
Gravity					
Type	=	*nom_element*	*nom_element*		
ImpactedComponentSEID	=			id	from *ou* to

Remarques

- **R/C/G/P/C/N = Risk / Consequence / Goods / People / Civilian Society / Natural Site**
- **Effet :** ajout, mise à jour ou suppression de gravity/complexity factor
- La valeur de *Type* portée par l'événement *Alert* indique quel type d'élément *(Risk* ou *Consequence)* doit être créé dans le modèle (si l'élément n'existe pas déjà)
- Si l'élément *GravityFactor* (ou *ComplexityFactor* est créé, alors il faut également créer *le lien* qui le relie à un des éléments de type *People, Goods, Civilian Society* ou *Natural Site.*
- Ce lien prend pour valeur d'attributs *from et to* les valeurs des identifiants du *Risk (Consequence)* créé et de l'élément impacté.
- la valeur de ImpactedComponentSEID est vide, alors cela signifie que le Risk (Consequence) est supprimé du modèle (et de facto tous les liens concernés par ce Risk/consequences sont aussi supprimés).

Type d'événement

OfferResource

		Modèle impacté : fieldModel.xml		
		Élements du modèle (et leurs caractéristiques) concernés		
		Partner	**G/P/C/N**	**Service**
Timestamp				
UniqueID				
Localisation	=	description	localisation	
Type	=	name	name	
Quantity		*0*		*0* id
Capability	=			*service*
Provider	=	id	id	

Remarques

- **G/P/C/N = Goods / People / Civilian Society / Natural Site**
- **Effet :** ajout, mise à jour ou suppression de goods/people/civilian society/natural site, partner ***ou*** ajout de partner et son service attaché
- La valeur de *Type* portée par l'événement *Resource* indique quel type d'élément *(Partner* ou *G/P/C/N* doit être créé dans le modèle (si l'élément n'existe pas déjà).
- Si la valeur de *Quantity* portée par l'événement *Resource* est égale à ***0***, cela signifie que l'élément concerné (côté modèle) est *supprimé.*
- Si l'événement concerne un Risk/Consequence existant dans le modèle, et que la valeur de ImpactedComponentSEID est vide, alors cela signifie que le Risk (Consequence) est supprimé du modèle (et de facto tous les liens concernés par ce Risk/consequences sont aussi supprimés).

Resource_status

		Modèle impacté : fieldModel.xml		
		Élements du modèle (et leurs caractéristiques) concernés		
		Service	**Partner**	**link**
Timestamp				
UniqueID				
Localisation				
Type	=	id		from *ou* to
Quantity		*0*		
State	=	state		
Provider	=		id	from *ou* to

Remarques

- **Effet :** ajout, mise à jour ou suppression de service
- La valeur de *Type* portée par l'événement *Resource* indique quel type d'élément *(Partner* ou *G/P/C/N* doit être créé dans le modèle (si l'élément n'existe pas déjà).
- Si la valeur de *Quantity* portée par l'événement *Resource_status* est égale à ***0***, cela signifie que le *Service* (côté modèle) est *supprimé.*
- Si l'élément *Service* est créé, alors il faut également créer *le lien* qui le relie à son *Partner.*
- Ce lien prend pour valeur d'attributs *from et to* les valeurs des identifiants du *Service* créé et du *Partner* concerné.

Annexe K

Feuille de style pour la seconde transformations XSL

Feuille model2model.xsl utilisée pour formater le contenu du modèle de la situation en vue de sa comparaison (étape de détection).

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3      version="1.0"
4      >
5      <xsl:output indent="yes" method="xml"/>
6
7      <xsl:template match="/">
8          <xsl:element name="situationModel">
9              <xsl:if test="crisisModel/civilianSociety">
10                 <xsl:call-template name="civilianSociety"/>
11             </xsl:if>
12
13             <xsl:if test="crisisModel/naturalSite">
14                 <xsl:call-template name="naturalSite"/>
15             </xsl:if>
16
17             <xsl:if test="crisisModel/people">
18                 <xsl:call-template name="people"/>
19             </xsl:if>
20             <xsl:if test="crisisModel/goods">
21                 <xsl:call-template name="goods"/>
22             </xsl:if>
23             <xsl:if test="crisisModel/risk">
24                 <xsl:call-template name="risk"/>
25             </xsl:if>
26             <xsl:if test="crisisModel/consequence">
27                 <xsl:call-template name="consequence"/>
28             </xsl:if>
29
30             <xsl:if test="crisisModel/characteristic">
31                 <xsl:call-template name="characteristic"/>
```

```
32              </xsl:if>
33              <xsl:if test="crisisModel/gravityFactor">
34                  <xsl:call-template name="gravityFactor"/>
35              </xsl:if>
36              <xsl:if test="crisisModel/complexityFactor">
37                  <xsl:call-template name="complexityFactor"/>
38              </xsl:if>
39
40              <xsl:if test="crisisModel/links">
41                  <xsl:call-template name="links"/>
42              </xsl:if>
43
44          </xsl:element>
45      </xsl:template>
46
47      <xsl:template name="rootElement">
48          <xsl:if test="/crisisModel">
49              <xsl:element name="situationModel">er</xsl:element>
50          </xsl:if>
51
52
53      </xsl:template>
54
55      <xsl:template name="civilianSociety">
56          <civilianSociety>
57              <xsl:for-each select="crisisModel/civilianSociety">
58                  <xsl:copy-of select="."/>
59              </xsl:for-each>
60          </civilianSociety>
61      </xsl:template>
62
63      <xsl:template name="naturalSite">
64          <naturalSites>
65              <xsl:for-each select="crisisModel/naturalSite">
66                  <xsl:copy-of select="."/>
67              </xsl:for-each>
68          </naturalSites>
69      </xsl:template>
70
71      <xsl:template name="people">
72          <people>
73              <xsl:for-each select="crisisModel/people">
74                  <xsl:copy-of select="."/>
75              </xsl:for-each>
76          </people>
77      </xsl:template>
78
79      <xsl:template name="goods">
80          <goods>
81              <xsl:for-each select="crisisModel/goods">
82                  <xsl:copy-of select="."/>
83              </xsl:for-each>
```

