



Coordination Models for Crisis Resolution: Discovery, Analysis and Assessment

Nguyen Le Tuan Thanh

► To cite this version:

Nguyen Le Tuan Thanh. Coordination Models for Crisis Resolution: Discovery, Analysis and Assessment. Artificial Intelligence [cs.AI]. Université Toulouse 3 Paul Sabatier, 2016. English. NNT : . tel-01525408

HAL Id: tel-01525408

<https://hal.science/tel-01525408>

Submitted on 20 May 2017

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.



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le 15/12/2016 par :

NGUYEN-TUAN-THANH LE

**Coordination Models for Crisis Resolution:
Discovery, Analysis and Assessment**

JURY

CHIHAB HANACHI	Professeur, Université Toulouse 1 Capitole	Directeur de Thèse
SERGE STINCKWICH	MCF, Université de Caen Normandie	Co-encadrant de Thèse
TUONG-VINH HO	Professeur, IFI - Université Nationale du Viêt Nam de Hanoï	Co-encadrant de Thèse
BERNARD ESPINASSE	Professeur, Université Aix-Marseille	Rapporteur
JULIE DUGDALE	MCF HDR, Université Grenoble Alpes	Rapporteur
FRANCIS ROUSSEAUX	Professeur, Université de Reims Champagne Ardennes	Examineur
HENDA HAJJAMI BEN GHEZALA	Professeure, Université de La Manouba	Examinatrice

École doctorale et spécialité :

MITT : Domaine STIC : Intelligence Artificielle

Unité de Recherche :

Institut de Recherche en Informatique de Toulouse (UMR 5505)

Coordination Models for Crisis Resolution: Discovery, Analysis and Assessment

Abstract:

This thesis is about coordination in multi-agent universes and particularly in crisis contexts.

Recently, we have witnessed an increasing number of crises, not only natural disasters (hurricane Katrina, Haiti earthquake, ...) but also man-made ones (Syrian refugees crisis in Europe, Arabic spring, rioting in Baltimore, ...). In such crisis, the different actors involved in the resolution have to act rapidly and simultaneously in order to ease an efficient control and reduce its impacts on the real world. To achieve this common goal as quickly and efficiently as possible, these actors (police, military forces, medical organizations, etc.) must join their respective resources and skills to collaborate and act in a coordinated way, most often by following a plan that specifies the expected flow of work between them.

By coordination, we mean all the work needed for putting resolution plans and all stakeholders' skills and resources together in order to reach the common goal (crisis resolution) in an efficient way. Crisis resolution plans are most of the time available in a textual format defining the actors, their roles and coordination recommendations in the different steps of crisis life-cycle (mitigation, preparedness, response and recovery). While plans in a printed document format are easy to manipulate by stakeholders when taken individually, they do not provide direct means to be analyzed, simulated, adapted or improved and may have various different interpretations. Therefore they are difficult to manage in real time and in a distributed setting. Given these observations, it becomes useful to model these textual plans to have an accurate representation of them, to reduce ambiguity and to support coordination between stakeholders and ease an efficient control and crisis resolution.

The goal of this PhD thesis is to contribute to **coordination engineering in crisis domains** by providing a comprehensive approach that considers both task and organizational aspects in a coherent conceptual framework. In this perspective, **our approach** combines Business Process and Multi-Agent paradigms and provides a mapping algorithm between their concepts. BPM (Business Process Modelling) provides an aggregate view of the coordination through the task aspect and so doing eases the validation, the simulation, the intelligibility of crisis resolution plans at design time and its monitoring at run time. The Multi-Agent

paradigm provides social abstractions (high-level interactions and organization structures) to model, analyse and simulate an organizational view of the coordination by representing the structure and the behaviour of the system being developed at a macro level, independently of the internal structure of agents (micro level).

The contribution of this thesis is a coordination framework, which consists of three related components:

- A design and development approach (design/discovery, analysis, simulation) that provides means (recommendations, formalisms, life-cycle, algorithms) to produce (agent and process-based) coordination models from a textual plan and/or event based log files,
- A mapping algorithm deriving BPMN process schemas onto multi-agent structures,
- Coordination evaluation metrics. We extend the works of Grossi and define formal metrics that allow the evaluation of the quality (efficiency, robustness and flexibility) of multi-agent system organizations.

We have applied this framework to the Ho Chi Minh City Tsunami resolution plan.

Keywords: Crisis and Disaster Management, Coordination, Multi-Agent System, Business Process Management.

Modèles de Coordination pour la Résolution des Crises: Découverte, Analyse et Evaluation

Résumé en français:

Cette thèse concerne la coordination en univers des multi-agents et en particulier dans des contextes de crise.

Nous assistons actuellement à un accroissement du nombre de crises, non seulement des catastrophes naturelles (ouragan Katrina, tremblement de terre d'Haïti, ...) mais aussi humanitaires (crise des réfugiés syriens en Europe, printemps Arabe, émeutes de Baltimore, ...). Dans toutes ces crises, les différents acteurs œuvrant pour leurs résolutions doivent agir rapidement et simultanément, afin de contrôler et réduire les impacts de la crise sur le monde réel. Pour atteindre cet objectif commun aussi rapidement et efficacement que possible, ces acteurs (police, forces militaires, organisations médicales) doivent mettre en commun leurs ressources et compétences respectives, collaborer et agir de manière coordonnée, le plus souvent en suivant un plan qui indique la répartition du travail et les échanges entre eux.

Par coordination, nous entendons tout le travail nécessaire pour élaborer ou adapter le plan de résolution de crises et mettre en commun les compétences et les ressources de tous les intervenants afin d'atteindre l'objectif commun (la résolution de crise) de manière efficace. Les plans de résolution de crise sont la plupart du temps disponibles dans un format textuel définissant les acteurs impliqués, leurs rôles et des recommandations pour leurs coordinations dans les différentes étapes du cycle de vie de crise (anticipation, préparation, résolution de la crise, retour d'expérience). Bien que les plans dans un format textuel soient faciles à manipuler par les intervenants pris individuellement, il n'existe pas de moyens directs pour les analyser, simuler, adapter, améliorer et ils peuvent avoir diverses interprétations, ce qui les rend difficile à gérer en temps réel et dans une situation où les acteurs sont géographiquement distribués. Au regard de ces observations, il devient indispensable de faire une modélisation de ces plans textuels afin d'en avoir une représentation précise, éviter toute ambiguïté, faciliter la coordination entre les intervenants et enfin faciliter la gestion ainsi que la résolution de crises.

L'objectif de cette thèse est de contribuer à **l'ingénierie de la coordination dans les domaines de crise** en fournissant une approche globale qui prend en compte sans les confondre et en les articulant deux aspects es-

sentuels de la gestion de crises: l’aspect tâche et l’aspect organisationnel. Dans cette perspective, **notre approche** combine deux paradigmes: BPM (Business Process Management) et les systèmes multi-agents, et fournit un algorithme de traduction entre leurs concepts. BPM fournit une vue globale de la coordination du point de vue de la tâche à réaliser et facilite la validation, la simulation et l’intelligibilité des plans de résolution de la crise au moment de leur conception tout en permettant le suivi de leur exécution. Le paradigme multi-agent fournit, quant à lui, des abstractions sociales (des interactions de haut niveau et des structures organisationnelles: rôles, relations et interactions entre acteurs) pour modéliser, analyser et simuler une vue organisationnelle de la coordination en représentant la structure et le comportement du système développé au niveau macro, indépendamment de la structure interne des agents à un niveau micro.

La contribution de cette thèse est un cadre conceptuel pour l’ingénierie de la coordination. Il se compose de trois propositions reliées entre elles:

- Une approche de conception et de développement (conception/découverte, analyse, simulation) qui fournit des moyens (recommandations, formalismes, cycle de vie, algorithmes) pour produire des modèles de coordination basés sur des agents et des processus à partir d’un plan textuel et/ou des évènements basés sur des fichiers logs.
- Un algorithme de transformation dérivant des schémas de processus BPMN en des structures organisationnelles multi-agents.
- Des indicateurs de mesure de l’évaluation de la coordination. Nous étendons pour cela les travaux de Grossi, puis définissons des métriques formelles permettant l’évaluation de la qualité (efficacité, robustesse et flexibilité) d’organisations multi-agents concrètes.

Nous avons mis en œuvre cette approche sur un exemple: le plan de réaction des autorités à un Tsunami sur la ville de Ho Chi Minh Ville (Vietnam).

Mots-clés: Gestion des Crises et des Catastrophes, Coordination, Système multi-agents, Gestion de processus.

DEDICATION

This dissertation is dedicated to my mother, **Thi-Mung NGUYEN**, and to my father, **Van-Dang LE**, who always believe and encourage me !

It is also dedicated to my daring wife, **Thi-Xuan-Khang VU**, and my little son, **The-An LE (Bin)**, for their love !

Preface and Acknowledgements

My research study has been performed in the SMAC team (Systèmes MultiAgents Coopératifs), at the IRIT Laboratory (Institut de Recherche en Informatique de Toulouse - UMR 5505), Université Toulouse I Capitole and Université Toulouse III Paul Sabatier, France. This work cannot succeed without the extraordinary guidance of my supervisors. I would like to give my special acknowledgement to Professor Chihab HANACHI, Associate Professor Serge STINCKWICH and Dr. Tuong-Vinh HO.

I always remember the first day I came to Toulouse three years ago. Professor HANACHI picked me up from the airport and helped me to get familiar with a new life in a new city. During my time in Toulouse, with his usual honesty, he always give me useful advices in the world of science.

I would like to appreciate the valuable support of Associate Professor STINCKWICH. His kindness and ambition help me to discover the beauty of my research. My stays in IRD (Bondy, Paris) were actually useful. I had new ideas each time I worked there and they helped accelerate my research. I also would like to express the gratitude to Dr. HO for accepting my candidate of this research. His support during my stay in Hanoi is noble.

I would like to express my appreciation to the reviewers of my thesis: Professor Bernard ESPINASSE at Universités Aix-Marseille and Associate Professor Julie DUGDALE at Laboratoire d'Informatique de Grenoble for their expertise and feedback.

I would like to thank the University of Science and Technology of Hanoi (USTH - Vietnam France University), Vietnam International Education Development - Ministry of Education and Training (VIED) and Consortium of USTH for their supports (finance, seminars, . . .) during my research in France.

My thanks go also to the colleagues of the University Toulouse I Capitole and IRIT Laboratory, especially the members of SMAC team: *Hanane*¹, *Sameh*, *Amal*, *Khalid*, etc. for sharing the moments of life. I would like to thank my friends at Colonel Roche: *Xuan-Hien TA*, *Van-Nhat PHAM*, *Thu-Hien NGUYEN*, *Trong-Trung LE*, *Minh-Quan PHAM* etc. for their kindness. Three years of life with plenty of souvenirs around friends are always in my heart.

¹Thank you for your hard work to perform the mapping Petri nets onto BPMN, see Appendix D.2

Contents

1	Introduction	14
1.1	Context	14
1.2	Problem addressed	15
1.3	Approach followed	17
1.4	Contributions	17
1.5	Outline of this thesis	18
I	State of the Art	20
2	Coordination in a Multi-agent Universe	22
2.1	Introduction	22
2.2	The Importance, Role and Need for Coordination	22
2.3	Coordination Models	26
2.4	Coordination Techniques	28
2.5	Coordination Model Evaluation Criteria	33
2.6	Conclusion	34
3	Coordination Models for Crisis Management	36
3.1	Introduction	37
3.2	Crisis Domain Analysis	37
3.2.1	Emergency vs Disaster vs Catastrophe	37
3.2.2	Crisis Management Life-cycle	39
3.3	Coordination Models for Crisis Management	41
3.3.1	Coordination in Crisis Management	41
3.3.2	Coordination Models for Crisis Management	41
3.3.2.1	Process-based Models	42
3.3.2.2	Organization-based Models	43
3.3.2.3	Multi-agent based Models	43
3.4	Comparison of Coordination Platforms	43
3.4.1	WORKPAD	44
3.4.2	SoKNOS	44
3.4.3	INDIGO	45
3.4.4	Sahana EDEN	45

<i>CONTENTS</i>	3
3.4.5 USHAHIDI	46
3.4.6 HAC-ER	46
3.4.7 Franke’s Framework	48
3.4.8 Comparison of Platforms	48
3.5 Conclusion	50
II Contribution	52
4 Overview of Our Approach	54
4.1 Introduction	54
4.2 The Engineering Lifecycle of Coordination Models	57
4.2.1 Advantages of Combining Workflow and Multi-Agent System Paradigms	59
4.2.2 Preparation of Coordination Data	60
4.2.3 Generation of Coordination Models	61
4.2.4 Transformation of Coordination Models	62
4.2.5 Analysis and Improvement of Coordination Models	62
4.3 Our contribution	64
4.3.1 A Scenario-based Approach	65
4.3.2 Model transformations	65
4.3.3 New Metrics and a Tool for Evaluating Organizations	65
4.4 Conclusion	67
5 Process-based Coordination Models	68
5.1 Introduction	69
5.2 Design of a Crisis Resolution Process from Scenarios	71
5.3 Identification of High-level Objects from Text	72
5.3.1 Principles	72
5.3.2 Illustration: Identifying Organizations, Tasks and their Relationships from HCMC’s Tsunami Response Plan	74
5.4 Building Scenarios	76
5.4.1 Scenario Structure	76
5.4.2 Generation of Scenarios in the Design Phase	79
5.5 Process Discovery	83
5.5.1 Process Representation using Petri nets	83
5.5.1.1 Petri nets	84
5.5.2 Process Discovery using the α -algorithm	84

5.5.2.1	Application to HCMC Scenarios	87
5.5.3	BPMN Representation of Stakeholder Validation	87
5.5.3.1	BPMN	88
5.5.3.2	Mapping Petri nets onto BPMN	89
5.6	Process Simulation, Analysis and Conformance Checking	91
5.6.1	Theoretical Verification	93
5.6.2	Process Deployment and Execution	93
5.6.2.1	Resource and Time Analysis	96
5.6.3	Process Simulation and Validation	96
5.6.3.1	Defining Simulation Parameters	96
5.6.3.2	Defining Simulation Scenarios	100
5.6.3.3	Simulation and Analysis of Scenarios	103
5.6.4	Process Conformance Checking and Organizational Discovery with the PROM Framework	105
5.6.5	The A4BP Tool for Measuring Process Complexity	108
5.7	Conclusion	112
6	Organization-based Coordination Models	114
6.1	Introduction	115
6.2	OCMAS vs ACMAS Models	116
6.2.1	The AGR Model	117
6.2.2	The BDI Agent Model	119
6.3	The Role Graph	120
6.3.1	Grossi's Definition	120
6.3.2	Extracting the Role Graph from Textual Rescue Plans	125
6.3.3	Extracting the Role Graph from AGR models	126
6.3.4	New Structural Metrics for dynamic situations	127
6.4	Extracting Organizational Models from Business Processes	129
6.4.1	Mapping a Process Model onto an AGR Model	131
6.4.2	Mapping a Process Model onto a Role Graph	132
6.4.3	Mapping a Process Model onto a BDI Agent Model	135
6.5	Case Study: the HCMC Tsunami Response Plan	137
6.5.1	Extracting the AGR Model from the Process Model	137
6.5.2	Extracting the Role Graph Model from the Process Model	139
6.5.3	Extracting the BDI Agent Model from the Process Model	141
6.6	Assessment of the HCMC Tsunami Response Plan	143
6.7	Conclusion	148

7 Conclusion, limits and Future Works	150
7.1 Conclusion	150
7.2 Limitations and Perspectives for Future Research	152
7.2.1 Involving Citizens in Crisis Resolution through Socio-media	152
7.2.2 Improving the Derivation of Standalone Agents from BPMN	153
Bibliography	156
 III Appendix	 166
A Personal Publications	168
B Glossary	169
B.1 Glossary	169
C Ontology for Crisis Management	170
C.1 Ontology for Crisis Management	170
D Source Code	173
D.1 Implementing the Framework for Assessing Organizational Struc- ture based on Role Graph in Moose	173
D.1.1 Introduction	173
D.1.2 AgentOrganizationEvaluation Code in Moose	173
D.1.2.1 AOERole Class	173
D.1.2.2 AOEOrganization Class	174
D.2 Implementing the Mapping from Petri nets to BPMN	174
D.3 Implementing the Mapping from BPMN to AGR	183
D.4 Implementing the Mapping from BPMN to BDI-Agent	192
E Résumé Long en Français	196
E.1 Introduction	196
E.1.1 Contexte	196
E.1.2 Problème adressé	197
E.1.3 Approches suivies	199
E.1.4 Contributions	200
E.1.5 Vue d'ensemble de cette thèse	200
E.2 Conclusion, limites et travaux futurs	202
E.2.1 Conclusion	202

E.2.2	Limitations et perspectives pour de futurs travaux	203
E.2.2.1	Impliquer des citoyens dans la résolution de crises en utilisant des réseaux sociaux	204
E.2.2.2	Améliorer la dérivation d'agents à partir de BPMN	205

List of Figures

1.1	Three interrelated aspects of Coordination	16
2.1	Parameters involved in coordination expressed as a mind map [Smith 2011]	25
2.2	Different coordination techniques [Le 2013]	28
3.1	Disaster of Tsunami in Japan (2011)	38
3.2	Life cycle of Crisis Management [Wallace 1985]	40
3.3	Vietnam's Crisis Early-Warning and Response System on top of Ushahidi platform	47
4.1	An extract from the textual Rescue Plan drawn up by the People's Committee of Ho Chi Minh City, Vietnam on 13 July 2012 (written in Vietnamese)	55
4.2	An extract from the translated version of the textual Rescue Plan drawn up by the People's Committee of Ho Chi Minh City, Viet- nam on 13 July 2012	55
4.3	Four layers of crisis management coordination	58
4.4	Lifecycle for evaluation and mapping from process models (<i>Petri nets</i> and <i>BPMN diagrams</i>) to organization models (<i>role graphs</i> , <i>ACMAS</i> and <i>OCMAS models</i>)	63
4.5	Our contribution (shown in red italics) based on our lifecycle so- lution for crisis management coordination	64
4.6	Meta-model of crisis coordination including the notions of: <i>cri- sis management, coordination, process, organization, multi-agent systems, environment</i> and <i>metrics</i>	66
5.1	Life-cycle of the engineering crisis process model	70
5.2	A scenario-based lifecycle for process design and validation from text	72
5.3	Meta-model of concepts involved in coordinating crisis manage- ment activities	73
5.4	Structure of a scenario [Rolland 1998]	78
5.5	Meta-model subset for process discovery	80
5.6	Petri nets representation of the HCMC tsunami response plan . .	87

5.7	Core subsets of BPMN elements (Source: http://www.bpmn.org)	88
5.8	Mapping between Petri nets and BPMN notation	90
5.9	BPMN representation of the HCMC tsunami response plan	92
5.10	The three YAWL perspectives	94
5.11	Task description	95
5.12	Execution and simulation of the tsunami response plan in the YAWL environment	96
5.13	Petri nets mined by ProM using α -algorithm on top of a 1000-cases event log	106
5.14	Sociogram showing the work transfer relation	107
5.15	The A4BP architecture	109
5.16	The A4BP interface	110
5.17	Visualization of process complexity in the HCMC tsunami rescue plan	110
5.18	Visualization of process activities based on time and resources fol- lowing the simulation of the post-HCMC tsunami rescue plan . .	111
6.1	The AGR meta-model proposed by [Ferber 2004]	119
6.2	Optimal values for the maximization of robustness, flexibility and efficiency as defined by Penserini et al. [Penserini 2009], and Grossi et al. [Grossi 2007]	124
6.3	The role graph built by stakeholders based on the tsunami response plan	125
6.4	Lifecycle of mapping the BPMN model to organizational models (role graph, ACMAS, OCMAS)	130
6.5	Pattern to detect the <i>Power relation</i> between two actors	134
6.6	Pattern to detect the <i>Coordination relation</i> between two actors . .	134
6.7	Pattern to detect the <i>Control relation</i> between two actors	134
6.8	Relation between the pool <i>IG</i> and the lane <i>LA</i>	139
6.9	Relation between the lane <i>LA</i> and lane <i>LCDF</i>	141
6.10	Role graph derived from the BPMN diagram for the HCMC tsunami response plan	143
6.11	Robustness of our organization compared to standard values . . .	145
6.12	Flexibility of our organization compared to standard values	146
6.13	Efficiency of our organization compared to standard values	147
C.1	Classes of our ontology for crisis management	171
C.2	Object properties of our ontology for crisis management	172

D.1	The Meta-model of Petri nets	181
D.2	The Meta-model of BPMN 2.0	182
D.3	The configuration of our transformation program	183
D.4	The visualization of page 1 of our input Petri nets model	184
D.5	The visualization of page 2 of our input Petri nets model	185
D.6	The output BPMN model of our transformation	186
D.7	The Meta-model of AGR according to [Ferber 2004]	187
D.8	The Meta-model of BDI-Agent according to [Michael 1987] and [Endert 2007]	192
E.1	Trois aspects interdépendants dus à la coordination	198

List of Tables

3.1	Comparison Table of Crisis Management Platforms (1)	49
3.2	Comparison Table of Crisis Management Platforms (2)	49
4.1	Advantages of coordination plan representations	61
5.1	Basic relations and constraints between tasks	74
5.2	Tasks and the corresponding actors in the tsunami response plan .	77
5.3	Relations between constraints and tasks	78
5.4	The representation of Scenario 1 (26 events) in an event log . . .	85
5.5	Relations between tasks in the α -algorithm	86
5.6	Probabilities of Choices (PC) of alternative tasks and Importance Factors (IF) of parallel tasks	98
5.7	Service Time (ST) of tasks used in the tsunami response plan based on the Truncated Normal Distribution	99
5.8	Waiting Time (WT) of tasks based on the Truncated Normal Dis- tribution	101
5.9	Availability and unit cost of resources in the tsunami response plan	102
5.10	Cost associated with Tasks	102
5.11	Number of resources allocated to tasks in the two scenarios	103
5.12	Comparison of Time Utilization (RU) for the two scenarios	104
5.13	Comparison of Resource Utilization (RU) for the two scenarios . .	104
5.14	Comparison of Total Cost (TC) for the two scenarios (€)	105
5.15	A4BP metrics	109
6.1	Mapping the business process to the AGR model for the HCMC tsunami response plan	137
6.2	Mapping the business process to the BDI Agent model for the HCMC tsunami response plan	142
6.3	Robustness (on the right) compared to standard values (on the left)	144
6.4	Flexibility (on the right) compared to standard values (on the left)	144
6.5	Efficiency (on the right) compared to standard values (on the left)	145

List of Algorithms

1	α -algorithm to create Petri nets [Van der Aalst 2011]	86
2	Algorithm to identify the role graph from textual plans	126
3	Mapping from the business process model to the AGR model . . .	133
4	Mapping the from business process to the role graph	136
5	Mapping the business process to the BDI Agent model	140

Listings

5.1	Textual HCMC abstract plan	75
5.2	Generating scenarios using the Logic program (Swi-Prolog syntax)	79
D.1	AOERole Class	173
D.2	AOEOrganization Class	174
D.3	Mapping from Petri nets to BPMN using ATL	174
D.4	Mapping from BPMN to AGR using ATL	183
D.5	Mapping from BPMN to AGR using ATL	192

CHAPTER 1

Introduction

Contents

1.1	Context	14
1.2	Problem addressed	15
1.3	Approach followed	17
1.4	Contributions	17
1.5	Outline of this thesis	18

1.1 Context

The last decade has witnessed an increasing number of crisis (Arabic spring, rioting in Baltimore, hurricane Katrina, Fukushima nuclear plant disaster and more recently the Syria refugees crisis in Europe, ...) of different nature (natural or industrial disasters, explosions of violence, tsunami, ...).

One of strong motivation of this work was to contribute the definition of an information system that may ease the management of crisis and more precisely the engineering of their corresponding response plan. Controlling the crisis, saving human lives and reducing its impacts or damage are some of the biggest challenges/desires in human history.

In particular, my country, Vietnam, given its geographic location in the South-East Asian region, could be hit by natural disasters such as tsunami or earthquake, even if the most annual threat is flood. The risk of tsunami or earthquake in Vietnam is not very large, but exists, and it is necessary to be prepared for these disasters [Ca 2008]. Well preparedness referring to efficient coordination among involved organizations and people will help avoid the unworthy damage.

In such crisis, the different actors involved in the resolution have to act rapidly and simultaneously in order to reduce its impacts on the real world. To achieve

this common goal as quickly and efficiently as possible, these actors (police, military forces, medical organizations, etc.) must join their respective resources and skills to collaborate and act in a coordinated way, most often by following a plan that specifies the expected flow of work between them.

1.2 Problem addressed

This thesis focuses on coordination in such multi-agent universe and particularly in crisis context. By *coordination*, we mean all the works needed for putting all these skills, resources and plan together in order to reach their common goal in an efficient way notably by reducing redundant actions and possible collisions between them. Coordination inefficiency could induce more casualties. As I am writing this thesis (October 2015), the European refugee crisis is happening. The inefficient coordination between European countries made hundreds of thousands of migrants cannot reach their desired locations and still stuck at the border with limited food and protection. In USA, when Hurricane Katrina happened in 2005, the poor coordination led to kill more than 1200 people, destroyed tens of billions of dollars and made hundreds of thousands homeless [Prizzia 2008]. Even when people were rescued, they did not have enough shelter or food provision [Franke 2011a].