```
84          </goods>
85      </xsl:template>
86
87      <xsl:template name="risk">
88          <risks>
89              <xsl:for-each select="crisisModel/risk">
90                  <xsl:copy-of select="."/>
91              </xsl:for-each>
92          </risks>
93      </xsl:template>
94
95      <xsl:template name="consequence">
96          <consequences>
97              <xsl:for-each select="crisisModel/consequence">
98                  <xsl:copy-of select="."/>
99              </xsl:for-each>
100         </consequences>
101     </xsl:template>
102
103     <xsl:template name="characteristic">
104         <characteristics>
105             <xsl:for-each select="crisisModel/characteristic">
106                 <xsl:copy-of select="."/>
107             </xsl:for-each>
108         </characteristics>
109     </xsl:template>
110
111     <xsl:template name="gravityFactor">
112         <gravityFactors>
113             <xsl:for-each select="crisisModel/gravityFactor">
114                 <xsl:copy-of select="."/>
115             </xsl:for-each>
116         </gravityFactors>
117     </xsl:template>
118
119     <xsl:template name="complexityFactor">
120         <complexityFactors>
121             <xsl:for-each select="crisisModel/complexityFactor">
122                 <xsl:copy-of select="."/>
123             </xsl:for-each>
124         </complexityFactors>
125     </xsl:template>
126
127     <xsl:template name="links">
128         <xsl:copy-of select="crisisModel/links"/>
129     </xsl:template>
130
131 </xsl:stylesheet>
```

Code K.1 – Fichier XSL formattant le contenu du modèle de la situation

Annexe L

Fichier de configuration d'Esper : esper.mise.cfg.xml

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <esper-configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3    xmlns="http://www.espertech.com/schema/esper"
4    xsi:schemaLocation="
5  http://www.espertech.com/schema/esper http://www.espertech.com/schema/esper/esper-
       configuration-3.0.xsd">
6    <!-- Event type : Measure -->
7    <event-type name="measureEvent">
8      <xml-dom root-element-name="Notify" schema-resource="wsdl/eventservice/standard/
           wsn/b-2.xsd"
9        default-namespace="http://docs.oasis-open.org/wsn/b-2">
10       <namespace-prefix prefix="ns18"
11         namespace="http://docs.oasis-open.org/wsn/b-2" />
12       <namespace-prefix prefix="ns2"
13         namespace="http://www.mines-albi.fr/nuclearcrisisevent" />
14       <xpath-property property-name="measureId"
15         xpath="ns18:NotificationMessage/ns18:Message/ns2:measureEvent/ns2:uid"
16         type="string" />
17       <xpath-property property-name="measureTimestamp"
18         xpath="ns18:NotificationMessage/ns18:Message/ns2:measureEvent/ns2:timestamp"
19         type="string" />
20       <xpath-property property-name="measureUnit"
21         xpath="ns18:NotificationMessage/ns18:Message/ns2:measureEvent/ns2:unit"
22         type="string" />
23       <xpath-property property-name="measureFrequency"
24         xpath="ns18:NotificationMessage/ns18:Message/ns2:measureEvent/ns2:frequency"
25         type="string" />
26       <xpath-property property-name="measureValue"
27         xpath="ns18:NotificationMessage/ns18:Message/ns2:measureEvent/ns2:value"
28         type="number" />
29       <xpath-property property-name="uncertaintyUnit"
30         xpath="ns18:NotificationMessage/ns18:Message/ns2:measureEvent/ns2:uncertainty
               /ns2:unit"
31         type="string" />
```

```
32        <xpath-property property-name="uncertaintyValue"
33          xpath="ns18:NotificationMessage/ns18:Message/ns2:measureEvent/ns2:uncertainty
                /ns2:value"
34          type="number" />
35        <xpath-property property-name="uncertaintyOrigin"
36          xpath="ns18:NotificationMessage/ns18:Message/ns2:measureEvent/ns2:uncertainty
                /ns2:origin"
37          type="string" />
38        <xpath-property property-name="measureLocation"
39          xpath="ns18:NotificationMessage/ns18:Message/ns2:measureEvent/ns2:situation/
                ns2:localisation"
40          type="string" />
41        <xpath-property property-name="measureSEID"
42          xpath="ns18:NotificationMessage/ns18:Message/ns2:measureEvent/ns2:situation/
                ns2:componentSEID"
43          type="string" />
44        <xpath-property property-name="measureName"
45          xpath="ns18:NotificationMessage/ns18:Message/ns2:measureEvent/ns2:situation/
                ns2:componentName"
46          type="string" />
47        <xpath-property property-name="measureType"
48          xpath="ns18:NotificationMessage/ns18:Message/ns2:measureEvent/ns2:situation/
                ns2:type"
49          type="string" />
50        <xpath-property property-name="measureImpactedSEID"
51          xpath="ns18:NotificationMessage/ns18:Message/ns2:measureEvent/ns2:situation/
                ns2:impactedComponentSEID"
52          type="string" />
53      </xml-dom>
54    </event-type>
55    <!-- Event type : Alert -->
56    <event-type name="alertEvent">
57      <xml-dom root-element-name="Notify" schema-resource="wsdl/eventservice/standard/
            wsn/b-2.xsd"
58        default-namespace="http://docs.oasis-open.org/wsn/b-2">
59        <namespace-prefix prefix="ns18"
60          namespace="http://docs.oasis-open.org/wsn/b-2" />
61        <namespace-prefix prefix="ns2"
62          namespace="http://www.mines-albi.fr/nuclearcrisisevent" />
63        <xpath-property property-name="alertId"
64          xpath="ns18:NotificationMessage/ns18:Message/ns2:alertEvent/ns2:uid"
65          type="string" />
66        <xpath-property property-name="alertTimestamp"
67          xpath="ns18:NotificationMessage/ns18:Message/ns2:alertEvent/ns2:timestamp"
68          type="string" />
69        <xpath-property property-name="alertDescription"
70          xpath="ns18:NotificationMessage/ns18:Message/ns2:alertEvent/ns2:description"
71          type="string" />
72        <xpath-property property-name="alertGravity"
73          xpath="ns18:NotificationMessage/ns18:Message/ns2:alertEvent/ns2:gravity"
74          type="string" />
75        <xpath-property property-name="uncertaintyUnit"
```

```
76          xpath="ns18:NotificationMessage/ns18:Message/ns2:alertEvent/ns2:uncertainty/
                ns2:unit"
77          type="string" />
78        <xpath-property property-name="uncertaintyValue"
79          xpath="ns18:NotificationMessage/ns18:Message/ns2:alertEvent/ns2:uncertainty/
                ns2:value"
80          type="number" />
81        <xpath-property property-name="uncertaintyOrigin"
82          xpath="ns18:NotificationMessage/ns18:Message/ns2:alertEvent/ns2:uncertainty/
                ns2:origin"
83          type="string" />
84        <xpath-property property-name="alertLocation"
85          xpath="ns18:NotificationMessage/ns18:Message/ns2:alertEvent/ns2:situation/ns2
                :localisation"
86          type="string" />
87        <xpath-property property-name="alertSEID"
88          xpath="ns18:NotificationMessage/ns18:Message/ns2:alertEvent/ns2:situation/ns2
                :componentSEID"
89          type="string" />
90        <xpath-property property-name="alertName"
91          xpath="ns18:NotificationMessage/ns18:Message/ns2:alertEvent/ns2:situation/ns2
                :componentName"
92          type="string" />
93        <xpath-property property-name="alertType"
94          xpath="ns18:NotificationMessage/ns18:Message/ns2:alertEvent/ns2:situation/ns2
                :type"
95          type="string" />
96        <xpath-property property-name="alertImpactedSEID"
97          xpath="ns18:NotificationMessage/ns18:Message/ns2:alertEvent/ns2:situation/ns2
                :impactedComponentSEID"
98          type="string" />
99      </xml-dom>
100   </event-type>
101   <!-- Event type : Demand -->
102   <event-type name="demandEvent">
103     <xml-dom root-element-name="Notify" schema-resource="wsdl/eventservice/standard/
            wsn/b-2.xsd"
104       default-namespace="http://docs.oasis-open.org/wsn/b-2">
105       <namespace-prefix prefix="ns18"
106         namespace="http://docs.oasis-open.org/wsn/b-2" />
107       <namespace-prefix prefix="ns2"
108         namespace="http://www.mines-albi.fr/nuclearcrisisevent" />
109       <xpath-property property-name="demandId"
110         xpath="ns18:NotificationMessage/ns18:Message/ns2:demandEvent/ns2:uid"
111         type="string" />
112       <xpath-property property-name="demandTimestamp"
113         xpath="ns18:NotificationMessage/ns18:Message/ns2:demandEvent/ns2:timestamp"
114         type="string" />
115       <xpath-property property-name="demandCapability"
116         xpath="ns18:NotificationMessage/ns18:Message/ns2:demandEvent/ns2:capability"
117         type="string" />
118       <xpath-property property-name="demandState"
```

```
119         xpath="ns18:NotificationMessage/ns18:Message/ns2:demandEvent/ns2:state"
120         type="string" />
121       <xpath-property property-name="uncertaintyUnit"
122         xpath="ns18:NotificationMessage/ns18:Message/ns2:demandEvent/ns2:uncertainty/
              ns2:unit"
123         type="string" />
124       <xpath-property property-name="uncertaintyValue"
125         xpath="ns18:NotificationMessage/ns18:Message/ns2:demandEvent/ns2:uncertainty/
              ns2:value"
126         type="number" />
127       <xpath-property property-name="uncertaintyOrigin"
128         xpath="ns18:NotificationMessage/ns18:Message/ns2:demandEvent/ns2:uncertainty/
              ns2:origin"
129         type="string" />
130       <xpath-property property-name="resourcesType"
131         xpath="ns18:NotificationMessage/ns18:Message/ns2:demandEvent/ns2:resources/
              ns2:type"
132         type="string" />
133       <xpath-property property-name="resourcesLocalisation"
134         xpath="ns18:NotificationMessage/ns18:Message/ns2:demandEvent/ns2:resources/
              ns2:localisation"
135         type="string" />
136       <xpath-property property-name="resourcesQuantity"
137         xpath="ns18:NotificationMessage/ns18:Message/ns2:demandEvent/ns2:resources/
              ns2:quantity"
138         type="number" />
139     </xml-dom>
140   </event-type>
141   <!-- Event type : Offer -->
142   <event-type name="offerEvent">
143     <xml-dom root-element-name="Notify" schema-resource="wsdl/eventservice/standard/
          wsn/b-2.xsd"
144       default-namespace="http://docs.oasis-open.org/wsn/b-2">
145       <namespace-prefix prefix="ns18"
146         namespace="http://docs.oasis-open.org/wsn/b-2" />
147       <namespace-prefix prefix="ns2"
148         namespace="http://www.mines-albi.fr/nuclearcrisisevent" />
149       <xpath-property property-name="offerId"
150         xpath="ns18:NotificationMessage/ns18:Message/ns2:offerEvent/ns2:uid"
151         type="string" />
152       <xpath-property property-name="offerTimestamp"
153         xpath="ns18:NotificationMessage/ns18:Message/ns2:offerEvent/ns2:timestamp"
154         type="string" />
155       <xpath-property property-name="offerCapability"
156         xpath="ns18:NotificationMessage/ns18:Message/ns2:offerEvent/ns2:capability"
157         type="string" />
158       <xpath-property property-name="offerProvider"
159         xpath="ns18:NotificationMessage/ns18:Message/ns2:offerEvent/ns2:provider"
160         type="string" />
161       <xpath-property property-name="uncertaintyUnit"
162         xpath="ns18:NotificationMessage/ns18:Message/ns2:offerEvent/ns2:uncertainty/
              ns2:unit"
```

```
163         type="string" />
164       <xpath-property property-name="uncertaintyValue"
165         xpath="ns18:NotificationMessage/ns18:Message/ns2:offerEvent/ns2:uncertainty/
              ns2:value"
166         type="number" />
167       <xpath-property property-name="uncertaintyOrigin"
168         xpath="ns18:NotificationMessage/ns18:Message/ns2:offerEvent/ns2:uncertainty/
              ns2:origin"
169         type="string" />
170       <xpath-property property-name="resourcesType"
171         xpath="ns18:NotificationMessage/ns18:Message/ns2:offerEvent/ns2:resources/ns2
              :type"
172         type="string" />
173       <xpath-property property-name="resourcesLocalisation"
174         xpath="ns18:NotificationMessage/ns18:Message/ns2:offerEvent/ns2:resources/ns2
              :localisation"
175         type="string" />
176       <xpath-property property-name="resourcesQuantity"
177         xpath="ns18:NotificationMessage/ns18:Message/ns2:offerEvent/ns2:resources/ns2
              :quantity"
178         type="number" />
179     </xml-dom>
180   </event-type>
181   <!-- Event type : Resources Status -->
182   <event-type name="resourcesStatusEvent">
183     <xml-dom root-element-name="Notify" schema-resource="wsdl/eventservice/standard/
          wsn/b-2.xsd"
184       default-namespace="http://docs.oasis-open.org/wsn/b-2">