Coordination in such a multi-agent universe raises several problems due to agent distribution, heterogeneity and autonomy of actions. Also, the universe could be open and new partners (e.g. non government organizations) could join and leave at any moment, and citizens could also influence the crisis resolution by acting in the real world or by propagating their opinion through social media such as *Twitter* or *Facebook* [Imran 2015]. Finally, a crisis is by nature a dynamic phenomena and could lead the actors to reconsider their plan to adapt it to unexpected events or a new form of crisis (e.g. a natural disaster, if ill-managed, could generate a social crisis).

Coordination is a complex issue since it has to take into account several interrelated aspects (Figure E.1):

- The *informational* aspect, which describes the universe of discourse in a common and shareable representation by the partners. It has to solve the semantic heterogeneity in information exchange that could appear between the different actors.
- The *organizational* aspect, which identifies the roles, actors, groups in-

volved in the crisis management and also the relationships (delegation, sub-contractor, coordination, control) that exist between them, but also the protocols that rule their interactions.

- The *task* aspect, which specifies the flow of work between actors. Most often, the plan is amenable to a process defining the tasks and their synchronization.

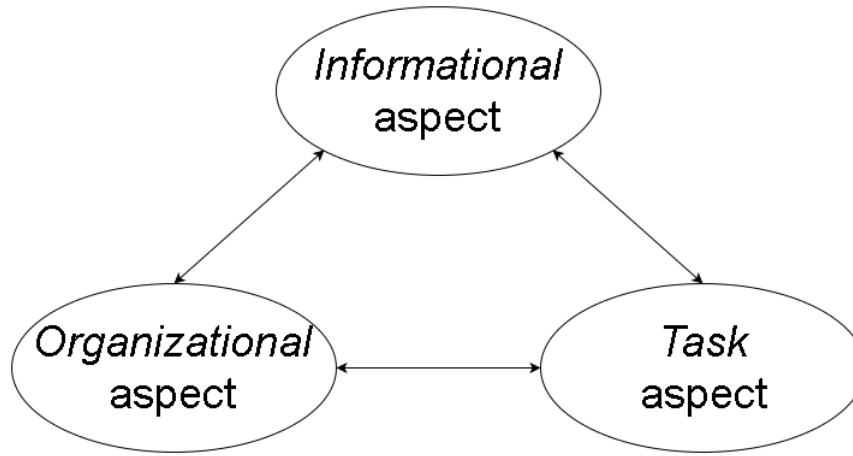


Figure 1.1: Three interrelated aspects of Coordination

In the more simple case, coordination is explicit and concrete models can be provided for each aspects. In some extreme situations, coordination is implicit and determined by a common goal shared by self-organized and interacting stakeholders [Divitini 2001]. In this case, models could be retrieved a posteriori by using process mining techniques [Van der Aalst 2012] to extract actors actions and interactions from a log of events. Obviously, there is a continuum of coordination scenarios between these two extremes.

Whatever the coordination scenario met, to deal with such a complex and dynamic universe, actors need computer-based information systems and tools easing their activities and supporting coordination. These enabling technologies allow actors to have an accurate vision of the crisis current state, to be aware of the past actions and to determine what could and should be done and by whom. From a computer science point of view, we need models and ways to express coordination with an adequate expressive power to cover all the software engineering life-cycle. Indeed, not only we need to specify and simulate these

models in the preparedness phase of the crisis management cycle but we should also support the coordination enactment during the resolution phase.

Many coordination models and platforms have been built for crisis management (e.g. WORKPAD [Catarci 2010] [Catarci 2011], SoKNOS [Paulheim 2009], INDIGO [Ahmad 2012] or USHAHIDI [Okolloh 2009], HAC-ER [Ramchurn 2015], ...) but often focused on one aspect only. The informational aspect has been widely treated by providing crisis ontologies [Bénaben 2008] or meta-models, on top of which shareable artefacts have been built (maps, reports, ...). The task aspect has also been investigated with an emphasize on task and resource allocation issues but current works limit themselves to the execution step of the software life-cycle. Regarding the organization aspect, main work follows the Agent Based Social Simulation (ABSS) approach [Dugdale 2013] mainly focused on the modeling of agents' interactions and on the question of how collective behavior, context awareness or emotions [Nguyen 2014] can emerge from self-organized actors.

1.3 Approach followed

The goal of this PhD thesis is to contribute to coordination engineering in crisis domain by providing a comprehensive approach that considers both organizational and task aspects in a coherent conceptual framework. In this perspective, our approach combines Business Process and Multi-Agent paradigms and provide mapping rules between their concepts. BPM (Business Process Modelling) provides an aggregate view of the coordination through the task aspect and so doing eases i) the validation, analysis, simulation and the intelligibility of crisis resolution plans (most often represented in a textual format in real life) at design time, ii) the monitoring of the crisis at run time. Multi-Agent paradigm provides social abstractions (high-level interactions and organization structures) to model, analyse and simulate an organizational view of the coordination by representing the structure and the behaviour of the system being developed at a macro level, independently of its internal structure (micro level). Also an agent centered view of MAS could be derived, by refinement, from the organization view and this former can be useful for simulation purpose as we will discuss it.

1.4 Contributions

Contributions of this work can be listed as follows:

- An informal approach (life-cycle, recommendations, meta-model, algorithms, ...) that guides designers in modelling a process-based representation of a crisis rescue plan. It is a three steps approach. We give first guidelines to derive the first process from a textual document describing roles and interactions between the involved stakeholders. This textual document considers often an idealistic version of the reality. Therefore, in a second step, we suggest to define different scenarios to cover more realistic situations. These scenarios could deviate from the ideal one and differ in term of available resources, performance, ... Then a process mining technique is used to derive a single process able to play different scenarios including the ideal one. This model can then be simulated, analysed, deployed and transformed as a Multi-Agent System. We have followed these guidelines to describe the Ho Chi Minh Tsunami resolution.
- An algorithm that allows the transformation of a BPMN model into a Multi-Agent one.
- A formal definition of metrics that allows the evaluation of organizational structure of a multi-agent system. In fact, we have extended the Grossi's framework [Grossi 2007]. While this latter evaluates organizational structures, our extension evaluates concrete (abstract) organizations i.e. really deployed ones.
- An implementation of organizational and process metrics.

1.5 Outline of this thesis

The thesis consists of two parts: *State of the art* and *Contribution*. It contains seven chapters whose contents are outlined as follows:

Chapter 1: Introduction

It presents our context, the problem being addressed and our approach. Then it summarizes my contribution before giving the thesis outline.

Part I: STATE OF THE ART

This part contains two chapters.

Chapter 2: Coordination in Multi-agent Universe

This chapter is devoted to coordination concept in multi-agent universe that is relevant for the investigation described in this thesis. It defines coordination notion and presents coordination models and techniques. Formal representations of coordination (ontology, meta-model) are also given.

Chapter 3: Coordination Models for Crisis Management

This chapter is dedicated to coordination models used in crisis management. Firstly, the crisis universe (terminology, ontology, meta-model, management life-cycle) is presented. Some coordination models (process-based, organization-based, multi-agent based models) are identified. Finally, a comparison of crisis management platforms is provided on top of these models and the existing insufficiency is underlined.

Part II: CONTRIBUTION

This part contains four chapters.

Chapter 4: Overview of Our Approach

It provides an overview of my approach by combining workflow and multi-agent system paradigms and then highlights our contribution through this thesis.

Chapter 5: Process-based Coordination Models

In this chapter, we describe the process aspect of our approach (process model, process simulation, process complexity and evaluation). It includes the guidelines to build process model from textual model. A case study in HCMC is given to illustrate our approach.

Chapter 6: Organization-based Coordination Models

We describe in this chapter the organization aspect of our approach. It includes the algorithms of mapping from process models to organization ones. The evaluation of organization coordination models is also provided.

Chapter 7: Conclusion & Future works

It concludes our thesis and discusses the open issues in this chapter.

In addition, we also provide four annexes. The first one contains the list of our publications during the research. While the second one is a glossary of technical terms, the third one is about the ontology for crisis management and the last one contains source codes developed in our work.

Part I

State of the Art

Coordination in a Multi-agent Universe

Contents

2.1	Introduction	22
2.2	The Importance, Role and Need for Coordination	22
2.3	Coordination Models	26
2.4	Coordination Techniques	28
2.5	Coordination Model Evaluation Criteria	33
2.6	Conclusion	34

2.1 Introduction

This first chapter aims to define the domain of coordination in a multi-agent universe, and outline the state of the art in modelling and coordination techniques. Here, agents are considered to be humans or software. First, we define coordination and show its importance and role in a multi-agent universe. Secondly, we explain in detail what a coordination model is, and the most frequently-used techniques. Finally, we identify some evaluation criteria for coordination models and those that are most relevant to our field of study.

2.2 The Importance, Role and Need for Coordination

Coordination is one of the key themes in the domain of multi-agent systems (MAS) and a frequent topic of discussion ([Lesser 2014], [El Fallah-Seghrouchni 2001],

[Omicini 2001], [Ferber 1995], [Nwana 1997]). While clearly this is because it covers many aspects (planning, protocols, negotiation and organization) at different levels (methodological, theoretical and technical), above all it is because this crucial function determines the usability of MAS; namely it *ensures the consistent and effective behaviour of system agents*. There are two aspects to coordination. The first is indispensable, and concerns *making it possible to carry out the expected service* (the overall objective of the system) by ensuring compliance with any constraints governing the provision of the service. As a MAS consists of autonomous agents, it is highly likely that in the absence of coordination, system behaviour becomes chaotic as agents are not aware of each other, obstruct each other, and behaviour is redundant (or at least lacks synergy) all of which means that the system's objective is not met. For proof, one need only observe the current management of Syrian refugees by the European Community, where each country has its own policy and a lack of coordination and shared regulations are contributing to political conflict between certain countries. From this perspective, coordination can be regarded as the counterpart to autonomy, aimed at countering natural disorder. The second aspect of coordination therefore aims to *optimize the overall performance of the system*, i.e., to improve efficiency by minimizing the resources (in terms of cost and processing time) needed, through promoting facilitative actions and synergies, and limiting redundant actions.

Specifically, coordination is necessary for several interrelated reasons ([Nwana 1997], [Jennings 1996]):

The lack of individual skills, resources or information. Agent specialization often means that the skills, resources and information that are available to each agent as an individual are insufficient for it alone to achieve the overall goal: this creates a need for pooling, sharing and exchange. In turn, this leads to mechanisms to allocate duties, develop access rules, and the sharing of resources and information exchange protocols between agents. As each agent has limited or specialized skills, their reasoning mechanisms may be relatively simple, which must be compensated by rich interactions between them. Coordination is needed here to regulate these interactions.

The division of labor to accomplish shared goals. Consider the case of volunteer citizens whose aim is to help in cooperative information acquisition tasks via social networks (crowdsourcing) in order to provide the most reliable and accurate global picture of a disaster (victims, affected locations, blocked transport routes, etc.) as seen in [Bhattacharjee 2011]. Ideally, it would be the case that everyone who participates in describing the situation has a well-defined

role, consistent with the overall objective, and that citizens act cooperatively, i.e., consistent with meeting the overall objective. Each citizen would carry out the specific task that was assigned to them, i.e., by acting opportunistically and taking advantage of observations made in the environment. Coordination is therefore necessary to allocate these roles and/or constrain their behaviour in order that it is clearly oriented towards the shared goal.

Interdependence between the actions of agents. Sometimes the tasks assigned to agents are interdependent. For example, consider a group of NGOs responsible for providing assistance to a group of recently-arrived refugees in a large city. Each NGO is specialized in a type of service (accommodation, education, legal aid, language training, health, supplies, etc.). It is clear that the various services offered by NGOs are mutually dependent: knowledge about the accommodation available may help in the provision of basic needs (food, blankets, mattresses), and health checks for children (current vaccinations). Where they will be living is a prerequisite for their education, while the presence of an interpreter facilitates all services even if not everyone needs help. These interdependencies between services require the actions of different NGOs to be synchronized. Figure 2.1, which the authors present as an ontology, represents the parameters involved in coordination and the relationships between them.

Efficiency. It can happen that several agents are specialized in the same domain. In this case, it is useless for them to carry out the same task. The work of one agent can be partially useful to another. In the context of the reception of refugees, some services are optional insofar as they facilitate the tasks that must be carried out, but are not essential. Thus the availability of an interpreter is optional but very useful in helping to identify the basic needs of refugees: the presence in the refugee population of a person who is familiar with the language of the host country means that the NGO does not have to find an interpreter, which clearly improves their performance.

The satisfaction of non-functional requirements. Grouping agents around a shared goal, as is found in crowd-sourcing, introduces non-functional constraints that apply to the whole system. Examples include the amount of time agents have to provide the service, its financial cost, etc. Here again, respecting constraints related to the quality of service requires coordinating the behaviour of agents.

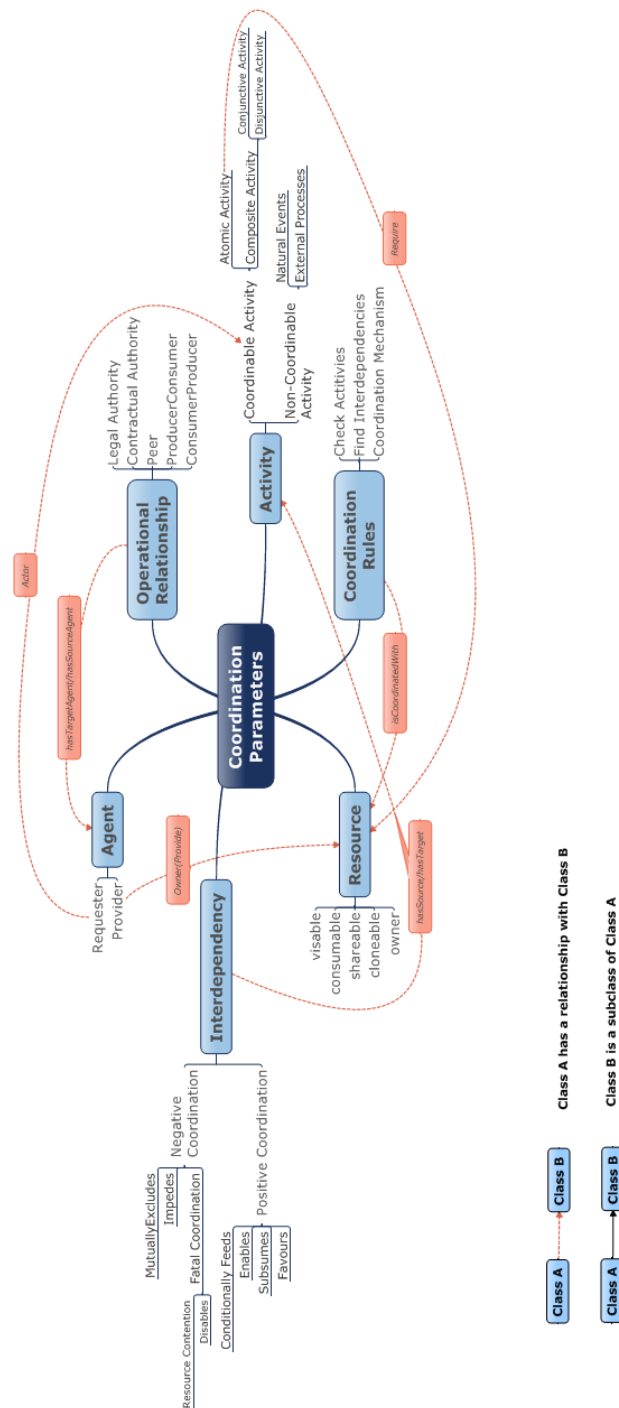


Figure 2.1: Parameters involved in coordination expressed as a mind map [Smith 2011]

2.3 Coordination Models

Coordination models can take different forms depending on their theoretical foundations. These choices are notably related to ([Ferber 1995], [Papadopoulos 2001], [El Fallah-Seghrouchni 2001]): the entities to be coordinated, the media and communication language that supports coordination, coordination protocols or rules, the language that is used to describe this coordination, and the chosen technique.

1. ***The entities to be coordinated.*** Some models make minimal assumptions about the internal structure of agents and simply identify the agents to be coordinated. Others impose constraints on the internal structure of agents and the specific topic of coordination, which can relate to actions, plans or decisions [El Fallah-Seghrouchni 2001]. One of the requirements at this level is not to have to “redesign” agents whenever coordination mechanisms change.
2. ***The media and the communication language.*** A coordination model is founded on communication between agents, which is based on two distinct elements (that are sometimes confused in languages such as Linda): the media and the communication language. In this case, the word *media* refers to the device used to acquire, make accessible and/or transmit the information useful for coordination. It can be a simple transmission channel or a shared space for exchanges that is more or less reactive (like blackboards [Engelmore 1988]) or coordination languages such as Linda [Ahuja 1986]. The word “media” can also refer to an intermediate agent who takes the role of *mediator* [Sycara 1999], when coordination is embodied in an individual component. The media corresponds to a component of the software architecture that supports coordination. The *language of communication* between agents, considered to provide the basic infrastructure for coordination [Papadopoulos 2001], corresponds to the syntax (and possibly the semantics) of the messages exchanged through this medium that acts as an intermediary, most often in languages such as FIPA-ACL [FIPA 2002] or KQML [Finin 1994].
3. ***Coordination protocols and rules.*** These are the laws or principles that govern the interaction between agents via the media. These laws must be specified and implemented using languages. At the operational level, these laws can be expressed using media access primitives and integrated into a host language, or be expressed in any desired, specific coordination

language, ideally declarative in order to preserve a certain degree of agent autonomy.

4. ***The coordination specification language.*** The general characteristics for a coordinating specification language are the following:
 - Ease of use for designers through the provision of high-level abstractions, allowing a hierarchical design and possibly the production of models that can be validated by system users.
 - An appropriate power of expression that enables the description of concepts and phenomena specific to coordination: coordination processes, plans, actions, decisions, resources, competition, and/or opening, etc.
 - Formal semantics that avoid ambiguities in the model and allow the analysis and validation of behavioural properties. The theories that are most commonly used to define formal semantics are automata, logic, graph theory, algebra and Petri nets.
 - Operational semantics that facilitate the correspondence between specification and implementation. This requirement often contradicts the first. In practice, when the aim is to provide high-level abstractions that can be manipulated by the designer, they are often difficult to implement directly in the machine.
5. ***Coordination techniques.*** There are many classifications of coordination techniques [Ferber 1995], [Nwana 1997], [El Fallah-Seghrouchni 2001]. [El Fallah-Seghrouchni 2001] offers a very detailed classification. Here we examine the general level; the following seven families are enough to provide an overview and help to position our work:
 - Coordination based on artefacts
 - Process-oriented coordination
 - Rule-based coordination
 - Coordination based on organizational structures
 - Coordination based on interaction protocols
 - Coordination of actions through planning
 - Negotiation

As Figure 2.2 illustrates, a coordination model may consist of one technique or combine several, and it can vary according to the context and form that coordination takes. The context can be static, in which case the agents remain the same throughout the coordination process, or dynamic, in which case agents are able to enter and leave during the process. The explicit form refers to a technique whereby the coordination schema is defined and accessible to actors (e.g., a process or plan), while the implicit form refers to the fact that there is no schema but only rules to be followed or constraints that must not be violated; here, all behaviour that follows the rules or does not violate constraints is acceptable.

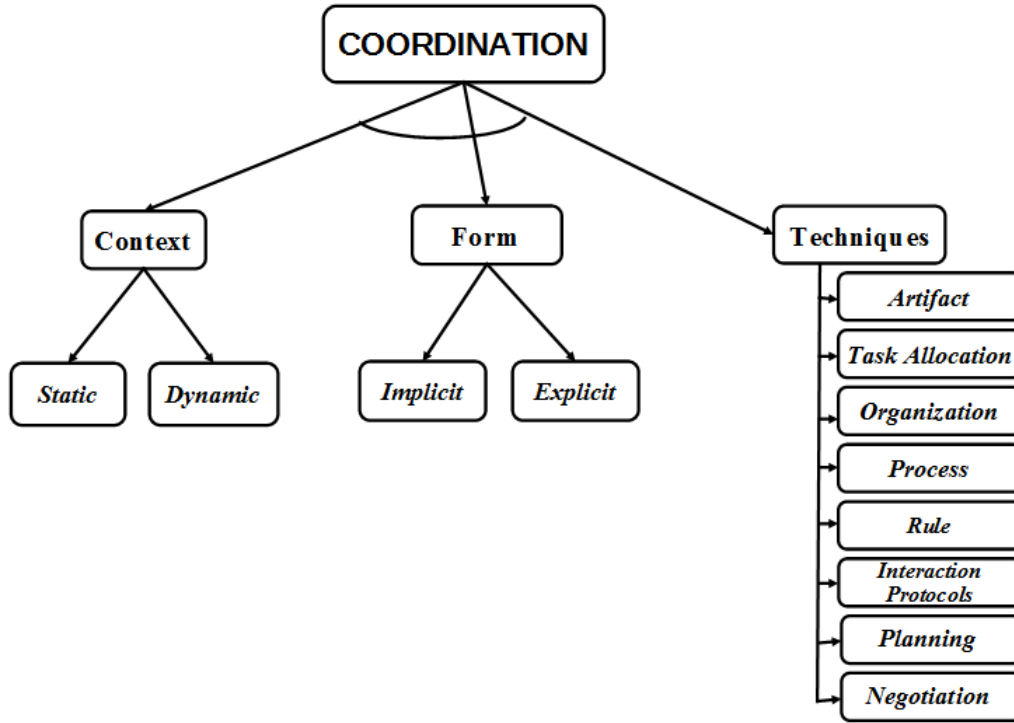


Figure 2.2: Different coordination techniques [Le 2013]

In the next section we address each technique in detail.

2.4 Coordination Techniques

This review of the state of the art does not claim to be complete or exhaustive. Here we present those techniques that shed light on our contribution.

Coordination by artefact. An artefact [Omicini 2008] is a shared artificial object (map, puzzle, blackboard, post-it note, etc.) that may or may not be automated, which reports the state and evolution of the activity of a group of agents (human and/or artificial) while at the same time regulating it. It acts as a mediator that visualizes the result and the context of the work carried out by group members, and constitutes a way to maintain team spirit and situation awareness. It is defined by: i) its structure; ii) functions that make it possible to add, modify and detect the artefact's elements; iii) the operating mode of these functions; and iv) coordination rules that regulate the actions of actors. These actions take account of concrete actions rather than communication acts.

For example, in a crisis context, geographic maps allow a crisis unit to visualize and identify the location of the actions performed by each actor (police, ambulance, army, etc.), planned actions, at-risk zones, etc., and consequently to assess the unfolding situation, the contribution of each action, and what remains to be done.

In the context of crowd-sourcing [Sarcevic 2012], this mechanism allows citizen volunteers to contact each other and to geographically visualize supply and demand for services (as was done for the Haiti crisis with Google Maps software), or even to view the extent and distribution of damage following a disaster (functionality found in Ushahidi¹ project).

Coordination by task allocation. This technique consists of dividing a list of subtasks among several agents derived from an initial task. There are two types of allocation: centralized or distributed. In the first case, there is a so-called *hierarchical* (or command) technique, whereby an Agent A decomposes the initial task, selects the best agents on the basis of their skills, and distributes sub-tasks to them. The second centralized technique is based on a *matchmaker* (MatchMaker, Broker) that puts agents who request and provide services into contact with each other. In the case of the distributed allocation of tasks, there are two main techniques: *allocation by network of acquaintances* and *allocation through call for proposals and bid evaluations* (e.g. contract net [Smith 1980]). Note that these techniques presuppose that the roles of agents are not necessarily fixed. For example, an agent can be a client at one point in time, and a service provider at another time. The dynamic allocation of roles to agents may itself be the subject of coordination techniques based on organizational structures.

Coordination based on organizational structures. This technique focuses on coordinating agents through an organizational structure that provides a

¹<https://www.ushahidi.com/>

space for interaction and sets rules or conventions that agents are subject to in this space. Compliance with these rules aims to organize the work of agents, limit conflicts and reduce communication costs. There are several forms of organization that differ according to, for example ([Ferber 1998], [Gasser 2001]):

- the subordination structure (hierarchical, egalitarian),
- the constitution of the organizational structure (predefined, emerging),
- the level of agent specialization (generalist, specialist),
- the extent of their competence (redundant or not),
- the degree of scalability of the organizational structure (fixed-role agents, agents with variable roles, rule changes).

Drawing on the theory of organizations and sociology, various techniques have been developed by integrating natural, powerful concepts that are as diverse as roles, groups, teams, rules, standards, commitments, responsibilities or permissions. However, organizational techniques are often structured around three key concepts: i) *the roles* found in the organization; ii) *the protocols* according to which these roles interact; and iii) mechanisms for the *allocation of roles* to agents. It should also be noted that MAS design methods have emerged that are focused on an organizational model (e.g., AALAADIN [Ferber 1998] and GAIA [Wooldridge 2000]) and are based on these three concepts. This is due to the triple role of organizational structures, which provide a way to understand an MAS, structure its design, and coordinate its implementation simultaneously. To the extent that an MAS might be deployed in an organization, the system must also have a model of the organization. Finally, it should be noted that this family of coordination mechanisms poses a dilemma: how, in practice, is it possible to define an effective organizational structure without altering the autonomous nature of agents?