185       <namespace-prefix prefix="ns18"
186         namespace="http://docs.oasis-open.org/wsn/b-2" />
187       <namespace-prefix prefix="ns2"
188         namespace="http://www.mines-albi.fr/nuclearcrisisevent" />
189       <xpath-property property-name="resourcesStatusId"
190         xpath="ns18:NotificationMessage/ns18:Message/ns2:resourcesStatusEvent/ns2:uid
              "
191         type="string" />
192       <xpath-property property-name="resourcesStatusTimestamp"
193         xpath="ns18:NotificationMessage/ns18:Message/ns2:resourcesStatusEvent/ns2:
              timestamp"
194         type="string" />
195       <xpath-property property-name="resourcesState"
196         xpath="ns18:NotificationMessage/ns18:Message/ns2:resourcesStatusEvent/ns2:
              state"
197         type="string" />
198       <xpath-property property-name="resourcesStatusProvider"
199         xpath="ns18:NotificationMessage/ns18:Message/ns2:resourcesStatusEvent/ns2:
              provider"
200         type="string" />
201       <xpath-property property-name="uncertaintyUnit"
202         xpath="ns18:NotificationMessage/ns18:Message/ns2:resourcesStatusEvent/ns2:
              uncertainty/ns2:unit"
203         type="string" />
```

```
204       <xpath-property property-name="uncertaintyValue"
205         xpath="ns18:NotificationMessage/ns18:Message/ns2:resourcesStatusEvent/ns2:
              uncertainty/ns2:value"
206         type="number" />
207       <xpath-property property-name="uncertaintyOrigin"
208         xpath="ns18:NotificationMessage/ns18:Message/ns2:resourcesStatusEvent/ns2:
              uncertainty/ns2:origin"
209         type="string" />
210       <xpath-property property-name="resourcesType"
211         xpath="ns18:NotificationMessage/ns18:Message/ns2:resourcesStatusEvent/ns2:
              resources/ns2:type"
212         type="string" />
213       <xpath-property property-name="resourcesLocalisation"
214         xpath="ns18:NotificationMessage/ns18:Message/ns2:resourcesStatusEvent/ns2:
              resources/ns2:localisation"
215         type="string" />
216       <xpath-property property-name="resourcesQuantity"
217         xpath="ns18:NotificationMessage/ns18:Message/ns2:resourcesStatusEvent/ns2:
              resources/ns2:quantity"
218         type="number" />
219     </xml-dom>
220   </event-type>
221   <!-- Event type : ActivityStatus -->
222   <event-type name="activityStatusEvent">
223     <xml-dom root-element-name="Notify" schema-resource="wsdl/eventservice/standard/
          wsn/b-2.xsd"
224       default-namespace="http://docs.oasis-open.org/wsn/b-2">
225       <namespace-prefix prefix="ns18"
226         namespace="http://docs.oasis-open.org/wsn/b-2" />
227       <namespace-prefix prefix="ns2"
228         namespace="http://www.mines-albi.fr/nuclearcrisisevent" />
229       <xpath-property property-name="activityStatusId"
230         xpath="ns18:NotificationMessage/ns18:Message/ns2:activityStatusEvent/ns2:uid"
231         type="string" />
232       <xpath-property property-name="resourcesStatusTimestamp"
233         xpath="ns18:NotificationMessage/ns18:Message/ns2:activityStatusEvent/ns2:
              timestamp"
234         type="string" />
235       <xpath-property property-name="resourcesStatus"
236         xpath="ns18:NotificationMessage/ns18:Message/ns2:activityStatusEvent/ns2:
              status"
237         type="string" />
238       <xpath-property property-name="uncertaintyUnit"
239         xpath="ns18:NotificationMessage/ns18:Message/ns2:activityStatusEvent/ns2:
              uncertainty/ns2:unit"
240         type="string" />
241       <xpath-property property-name="uncertaintyValue"
242         xpath="ns18:NotificationMessage/ns18:Message/ns2:activityStatusEvent/ns2:
              uncertainty/ns2:value"
243         type="number" />
244       <xpath-property property-name="uncertaintyOrigin"
```

```
245         xpath="ns18:NotificationMessage/ns18:Message/ns2:activityStatusEvent/ns2:
                uncertainty/ns2:origin"
246         type="string" />
247       <xpath-property property-name="activityId"
248         xpath="ns18:NotificationMessage/ns18:Message/ns2:activityStatusEvent/ns2:
                activity/ns2:activityID"
249         type="string" />
250     </xml-dom>
251   </event-type>
252   <!-- Event type : Instruction -->
253   <event-type name="instructionEvent">
254     <xml-dom root-element-name="Notify" schema-resource="wsdl/eventservice/standard/
            wsn/b-2.xsd"
255       default-namespace="http://docs.oasis-open.org/wsn/b-2">
256       <namespace-prefix prefix="ns18"
257         namespace="http://docs.oasis-open.org/wsn/b-2" />
258       <namespace-prefix prefix="ns2"
259         namespace="http://www.mines-albi.fr/nuclearcrisisevent" />
260       <xpath-property property-name="instructionId"
261         xpath="ns18:NotificationMessage/ns18:Message/ns2:instructionEvent/ns2:uid"
262         type="string" />
263       <xpath-property property-name="instructionTimestamp"
264         xpath="ns18:NotificationMessage/ns18:Message/ns2:instructionEvent/ns2:
                timestamp"
265         type="string" />
266       <xpath-property property-name="instructionMessage"
267         xpath="ns18:NotificationMessage/ns18:Message/ns2:instructionEvent/ns2:
                instructionMessage"
268         type="string" />
269       <xpath-property property-name="uncertaintyUnit"
270         xpath="ns18:NotificationMessage/ns18:Message/ns2:instructionEvent/ns2:
                uncertainty/ns2:unit"
271         type="string" />
272       <xpath-property property-name="uncertaintyValue"
273         xpath="ns18:NotificationMessage/ns18:Message/ns2:instructionEvent/ns2:
                uncertainty/ns2:value"
274         type="number" />
275       <xpath-property property-name="uncertaintyOrigin"
276         xpath="ns18:NotificationMessage/ns18:Message/ns2:instructionEvent/ns2:
                uncertainty/ns2:origin"
277         type="string" />
278       <xpath-property property-name="activityId"
279         xpath="ns18:NotificationMessage/ns18:Message/ns2:instructionEvent/ns2:
                consequence/ns2:activityID"
280         type="string" />
281     </xml-dom>
282   </event-type>
283   <!-- Event type : Report -->
284   <event-type name="reportEvent">
285     <xml-dom root-element-name="Notify" schema-resource="wsdl/eventservice/standard/
            wsn/b-2.xsd"
286       default-namespace="http://docs.oasis-open.org/wsn/b-2">
```