Process-based coordination. A process is a coordinated set of tasks implemented by actors (human or material) within an organization to achieve a specific result. It is described by a schema (model) that represents, in a form that supports automatic manipulation, various interconnected perspectives [Van der Aalst 2004].

The perspectives that are generally taken into account are behavioural, organizational and informational [Van der Aalst 2003a]. The behavioural perspective

provides details of tasks and their coordination (linkage rules or control structures). This perspective can be compared to the idea of the plan, as a process describes all of the possible ways to achieve the result. A path therefore corresponds to a plan. The organizational perspective structures the actors involved in the implementation of the process into groups that can play the same role, or groups belonging to a single organizational unit. It also shows the relationships between the actors themselves (hierarchy, delegation, responsibility). The informational perspective represents the structure of forms, documents and data that are used and produced by the process. This information is important because its existence and value may determine the conditions for task execution.

A process is described using a notation (the Business Process Model and Notation standard for example) or a formalism like High Level Petri nets [Van der Aalst 1998]. The advantage of using processes is that dedicated tools are available for their implementation: for example workflow management systems such as Bonita, W4 and YAWL.

Rule-based coordination. Rules are a declarative way to express coordination. The most common form is an expression such as “when <event> if <condition> then <action>”. Coordination is expressed by a set of rules without specifying the control structure needed to activate them; this role is entrusted to an independent inference engine that reasons on the basis of these rules [Anicic 2010]. The course of events decides the order that rules are invoked in and priority mechanisms are used to resolve potential conflicts in the choice of rules to be executed. The combination of events and conditions can express: i) complex situations occurring over time (versus an instantaneous event); and ii) time constraints linking these situations (superposition, precedence, inclusion, etc.) such as Allen’s interval algebra for reasoning about temporal relations found in the ETALIS language [Anicic 2010].

Coordination based on interaction protocols. This technique attempts to coordinate agents by constraining their conversations, i.e. interaction sequences in causal relationships. It focuses on conversations that unfold in line with a predefined schema that determines in particular: i) the purpose of the conversation; ii) the role of each participant; iii) the type of interventions, or elementary interactions that they can carry out; and iv) intervention rules that specify the circumstances in which an agent playing a certain role can (or must) carry out a certain type of intervention.

Thus, a conversation is considered as an occurrence of an Interaction Protocol. Agents remain free to choose when and how to intervene, but their choices

are constrained by the rules of the conversation. Very often, these protocols are simply the computerized adaptation of widely-used procedures found in society, which are optimized to achieve a specific purpose. Auction protocols, voting systems, the Contract Net protocol, or other negotiation protocols are some examples ([Noriega 1999], [Smith 1980]). These protocols can be specified in a distributed manner, i.e., they can be dissolved into each role or, conversely, isolated and specified as an individual component that agents can share and use as an ontology [Hanachi 2004], [Cranefield 2005].

Much work has been developed around protocols for two reasons:

- Firstly, they are a mechanism that combines well with all other mechanisms. Task sharing, negotiation and planning may, at various stages, benefit from the deployment of protocols;
- However, they also make it possible to give a formal semantic to performative agent communication languages such as FIPA-ACL and KQML. In practice, the emission of a message containing a performative transmits the intent of the message and triggers a specific behaviour in both the transmitter and the receiver. The behaviour that defines a performative semantic is easily described by a protocol. Although few attempts have been made to use a formal language (Logic [Pitt 2000], Petri nets [Hanachi 2004]), to model this protocol, it is possible to provide a formal semantic for languages such as KQML or FIPA-ACL.

Coordination of actions by multi-agent planning. A plan is a partially-ordered set of actions and interactions with a purpose, which can be broken down into partial plans. Partial plans can be combined into a global plan following the resolution of potential conflicts between them, and any necessary optimizations. The objective of multi-agent planning is to coordinate the action plans of several agents and avoid conflicts. Planning can be summarized by three steps [Ferber 1995]:

1. the construction of a plan or several partial plans,
2. the coordination of partial plans,
3. the coordinated execution.

There are several forms of planning, depending on whether the steps are performed by one or more agents, whether they are sequential or interlaced, and

whether they are part of a static or dynamic environment. There are essentially three forms of planning that can unfold. Central planning corresponds to a scenario in which a planning agent builds a global plan, then parallelizes and synchronizes it into partial plans that it allocates to different agents for coordinated implementation. The synchronization and allocation can be dynamic (e.g., allocation by tender) or static. Coordinated planning is a technique in which each agent constructs their partial plan; upon receipt of these plans a coordinator modifies, summarizes and synchronizes them, and resolves potential conflicts to produce a global plan. Then, each agent executes their modified partial plan in coordination with the others. Finally, distributed coordination is a technique in which all of the steps are distributed. Each agent creates, negotiates, modifies and executes the actions in their plan as a function of their interactions. This scenario requires that each agent has knowledge of the other agents and exchanges information (plans, goals, etc.) with them [Durfee 1988].

Negotiation aims to resolve or prevent potential conflicts between agents. Conflicts can involve access to resources, different solutions to the same problem, etc. In the case of a conflict, agents will talk to each other to reach an agreed compromise. Although we provide no further details on this family of techniques, it should be noted that conversations implemented to negotiate may also be based on protocols.

To conclude, it should be noted that coordination methods often combine several of these techniques, and that procedures exist to select, in real time, the best technique. Organizational structures and processes are entirely appropriate as a way to understand high-level coordination, which is much sought-after in crisis management information systems. Interaction protocols also seem to be essential, as they underlie most other techniques, including organizational structures.

2.5 Coordination Model Evaluation Criteria

There are a multitude of criteria available to evaluate coordination models and it is difficult – if not even a waste of time – to examine all of them. In practice, their relative importance depends on the modelling context, specifically two main parameters:

- *The domain of application.* For example, in the context of Internet applications [Omicini 2001] (crowdsourcing in crises in our case) the focus is

on criteria such as scalability, interoperability, safety, reliability, efficiency, robustness, flexibility or confidence in the model.

- *The life-cycle phase.* In the design phase, the focus is on criteria such as: the quality of the coordination specification language (expressive power, abstraction, theoretical foundations, validation capacity); the extent of the constraints it imposes on the system architecture and that of agents (centralization/ distribution/ indifferent, the media and communication language); completeness (i.e., its ability to consider all aspects of the intended application); or transposition (i.e., the qualities that cause the model to be produced, or whether one of its components is reusable and/ or can be coupled). At the execution level, the focus is on the model's efficiency (speed, conflict avoidance, etc.), robustness (operating in degraded mode, fault tolerance), responsiveness to change, communication costs or adaptability. Although these criteria are measurable at runtime, simulations make it possible for them to be taken into account from the design stage.

2.6 Conclusion

In this chapter, we first considered the importance and the need for coordination between agents (human, organization, ...) in a multi-agent universe when individual skills, resources or information are insufficient. Coordination helps to improve the division of labor to accomplish shared goals, to synchronize interdependent actions, to satisfy non-functional constraints, and therefore coordination aims at optimizing the overall efficiency of the system.

We have also defined a conceptual framework to specify coordination models. It consists in five parameters: i) the entities to be coordinated; ii) the media and communication languages supporting coordination; iii) coordination protocols or rules; iv) the language used to describe this coordination; and v) the chosen techniques (among artefacts, processes, organizations, protocols, ...). The values of these parameters have to be chosen according, to the coordination context (static or dynamic), and its type (implicit or explicit).

In the next chapter, based on the background presented in this chapter, we will focus on coordination models used in the specific domain of crisis management.

Coordination Models for Crisis Management

Contents

3.1	Introduction	37
3.2	Crisis Domain Analysis	37
3.2.1	Emergency vs Disaster vs Catastrophe	37
3.2.2	Crisis Management Life-cycle	39
3.3	Coordination Models for Crisis Management	41
3.3.1	Coordination in Crisis Management	41
3.3.2	Coordination Models for Crisis Management	41
3.3.2.1	Process-based Models	42
3.3.2.2	Organization-based Models	43
3.3.2.3	Multi-agent based Models	43
3.4	Comparison of Coordination Platforms	43
3.4.1	WORKPAD	44
3.4.2	SoKNOS	44
3.4.3	INDIGO	45
3.4.4	Sahana EDEN	45
3.4.5	USHAHIDI	46
3.4.6	HAC-ER	46
3.4.7	Franke's Framework	48
3.4.8	Comparison of Platforms	48
3.5	Conclusion	50

3.1 Introduction

Large scale catastrophes or crises obviously involve the participation of numerous organisations. It thus forces the individuals involved to interfere and interact with each other. The coordination between stakeholders could therefore be considered as a proper solution (described in Chapter 2) in the context of uncertain and chaotic situations after a disaster. However, it could also raise problems in the case of difficulty to perform effectively in crisis situations. The coordination could then be identified as a critical factor of failure in crisis response.

In this chapter, we present the state of the art on existing platforms devoted to crisis management, from a coordination point of view. In section 3.2, we provide a brief analysis of the crisis domain, then in section 3.3, we highlight the importance of coordination in a crisis situation. Then (section 3.4), we focus on crisis management platforms enabling coordination and compare them according to different criteria.

3.2 Crisis Domain Analysis

In this section, we try to perform a thorough analysis of the crisis domain. First, we define the principal terms in this domain, then we introduce the life-cycle used in a situation of crisis management.

Nowadays, we have to face an increasing number of crises, including *natural or industrial disasters* (e.g. Ebola outbreak in 2014, Nepal earthquake in 2015, Tianjin explosions in 2015, etc.), *man-made disasters* (e.g. Ukrainian conflict of 2013, Greek's economic crisis in 2015 or the European Migrant/Refugee Crisis in 2015, etc.) and even *combined natural & man-made disasters* (e.g. the triple - earthquakes, tsunamis and nuclear - disaster in Japan in 2011) as shown in Figure 3.1¹.

3.2.1 Emergency vs Disaster vs Catastrophe

There is no complete accepted definition of the crisis-related terms in the community. And the researcher could indiscriminately use the terms of emergency, disaster or catastrophe, etc. [Dugdale 2010].

We however try in this section to distinguish between these terms by giving the definitions and to classify them according to their impact of gradation.

¹Source: <http://bernews.com/2012/02/catastrophes-take-toll-on-partnerre/>



Figure 3.1: Disaster of Tsunami in Japan (2011)

- Emergency is an event that interferes with the usual/daily tasks/missions of an organization [Franke 2011a]. Some examples of emergencies are: a road traffic accident, an airplane crash, etc. The emergency response is generally taken in charge by public security organizations such as a fire brigade or the police that already have precise/predefined tasks and objectives about this event response. The number of actors/stakeholders mobilized during an emergency is generally limited [Franke 2011a].
- A disaster is a situation that takes place on a larger scale than an emergency. It can affect the life of more people for a long time. Some examples of disasters are: fire in an industrial area, floods, etc. The usual daily actions (going out, working or go shopping) then become of secondary importance compared to disaster response. During a disaster, several actors/organizations are potentially mobilized. The tasks of these actors are more complex than the ones performed in an emergency and can not necessarily be anticipated. Contrary to an emergency, private actors may complement the public organizations in the field of security (non-governmental humanitarian organizations, for example). The duration of a disaster is generally large.

- A catastrophe is characterised by its extreme impact harming both people and infrastructure [Franke 2011a]. The stakeholders involved in the response must be affected. Moreover, the communication and transport infrastructure is heavily damaged. Some examples of catastrophes are: Nuclear bombings on Hiroshima and Nagasaki (1945), Haiti earthquake (2011), ...

According to its scale, a situation or an event classified as *Emergency*, *Disaster* or *Catastrophe* requires different degrees of coordination between stakeholders through time. In our opinion, the term *Crisis* is the unstable situation that requires urgent assistance after a *emergency*, a *disaster* or a *catastrophe*.

The notion of *Risk* is different from previous terms since it is defined as the possibility of a future event with negative consequences [Bénaben 2008] [Turnbull 2010]. Most of the day life activities contain potential risks e.g. the possibility for two arbitrary cars of being involved in a car crash when moving on the road. Unlike the other terms defined previously, the occurrence of a risk is normally defined during the preparedness phase of the crisis life-cycle (as detailed in the section below).

3.2.2 Crisis Management Life-cycle

Coordination in crisis situations needs to be managed in order to improve the efficiency and effectiveness of response activities. Crisis management can be defined as *the process by which a business or other organization deals with a sudden emergency situation* [Turnbull 2010].

A life-cycle allows to determine different phases of a process and to understand how these different phases are linked together in time. We could find the notion of life-cycle in numerous scientific domains: life life-cycle, software life-cycle, project life-cycle, etc.

There is no standard life-cycle for crisis management. The most accepted one contains the following four phases which are shown in Figure 3.2 [Wallace 1985] [Franke 2011a] [Paulheim 2009].

1. **Mitigation** phase can be considered as a long-term first step. In this phase, we perform the long-term policy regulation with the participation of authorities, as well as provide the training and education on crisis-related issues to all concerned parties. Possible risks and vulnerabilities should be taken into account during this phase to design infrastructure (e.g. building

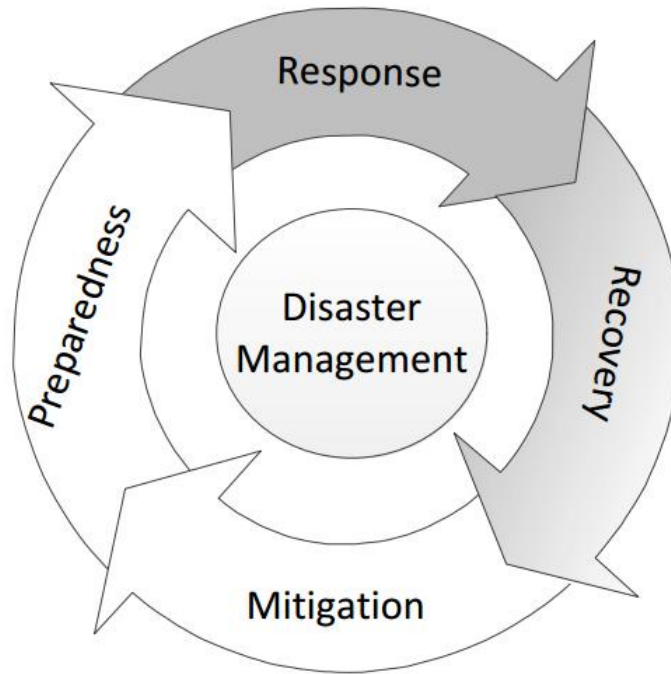


Figure 3.2: Life cycle of Crisis Management [Wallace 1985]

higher dams to protect from a flood), transfer risks (e.g. insurance) or distribute population to reduce impact. Scenarios are built and simulated in reality (i.e. a rehearsal as the one that happened in Da Nang city, Vietnam on 15 May 2011²) or computationally.

2. **Preparedness** phase can be considered as a short-term step (in contrast to the Mitigation phase). This phase is triggered when the notification about a specific upcoming disaster is received. Authorities prepare necessary resources, materials and analyze possible vulnerabilities, risks, scenarios as well as develop precise plans for their response. First responders and local citizens are notified and maintain contact with other involved organizations.
3. **Response** phase happens during the crisis. In this phase we have to mobilize and coordinate during an emergency the activities of various stakeholders to struggle against serious situations. Saving human lives, minimizing

²Live Simulation to test tsunami early warning system in Da Nang City <http://english.vietnamnet.vn/fms/society/7941/vietnam-tests-tsunami-early-warning-system.html>

impact and damages are the most important duties of this phase. Other tasks include warning, evacuation, shelter, feeding, search and rescue, etc.

4. **Recovery** phase is performed after the crisis. In this phase, efforts are focused on the return to normal life: damage assessment, restoration of buildings, restoration of community services, donation and aid management, etc.

[La Rosa 2009] proposes an alternative life-cycle with five phases. The additional phase is called *Planning* and can be included in the *Mitigation* phase. In our work, we will be focusing on the usual four-phase life-cycle described above, because of its clarity and coherence.

3.3 Coordination Models for Crisis Management

Why do we need to coordinate in crisis management? What is the goal of coordination? This section will answer these questions and present some specific coordination models used for crisis management.

3.3.1 Coordination in Crisis Management

Managing a crisis is a massive task to be achieved, overwhelming the capacity of one stakeholder [Prizzia 2008]. Involved organizations have to work together. Coordination takes advantage of the expertise, capability and resources of the various stakeholders. For those reasons, coordination is recognized as a central concern in crisis management [Chen 2008] [Hanachi 2012]. Indeed, coordination is required to ensure the coherent behaviour of the cooperative and distributed actors involved in the crisis resolution.

The goal of coordination in crisis management is multifaceted: managing dependencies between activities performed by stakeholders, orchestrating actions and interactions, avoiding collisions and redundancies, making precise decisions and responding in real-time, saving human lives, reducing the damage rapidly, etc.

3.3.2 Coordination Models for Crisis Management

A coordination model is useful in a crisis context since it facilitates the interdependence between stakeholders, supports the achievement of common goals

(e.g. saving victims), resources (e.g. vehicles, food, houses for victims,...) and sharing of competencies (e.g. medical, carriers,...). Different techniques have to be exploited and/or combined: organizational structure, contracting, negotiating, planning, shared artefacts. In this work, we follow a threefold approach combining *Processes*, *Organizational Structure* and *Multi-agent Systems*.

Most of the existing works regarding coordination models that represent plans, considers event-based, artefact-based or process-aware models. [Ma 2012] proposes a coordination model of emergency response operations via an event-based awareness mechanism. Activities are represented as shared plans and a plan is updated each time an activity state evolves. Users subscribe to events (meaningful states transitions of activity, goal or resource) and are notified as soon as an event interesting to them occurs. [Franke 2013] proposes an inter-organizational approach without the definition of a central process but by allowing organizations to share selected activities and by providing means for detecting and handling conflicts when the state of an activity changes. Each activity has a life-cycle similar to a micro-process.

3.3.2.1 Process-based Models

Considering the opportunities offered by such approach in the context of BPM frameworks, [Van der Aalst 2003b] provides techniques, tools and notations for process engineering including: design, analysis, simulation, execution through a workflow system, monitoring, adaptation and process mining. Process-oriented techniques can be considered as a combination of plans within an organizational structure. Processes are somehow implicit and are neither considered as the main component of the coordination nor are they engineered during their whole life-cycle. The advantage of process-oriented coordination is to provide visibility on the whole crisis evolution: past, present and future activities and their relationships.

Some early works try to apply process modelling for crisis management. For example, [Franke 2011a] made relations between activities explicit by specifying their control-flow. Another example is described in [Le 2013] or [Le 2015] that provides a process model extracted from textual rescue plans. The most advanced practical work is the one developed in the context of Workpad project [Catarci 2011]. It shows the benefits of building a Process-aware Information System for emergency management, and proposes an architecture and a system to support the execution of emergency management processes. Rescue operators are supposed to be equipped with mobile personal assistants and their work is orchestrated

by a workflow system called ROME4EY hosted on the device of team leaders. This work focuses on the design and execution phases of process life-cycle and aims at supporting team works and not the global disaster plan as we consider. [Sell 2009] showed how to use a workflow management system to manage emergency plans. This work is closer to ours but does not address the whole process life-cycle, since it focuses on resource management and delegation functionality, and does not deal with a specific plan as we do with the HCMC one.

In [Franke 2011b], the authors proposed a very detailed review of process management systems supporting disaster response scenarios. However, one main drawback of these systems is to support the real-time crisis management, while we consider the whole life-cycle of the process including the simulation and validation steps.

3.3.2.2 Organization-based Models

Coordination in crisis response involves the participation of numerous stakeholders. Each plays a certain role and has a certain relationship with the others. Organization models or organizational structure (role, relationship, resource allocation) for crisis management are studied in several research efforts [Le 2015].

3.3.2.3 Multi-agent based Models

Coordination in multi-agent systems, such as the one met in crisis management, has been deeply studied in the literature [Omicini 2001] [Smith 2011]. Stakeholders in crisis management can be considered as agents in MAS (Multi-Agent Systems). The problem of coordinating their behaviour has been regularly addressed [Lesser 2014]. [Nwana 1996] introduces the following reasons as why the agents have to coordinate: i) to solve the entire problem which overwhelms the capability or resources of individuals; ii) to meet global constraints; iii) to handle the dependencies between stakeholders' activities; iv) to prevent circumstances of disorder or conflicting actions; v) to profit from the expertise or resources of various stakeholders and vi) to improve the overall efficiency.

3.4 Comparison of Coordination Platforms

The existing coordination platforms are examined according to their ability of discovery, analysis and assessment (efficiency). We first identify the main disaster

management platforms that deal with coordination, we then define criteria that will be used to compare them.

3.4.1 WORKPAD

Workpad³ project [Catarci 2010] was funded by European Commission from 2006 to 2009. Its purpose was to build an adaptive Peer-to-Peer software infrastructure for supporting collaborative work of human operators in emergency/disaster scenarios. Workpad supports collaborative work of human operators during emergency/disaster through mobile devices. Based on their case studies (Civil Protection works in Italy) and user requirements, the project identifies two classes of users: back-end and front-end users. Front-end users are sent to affected areas in order to manage an emergency, while back-end users include the control rooms and headquarters managing the front-end users. A showcase of the project was performed in June 2009 in the town of Pentidattilo (Calabria, Italy). The framework can adapt to changing situations during a disaster.

Workpad can be used in a dynamic context and the coordination follows an explicit form. It combines processes and shared artifacts techniques. It is useful for the preparedness and response phases.

3.4.2 SoKNOS

SoKNOS⁴ (Service-orientierte Architekturen zur Unterstützung von Netzwerken im Rahmen Öffentlicher Sicherheit, German for Service Oriented Architectures for supporting Networks in Public Security) project was funded by the German Federal Ministry of Education and Research [Paulheim 2009] [Babitski 2011]. The main aim of this project was to obtain and integrate heterogeneous sources of information from existing distributed systems (weather, traffic information, ...), that would enable emergency organizations to collaborate efficiently. Decision makers are then able to visually explore the situation by combining different methods and collaboratively solve problems by creating plans.

SoKNOS can be used in a dynamic context with an implicit form of coordination. It combines processes and organization techniques. SoKNOS supports the response and the recovery phases from natural and socio-technical disasters. It is flexible and can adapt to change in current situations.

³<http://www.dis.uniroma1.it/~workpad/>

⁴<http://www.soknos.de/> or <http://www.quizover.com/oer/course/soknos-using-semantic-technologies-in-by-florian-probst-videolectures>

3.4.3 INDIGO

INDIGO⁵ project [Ahmad 2012] was developed by an European consortium (funded by the European Commission through the Security Research program under FP7) leaded by the DIGINEXT company. It is a decision support system for crisis management based on a networked 3D visualization system, simulation tools and mobile communication technologies that provide various scenarios for training of first responders, strategic and operational managers.

INDIGO can be used in static context and under explicit form. It combines organization and planning techniques. This project was applied for the industrial natural gas storage in France in collaboration with local fire-fighters and in a training session in Stockholm (2012). It is useful for mitigation and preparedness phases. It is flexible but has limited adaptation capabilities.

3.4.4 Sahana EDEN

Sahana EDEN⁶ (Emergency Development ENvironment for Rapid Deployment Humanitarian Response Management) [Chan 2012] is an open-source software tool developed by the Sahana Software Foundation⁷ (a non-profit organization). EDEN provides a set of web-based disasters management applications: registry of missing persons, request/pledge management, situation awareness, volunteer coordination, ...

It can be used in both dynamic and static contexts. The form of coordination is implicit and the technique used is shared artifact. This mature platform has been deployed on numerous disasters around the world since more than ten years: 2004 Indian Ocean Tsunami, 2010 Haiti Earthquake, 2013 Typhoon Haiyan in the Philippines, the Resource Management System of IFRC (International Federation of Red Cross and Red Crescent Societies) or 2015 earthquake in Nepal⁸.

It is useful for the preparedness, response and recover phases. Sahana EDEN is designed to be rapidly configured and modified according to the user needs within disaster management.

⁵<http://indigo.diginext.fr/EN/index.html>

⁶<http://eden.sahanafoundation.org/>

⁷<http://sahanafoundation.org/>

⁸<http://nepal.sahana.io/>

3.4.5 USHAHIDI

Ushahidi⁹ [Okolloh 2009] is developed by Ushahidi, Inc. (a non-profit software company) that develops free and open-source software for information collection, visualization and interactive mapping. It allows citizens to report crisis-related data from several sources: the Internet, mobile phones, etc. Ushahidi allows to aggregate and display the collected information on an interactive map. In the context of this PhD thesis, we have deployed an instance of Ushahidi for Vietnam¹⁰, called *Vietnam's Crisis Early-Warning and Response System*, as shown in Figure 3.3.

The coordination form is implicit and based on the input of volunteer citizens using the platform. The coordination technique used is shared artifacts (maps, reports, ...). Maps can contain different icons, pictures or videos of the current crisis, symbols like circles with numbers of crisis occurrences, etc.

It has been deployed for tracking several real situations: Haiti earthquake¹¹, Ebola outbreak¹², Syrian revolution¹³, etc.

Ushahidi is useful for dynamic contexts, mainly during the crisis response phase. However, it can be also used during recovery phase to analyse damages. Ushahidi is flexible because the software can be adapted to different situations and missing crisis categories can be added if needed.

3.4.6 HAC-ER

HAC-ER¹⁴ [Ramchurn 2015] is an acronym for Human-Agent Collectives for Emergency Response. It is a disaster response system to collaboratively plan and carry out tasks in teams (UAV and responders on the ground). It gathers situational awareness information using crowdsourcing (reports) and UAV that could be visualized with the help of a heatmap. HAC-ER then coordinates UAV with first responders and employs a provenance tracking and analysis tool. Its users are emergency responders (tactical and operational levels).

HAC-ER can be used in dynamic contexts and under implicit form. The coordination technique is shared artifacts. It was applied during the Haiti Earthquake

⁹<https://www.ushahidi.com/>

¹⁰<http://vcers.byethost7.com/ushahidi-2.7.4/>

¹¹<http://blog.ushahidi.com/2012/01/12/haiti-and-the-power-of-crowdsourcing/>

¹²<http://www.ushahidi.com/2014/11/04/tracking-ebola-crisisnet-ushahidi-platform>

¹³<http://blog.crisis.net/syrian-social-media-journalists-secret-weapon-in-the-crisis-data-revolution>

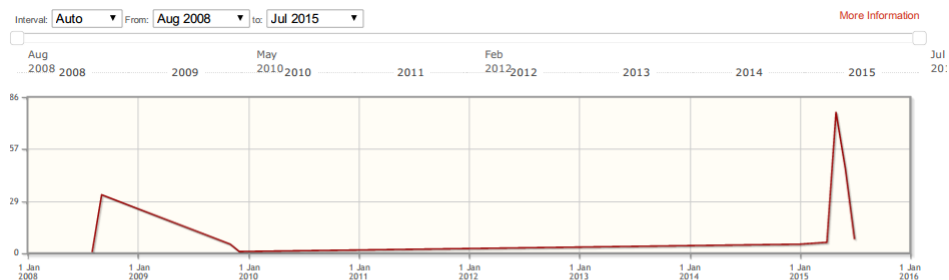
¹⁴Human-agent collectives in action <https://vimeo.com/119525848>

VCERS

Vietnam's Crisis Early-Warning and Response System

[SUBMIT A REPORT](#)OTHER DEPLOYMENTS [\[HIDE\]](#)

Welcome to Vietnam's Crisis Early-Warning and Response System portal. This is an experimental version deployed by Nguyen Tuan Thanh LE in order to help you report information about any disaster, may affect Vietnam. I am thankful for your contribution !

[HOME](#) [REPORTS](#) [SUBMIT A REPORT](#) [GET ALERTS](#) [CONTACT US](#)
**Official & mainstream news**

TITLE	SOURCE	DATE
Green alert for tropical cyclone LINFA-15....	Global Diaster...	Jul 4 2015
Green alert for tropical cyclone NANGKA-15....	Global Diaster...	Jul 4 2015
Green alert for tropical cyclone CHAN-HOM-15....	Global Diaster...	Jul 4 2015
Green alert for tropical cyclone RAQUEL-15....	Global Diaster...	Jul 4 2015
Viet Nam: Closer to bringing drinking water...	Vietnam - Unite...	Jul 3 2015
Orange earthquake alert (Magnitude 6.1M,...	Global Diaster...	Jul 3 2015

Reports

TITLE	LOCATION	DATE
Green alert for tropical cyclone CHAN-HOM-15....	Marshall Islands + Japan	Jul 4 2015
Green alert for tropical cyclone NANGKA-15....	Marshall Islands	Jul 4 2015
Green alert for tropical cyclone LINFA-15....	Philippines, China	Jul 4 2015
Green alert for tropical cyclone RAQUEL-15....	Solomon Islands	Jul 4 2015
Orange earthquake alert (Magnitude 6.1M,...	Philippines	Jul 3 2015
Green earthquake alert (Magnitude 6.4M,...	China	Jul 3 2015

Figure 3.3: Vietnam's Crisis Early-Warning and Response System on top of Ushahidi platform

in 2010. It is useful for response and preparedness phases. This tool is flexible and can be adapted to the current situation.

3.4.7 Franke's Framework

Franke's framework for (temporal) coordination in dynamic situations (such as crisis management) is part of the SoKNOS project [Franke 2011a] [Franke 2013]. This framework provides an open and distributed collaboration environment for coordinating the activities in dynamic situations (i.e. goals change frequently) by modelling activities with states and their temporal dependencies/constraints while detecting/highlighting the deviations from what has been done and what was expected to be done. Its users are disaster managers.

This framework can be used in dynamic contexts and under implicit form. It combines process-based and artifact-based techniques. It is useful for the response phase. It is flexible and can adapt in the situation at hand.

3.4.8 Comparison of Platforms

We want to be able to compare the coordination models in crisis management along the following axis: discovery, analysis and assessment (efficiency). We introduce here criteria to compare the previous platforms based on the coordination parameters, the perspectives addressed, the crisis management life-cycle and their adaptation or flexibility:

- **Coordination Parameters** - Which crisis context does the platform support: a *static* or a *dynamic* one? Which coordination form does the platform support: an *explicit* or an *implicit* one? Which coordination techniques does the platform support: *Contracts*, *Organization*, *Negotiation*, *Planning*, *Shared Artifacts*, *Interaction Protocols*, *Processes* or *Rule-based*?
- **Perspectives Addressed** - Does the platform support an information model, an organizational model or a task model?
- **Coordination Engineering Support**¹⁵ - Does the platform provide means to support coordination model engineering (design, simulation, validation, ...)?

¹⁵“++” means “fully supported”; “+” means “partially supported”; “-” means “not presented”

- **Crisis Management Life-cycle** - Which phases of the Crisis Management Life-cycle does the platform support?
- **Adaptation or Flexibility** - Does the platform can adapt to modifications of the situation/environment? Is the platform flexible?

Tables 3.1 and 3.2 show the results of our comparison.

Table 3.1: Comparison Table of Crisis Management Platforms (1)

	Coordination Parameters			Perspective Addressed			Coordination Engineering Support		
	Context	Form	Techniques	Infor. Model	Org. Model	Task Model	Design	Simulation	Analysis
WORKPAD	Dynamic	Explicit	Process & Share artifact	X		X	+	+	-
SoKNOS	Dynamic	Implicit	Process & Organization	X	X	X	+	+	-
INDIGO	Static	Explicit	Planning & Organization	X	X		++	++	+
SAHARA	Static & Dynamic	Implicit	Shared artifact				+	-	+
USHAHIDI	Dynamic	Implicit	Shared artifact				+	-	+
HAC-ER	Dynamic	Implicit	Shared artifact	X			+	++	+
FRANKE's	Dynamic	Implicit	Process & Shared artifact			X	+	-	-

Table 3.2: Comparison Table of Crisis Management Platforms (2)

	Crisis Management Life-cycle				Adaptation Flexibility
	Mitigation	Preparedness	Response	Recovery	
WORKPAD		X	X		Adapt.
SoKNOS			X	X	Adapt. & Flex.
INDIGO	X	X			Flex.
SAHARA		X	X		Adapt. & Flex.
USHAHIDI			X	X	Adapt. & Flex.
HAC-ER		X	X		Adapt. & Flex.
FRANKE's		X			Adapt. & Flex.

According to the comparison tables, most existing coordination frameworks have clear shortcomings. However, the SoKNOS platform is the most complete one since it takes into account all the perspectives and it couples the main coordination techniques: process and organization. However, most of them only have either a very weak engineering support or none, while this support is necessary

to ease coordination model design, simulation and validation. This property is of paramount importance to guarantee the deployment of verified and validated systems because of the serious consequences that may occur in a crisis situation. We can notice in our comparison that the INDIGO framework which has a good coordination engineering support, does not cover the response phase (see Table 3.2) and does not take into account the task perspective (see Table 3.1).

In the following chapters, we will show how our solution overcomes this engineering problem while meeting most of the other requirements.

3.5 Conclusion

In this chapter, we have introduced the specificities of coordination in the crisis domain. We showed how coordination models intervened in crises and we provided a comparison of existing coordination platforms.

We also defined the crisis-related terms (emergency, disaster and catastrophe) and presented a well-known life-cycle for crisis management consisting of four phases: mitigation, preparedness, response and recovery. Then, we presented different coordination models for crisis management built on top of process, organization and multi-agent aspects. Finally, we provided a comparison of coordination platforms for crisis management (*WORKPAD*, *SoKNOS*, *INDIGO*, *Sahara EDEN*, *USHAHIDI*, *HAC-ER* and *Franke's Framework*) and revealed the insufficiency of existing works.

In the next chapter, we will give more details about our approach for engineering coordination models. We will explain the advantages of combining workflow and multi-agent system paradigms. We will then provide a life-cycle from data preparation then model generation and model transformation to model analysis. Finally, we will highlight the contributions of the research presented in this thesis.

Part II

Contribution

Overview of Our Approach

Contents

4.1 Introduction	54
4.2 The Engineering Lifecycle of Coordination Models	57
4.2.1 Advantages of Combining Workflow and Multi-Agent System Paradigms	59
4.2.2 Preparation of Coordination Data	60
4.2.3 Generation of Coordination Models	61
4.2.4 Transformation of Coordination Models	62
4.2.5 Analysis and Improvement of Coordination Models	62
4.3 Our contribution	64
4.3.1 A Scenario-based Approach	65
4.3.2 Model transformations	65
4.3.3 New Metrics and a Tool for Evaluating Organizations	65
4.4 Conclusion	67

4.1 Introduction

Generally, coordination plans (reports, guidelines, etc.) that are drawn up between crisis stakeholders (i.e. government organizations, NGOs, public or private actors) are presented in the format of a textual document, such as the one¹ shown in Figure 4.2.

¹Decision No. 3558/QĐ-UBND enacted by People Committee of HCMC on 13 July 2012 about *Plan to Prevent, Respond and Overcome Earthquake, Tsunami* <http://www.phongchonglutbaotphcm.gov.vn/?id=51&cid=4395>

ỦY BAN NHÂN DÂN CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
THÀNH PHỐ HỒ CHÍ MINH **Độc lập - Tự do - Hạnh phúc**

PHƯƠNG AN
Phòng ngừa, ứng phó, khắc phục hậu quả động đất, sóng thần
trên địa bàn thành phố

(Ban hành kèm theo Quyết định số 3558/QĐ-UBND
ngày 13 tháng 7 năm 2012 của Ủy ban nhân dân thành phố)

PHẦN I
MỤC ĐÍCH VÀ YÊU CẦU

I. SỰ CẦN THIẾT:

Động đất rất khó dự báo trước, từ lúc xuất hiện các dấu hiệu xảy ra động đất đến khi xảy ra động đất chỉ trong thời gian rất ngắn nên rất khó cho các nhà khoa học để dự báo chính xác thời điểm và vị trí xảy ra động đất. Khả năng dự báo chỉ dựa vào tài liệu thống kê tần suất xảy ra động đất trong lịch sử.

Đối với sóng thần gây ra do động đất ở xa, thời gian lan truyền của sóng thần từ khu vực xảy ra động đất tới vùng bờ là thời gian tối đa để vận hành hệ thống cảnh báo và di tản người dân đến nơi an toàn.

Figure 4.1: An extract from the textual Rescue Plan drawn up by the People's Committee of Ho Chi Minh City, Vietnam on 13 July 2012 (written in Vietnamese)

PART I: PURPOSES AND REQUIREMENTS

I. THE NECESSITY

Earthquakes are difficult to predict, since the signs of its appearance until the time of its occurrence is a very short time. So it is difficult for scientists to predict the exact time and location. Predictability only depends on document frequency statistics earthquake in history.

For a tsunami caused by a distant earthquake, its propagation time from the earthquake area to the coast is the maximum time to operate the warning system and evacuate people to the safe places.

So, this plan focuses on developing the basic situation and assigning the roles, common tasks for the Departments and the Divisions involved in the preparation of forces and means; the organizations of communications; the rehearsal of situations; the evacuation and the relocation of people to ensure the effectiveness of search and rescue work, to overcome the consequences of an earthquake/tsunami for the aim of timely response in every situation.

II. THE PURPOSES

1. The plan ensures the unified direction, the close coordination between the organizations in order to

Figure 4.2: An extract from the translated version of the textual Rescue Plan drawn up by the People's Committee of Ho Chi Minh City, Vietnam on 13 July 2012

In a crisis, community leaders use them to manage the situation and issue commands to first responders. This hierarchical coordination (i.e. master-slave cooperation) leads to time-consuming and inefficient coordination. Moreover, it is difficult to compare plans in order to analyse and evaluate the quality of coordination between stakeholders, for example.

In this chapter we propose an approach for modelling, simulating and analysing coordination between crisis management stakeholders based on the combination of two perspectives: *process* and *organizational*, which are discussed in more detail in Chapters 5 and 6, respectively.

Chapter 5 presents a process perspective of coordination in a crisis situation based on the Petri net formalism and the Business Process Model and Notation (BPMN). A scenario-based lifecycle approach [Rolland 1998] is proposed which is compliant with a crisis meta-model that we define. We apply the approach to process design and validation in order to identify top-level objects and the links between them based on crisis plans in text format. Scenarios are generated that make it possible to discover processes by means of Petri nets using the process mining α -algorithm. A BPMN diagram that integrates organizational aspects is derived from the discovered Petri nets and additional information provided by stakeholders. This diagram supports scenario-based (i.e. “what-if”) simulations and analysis, together with complexity metrics.

Chapter 6 is devoted to the organizational perspective of crisis coordination, which is abstracted using Multi-Agent Models (MAM). A Multi-Agent System consists of intelligent agents that are able to communicate and interact with each other in an environment in order to achieve their goals. We classify multi-agent systems into two categories: Agent-centered Multi-Agent Systems (ACMAS) such as Belief–Desire–Intention (BDI) agent that focus on individuals; and Organization-centered Multi-Agent Systems (OCMAS) such as the Agent–Group–Role (AGR) model that are focused on social/organisational aspects. While Chapter 6 focuses on OCMAS, we also show how to generate ACMAS models from BPMN diagrams. We assess the organizational structure of the coordination plan using a role graph [Grossi 2007], and show how organizational models (AGR and role graphs) can be derived from the process models described in Chapter 5. These are then refined to obtain an ACMAS view represented using BDI agents. These model transformations are implemented in the ATL platform.

In this chapter, we briefly introduce our approach to crisis management coordination models (shown in Figure 4.3) and outline our contributions. Our

approach is structured into four hierarchical layers: i) *paradigms*, ii) *views*, iii) *languages* and iv) *evaluation*.

With respect to the top layer, crisis management coordination can be divided into two paradigms: the *workflow paradigm* (Chapter 5) and the *multi-agent paradigm* (Chapter 6).

Regarding the second layer (views), the workflow paradigm takes a process view, while the multi-agent paradigm can take an agent- and/or organization-centered view.

In the third layer, each view is represented by specific languages (i.e. models). For the process view (Chapter 5) this takes the form of either Petri nets (Section 5.5.1) or a BPMN diagram (Section 5.5.3). The agent-centered view uses a BDI-type agent (Section 6.2.2), while the organization-centered view relies upon an AGR model (Section 6.2.1) and/or a role graph (Section 6.3). Transformations are possible between views: from the process to the organization view and/or agent-based view (Section 6.4).

Finally, the fourth layer concerns analysis and evaluations. For processes (Chapter 5), we verify their properties (reachability, termination), evaluate their complexity using metrics, and simulate resources and time. For organizational structures (Sections 6.3.1 and 6.3.4), we evaluate their flexibility, robustness and efficiency using metrics such as *completeness*, *connectedness*, and *univocity*. For agents, the analysis could extend to behaviour complexity (reactive, deliberative, etc.), but we did not investigate this aspect.

4.2 The Engineering Lifecycle of Coordination Models

In the general domain of coordination (cf. *Context - Form - Techniques* shown in Figure 2.2, Chapter 2), we limit ourselves to the *dynamic context* and *explicit form*, and apply *process* and *organizational* techniques for crisis management. Our four-layer approach (cf. Figure 4.3) is described in detail and is based on a five-step lifecycle (cf. Figure 4.5).

The first step is *Preparation of Coordination Data*. It aims to collect crisis-related data that can be classified into two categories: i) *explicit forms* (e.g. textual document, official guidelines, human perceptions, expert knowledge, structured event logs); and ii) *implicit forms* (e.g. unstructured data from social networks or crisis websites). Here, our work focuses on *explicit forms*.

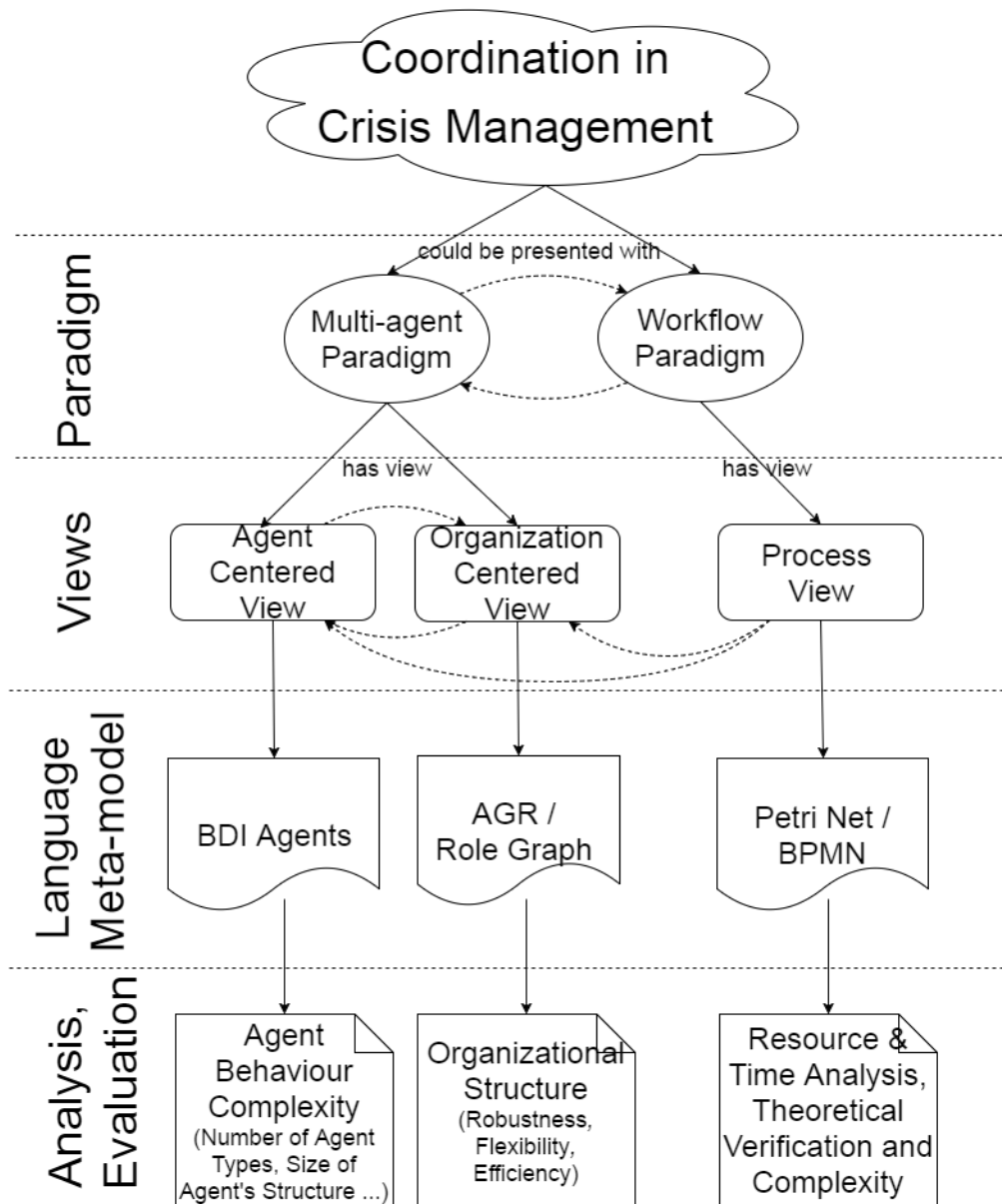


Figure 4.3: Four layers of crisis management coordination

The second step is called *Generation and Discovery of Coordination Models*. Based on the data collected in the first step, process models for crisis coordination are built using the workflow paradigm (see Chapter 5). The modelling can be performed manually by stakeholders or experts, semi-automatically, or discovered automatically from historic event logs [Rozinat 2009]. Organizational and agent models can also be built from these data.

The third step is called *Transformation of Coordination Models*. In this step, we provide mapping algorithms to automatically derive MAM from a process model. Consequently, we automatically obtain both *ACMAS* and *OCMAS* models (see Chapter 6) rather than carrying out this work manually.

The fourth step, called *Simulation of Coordination Models*, consists of simulating the three previous models to better understand and improve them. For example, simulations can be performed on both the process model (i.e. discrete-event simulation, see Section 5.6) and/or MAM (at both the agent and organizational levels).

Finally, the fifth step is called *Analysis, Evaluation and Improvement of Coordination Models*. The idea here is to analyse the three coordination models and their simulations in order to evaluate and improve them. We define static and dynamic evaluation criteria (Sections 5.6.5 and 6.6) that can provide useful recommendations for the authorities in order to improve crisis management coordination.

4.2.1 Advantages of Combining Workflow and Multi-Agent System Paradigms

The workflow paradigm (Chapter 5) expresses informational and organizational aspects. Informational models describe the forms, documents and data used and produced by a workflow. Process models define tasks and their coordination and can be described using languages such as *Petri nets* (Section 5.5.1, Chapter 5), *BPMN* diagrams (Section 5.5.3, Chapter 5), *State Charts* or *Event-driven Process Chains*. Organizational models define the allocation of roles, resources, authorizations, the delegation of tasks, etc. Workflow techniques are the most useful way to present crisis resolution plans because:

- Workflow models can capture emergency plans, as demonstrated in [Sell 2009].
- Once the plans are modelled as processes, they can be implemented, simulated, then analysed or executed in a workflow management system such as

YAWL² (Section 5.6.2). Mining tools such as PROM³ (Section 5.6.4) can be used to discover new models from event logs and to check compliance with pre-defined models.

- Workflow provides a variety of descriptive languages at different levels of abstraction. Petri nets can be used for theoretical analysis and simulation, unlike the BPMN specification that is used for validation by end-users. In addition, tools such as YAWL offer a specific notation and provide operational models.

On the other hand, the multi-agent paradigm (Chapter 6) can be used to present both individual (ACMAS), and social/organizational (OCMAS) aspects. While ACMAS provides realistic simulations that take into account specific, individual behaviours, OCMAS provides a way to represent and evaluate high-level organizational and social aspects. Finally, the coupling of these two paradigms is possible, thanks to their shared concepts (organization, activity, roles, etc.).

4.2.2 Preparation of Coordination Data

Crisis management is a real, complex system with a huge data to be considered. Crisis data comes from diverse sources and in different formats including: *unstructured* (e.g. a written plan), *semi-structured* (e.g. reports from crisis websites, tweets, comments on Facebook), or *well-structured* (e.g. event logs). Each format requires an appropriate method to mine the value in the raw data. The usual flow is from *unstructured*, to *semi-structured*, to *structured data*. The target format (i.e. structured) is then used as the input for automatic modelling and simulation techniques.

However, not all unstructured data can be transformed into a structured format. Response plans in text format are one of the most difficult formats to work with, despite their widespread use in crisis management (Figure 4.2). These plans do not contain any meta-data to support their processing. Therefore, guidelines are usually processed manually based on the expertise of stakeholders [Le 2013], [Le 2015].

In our work, in addition, to collect crisis reports from websites, we deployed an instance of Ushahidi (Section 3.4.5, Chapter 3), Vietnam’s Crisis Early-Warning

²Yet Another Work-flow Language <http://www.yawlfoundation.org/>

³Process Mining tool <http://www.promtools.org/prom6/>

and Response System (VCERS⁴).

Unlike textual plans, *event logs* are expressed in a well-structured format and are generated by real world crises and/or training exercises. Thus, they contain comprehensive information about time, activities, actors, receivers, etc., that facilitates automatic post-processing. One of the best-known techniques for processing event logs is *Process Mining*, introduced by [Van der Aalst 2005] (Section 5.6.4, Chapter 5).

4.2.3 Generation of Coordination Models

The crisis data collected in the previous step is then used to build coordination models based on *process*, *organizational* and *agent* views. The advantages of some of the modelling languages used in crisis management are given in Table 4.1.

Table 4.1: Advantages of coordination plan representations

Models	Description and Advantages
Petri nets	A directed bipartite graph based on resources (tokens), formal semantics, supporting behavioural analysis and macro-simulations of processes
BPMN Model	An understandable and aggregate representation of stakeholders' behaviour supporting macro-analysis and macro-simulation of processes [Le 2013]
Role Graph Model	Expressing dependency between the roles and enabling analysis such as robustness, flexibility and efficiency of organization structure [Le 2015] [Grossi 2007]
BDI Agent Model	A typical ACMA representation enabling micro simulations of agents [Endert 2007]
AGR Model	A complete OCMAS representation enabling macro (organization) and micro (agents) simulation of [Ferber 2004]

Process models (e.g. Petri nets or BPMN diagrams) provide an intuitive and graphical view that can be understood by decision-makers or end-users. They

⁴Vietnam's Crisis Early-Warning and Response System <http://vcers.byethost7.com/ushahidi-2.7.4/>

support macro-simulations and the analysis of resource and time utilization, resource allocation strategy, complexity or total costs within processes (Section 5.6).