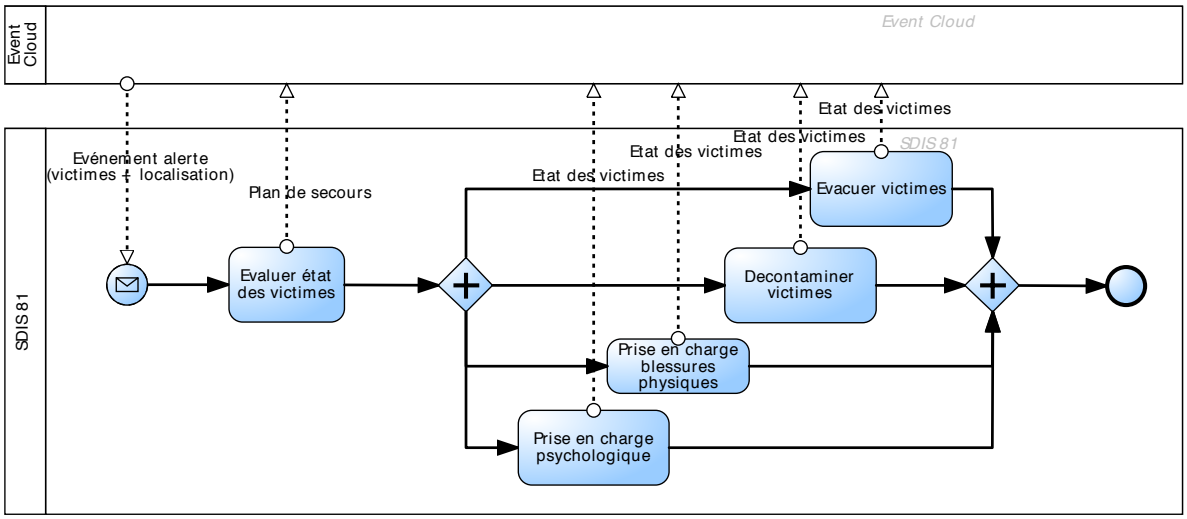
```
287       <namespace-prefix prefix="ns18"
288         namespace="http://docs.oasis-open.org/wsn/b-2" />
289       <namespace-prefix prefix="ns2"
290         namespace="http://www.mines-albi.fr/nuclearcrisisevent" />
291       <xpath-property property-name="reportId"
292         xpath="ns18:NotificationMessage/ns18:Message/ns2:reportEvent/ns2:uid"
293         type="string" />
294       <xpath-property property-name="reportTimestamp"
295         xpath="ns18:NotificationMessage/ns18:Message/ns2:reportEvent/ns2:timestamp"
296         type="string" />
297       <xpath-property property-name="reportResultMessage"
298         xpath="ns18:NotificationMessage/ns18:Message/ns2:instructionEvent/ns2:
              resultMessage"
299         type="string" />
300       <xpath-property property-name="uncertaintyUnit"
301         xpath="ns18:NotificationMessage/ns18:Message/ns2:instructionEvent/ns2:
              uncertainty/ns2:unit"
302         type="string" />
303       <xpath-property property-name="uncertaintyValue"
304         xpath="ns18:NotificationMessage/ns18:Message/ns2:instructionEvent/ns2:
              uncertainty/ns2:value"
305         type="number" />
306       <xpath-property property-name="uncertaintyOrigin"
307         xpath="ns18:NotificationMessage/ns18:Message/ns2:instructionEvent/ns2:
              uncertainty/ns2:origin"
308         type="string" />
309       <xpath-property property-name="activityId"
310         xpath="ns18:NotificationMessage/ns18:Message/ns2:instructionEvent/ns2:
              consequence/ns2:activityID"
311         type="string" />
312     </xml-dom>
313   </event-type>
314   <!-- Imports -->
315   <auto-import import-name="java.lang.Double" />
316   <auto-import import-name="java.math.*" />
317   <!-- <plugin-loader name="EsperIOHTTPAdapter" class-name="com.espertech.esperio.
          http.EsperIOHTTPAdapterPlugin">
318     <init-arg name="esperio.http.configuration.file" value="file:///Users/barthe/
          Documents/workspaces/workspaceHelios/EsperTestXML/etc/esperio-http-sample-
          config.xml"
319     /> </plugin-loader> -->
320 </esper-configuration>
```