The *role graph* model focuses on dependencies and enables the analysis of organizational structure (robustness, flexibility and/or efficiency) (Section 6.3).

The *BDI Agent model* (consistent with the ACMAS paradigm) supports micro-simulations (i.e. individuals). Finally, the *AGR model* is consistent with the OC-MAS paradigm, and provides both macro- and micro-simulations [Ferber 2004] (Section 6.2).

4.2.4 Transformation of Coordination Models

The different coordination models in a rescue plan can be built independently. However, they can also be mapped from one to another (e.g. Petri nets to BPMN diagrams, BPMN diagrams to role graphs, BPMN diagrams to BDI Agents, BPMN diagrams to AGR models, etc.). Here, we propose some guidelines and/or algorithms to facilitate these transformations (Sections 5.5.3.2 and 6.4). In our work, process models are the source of mappings. MAM (e.g. role graph, BDI Agent model, or the AGR model) are then derived from these models. The mappings help to develop a multi-view of crisis management, and support post-processing (analysis, simulation), as shown in the first part of Figure 4.4.

These transformations connect an *activity-event* and a *multi-agent* paradigm. More precisely, we use a Model-to-Model transformation based on ATL⁵ to map one meta-model to another (for example, taking the Petri net meta-model as the source and a BPMN meta-model as the destination, or taking a BPMN meta-model as the source and mapping to a AGR meta-model as the destination).

Regarding mapping from process models to ACMAS models, certain agent notations can be derived directly from process models, these include: *Environment*, *Agents*, *Agent behaviour / activities*, and *Agent interactions / communications*. However, abstract and/or mental concepts (such as *Agent objectives / motivations*, *Agent aptitudes*, *Agent beliefs* or *Agent skills*) do not exist in process models and cannot be derived in the multi-agent world.

4.2.5 Analysis and Improvement of Coordination Models

The combination of process models and Multi-Agent System models makes it possible to carry out diverse analysis based on the strengths of two representations

⁵ATL Transformation Language <https://eclipse.org/atl/>

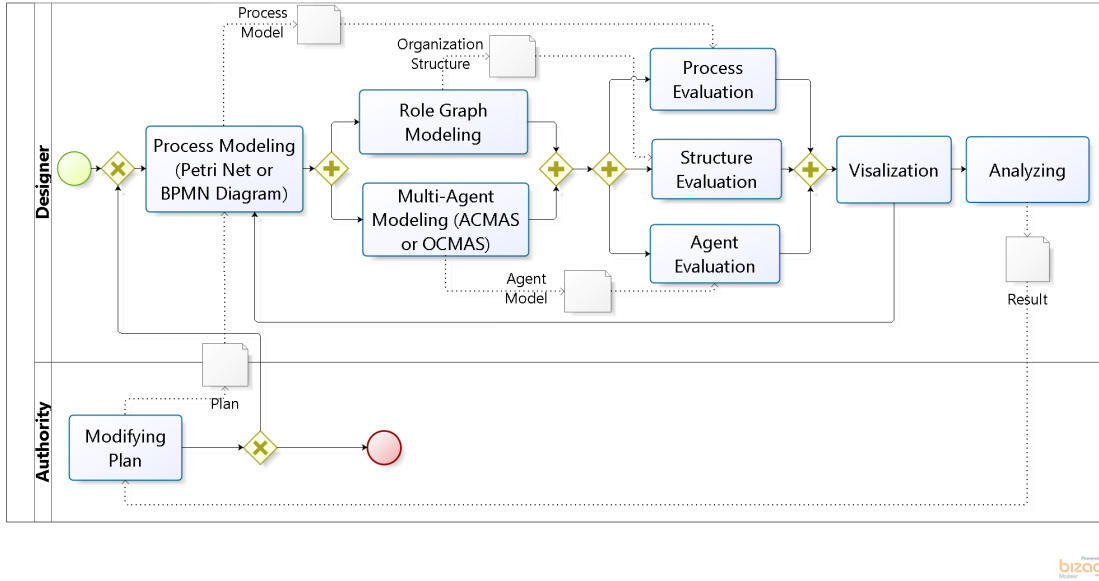


Figure 4.4: Lifecycle for evaluation and mapping from process models (*Petri nets* and *BPMN diagrams*) to organization models (*role graphs*, *ACMAS* and *OCMAS models*)

(e.g. control flow complexity for process models, organizational structure qualities for OCMAS models) in order to improve the quality of coordination between stakeholders [Cardoso 2008] [Grossi 2007]. Process models are used to capture the activities and messages exchanged between the actors involved in a crisis, while an OCMAS model is used to represent their roles, interactions and the organizational structure.

The second part of Figure 4.4 shows our evaluation and validation approach with respect to the designer and the authority (Sections 5.6.5 and 6.6). The evaluations of process and organizational aspects support each other, as the first abstracts the coordinated behaviour of actors, while the latter abstracts their relationships (control, coordination, power, etc.) which influences the efficiency, robustness and flexibility of a crisis rescue plan.

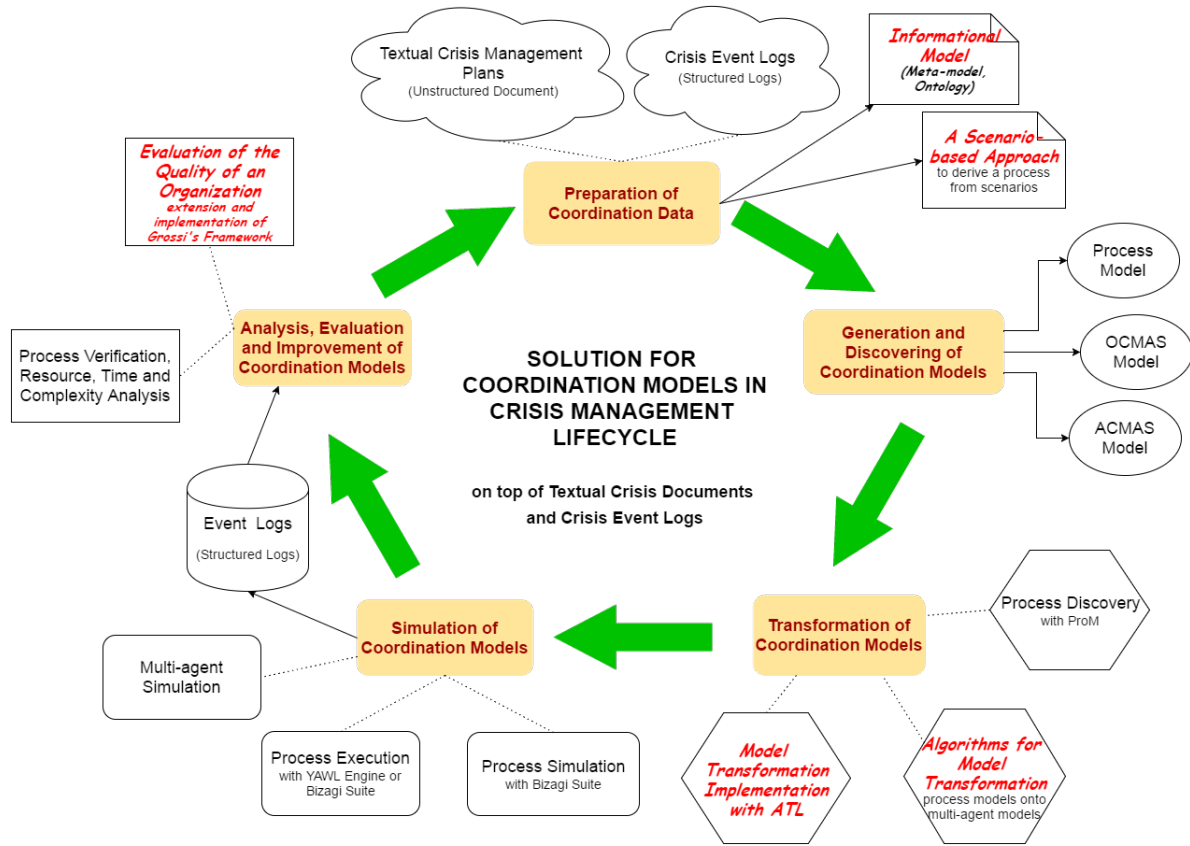


Figure 4.5: Our contribution (shown in red italics) based on our lifecycle solution for crisis management coordination

4.3 Our contribution

Our lifecycle approach provides a complete solution for the explicit coordination of crisis management. It describes the coordination models that can be extracted from textual rescue plans and/or event logs. Figure 4.5 highlights our contributions (Section 4.2). In this figure, the novelty of our work is shown by the words highlighted in red italics.

In the following sections, we describe in detail our three main contributions.

4.3.1 A Scenario-based Approach

Our first contribution concerns the *Crisis Perception* phase (Section 5.3, Chapter 5). It consists in the definition of a scenario-based approach (a crisis meta-model, guidelines, recommendations, etc.) to identify actors, their interactions and tasks, and to provide scenarios that help to derive the crisis resolution process. We developed a Prolog program that is able to generate scenarios from the meta-model, together with a general coordination meta-model that provides an overview of all the concepts involved in crisis management (Figure 4.6), and which links the agent and workflow paradigms. Furthermore, our work refines the activity coordination aspect (Figure 5.3, Chapter 5) and builds scenarios. The general meta-model shown in Figure 4.6 extends the work of [Smith 2011] and has been implemented as an ontology in the Protégé⁶ framework (Appendix C.1).

4.3.2 Model transformations

Our second contribution is in the model transformation phase (Section 6.4, Chapter 6). We developed and implemented (using the ATL platform) transformation algorithms to derive OCMAS and ACMAS models from process models.

4.3.3 New Metrics and a Tool for Evaluating Organizations

Our third contribution concerns the analysis, evaluation and improvement of the coordination phase (Section 6.6, Chapter 6); in particular, the evaluation of the organizational dimension of a crisis plan. We developed a Smalltalk application to implement the framework developed by Grossi et al. [Grossi 2007] for evaluating an organizational structure (in terms of robustness, flexibility and efficiency). More precisely, we implemented metrics in the Pharo⁷ environment. The source code for our application, named the *AgentOrganizationEvaluation-Model*, is included as Appendix D.1.

In addition, we extended the Grossi theoretical framework to concrete, real-world organizations. While an abstract organization is theoretical (consisting of actors and the *a priori* links between them), a concrete organization takes into account real links between actors in an actual situation.

⁶Protégé framework <http://protege.stanford.edu/>

⁷Pharo environment <http://pharo.org/>

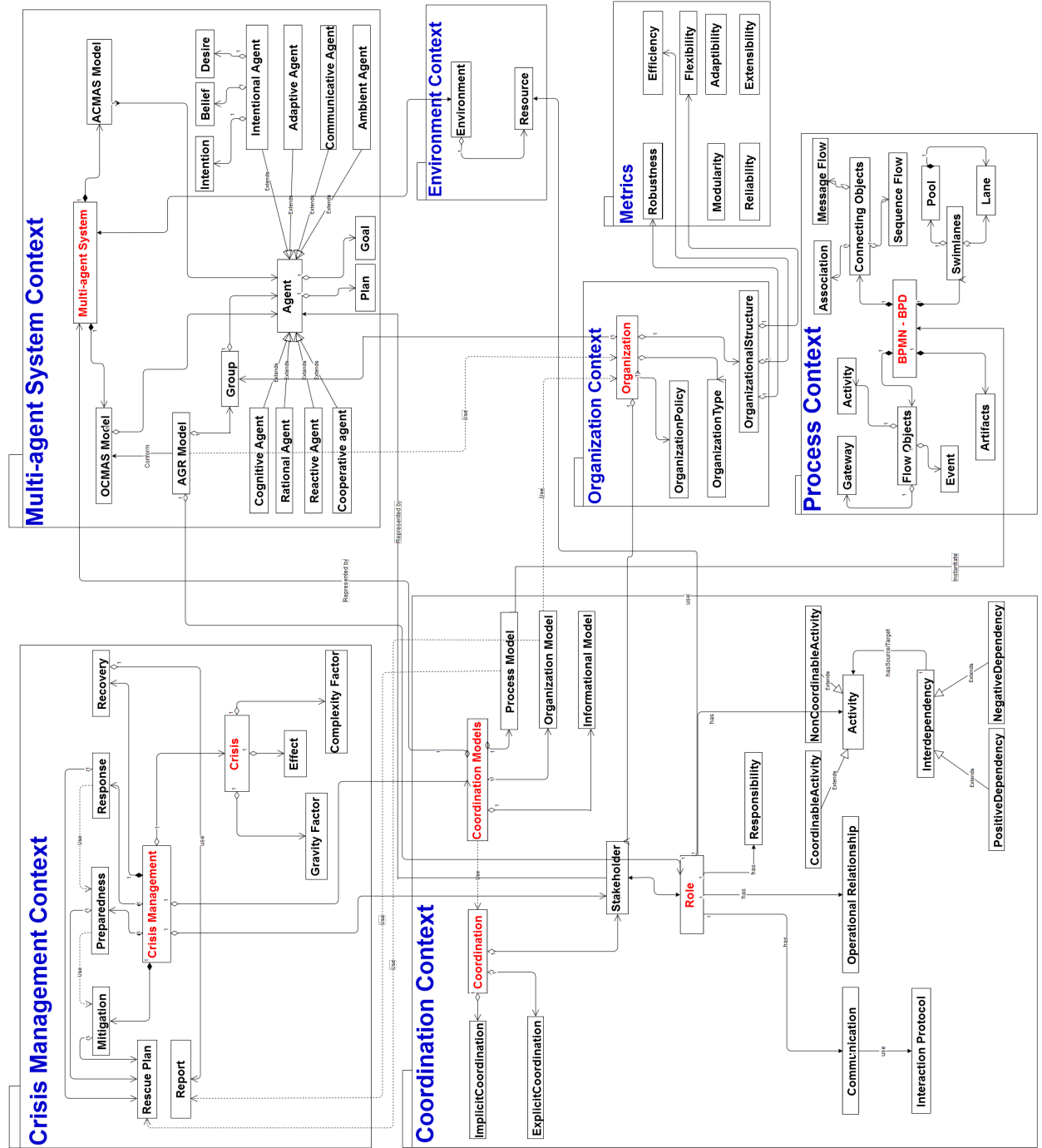


Figure 4.6: Meta-model of crisis coordination including the notions of: *crisis management*, *coordination*, *process*, *organization*, *multi-agent systems*, *environment* and *metrics*

4.4 Conclusion

In this chapter, we presented an overview of our work and our contributions to the problem of crisis management coordination.

Section 4.2 outlines our approach, which consists of four hierarchical layers: *paradigms*, *views*, *languages* and *evaluations*. We then describe our work in detail based on a five-step lifecycle: i) preparation of coordination data; ii) generation and discovery of coordination models; iii) transformation of coordination models; iv) simulation of coordination models; and v) analysis, evaluation and improvement.

Section 4.3 highlights our three main contributions: i) a new scenario-based approach to derive a process from scenarios based on a crisis coordination meta-model; ii) algorithms designed to transform process models into MAM; and iii) the implementation of new metrics to evaluate the quality of an organization.

Our approach and contributions are described in more detail in the next two chapters. Chapter 5 is dedicated to the process perspective, while Chapter 6 discusses the organizational perspective, based on the multi-agent paradigm.

Process-based Coordination Models

Contents

5.1	Introduction	69
5.2	Design of a Crisis Resolution Process from Scenarios	71
5.3	Identification of High-level Objects from Text	72
5.3.1	Principles	72
5.3.2	Illustration: Identifying Organizations, Tasks and their Relationships from HCMC's Tsunami Response Plan	74
5.4	Building Scenarios	76
5.4.1	Scenario Structure	76
5.4.2	Generation of Scenarios in the Design Phase	79
5.5	Process Discovery	83
5.5.1	Process Representation using Petri nets	83
5.5.1.1	Petri nets	84
5.5.2	Process Discovery using the α -algorithm	84
5.5.2.1	Application to HCMC Scenarios	87
5.5.3	BPMN Representation of Stakeholder Validation	87
5.5.3.1	BPMN	88
5.5.3.2	Mapping Petri nets onto BPMN	89
5.6	Process Simulation, Analysis and Conformance Checking	91
5.6.1	Theoretical Verification	93
5.6.2	Process Deployment and Execution	93
5.6.2.1	Resource and Time Analysis	96
5.6.3	Process Simulation and Validation	96

5.6.3.1	Defining Simulation Parameters	96
5.6.3.2	Defining Simulation Scenarios	100
5.6.3.3	Simulation and Analysis of Scenarios	103
5.6.4	Process Conformance Checking and Organizational Discovery with the PROM Framework	105
5.6.5	The A4BP Tool for Measuring Process Complexity	108
5.7	Conclusion	112

5.1 Introduction

This chapter explores a process-oriented view of coordination in a crisis, and proposes a tool-based method to facilitate the engineering of these processes. As we have indicated in Section 5.2, a crisis resolution plan, usually in text format, can be abstracted by a process. In this chapter we will show how the process paradigm is useful in the design, analysis and simulation of a crisis resolution plan, at a high enough level of abstraction to facilitate its validation by the stakeholders and the authorities responsible for its preparation and implementation. This high level of abstraction is deliberate, as we do not consider the process as a prescriber object (driver) but as a guide to resolve the crisis. This is how it is understood by stakeholders who have room for manoeuvre to cope with unforeseen situations. It allows these stakeholders to identify the principal landmarks that punctuate crisis management, while maintaining control and without risking challenges, given their responsibilities.

A second focus is on the coordination of the activities of actors regardless of how each activity is carried out, given that it depends on their professional knowledge. A high level of abstraction does not exclude the definition of rigorous models and their verification.

The novelty of our approach lies in:

1. The design itself, which is based on scenarios consistent with a crisis meta-model.
2. The design is developed using a model discovery technique from scenarios based on mixed sources of information (human, textual, artificial, etc.)

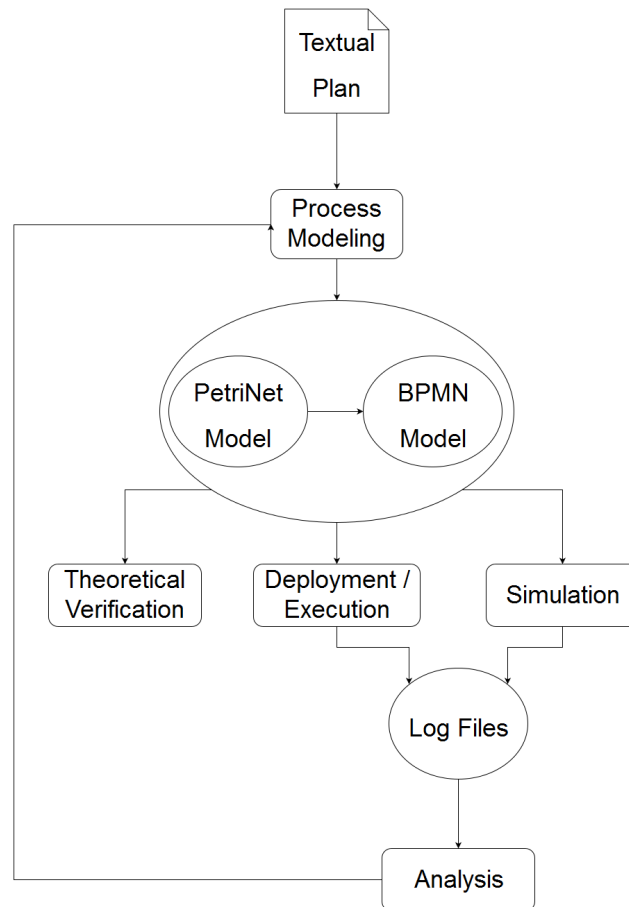


Figure 5.1: Life-cycle of the engineering crisis process model

5.2. DESIGN OF A CRISIS RESOLUTION PROCESS FROM SCENARIOS⁷¹

and known *a priori*. In the literature, discovery is most often based on *a posteriori* historical elements.

3. We exploit both Petri nets and Business Process Model and Notation (BPMN) formalisms.
4. Our approach covers the processes life-cycle (Figure 5.1) and assures interoperability between phases due to the choice of notation.

We illustrate our ideas with the crisis management plan prepared by Ho Chi Minh city officials (HCMC) in Vietnam to handle a tsunami. An excerpt is given in Figure 4.2 and the complete document is available from the website of *Committee for Flood-Storm Prevention and Search-Rescue* of HCMC¹.

The rest of the chapter is organized as follows. We first describe, in Section 5.2, the modelling method that we propose to derive a collaborative crisis resolution process starting with a text. It is based on crisis meta-model scenarios. Four steps in this method will be detailed in following sections. In Section 5.3, we identify high-level objects and the relationships between them from textual rescue plans. Then in Section 5.4, we build scenarios based on the structure defined by [Rolland 1998]. These scenarios are used to discover process models in Section 5.5. The process is represented by two complementary languages: (i) Petri nets [Murata 1989] that can be discovered from scenarios and have the advantage of allowing the verification of the plan's formal properties; and (ii) BPMN, which makes it easier for stakeholders to validate. In Section 5.6 we show how it is possible to simulate and analyse process models with the tools and languages described above. The process conformance checking and measuring process complexity are also performed on these models.

5.2 Design of a Crisis Resolution Process from Scenarios

The design method that we propose (Figure 5.2) has the following characteristics:

1. It consists of four stages with possible iterations following validation by stakeholders;

¹Decision No. 3558/QĐ-UBND enacted by People Committee of HCMC on 13 July 2012 about *Plan to Prevent, Respond and Overcome Earthquake, Tsunami* <http://www.phongchonglutbaotphcm.gov.vn/?id=51&cid=4395>

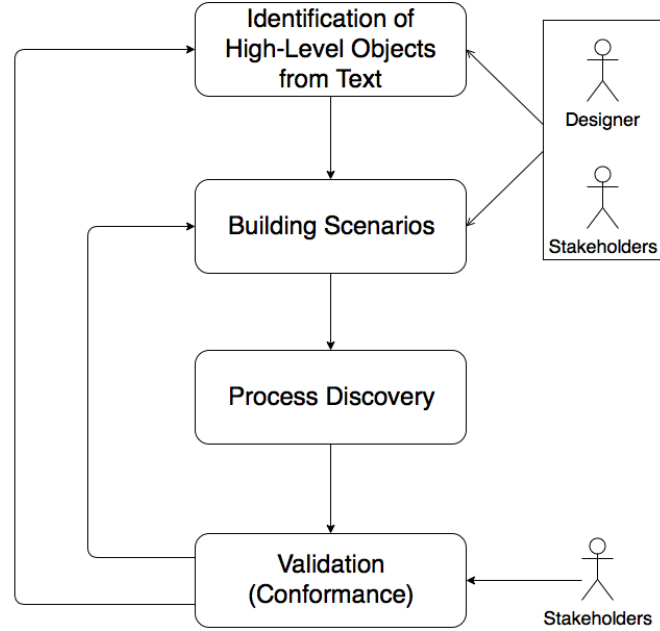


Figure 5.2: A scenario-based lifecycle for process design and validation from text

2. The identification of high-level objects is based on a simple crisis meta-model (Figure 5.2);
3. It is organized around scenarios that correspond to possible crisis resolution plans;
4. It exploits a process-mining technique, in this case the α -algorithm [Van der Aalst 2011], to derive the crisis resolution process from scenarios expressed in tabular form.

We present each of these steps in detail below.

5.3 Identification of High-level Objects from Text

5.3.1 Principles

The designer must extract high-level text objects and the relationships between them. It can be based on the meta-model proposed in Figure 5.3. This fairly simple model can be used to extract key concepts that describe the coordination of activities. Although more detailed models can be found in the literature

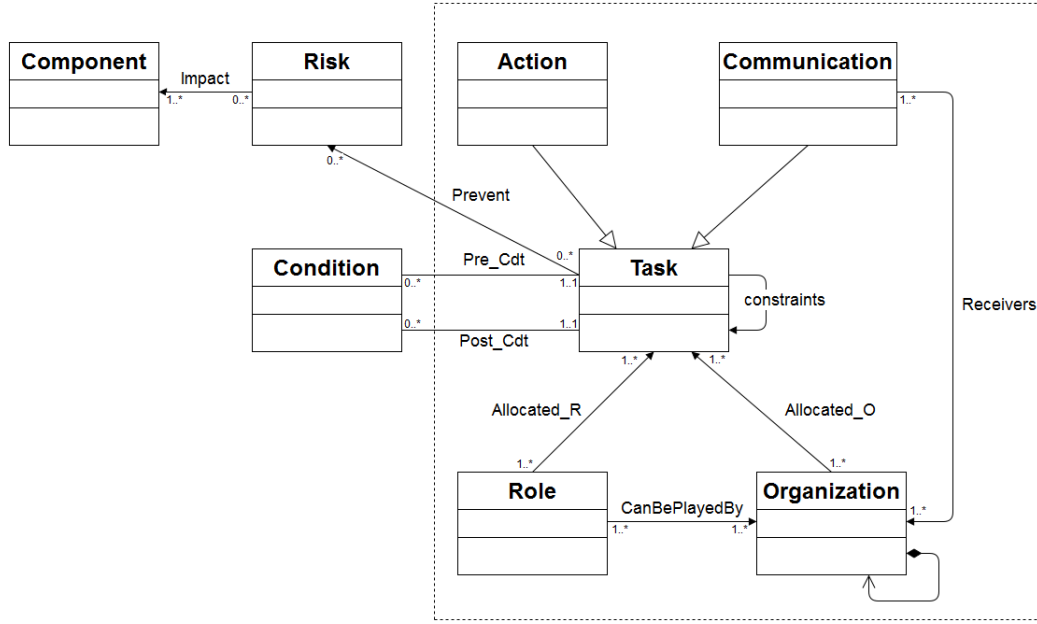


Figure 5.3: Meta-model of concepts involved in coordinating crisis management activities

[Bénaben 2008], they can be difficult to use because it is impossible in practice to have access to all of the theoretical information they contain (probability or severity of risk, etc.).