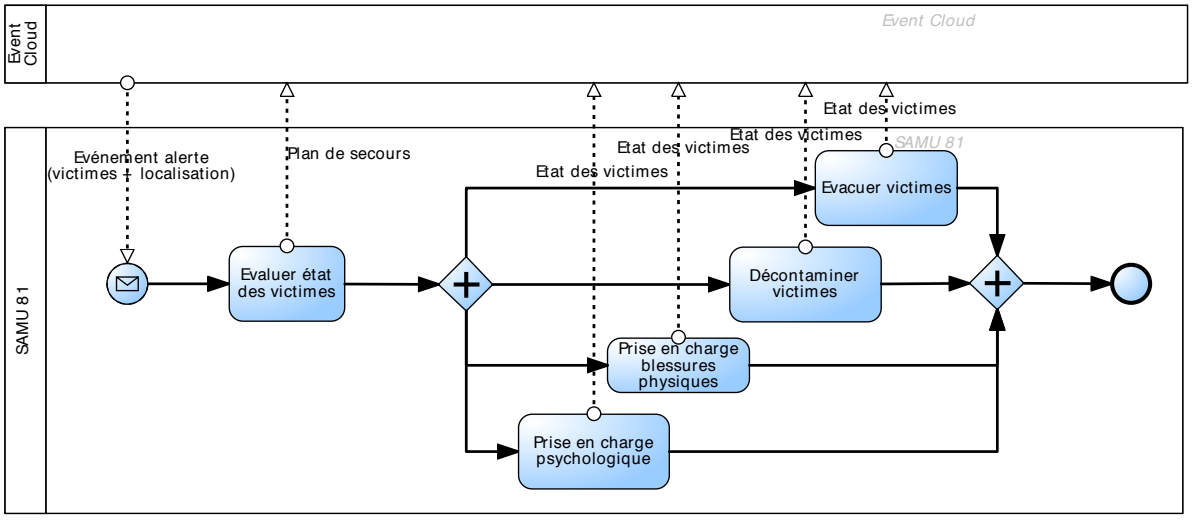
Code L.1 – Fichier de configuration des types d'événements dans Esper