Here, we limit the presentation to high-level concepts that are easily identifiable in the text and which make it possible to derive simple process models that give an aggregated view of the plan. Designers, working with the stakeholders, can then refine the process by providing scenarios. Although, the approach assumes a manual textual analysis, automatic text analysis techniques are emerging [Viorica Epure 2015] and could be used to extract the elements of the meta-model.

The meta-model records *Tasks*, *Roles* and/or the *Organizations* that are responsible. An organization can contain others, and in this case there is a hierarchical relationship between them. *Tasks* achieve objectives that reduce or resolve *Risks* (potential or actual). *Tasks* have *constraints* (precedence or choice) and the causal links: the effect of a *Task* (*post-condition*) can be exploited by another (*pre-condition*). *Constraints* and causal links are given in the table 5.1. *Tasks* can take two forms (*communication* or *action*). In the second case, the recipient of the communication is recorded.

Table 5.1: Basic relations and constraints between tasks

C1) Before(A,B)	<i>A</i> should occur before <i>B</i> .
C2) Choice(A,B)	Either <i>A</i> or <i>B</i> can be performed. It is assumed that the choice is the result of a prior decision.
C3) Fill(T1,p,T2)	<i>T1</i> produces <i>p</i> that is used by <i>T2</i> , and <i>T1</i> should occur before <i>T2</i> . <i>p</i> is a post-condition of <i>T1</i> and pre-condition of <i>T2</i> .
C4) Parallel(T1,T2)	<i>T1</i> could be performed in parallel with <i>T2</i> .

5.3.2 Illustration: Identifying Organizations, Tasks and their Relationships from HCMC's Tsunami Response Plan

HCMC's tsunami response plan names over thirty organizations, each of which has numerous tasks. For clarity, we grouped organizations with similar responsibilities or missions into a more abstract organization. Some organizations have a similar role: for example, both police and the military are the first-line responders and share the mission of informing and evacuating people, and moving the injured to safety. Here we do not present these different roles in detail, as we focus on tasks and organizations.

The *Local Administration* groups four organizations: 1) *The Committee for Flood-Storm Prevention and Search-Rescue of HCMC*; 2) *The People's Committee of Districts, Communes and Towns*; 3) *The Chairman of People's Committee of Districts, Communes and Towns*; and 4) *The Command Centre for the Counter-Flooding Program*.

The *Military* represents three organizations: 1) *The HCMC High command*; 2) *The HCMC Border Guard High Command*; and 3) *The Border Guard Forces*.

The *Police* represents two organizations: 1) *The HCMC Police*; and 2) *The Department of Fire Prevention and Firefighting*.

The *Local Civil Defence Forces* represents two organizations: 1) *The Local Civil Defence Forces*; and 2) *The HCMC Young Volunteers Force*.

The *Communication Unit* groups three organizations: 1) *The Department of*

Information and Communication; 2) *The HCMC Television Station*; and 3) *The Radio Voice of HCMC People*.

Finally, the *Health and Red Cross* groups three organizations: 1) *The Department of Health*; 2) *The Center for Preventive Medical*; and 3) *The HCMC Red Cross*.

Seven organizations are involved in the response and search-and-rescue phase:

- O1: Institute of Geophysics (Vietnam Academy of Science and Technology)
- O2: Local Administration
- O3: Military
- O4: Police
- O5: Local Civil Defence Forces
- O6: Communication Unit
- O7: Health and Red Cross

We translated the response plan from Vietnamese into English and the following summary shows organizations, their tasks and the relations among tasks: a textual HCMC abstract plan.

Listing 5.1: Textual HCMC abstract plan

1	If the risk of a tsunami that may affect the areas of Can Gio (Ho Chi Minh City, Vietnam) is detected (T1), the Institute of Geophysics (O1) will inform (T2) the city's local administration (O2) about the time, place and predicted level of the tsunami so that it can prepare to respond appropriately to the disaster and minimize the number of victims, prepare food stocks, etc.
2	
3	After receiving the tsunami warning (T3), the local administration (O2) will lead and mobilize their available forces, materials, facilities (e.g. car, trucks, canoes, boats) (T4) to support search and rescue in order to minimize damage. It will also direct evacuations (T5) with the help of several functional units such as the military (O3), police (O4), local civil defence forces (O5), the communication unit (O6), health and Red Cross organizations (O7), etc.
4	
5	Local civil defence forces (O5) patrol streets and residential areas to inform citizens using portable loudspeakers (T6). People will move to safe places under the guidance of the military (O3) and the police (O4). Meanwhile, if the media infrastructure is working, the communication unit (O6) will broadcasts a tsunami warning (T7) on radio and television

channels to inform people both onshore and offshore of the approaching risk. The military (O3) is the core body responsible for evacuating coastal populations (T8) to safe shelters with the cooperation of the police (O4) (T8'). At the same time, the military (O3) will use whistles, alarms, and fires (T9) to warn ships and fishermen (T10) that they should return to shore and store their boats in safe locations.

The police (O4) have the main responsibility for protecting citizens' property (T11) and ensuring public order and safety (T12) to avoid disorder (e.g. transportation, looters, etc.). The health and Red Cross organizations unit (O7) mobilizes doctors, nurses, rescue teams, facilities, and the equipment to support hospitals (T13). During the evacuation, it has task of performing first aid (T14) on the injured. They must call for an ambulance (T15) to transport victims in a serious condition to hospital.

When the tsunami has passed, the Institute of Geophysics (O1) notifies (T16) the HCMC local administration (O2). After receiving this message (T17), the local administration (O2) directs functional units (T18) to address the aftermath. The communication unit (O6) proposes ways to recover communication systems (T19). Meanwhile, the military (O3) and police (O4) coordinate their activities to identify damage (T20) (T20'): collapsed buildings, dead/ injured people, etc. The military (O3) searches for any fishermen (T21) at sea. The police (O4) ensures social order and safety (T22) by providing temporary accommodation for the homeless. Health and Red Cross organizations (O7) support health services, and help in disease prevention (T23) in affected areas. It also verifies DNA samples from anonymous victims (T24) of the tsunami. Finally, once all functional units have finished their work, the local administration (O2) ends the tsunami response (T25).

Table 5.2 provides a summary of tasks and actors.

The constraints and causal relations between tasks can be identified in the text by temporal terms such as “*meanwhile*”, “*so that*”, “*after*”, “*finally*”, and “*at the same time*”. This led to the results shown in Table 5.3.

5.4 Building Scenarios

5.4.1 Scenario Structure

Our method focuses on scenarios following the structure defined by [Rolland 1998]. A scenario is defined by its *purpose*, *lifecycle*, *form* and *content* (cf. Figure 5.4). In this context, they are defined as follows:

- The *purpose* is to cover the various cases that may arise from the text that defines the plan. These scenarios should allow us to model the plan as a process. This process must be applicable to each scenario.

Table 5.2: Tasks and the corresponding actors in the tsunami response plan

Tasks	Organizations
T1: Detect the risk of tsunami	O1: Institute of Geophysics
T2: Issue tsunami warning	O1: Institute of Geophysics
T3: Receive tsunami warning	O2: Local Administration
T4: Mobilize forces, materials, facilities	O2: Local Administration
T5: Direct evacuation	O2: Local Administration
T6: Inform local population using portable speakers	O5: Local Civil Defence Forces
T7: Broadcast tsunami information	O6: Communication unit
T8: Evacuate the local population	O3: Military
T8': Help to evacuate the local population	O4: Police
T9: Warn shipping	O3: Military
T10: Tell fishermen to move to safe places	O3: Military
T11: Protect private property	O4: Police
T12: Ensure the order and safety	O4: Police
T13: Mobilize doctors, nurses, rescue teams, facilities, equipments	O7: Health & Red Cross
T14: Perform first aid	O7: Health & Red Cross
T15: Call an ambulance	O7: Health & Red Cross
T16: Notify the end of the tsunami	O1: Institute of Geophysics
T17: Receive the notification that the tsunami has ended	O2: Local Administration
T18: Begin recovery	O2: Local Administration
T19: Recover communication systems	O6: Communication Unit
T20: Identify damage	O3: Military
T21: Search for lost fishermen	O3: Military
T20': Help to identify damage	O4: Police
T22: Ensure public order and safety	O4: Police
T23: Support health services, disease prevention	O7: Health & Red Cross
T24: Verify ADN samples of anonymous victims	O7: Health & Red Cross
T25: End tsunami response	O2: Local Administration

Table 5.3: Relations between constraints and tasks

T1 fills T2
T2 fills T3
T3 before T4, T5
T4 T5
T4, T5 before T6, T7, T8, T9, T10, T8', T11, T12, T13, T14, T15
T6 T7 T8 T9 T10 T8' T11 T12 T13 T16
T13 before T14, T15
T14 or T15
T16 before T17
T6, T7, T8, T9, T10, T8', T11, T12, T13, T14, T15, T16, before T17
T17 before T18
T18 before T19, T20, T21, T20', T22, T23, T24
T19 T20 T21 T20' T22 T23 T24
T19, T20, T21, T20', T22, T23, T24 before T25

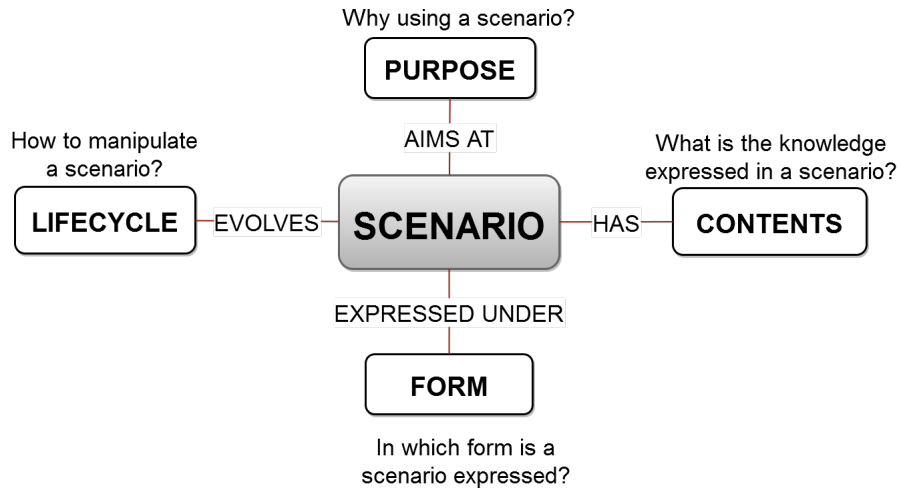


Figure 5.4: Structure of a scenario [Rolland 1998]

- The *lifecycle* is: *creation*, *refinement*, *reduction*, *transformation*, and *simulation*. Not only must scenarios make it possible to understand the collaborative crisis resolution process, but they must then be modified or refined to allow for simulation and their analysis.
- The *form* varies as a function of the stages of the method. It is based on the elements of the meta-model shown in Figure 5.3, but may need to be reduced, refined or transformed as a function of the steps. Therefore, in the concept identification stage, we take into account the full meta-model. For stage 3 of the design (process discovery), we use a simplified form expressed in tabular format and a subset of the meta-model. In the simulation and analysis stages, we consider the cardinality of the actors, the estimated time to execute tasks, etc.
- The *content* contains the input and output parameters. The input parameters express for example the *arrival rate of new instances*, *probabilities for choices*, *end-to-end process time* or *resource constraints* etc. On the other hand, the time aspects (i.e minimum, maximum, average process time of tasks) or resource utilization can be considered as output parameters.

5.4.2 Generation of Scenarios in the Design Phase

The process discovery only requires a subset of the full meta-model (Figure 5.3), and is summarized in Figure 5.5. The aim is to produce several scenarios covering the different alternatives, and in which tasks respect the inter-task relations and constraints (see Table 5.3). These scenarios can either be directly proposed by crisis management stakeholders, or be suggested by the scenario generator. The most realistic scenarios can be selected by stakeholders.

The following code shows a logical formulation as a Prolog program capable of generating scenarios. Basic predicates are derived directly from the meta-model and deduction rules are then produced to generate scenarios given the constraints and relations between tasks. The more accurate result can be obtained by adding more constraints and relationships (*fill relation*, *before temporal constraints* and *choice relation*) into this program.

Listing 5.2: Generating scenarios using the Logic program (Swi-Prolog syntax)

```

1 /* GENERATING SCENARIOS
2 This program generates scenarios composed of a sequence of tasks verifying the relation and
   constraints between them, notably: fill relation, before constraints and choice relation.
```

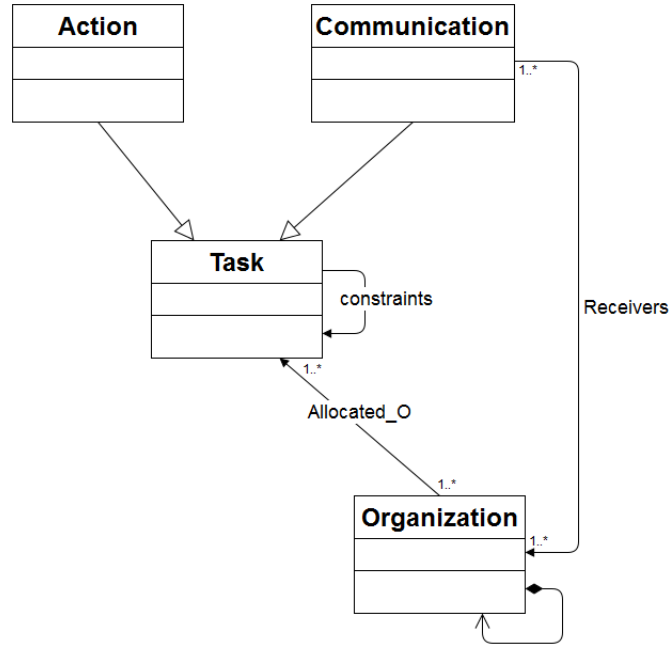


Figure 5.5: Meta-model subset for process discovery

```

3  The choice relation has been simplified as explained in the code */
4
5  /* FACTS COMPLIANT WITH THE META-MODEL */
6  /******
7  /* We first declare the facts corresponding to the relations and constraints in a format compliant
   with the meta-model */
8
9  /* fill relation */
10 fill(t1,d1,t2).
11 fill(t2,d2,t3).
12
13 /* before temporal constraint: we express the constraint between lists */
14 lbefore([t3], [t4,t5]).
15
16 lbefore([t4],[t6,t7,t8,t9,t10,t8p,t11,t12,t1314,t1315,t16]).
17 lbefore([t5],[t6,t7,t8,t9,t10,t8p,t11,t12,t1314,t1315,t16]).
18 lbefore([t6,t7,t8,t9,t10,t8p,t11,t12,t1314,t1315,t16],[t17]).
19
20 lbefore([t17],[t18]).
21 lbefore([t18],[t19,t20,t21,t20p,t22,t23,t24]).
22 lbefore([t19,t20,t21,t20p,t22,t23,t24],[t25]).
23

```

```

24  /* we deduce the before relation between two atomic tasks */
25
26  /* choice relation */
27  choice(t1314,t1315). /* we do not use this predicate, we merge t13 with t14 or t15 */
28      /* and create two atomic tasks t1314 and t1315 */
29
30  /* all the parallel list of tasks */
31  parallel(pa1,[t4,t5]).
32  parallel(pa2, [t6,t7,t8,t9,t10,t8p,t11,t12,t1314,t16]).
33  parallel(pa3, [t6,t7,t8,t9,t10,t8p,t11,t12,t1315,t16]).
34  parallel(pa4, [t19,t20,t21,t20p,t22,t23,t24]).
35
36  /* DEDUCTION RULES */
37  /******
38  /* we deduce the before predicate between two tasks from the lbefore predicate */
39  /* which is applied to lists */
40
41  before(X,Y):– lbefore(L1,L2),
42      member(X,L1),
43      member(Y,L2).
44
45  /* We know represent the requires relation by a rule combining all the
46  previous constraints and relations. S2 requires S1 if :
47  – S1 fills S2, or
48  – S1 should start before S2 */
49
50  requires(T1,T2):–fill(T1,_,T2).
51  requires(T1,T2):–before(T1,T2).
52
53  /* We determine the starting and ending tasks */
54  starting_task(T):–requires(T,_, not(requires(_,T))).
55  ending_task(E):–requires(_,E), not(requires(E,_)).
56
57  /* This following recursive rule builds a scenario as a list of ordered tasks where two any
successive tasks have a "requires" relation between them. Each task could appear at
maximum one time in the scenario: each new task could be integrated in the scenario only if
it doesn't already appear in the "Partial" scenario" */
58
59  scenario(Task, Task, _, [Task]). /* trivial case: a scenario of one task */
60  scenario(Start, End, Partial, [Start | Scenario]) :– /* recursive case */
61      requires(Start, X),
62      not(member(X, Partial)),
63      scenario(X, End, [X | Partial], Scenario).
64
65  /* We compute here a scenario going from a starting tasks S to an ending one E and the scenario

```

```

        is returned in R */
66 completescenario(S,E,R):—
67   starting_task(S),
68   ending_task(E),
69   scenario(S,E,[S],Sc), /* we obtain a scenario Sc */
70   all_scenario(Sc,R). /* we complete Sc by parallel set of tasks and
71                        perform all the possible permutations */
72
73 /* We perform all the scenario R from one scenario S */
74 all_scenario(S,R):—all_parallel(S, R1), /* we add the parallel tasks */
75   all_permutation(R1,R). /* we perform all the possible permutations
76                          inside each set of parallel tasks */
77
78 /* Treatment of parallel tasks: replace a task by the set of parallel tasks to which it belongs */
79 all_parallel([],[]).
80
81 all_parallel([E|Q],[E|E2]) :—
82   findall(L,parallel(_,L),List),
83   not(find(List,E,_)),
84   all_parallel(Q,E2).
85
86 all_parallel([E|Q],[Lout|E2]) :—
87   findall(L,parallel(_,L),List),
88   find(List,E,Lout),
89   all_parallel(Q,E2).
90
91 find([Lin|_],E,Lin):—member(E,Lin).
92 find([Lin|T],E,Lout):—not(member(E,Lin)),find(T,E,Lout).
93
94 /* Permutation of inside the parallel sets of a scenario */
95 permutation(T,T). /* trivial case: an atomic task is not per-mutated */
96 permutation(L,[T|Q]) :— /* permutation */
97   dif(L,[T|Q]),
98   select(T,L,L1),
99   permutation(L1,Q).
100
101 all_permutation(L,R) :— /* all possible permutations of L is performed in R */
102   maplist(permutation,L,R).

```

The execution of the task “*completescenario(S,E,Sc)*” generates all possible scenarios (*Sc*) beginning with *S* tasks and ending with terminal *E* tasks. With the library included in the Prolog program, we obtain results in the form shown in Table 5.3, where tasks are shown in chronological order. The generator produces thousands of scenarios. However, not all of them are realistic in terms of

resources, and interested stakeholders can reduce them or add others that they have experienced. Here, we extract six to illustrate our discussion. They are arbitrary but respect the constraints and relations, and are sufficiently representative of the various cases.

Excerpts from scenarios are:

1. Scenario 1 (26 events): T1. T2. T3. T5. T4. T11. T8. T10. T9. T13. T7. T14. T12. T8'. T6. T16. T17. T18. T22. T20. T19. T24. T20'. T23. T21. T25.
2. Scenario 2 (26 events): T1. T2. T3. T4. T5. T9. T6. T7. T11. T8. T8'. T12. T13. T14. T10. T16. T17. T18. T20'. T23. T22. T21. T19. T20. T24. T25.
3. Scenario 3 (26 events): T1. T2. T3. T4. T5. T8'. T8. T13. T14. T10. T6. T12. T11. T9. T7. T16. T17. T18. T20. T22. T21. T19. T24. T20'. T23. T25.
4. Scenario 4 (26 events): T1. T2. T3. T5. T4. T8. T7. T9. T6. T11. T12. T13. T10. T14. T8'. T16. T17. T18. T21. T20. T23. T24. T20'. T19. T22. T25.
5. Scenario 5 (26 events): T1. T2. T3. T4. T5. T7. T6. T12. T10. T8. T8'. T9. T13. T14. T11. T16. T17. T18. T20. T23. T19. T21. T24. T20'. T22. T25.
6. Scenario 6 (26 events): T1. T2. T3. T5. T4. T7. T8'. T13. T11. T12. T8. T6. T15. T10. T9. T16. T17. T18. T20. T21. T24. T19. T20'. T22. T23. T25.

5.5 Process Discovery

5.5.1 Process Representation using Petri nets

The constraints and causal relationships between tasks allow us to generate, and then select crisis response scenarios. They are linear, while the aim is to synthesize all of these scenarios in one process capable of executing each of these scenarios and clearly explaining the choices and parallelism between tasks. We use the α -algorithm [Van der Aalst 2011] to infer a Petri nets (Definition 1) based on a

log file W . While the α -algorithm usually works on a log file of traces of previous executions, the novelty of our work lies in the fact of using a set of potential scenarios.

5.5.1.1 Petri nets

A Petri net is a directed bipartite graph with two types of nodes: *places* and *transitions*. Graphically, places are represented by circles and transitions by rectangles. An arc can only connect a place to a transition or a transition to a place. In the first case, the place is called an input place of the transition, and an output place in the second case. The formal definition of classical Petri nets is as follows:

Definition 1 (*Petri net*) A Petri net is a triplet $N = (P, T, F)$, where P is a finite set of places, T is a finite set of transitions such that $P \cap T = \emptyset$, and $F \subseteq (P \times T) \cup (T \times P)$ is a set of directed arcs, called the flow relation.

Usually, the behaviour of a system is modelled as a scheduling of operations that consume or produce resources. The states of resources are described by places, and the actions by transitions. The availability of n resources in a place p is represented by n tokens (black dots) in p . At a given time, the state of the system is defined by the distribution of tokens over places, also called “marking”. The system dynamic is described by the execution of transitions, which moves tokens from input places to output places according to the following rules:

1. A transition t is said to be enabled under a given marking if each input place p of t has at least one token. An enabled transition may occur.
2. If an enabled transition t occurs, then t removes one token from each input place and deposits one token in each output place.

5.5.2 Process Discovery using the α -algorithm

An event log includes different scenarios, also called “cases”. In our context, we use the logs in tabular form and expressed under the format: (*ScenarioId*, *Task*, *Performer*, *Receiver(s)*, *Timestamps*). The *Performer* is the organization that performed a task, while the *Receiver* is the possible receivers of a task if this task refers the interaction between organizations. Table 5.4 illustrates a scenario represented in an event log.

Table 5.4: The representation of Scenario 1 (26 events) in an event log

S.Id	Task	Performer	Receiver(s)	Timestamps
1	T1	Inst. of Geo.	-	2016-02-25 16:04:20
1	T2	Inst. of Geo.	-	2016-02-25 17:04:20
1	T3	Local Admin.	-	2016-02-25 18:04:20
1	T5	Local Admin.	-	2016-02-25 19:04:20
1	T4	Local Admin.	-	2016-02-25 20:04:20
1	T11	Police	-	2016-02-25 21:04:20
1	T8	Military	-	2016-02-25 22:04:20
1	T10	Military	-	2016-02-25 23:04:20
1	T9	Military	-	2016-02-26 00:04:20
1	T13	Health & Red Cross	-	2016-02-26 01:04:20
1	T7	Communication Unit	-	2016-02-26 02:04:20
1	T14	Health & Red Cross	-	2016-02-26 03:04:20
1	T12	Police	-	2016-02-26 04:04:20
1	T8'	Police	-	2016-02-26 05:04:20
1	T6	Local Civil D. F.	-	2016-02-26 06:04:20
1	T16	Inst. of Geo.	-	2016-02-26 07:04:20
1	T17	Local Admin.	-	2016-02-26 08:04:20
1	T18	Local Admin.	-	2016-02-26 09:04:20
1	T22	Police	-	2016-02-26 10:04:20
1	T20	Military	-	2016-02-26 11:04:20
1	T19	Communication Unit	-	2016-02-26 12:04:20
1	T24	Health & Red Cross	-	2016-02-26 13:04:20
1	T20'	Police	-	2016-02-26 14:04:20
1	T23	Health & Red Cross	-	2016-02-26 15:04:20
1	T21	Military	-	2016-02-26 16:04:20
1	T25	Local Admin.	-	2016-02-26 17:04:20

Table 5.5: Relations between tasks in the α -algorithm

Direct succession	$x > y$ iff for some case x is directly followed by y
Direct causality	$x \rightarrow y$ iff $x > y$ and not $y > x$
Parallel	$x \parallel y$ iff $x > y$ and $y > x$
Choice	$x \neq y$ iff not $x > y$ and not $y > x$

The algorithm is based on the inheritance relationship between tasks in which it infers three other relations (see Table 5.5). The *direct succession* relation between two tasks is more restrictive than the *before* task discussed in Section 5.3.1 as it indicates a succession relationship without intermediaries.