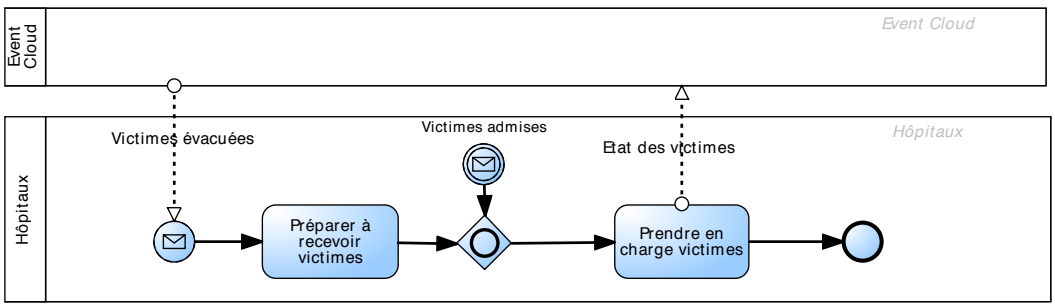
Annexe M

Processus de réponse à la crise (cas d'étude simplifié)

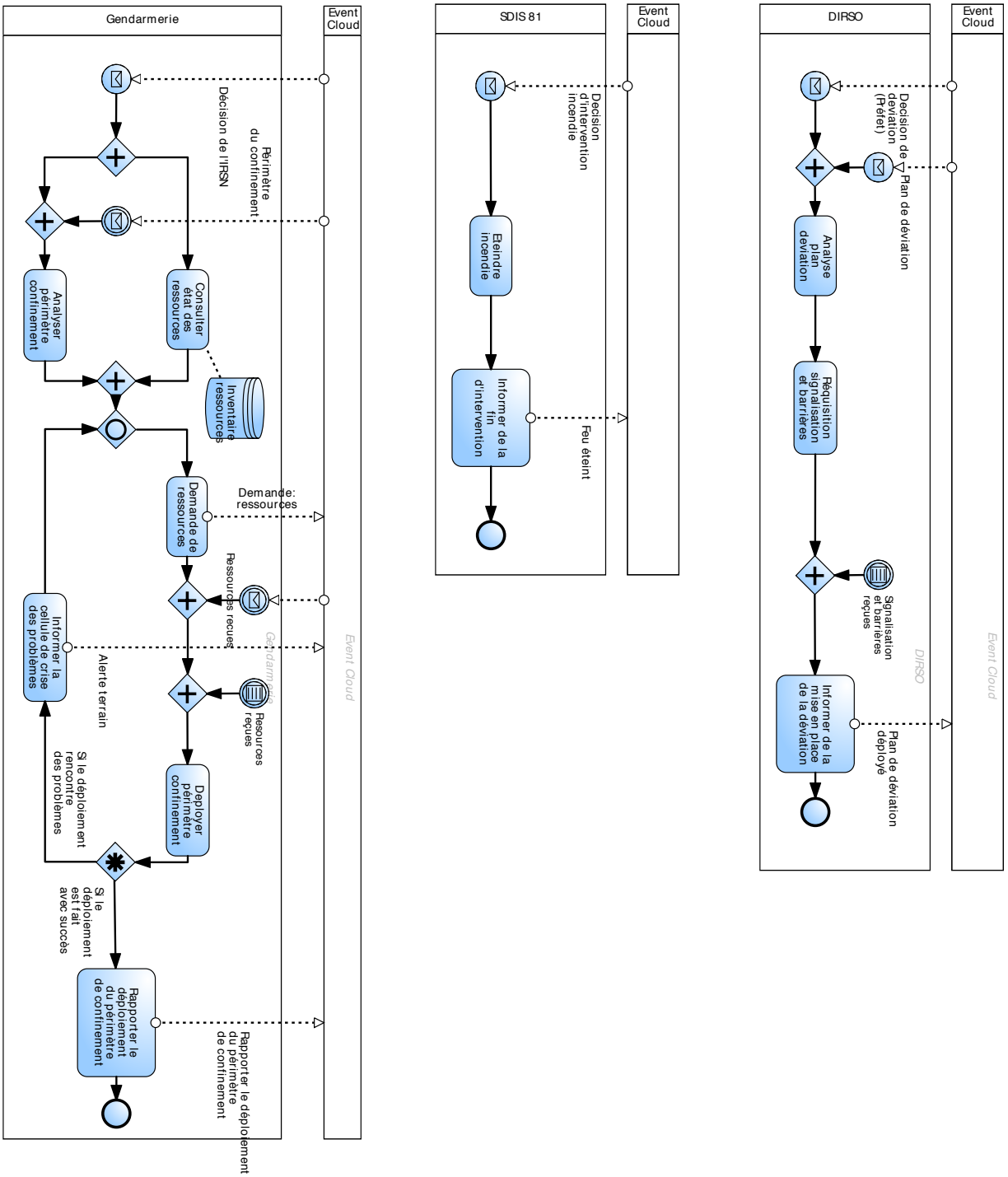
operationnel_assistance_secours



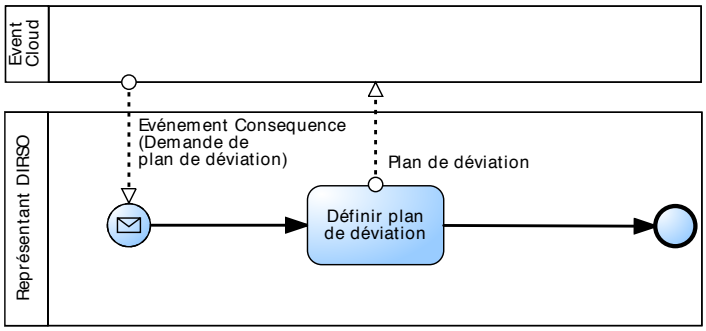


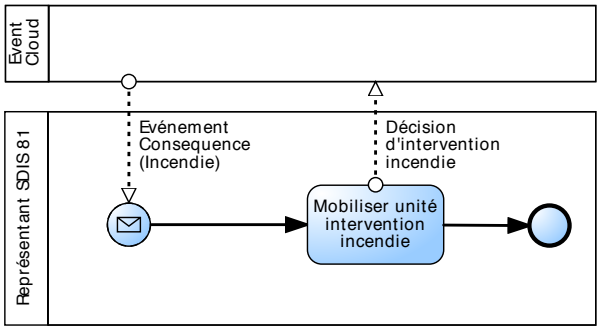


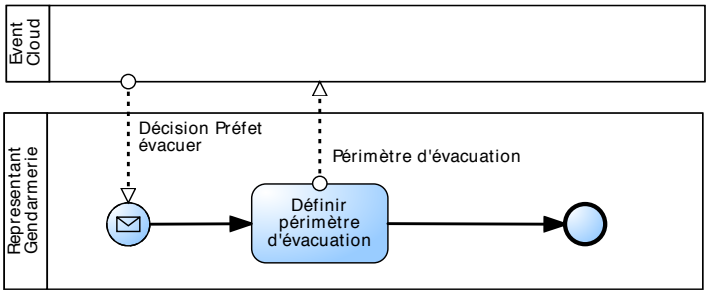
operationnels

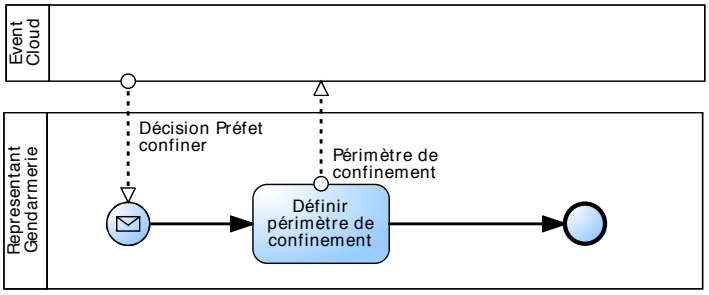


Processus_t0









Annexe N

Règles métier implémentées dans Esper (cas d'étude simplifié)

Détection d'un incendie La détection d'un incendie peut se faire en corrélant les mesures de températures relevées par plusieurs capteurs situés à une distance raisonnable les uns des autres. La règle ci-dessous indique au service MISECep (qui encapsule le moteur de CEP Esper) qu'en cas de réception d'événements de type *Mesure* faisant état d'une température supérieure à 100°C et émis par deux capteurs, ces capteurs étant situés à proximité les uns des autres, dans un laps de temps de 10 min [1], alors MISECep doit créer et émettre un nouvel événement de type *Alerte*, indiquant une conséquence [2] nommée incendie et localisée dans le même secteur que les capteurs. A et B représentent deux événements de type *Mesure* reçus par MISECep.