We now show the steps of the algorithm. W is a scenario file on T (all tasks). $\alpha(W)$ is built according to the Algorithm 1. The first line constructs all transitions based on tasks that appear in the log file W . Lines 2 and 3 respectively calculate T_I and T_O . T_I designates all tasks that begin a case (scenario). T_O designates all tasks that end it. Line 4 calculates the set X_W of pairs of tasks (A, B) whose elements are causally related. Tasks in A can have different relations between them, as is the case for B . Line 5 calculates a minimal subset Y_W of X_W . Line 6 calculates the places P_W that link the pairs in the overall set of Y_W transitions. Line 7 calculates the arcs and line 8 returns the expected result (P_W, T_W, F_W) .

Algorithm 1 α -algorithm to create Petri nets [Van der Aalst 2011]

Input: W (*LogFile*)

Output: $PetriNet(P_W, T_W, F_W)$

- 1: $T_W = \{t \in T \mid \exists \sigma \in W \ t \in \sigma\},$
 - 2: $T_I = \{t \in T \mid \exists \sigma \in W \ t = first(\sigma)\},$
 - 3: $T_O = \{t \in T \mid \exists \sigma \in W \ t = last(\sigma)\},$
 - 4: $X_W = \{(A, B) \mid A \subseteq T_W \wedge B \subseteq T_W \wedge \forall_{a \in A} \forall_{b \in B} a \rightarrow_W b \wedge \forall_{a_1, a_2 \in A} a_1 \#_W a_2 \wedge \forall_{b_1, b_2 \in B} b_1 \#_W b_2\},$
 - 5: $Y_W = \{(A, B) \in X \mid \forall_{(A', B') \in X} A \subseteq A' \wedge B \subseteq B' \Rightarrow (A, B) = (A', B')\},$
 - 6: $P_W = \{p_{(A, B)} \mid (A, B) \in Y_W\} \cup \{i_W, o_W\},$
 - 7: $F_W = \{(a, p_{(A, B)}) \mid (A, B) \in Y_W \wedge a \in A\} \cup \{(p_{(A, B)}, b) \mid (A, B) \in Y_W \wedge b \in B\} \cup \{(i_W, t) \mid t \in T_I\} \cup \{(t, o_W) \mid t \in T_O\},$ and
 - 8: $\alpha(W) = (P_W, T_W, F_W).$
-

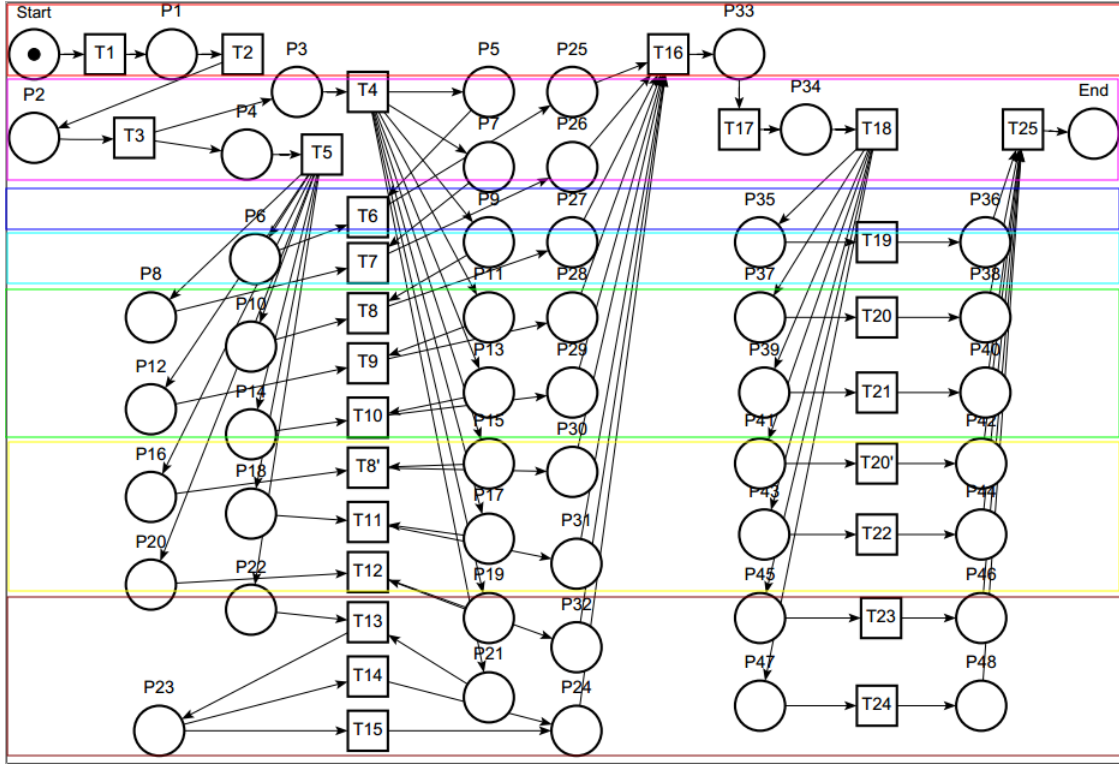


Figure 5.6: Petri nets representation of the HCMC tsunami response plan

5.5.2.1 Application to HCMC Scenarios

Figure 5.6 shows the application of the α -algorithm to the six scenarios presented in Section 5.4.2.

5.5.3 BPMN Representation of Stakeholder Validation

A BPMN representation of the plan is easier for stakeholders to validate as it integrates an organizational perspective that is not found in conventional Petri nets; at the same time, the control structure is more readable. In our context, there are two ways to obtain a BPMN representation:

1. By mapping the Petri nets discovered from the scenarios onto a BPMN graph and complete it with organizational elements (see Section 5.5.3.2);
2. Analyse the text and/or scenarios and draw it directly.

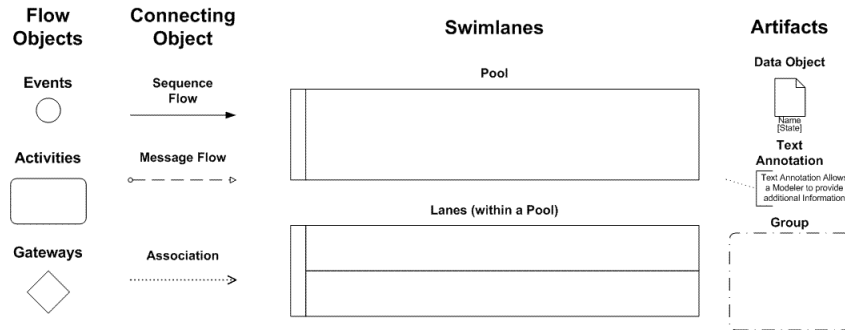


Figure 5.7: Core subsets of BPMN elements (Source: <http://www.bpmn.org>)

Here, we illustrate the first option.

5.5.3.1 BPMN

BPMN was developed by the Object Management Group (OMG) to model business processes. The main advantage is that it is both easy to use for designers and easily understandable by end-users or stakeholders. Many open-source and commercial business process tools (e.g. *jBPM*, *BonitaSoft*, *Camunda*, *Activiti Modeler*, *Bizagi Modeler*) support the notation and can both draw and simulate models. Here, we consider a core subset of BPMN elements (Figure 5.7).

There are four main categories of elements:

1. Basic objects

- *Events*: An event corresponds to things that happen instantaneously. They are depicted by a circle that expresses the start or end of a process, or an intermediate event (i.e. *start event* by a green circle, while *end event* by a red one).
- *Activities*: An activity expresses a task performed by an actor. It is represented by a round-cornered rectangle.
- A *Gateway* controls the flow of execution of the process. It is represented by a diamond that illustrates inclusive parallelism (indicated by a “+”), or an exclusive choice (indicated by an “×”).

2. Connecting Objects are used to connect *Basic Objects* (*Events*, *Activities* and *Gateways*)

- A *Sequence Flow* connects basic objects to express a sequential flow of work. It is depicted by a directed arc.
 - A *Message Flow* is represented by a dash arrow connecting two activities and/or events (with a message symbol) to express the transmission of messages.
 - An *Association* connects an *Activity* to a *Data Object*.
3. **Swimlanes** are used to structure the process from an organizational perspective. They are represented by rectangular boxes:
- A *Pool* represents an organization within which a process is executed.
 - A *Lane* is a sub-division of an organization (role, department, ...). It is shown within a pool.

4. Artefacts

- A *Data Object* represents data used or produced by activities. It connects with *Activities* through *Associations*.
- A *Group* is a way to gather objects for documentation purpose. It is represented by a rectangle drawn with a dashed line.
- An *Annotation* is a comment that provides additional information to the reader. Both *Groups* and *Annotations* have no influence on the execution of the process.

5.5.3.2 Mapping Petri nets onto BPMN

Figure 5.8 shows the relationship between Petri nets and BPMN notation.

Consequently, the Petri nets shown in Figure 5.6 can be easily transformed into the BPMN representation shown in Figure 5.9. The Petri net formalism is used to verify the theoretical properties (reachability of particular states, termination, liveness, etc.) and for simulations. Next, we describe the adapted BPMN model and note some temporal constraints. We only consider two pools as we only have two organizations. The BPMN model is considered as a shared artefact that could be used for negotiation of resources or improve coordination for similar tasks done by different organizations.

The mapping from Petri net to BPMN is implemented using ATL² technology on top of two corresponding meta-models (see Section D.2 of Appendix D). In

²ATL Transformation Language <https://wiki.eclipse.org/ATL>









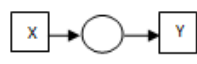
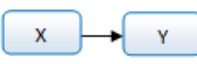
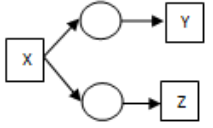
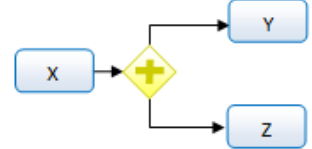
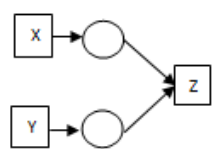
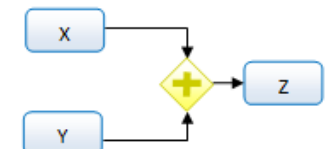
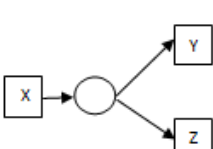
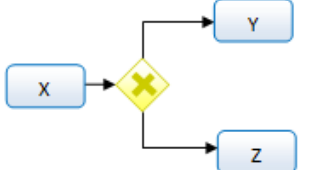
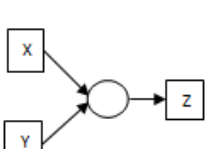
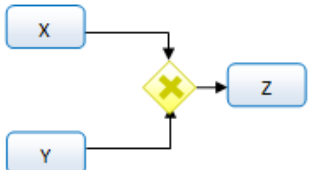
Petri Nets		BPMN	
Place		Event	
Starting Place		Start Event	
Ending Place		End Event	
Transition		Activity	
Sequential		Sequence Flow	
Parallelism		Parallel Gateway	
Synchronization		Synchronization	
Exclusive choice		Exclusive Gateway	
Merge		Merge	

Figure 5.8: Mapping between Petri nets and BPMN notation

fact, the source model of the mapping (cf. the Petri net) is expressed under PNML³ format which contains two separated *pages* (i.e. processes).

Eight stakeholders are identified from the Petri net with reference to the identification of high-level objects from text (cf. Section 5.3). They are depicted by two pools (corresponding to two *pages* of the PNML file) and six lanes connected by the task flows and mutual interactions.

Some parallel structures between tasks are detected from the Petri net according to the mapping table shown in Figure 5.8, e.g. the tasks in [T3, T4, T5] correspond to [X, Y, Z] respectively and similar to [T4, T6, T7], [T5, T6, T7], [T4, T7, T8], [T5, T7, T8], ... We notice that the *Military*, *Police* and *Health and Red Cross* organizations are supposed to perform their tasks in parallel, so they require probably more resources and high responsibility. In this case, each organization should be distributed over the parallel tasks according to a given policy (proportional distribution, distribution according to the importance given to each task, ...).

In addition, an exclusive choice structure is detected from the Petri net (Figure 5.6) regarding to the relationship of the tasks T13, T14 and T15. The *Health and Red Cross* organization has to choose to carry out only one task among two possible ones.

If a task can be performed by two different actors, it is duplicated instead of using an abstract one. In the BPMN diagram shown in Figure 5.9, we duplicate the task *Evacuate people*, that may be performed by two actors (*Military* and *Police*), to become two tasks *T8* and *T8'* respectively.

5.6 Process Simulation, Analysis and Conformance Checking

In this section, we propose some recommendations for the analysis and simulation of crisis management processes. This is based on a combination of complementary tools and languages. The simulation and process analysis can be performed in several ways as a function of objectives and the support language:

- We use the Petri net formalism to verify the theoretical properties of the model, such as the accessibility of certain states, blocking states or termination of the process.

³Petri Net Markup Language <http://www.pnml.org/>

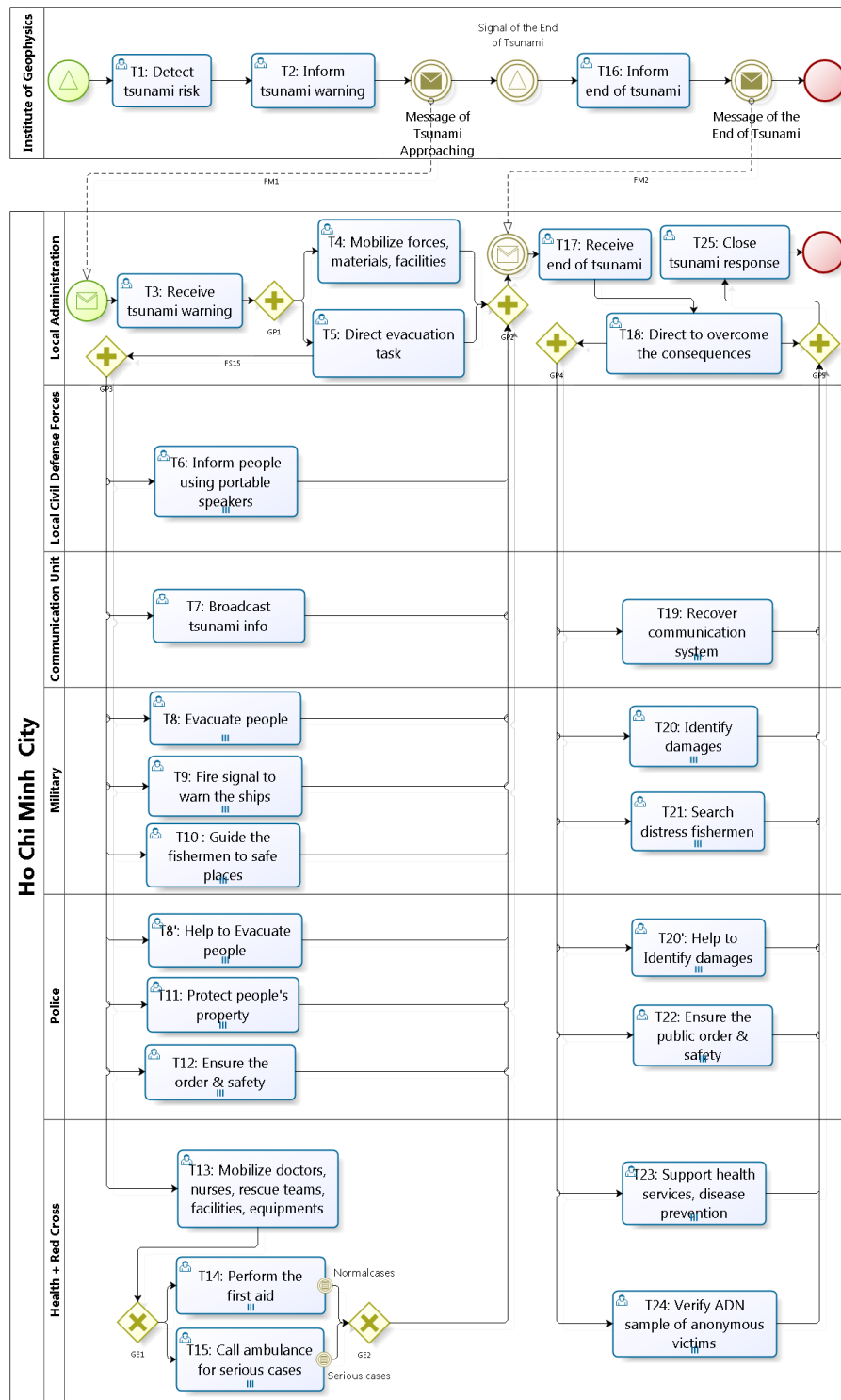


Figure 5.9: BPMN representation of the HCMC tsunami response plan

- We use a workflow management system (e.g. YAWL, BONITA) prototype to validate the geographical distribution of the process and the orchestration of actors. This is important in order to be able to perform crisis simulation exercises with actors in the field and validate their coordination.
- BPMN simulators can be used for model calibration (in terms of resources) and the examination of various execution scenarios. In the latter case, it is possible to simulate several scenarios by adjusting the workforce and their workload. We will illustrate this here.
- We implement the metric-based BPMN model analysis tool proposed by [Cardoso 2008] to analyse the complexity of processes that can determine actors' acceptance. The tool is called *Assessment for Business Process* (A4BP) [Peralta 2015]. This type of analysis can lead to the identification of a *pathology* in a model, which can lead to its transformation.

5.6.1 Theoretical Verification

To verify the theoretical properties of the model (such as reachability) in some states, the termination of the process, the liveness (cf. all paths are possible), we use the Petri net formalism. One way to study this is to build the reachability graph that makes it possible to deduce its properties by simple inspection. In our context, we deduce from the reachability graph the network language $T1.T2.T3 (T4 \parallel T5). (T6 \parallel T7 \parallel T8 \parallel T9 \parallel T10 \parallel T8' \parallel T11 \parallel T12 \parallel (T13. (T14 \parallel T15)). T16.T17.T18. (T19 \parallel T20 \parallel T21 \parallel T20' \parallel T22 \parallel T23 \parallel T24). T25$.

The soundness property includes three sub-properties:

1. The process, when started, can always complete;
2. When the process ends, there should be no other tasks still running;
3. The process should not contain tasks that will never be executed (no dead transitions).

5.6.2 Process Deployment and Execution

Distributed execution is interesting because it can validate the coordination of actors who may be geographically distributed in the field (different space, same time). Control of coordination may be delegated to the workflow management system and/ or supervised by the crisis unit who can also collect feedback. We

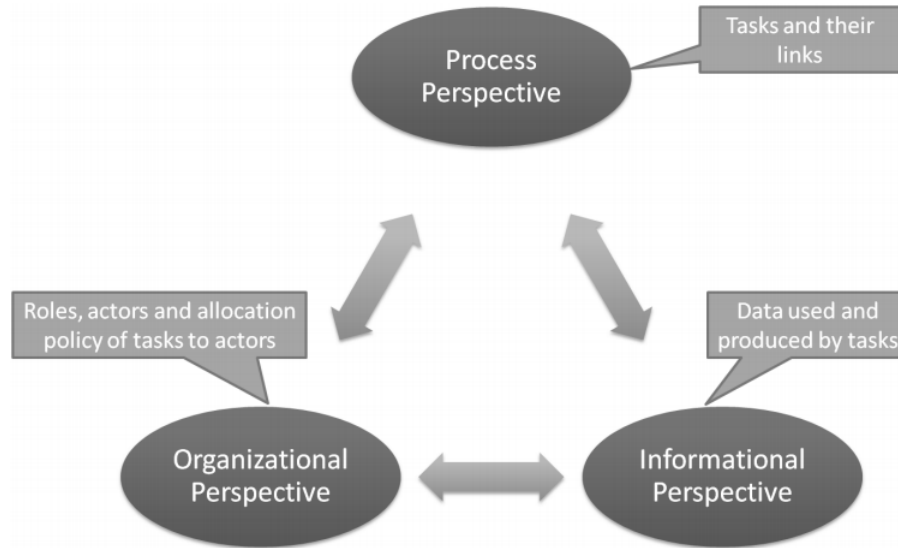


Figure 5.10: The three YAWL perspectives

implemented a prototype management process for the HCMC tsunami in the YAWL⁴ workflow management system. We chose this system because processes are modelled in a language derived from Petri nets. Actors may be geographically distant and communicate with the crisis unit via the electronic exchange of forms showing the actions they have to perform (received from the crisis unit) and reporting of their activities (sent to the crisis unit). During execution, the system records information in log files that can be used for analysis in process mining (see Section 5.6.4). The state of the process can also be visualized: completed tasks, pending tasks, etc. YAWL includes three perspectives (Figure 5.10). The *Process Perspective* concerns tasks and the links between them; the *Informational Perspective* concerns the data used and produced by each task; and the *Organizational Perspective* defines the roles in the process, the actors playing these roles, and the allocation of tasks to actors. However, the organizational aspect remains poor within YAWL and is better addressed by the agent perspective (see Chapter 6).

YAWL provides an engine that automatically allocates tasks to actors through their worklists, based on a predefined policy. When the engine is informed that a task has been completed, it automatically moves the process on. Actors can access their worklist via a web browser, and can check their allocated tasks,

⁴Yet Another Work-flow Language, version 5.7, <http://www.yawlfoundation.org/>

The screenshot shows the YAWL (Yet Another Workflow Language) software interface. At the top, there is a blue header bar with the YAWL logo (a sequence of icons: a play button, a square with 'Y', a circle with 'A', a square with 'W', a circle with 'L', and a square with a play button) and the text "www.yawlfoundation.org" on the left and "Leading the World in Process Innovation" on the right. Below the header, the text "Edit Work Item: 6.1" is displayed. The main part of the interface is a dialog box titled "Detecttsunami". Inside the dialog box, there are four input fields: "Level:", "Place:", "Time:", and "Date:". The "Date:" field is pre-filled with "38/02/2016" and has a calendar icon to its right. Below the input fields, there are three buttons: "Cancel", "Save", and "Complete".

Figure 5.11: Task description

decide whether to accept them or not, and report when they are completed. The allocation of tasks can also be set to resolve possible conflicts (e.g. several actors working on the same task) or reallocate tasks to other actors if there is a lack of resources. Task allocation can be managed by the engine or supervised by a particular actor; the crisis unit in our case. The figures below show two screenshots of the YAWL implementation (Figure 5.11 for description of a task by filling its attributes and Figure 5.12 for assigning a task to an actor).

Although we could not test the prototype with real actors, the implementation is interesting for two reasons. First, it makes it possible to test distributed coordination and collect the views of actors in the field at the same time as those of in the crisis unit. The second observation is that if the crisis unit is fully computerized, the tool could support monitoring of the coordination of actors while providing information that facilitates group awareness. Information exchange is contextualised by the task it is related to and the geographical location of the issuer. Coupling this with a geographic information system would provide a visual picture of the state of the crisis: what has been done and what remains to be done, by whom and where.

Process simulation, when it involves stakeholders could complement real-world exercises and improve the preparedness phase, underline coordination and resource issues, and improve the plan. In the following subsections, we provide parameters and scenarios that can simulate a BPMN diagram (Figure 5.9).

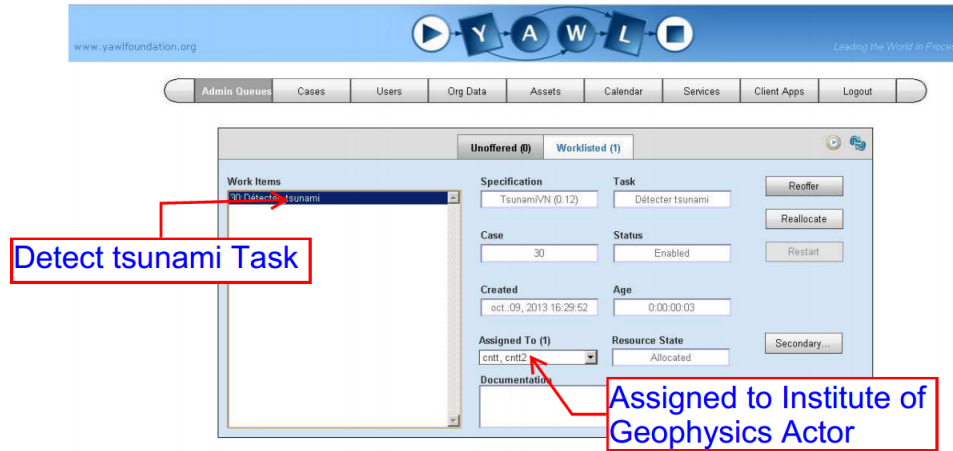


Figure 5.12: Execution and simulation of the tsunami response plan in the YAWL environment

5.6.2.1 Resource and Time Analysis

A process cannot be disconnected from real constraints (time needed to carry out tasks, costs, available workforce, etc.). This notably includes being able to estimate the response time to a crisis as a function of the workforce (available resources). Many questions arise, in particular in the preparedness phase of the crisis. How long will it take to resolve the crisis? What will happen, for example, if a tsunami arrives after an earthquake and staff are already occupied? How to anticipate an overburdened workforce and the risk of breakdown? What is the best way to allocate the workforce to parallel tasks? Should we recruit volunteers? It is therefore necessary to provide stakeholders with a way to simulate different scenarios (optimistic, pessimistic, or normal), and what-if analyses to better calibrate resources. Some of these questions may also arise during the resolution phase, in relation to the part of the plan that remains to be executed, and therefore facilitate decisions.