SI A.id $\approx$ B.id
ET A.localisation $\sim$ B.localisation
ET A.unit = °C
ET B.unit = °C
ET A.value $\geqslant$ 100
ET B.value $\geqslant$ 100
TOUTES LES10 minutes
==>Alerte{Consequence ; Incendie ; A.localisation}

Détection d'un risque de contamination La détection d'un risque de contamination peut se faire en corrélant les mesures de radiation moyenne dans l'air ambiant relevées par plusieurs capteurs (qu'ils soient sur site ou embarqué à bord d'un véhicule), ainsi que les données éoliennes (vitesse et direction du vent), et les données topologiques et urbaines (présence de bâtiments d'habitations, d'accueil du public, etc.).
La règle ci-dessous indique au service MISECep (qui encapsule le moteur de CEP Esper) qu'en cas de réception d'événements de type *Mesure* faisant état d'une vitesse du vent

1. On « réinitialise » la fenêtre temporelle toutes les 10 minutes en la vidant de tous les événements qu'elle contient.
2. Pour pouvoir ensuite faire le lien avec le modèle de la crise et mettre à jour le modèle *terrain*

supérieure à 15 km/h, d'un taux de radiation moyen de 2 mSv/h, et d'une direction du vent de type nord-ouest pendant 5 minutes[3], alors MISECep doit créer et émettre un nouvel événement de type *Alerte*, indiquant un risque[4] nommée contamination et localisée dans les secteurs habités (le nuage de particules radioactives se dirigeant alors vers des zones peuplées, le cadastre indiquant la présence d'habitations et d'une école primaire dans cette direction). A, B, C représentent trois événements de type *Mesure* reçus par MISECep : A représente une mesure de la radiation moyenne dans l'air ambiant, B une mesure de la vitesse du vent, C une mesure de la direction du vent.

SI A.unit = mSv/h
ET B.unit = km/h
ET C.unit = direction
ET A.value $\geqslant$ 2
ET B.value $\geqslant$ 15
ET C.value = nno
PENDANT 5 minutes
==>Alerte{Risque ; Contamination ; C.value}

Détection d'une panne à l'appel d'un service La détection d'une panne à l'appel d'un service se fait en observant l'absence d'événement statuant un état « wip » pour le service appelé.
La règle ci-dessous indique au service MISECep (qui encapsule le moteur de CEP Esper) qu'en cas de non réception d'événements de type *Activité* faisant état d'un statut égal à « wip », dans les 5 minutes[5] après l'appel du service par le SIM (le SIM émettant un événement pour informer MISECep de l'appel d'un service dans le déroulement de l'orchestration des services), alors MISECep doit créer et émettre un nouvel événement de type *Activité*, indiquant un état « failed ». A, B représentent respectivement deux événements de type *Demande* et *Activité* attendus par MISECep : A représente l'appel du service par le SIM, B le changement de statut du service appelé (son statut passe de « wait » à « wip » si tout va bien).

POUR TOUT NON { **SI** A.activityId = B.activityId
ET B.status = wip
PENDANT 5 minutes} ==>Activite{B.activityId ; failed}

3. La fenêtre temporelle est de 5 min et « glisse » au cours du temps, les événements sont supprimés de la fenêtre dès qu'ils y sont présents plus de 5 minutes.
4. Pour pouvoir ensuite faire le lien avec le modèle de la crise et mettre à jour le modèle *terrain*
5. Choix de durée totalement arbitraire pour l'exemple.

Prise en charge de l'agilité de workflows collaboratifs
par une approche dirigée par les événements

Les organisations participent à des collaborations pour faire face à un environnement en perpétuelle évolution (mondialisation, crise, etc.). Or, la nature instable de la collaboration et de son environnement peuvent compromettre la pertinence des processus définis pour atteindre les objectifs collectifs. Deux problématiques émergent alors de ce constat : comment détecter l'inadéquation des processus exécutés par rapport aux objectifs poursuivis à l'instant t ? Comment redéfinir la meilleure réponse possible à l'instant t, dans un laps de temps proche du temps réel (en fonction de la situation de collaboration, de l'avancée de l'exécution des processus, et l'état des acteurs et ressources). La problématique scientifique relève de l'apport d'agilité aux processus collaboratifs. Ces travaux de thèse ambitionnent de répondre à ces questions en proposant (i) une définition de *l'agilité* des processus collaboratifs, (ii) la mise en place d'une *architecture orientée services dirigée par les événements* (ED-SOA), afin de prendre en compte les événements émis par la collaboration et son environnement, (iii) un algorithme de mesure de distance entre le modèle de la situation collaborative telle qu'elle devrait être et le modèle de la situation telle qu'elle est réellement *(détection)*, et enfin (iv) un algorithme de recommandation d'*adaptation* des processus collaboratifs. L'implémentation des mécanismes de l'agilité des processus collaboratifs a donné naissance à un prototype open-source. Les résultats de cette thèse s'inscrivent également au sein des projets ANR SocEDA et EUR FP7 PLAY qui proposent chacun une gestion des processus collaboratifs dans le contexte d'une plateforme dirigée par les événements.
Mots-clés : Crise — EDA — Ontologie — Orchestration — Chorégraphie — Médiation

---

Ensuring the agility of collaborative workflows through an event driven approach

It is known that organizations have to take part into collaborations to face un unstable world. But, the unstable nature of the collaboration and of its environment can threaten the accuracy of the processes. We can extract two main issues from this context : (i) How to detect the instant where the collaborative processes does not match with the current crisis situation ; (ii) How to redefine the best possible response at time t, in real time, depending of the collaborative situation, the processes execution progress, the state of the resources and collaboration's partners. The scientific problem is focused on the agility of collaborative processes. This thesis aspires to solve these issues by proposing (i) a definition for collaborative processes agility, (ii) the definition and the realization of an Event-Driven Architecture layer among the SOA architecture of the information system of the collaboration, in order to take in account the events emitted by the collaboration and its field to update the collaborative situation models. Then (iii) an algorithm to measure the distances between the model of the crisis situation at is it supposed to be when the processes are executed, and the model of the crisis situation as it really is, and (iv) a tool for decision support and a redesign of the crisis response processes and choreography. These results are in line with the ANR SocEDA project and the EUR FP7 PLAY project which aim at providing management of collaborative processes, in an event-driven context. The mechanisms to ensure collaborative processes agility are implemented as an open-source prototype.
Keywords : Crisis — EDA — Ontology — Orchestration — Choreography — Mediation