5.6.3 Process Simulation and Validation

5.6.3.1 Defining Simulation Parameters

In order to perform such simulations, in addition to the process model (see Figure 5.9), extra information (quantity of resources, time constraints etc.) is needed to define accurate scenarios. [Van der Aalst 2008] states that a simulation consists of four input parameters:

5.6. PROCESS SIMULATION, ANALYSIS AND CONFORMANCE CHECKING97

1. *Arrival of new cases* expresses the law of new process instances (i.e. the time interval between arrivals, the maximum number of instances). It is rare that one tsunami is followed by another, but choosing several cases lets us explore the different alternatives included in the process and the potential workload of actors.
2. *Probability of choices* indicates the probability given to alternative choices in the plan (i.e. conflicting transitions in Petri nets or exclusive gateways in BPMN).
3. *Service time* (or process time for tasks) expresses the time need for task completion. A distribution law could be applied, based on what has been reported during exercises or past crises.
4. *Availability of resources* corresponds to the number of mobilized people for each organizations or department, as well as the distribution of resources over the tasks. Through this parameter, we could evaluate the effect of resource quantity and distribution over the process performance.

Current BPMN simulators lack factors such as actors' capacities, priorities or task priorities, although some vendors (e.g. Bizagi, BonitaSoft) provide simulators that implement such parameters. Here, we added a parameter that refers to the importance of a task in terms of rescue or recovery. The higher this factor, the more the important the task in saving people or things. As it influences crisis resolution, performance is included in our simulation to allocate resources to parallel tasks (although it can also be applied to other types of tasks).

The two parameters *arrival of new cases* and *service time* may have different distributions (discrete or continuous) and generate random values such as the Poisson Distribution, the Uniform Distribution, the Negative Exponential Distribution, the Triangular Distribution, the Erlang Distribution, etc. Here, we apply the Truncated Normal Distribution, a continuous distribution that is used in nature where values are evenly distributed around a mean.

The case study focuses on training for stakeholders who must follow tsunami response plans. This is broken down into three phases:

1. Receiving the tsunami warning;
2. Evacuation before the tsunami arrives; and
3. Recovery.

Table 5.6: Probabilities of Choices (PC) of alternative tasks and Importance Factors (IF) of parallel tasks

Tasks	PC	Tasks	IF
T14/T15	0.9/0.1	T4/T5	0.3/0.7
		T8/T9/T10	0.8/0.1/0.1
		T8'/T11/T12	0.1/0.3/0.6
		T20/T21	0.8/0.2
		T20'/T22	0.1/0.9
		T23/T24	0.5/0.5

The shorter the first and especially the second phase, the more lives are saved. An important aim of the simulation is to minimize the total process time. We assume that the time needed for the three phases of training is about 15 minutes, 2 hours and 20 hours respectively making a total of about 22 hours and 15 minutes. However, the actual execution time is determined by the distribution function. In addition, the simulation at the macro level cannot determine the number of victims (deaths, injuries). The cost of a task is another factor that affects the number of victims: the less an activity can save lives, the more it costs; here the idea is to decrease the total cost of the process.

Assuming there are 10 consecutive training sessions, the *Maximum Arrival Count* parameter of *Start Event* is set by 10. The *Arrival Interval* parameter follows the normal distribution, with a mean (μ) of 22 hours 15 minutes, and a standard deviation (σ) of three hours.

The *Probabilities of Choices* of alternative tasks and the *Importance Factors* of parallel tasks are shown in Table 5.6. Resources allocated to tasks could be based on task's importance: *more important tasks are provided with more resources*.

Service Time follows the Truncated Normal Distribution shown in Table 5.7 with a base time unit of the hour. Assuming the duration of the tsunami follows a Truncated Normal Distribution with a mean (μ) of 15 minutes, the standard deviation (σ) is three minutes, the minimum is three minutes and the maximum is 30 minutes.

Waiting Time also follows the Truncated Normal Distribution. The aim is to minimize this parameter, as the situation is urgent. Table 5.8 shows waiting time values. We also model seven roles (actors) as follows: The Institute of Geophysics (IG), the Local Administration (LA), Military (M), Police (P), the Local Civil

Table 5.7: Service Time (ST) of tasks used in the tsunami response plan based on the Truncated Normal Distribution

Task	Service Time (Hours)	Task	Service Time (Hours)
T1	$\mu = 0.2, \sigma = 0.05$ Min = 0, Max = 0.3	T2	$\mu = 0.1, \sigma = 0.02$ Min = 0, Max = 0.2
T3	$\mu = 0.1, \sigma = 0.02$ Min = 0, Max = 0.2	T4	$\mu = 0.75, \sigma = 0.25$ Min = 0, Max = 2
T5	$\mu = 0.5, \sigma = 0.16$ Min = 0, Max = 1	T6	$\mu = 1, \sigma = 0.3$ Min = 0, Max = 2.5
T7	$\mu = 1, \sigma = 0.16$ Min = 0, Max = 2.5	T8	$\mu = 1.5, \sigma = 0.5$ Min = 0, Max = 3.5
T8'	$\mu = 1.5, \sigma = 0.5$ Min = 0, Max = 3.5	T9	$\mu = 1, \sigma = 0.15$ Min = 0, Max = 2.5
T10	$\mu = 1, \sigma = 0.16$ Min = 0, Max = 2.5	T11	$\mu = 1.5, \sigma = 0.5$ Min = 0, Max = 3.5
T12	$\mu = 1, \sigma = 0.15$ Min = 0, Max = 2	T13	$\mu = 0.75, \sigma = 0.1$ Min = 0, Max = 2
T14	$\mu = 1, \sigma = 0.3$ Min = 0, Max = 2.5	T15	$\mu = 1, \sigma = 0.15$ Min = 0, Max = 2.5
T16	$\mu = 0.1, \sigma = 0.02$ Min = 0, Max = 0.2	T17	$\mu = 0.1, \sigma = 0.02$ Min = 0, Max = 0.2
T18	$\mu = 0.5, \sigma = 0.15$ Min = 0, Max = 1	T19	$\mu = 20, \sigma = 1$ Min = 0, Max = 30
T20	$\mu = 20, \sigma = 2$ Min = 0, Max = 30	T21	$\mu = 20, \sigma = 1$ Min = 0, Max = 30
T20'	$\mu = 20, \sigma = 1$ Min = 0, Max = 30	T22	$\mu = 20, \sigma = 0.5$ Min = 0, Max = 30
T23	$\mu = 20, \sigma = 2$ Min = 0, Max = 30	T24	$\mu = 20, \sigma = 1$ Min = 0, Max = 30
T25	$\mu = 0.5, \sigma = 0.15$ Min = 0, Max = 1.5		

Defence Forces (LCDF), the Communication Unit (CU), and Health & Red Cross services (HR).

The total *number of human resources* and the unit cost (cost per hour) for each organization are shown in Table 5.9. For clarity, it does not take into account other non-human resources such as transport (e.g. ambulances, fire trucks, canoes), or machinery (e.g. epidemic prevention sprayers, GPS devices).

In addition, Table 5.10 shows the cost associated with each activity.

Expected outputs are:

1. *Time utilization* represents the total time needed for the tsunami response process, average time, average waiting time, minimum or maximum time for each task; and
2. *Resource utilization* expresses the distribution of resources for each actor.

Here we use the Bizagi Modeller⁵ to model and simulate the BPMN diagram (see Figure 5.9).

5.6.3.2 Defining Simulation Scenarios

The process model is also examined by defining scenarios and analysing simulations. Following [Rolland 1998], a scenario is defined by four components: its *purpose*, *lifecycle*, *form* and *content* (Figure 5.4). Regarding *purpose*, the crisis simulation aims to answer two questions:

1. What is best way to efficiently allocate human resources to tasks?
2. What is the best resource allocation strategy?

The *content* and *form* of scenarios are defined by services time (in minutes), the number of actors (in positive integer), the probability of alternative tasks, and the importance factor (as a percentage).

The number of resources allocated to tasks was modified and two scenarios were created based on two resource allocation strategies for parallel and alternative tasks:

- *Scenario 1*: This is called *Uniform Allocation of Resources*. An equal number of resources are allocated to parallel and alternative tasks (R/N where R is the number of available resource and N is the number of parallel or

⁵<http://www.bizagi.com/>

Table 5.8: Waiting Time (WT) of tasks based on the Truncated Normal Distribution

Task	Waiting Time (Hours)	Task	Waiting Time (Hours)
T1	$\mu = 0.01, \sigma = 0.01$ Min = 0, Max = 0.5	T2	$\mu = 0.03, \sigma = 0.01$ Min = 0, Max = 0.5
T3	$\mu = 0.05, \sigma = 0.02$ Min = 0, Max = 0.5	T4	$\mu = 0.03, \sigma = 0.01$ Min = 0, Max = 0.5
T5	$\mu = 0.05, \sigma = 0.01$ Min = 0, Max = 0.5	T6	$\mu = 0.06, \sigma = 0.02$ Min = 0, Max = 0.5
T7	$\mu = 0.04, \sigma = 0.01$ Min = 0, Max = 0.5	T8	$\mu = 0.02, \sigma = 0.01$ Min = 0, Max = 0.5
T8'	$\mu = 0.02, \sigma = 0.01$ Min = 0, Max = 0.5	T9	$\mu = 0.03, \sigma = 0.01$ Min = 0, Max = 0.5
T10	$\mu = 0.03, \sigma = 0.01$ Min = 0, Max = 0.5	T11	$\mu = 0.04, \sigma = 0.02$ Min = 0, Max = 0.5
T12	$\mu = 0.03, \sigma = 0.02$ Min = 0, Max = 0.5	T13	$\mu = 0.05, \sigma = 0.02$ Min = 0, Max = 0.5
T14	$\mu = 0.02, \sigma = 0.01$ Min = 0, Max = 0.5	T15	$\mu = 0.05, \sigma = 0.03$ Min = 0, Max = 0.5
T16	$\mu = 0.05, \sigma = 0.02$ Min = 0, Max = 0.5	T17	$\mu = 0.03, \sigma = 0.01$ Min = 0, Max = 0.5
T18	$\mu = 0.05, \sigma = 0.01$ Min = 0, Max = 0.5	T19	$\mu = 0.04, \sigma = 0.01$ Min = 0, Max = 0.5
T20	$\mu = 0.05, \sigma = 0.02$ Min = 0, Max = 0.5	T21	$\mu = 0.05, \sigma = 0.01$ Min = 0, Max = 0.5
T20'	$\mu = 0.05, \sigma = 0.02$ Min = 0, Max = 0.5	T22	$\mu = 0.04, \sigma = 0.01$ Min = 0, Max = 0.5
T23	$\mu = 0.05, \sigma = 0.01$ Min = 0, Max = 0.5	T24	$\mu = 0.05, \sigma = 0.02$ Min = 0, Max = 0.5
T25	$\mu = 0.02, \sigma = 0.01$ Min = 0, Max = 0.5		

Table 5.9: Availability and unit cost of resources in the tsunami response plan

Resource	Availability	Unit Cost (€)
Institute of Geophysics	10	15
Local Administration	100	18
Local Civil Defence Forces	200	15
Communication Unit	150	16
Military	6000	20
Police	3000	20
Health & Red Cross	2000	18

Table 5.10: Cost associated with Tasks

Task	Cost (€)	Task	Cost (€)
T1	15	T2	5
T3	5	T4	100
T5	50	T6	200
T7	150	T8	800
T8'	100	T9	80
T10	120	T11	600
T12	200	T13	100
T14	500	T15	80
T16	5	T17	5
T18	50	T19	700
T20	750	T21	100
T20'	120	T22	550
T23	400	T24	500
T25	50		

Table 5.11: Number of resources allocated to tasks in the two scenarios

Task	Scen. 1	Scen. 2	Task	Scen. 1	Scen. 2
T1	10	10	T14	1000	1800
T2	10	10	T15	1000	200
T3	100	100	T16	10	10
T4	50	30	T17	100	100
T5	50	70	T18	100	100
T6	200	200	T19	150	150
T7	150	150	T20	3000	4800
T8	2000	4800	T21	3000	1200
T9	2000	600	T20'	1500	300
T10	2000	600	T22	1500	2700
T8'	1000	300	T23	1000	1000
T11	1000	900	T24	1000	1000
T12	1000	1800	T25	100	100
T13	2000	2000			

alternative tasks performed by an organization). In this case, no assumptions are made about the probability of choices or the importance of tasks. For example, one assumption is that there are 6000 military personnel who have three parallel tasks. Each task is allocated 2000 ($6000/3$) personnel. Other tasks are allocated the maximum number of resources.

- *Scenario 2: Important Focus Allocation of Resources* is based on the distribution parameter assigned to parallel or alternative tasks (i.e. **R*PC** or **R*IF**, see Table 5.6). These numbers are provided by the authority as a function of the importance factor or the choice probability they give to each parallel or alternative task. In contrast, we allocate the maximum number of resources to all of the other tasks.

The number of human resources allocated to each task in the scenarios are shown in Table 5.11.

5.6.3.3 Simulation and Analysis of Scenarios

As mentioned above, the simulations show: i) *time needed*; ii) *resource use*; and iii) *costs*. Time is the key factor as it determines the number of lives that can be

Table 5.12: Comparison of Time Utilization (RU) for the two scenarios

	Min. Time	Max. Time	Avg. Time	Total Time
Scen. 1	1d 4h 38m 24s	3d 3h 3m	1d 19h 23m 24s	<i>110d 23h 21m</i>
Scen. 2	1d 4h 38m 24s	2d 4h 43m 48s	1d 19h 31m 48s	<i>108d 9h 51m</i>

Table 5.13: Comparison of Resource Utilization (RU) for the two scenarios

Organization	RU in Scenario 1	RU in Scenario 2
Military	29.67 %	29.82 %
Police	29.98 %	29.61 %
Health & RC	30.94 %	31.47 %
Local Admin.	3.10 %	3.04 %
Commu. Unit	29.85 %	29.85 %
LCDF	1.47 %	1.47 %
Inst. of Geo.	0.68 %	0.68 %

saved.

Table 5.12 shows the time needed (*minimum, maximum, average and total*) for 10 consecutive training sessions. Minimum time in both scenarios is the same (1 day, 4 hours, 38 minutes, 24 seconds). However, maximum and total time (for 10 process) is lower in Scenario 2 (*Important Focus Allocation of Resources*) than Scenario 1 (*Uniform Allocation of Resources*). Therefore, we can conclude that Scenario 2 is more effective (could save more lives) than Scenario 1 with respect to the most important aspect, namely time. This result suggests that authorities should focus on task priorities when allocating resources.

Two outputs are considered: i) *resource utilization* (Table 5.13), and ii) *cost* (Table 5.14). These figures show little difference between the two scenarios in both cases. Scenario 1 (*Uniform Allocation of Resources*) is slightly more effective than Scenario 2 (*Important Focus Allocation of Resources*): €47,193,167.29 compared to €47,299,636.69.

Table 5.14: Comparison of Total Cost (TC) for the two scenarios (€)

Organization	TC in Scenario 1	TC in Scenario 2
Military	25,633,088.6	25,763,073.4
Police	12,950,783.2	12,791,661.2
Health & RC	8,020,726.2	8,157,112.92
Local Admin.	40,167.72	39,387.6
Commu. Unit	515,845.12	515,845.12
LCDF	31,825.05	31,825.05
Inst. of Geo.	731.4	731.4
- Total -	<u>47,193,167.29</u>	<u>47,299,636.69</u>

5.6.4 Process Conformance Checking and Organizational Discovery with the PROM Framework

Within a workflow management system such as YAWL, process executions are recorded in log files. A log file contains several cases (scenarios) of the same model. This type of file can be used to discover process and organizational models, and more precisely to:

1. Identify and understand new models that deviate from those prescribed and may constitute good practice;
2. Analyse the coherence of *a priori* models (prescribed processes) and performed models (actual processes); and
3. Improve existing models by analysing performance data (costs, duration, etc.).

PROM⁶ software provides this set of features [Van der Aalst 2011] and includes components such as the α -algorithm, a heuristics miner, a social network miner, etc. Figure 5.13 shows the Petri net discovered by PROM using α -algorithm on top of a 1000-cases event log generated by the using of YAWL. The discovered Petri net is similar/compliant to the one built at design time (see Figure 5.6).

Let us now detail the discovery of the organizational aspect.

⁶ProM tool <http://www.promtools.org/prom6/>

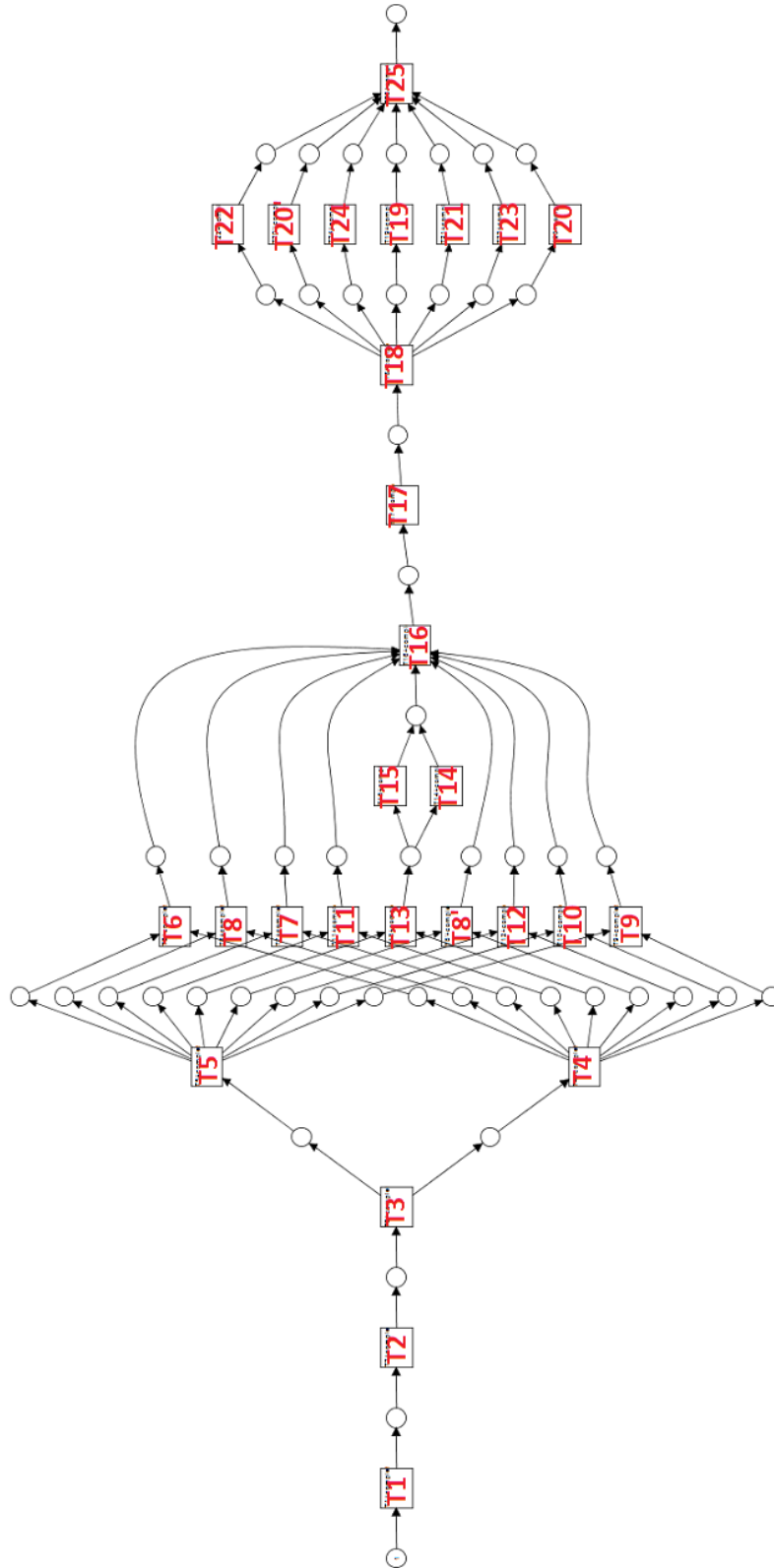


Figure 5.13: Petri nets mined by ProM using α -algorithm on top of a 1000-cases event log

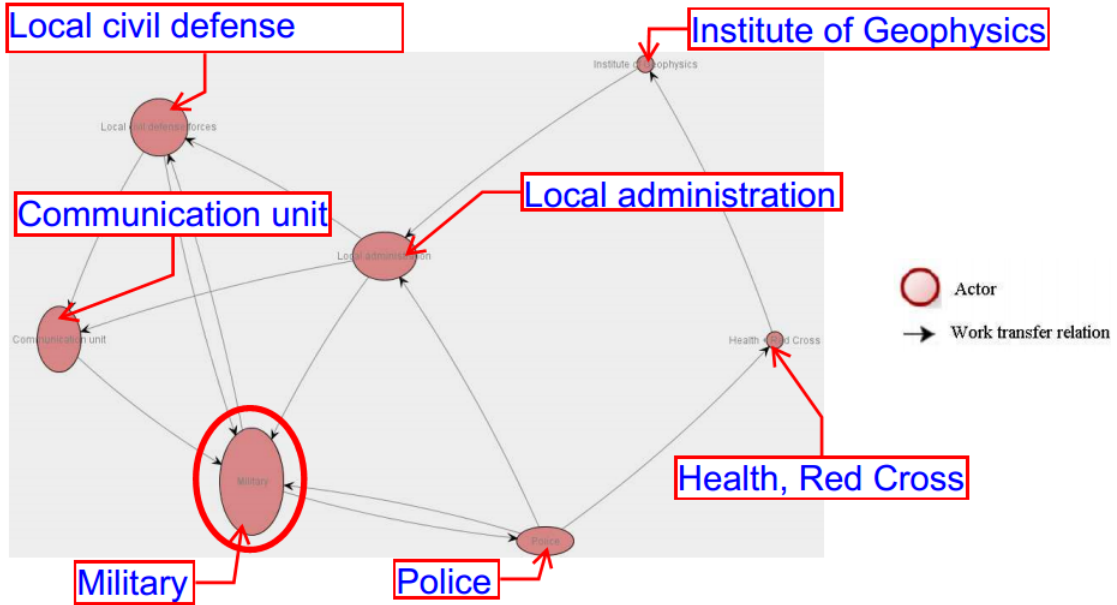


Figure 5.14: Sociogram showing the work transfer relation

From the organizational point of view, PROM is able to discover *sociograms*. A sociogram is a graph (P, R) where P is the set of actors and R is the relation between actors. Here, it represents the work transfer from an actor A to an actor B. The graph may be weighted to reflect the importance of relationships. In this case, a function W is added to assign a value to each element of R . This relationship may also represent different kinds of collaboration: *work transfer*, *collaboration on common cases*, and *identical profile among actors*. This sociogram is similar to the role graph (see Section 6.3 in the next chapter) which is unfortunately not supported by PROM.

Figure 5.14 presents the sociogram that was created using the *Mine for a Handover-of-Work Social Network* plugin. It describes the *work transfer relation* between seven actors, and can be used to analyse the structure of interactions between actors, their involvement in the process, and task dependencies. In a case (scenario) a work transfer from a person i to a person j is achieved if there are two successive activities where the first is performed by i and the second by j . In Figure 5.14, the more an actor is involved in the work transfer relation, the more greater the ellipse.

5.6.5 The A4BP Tool for Measuring Process Complexity

We implemented process complexity metrics in the *Assessment for Business Process* tool (A4BP⁷) [Peralta 2015], which is developed in the Pharo⁸ environment. A4BP allows developers, engineers, process managers and end-users to model, analyse and visualize multiple business process perspectives. It can be used by a rescue plan analyst in two different ways:

1. Based on default, predefined visualizations; or
2. To build a new visualization using the scripting engine, which is based on Roassal⁹.

Figure 5.15 shows the A4BP architecture, which consisting of:

1. A *meta-model process engine* that parses process definitions, builds an object model of BPMN instances and calculates quality metrics;
2. A *simulation engine* based on BPSim¹⁰, which is an extension of BPMN to configure simulations, define scenarios and capture results based on five dynamic perspectives: *Time*, *Control*, *Resources*, *Cost* and *Task Priority*;
3. A *front-end environment* using Roassal, which is an agile visualization engine to produce dynamic visualizations using business process elements.

A4BP supports six metrics (Table 5.15). n_1 is the number of unique activities, splits and joins, and control-flow elements; n_2 is the number of unique data variables manipulated by the process and its activities; N_1 and N_2 are respectively the total number of elements and data occurrences [Cardoso 2006] [Cardoso 2008].

Figure 5.16 shows the A4BP interface once a process model has been loaded. It shows the structure of the process, the list of metrics and their visualization, and makes it possible to define a simulation scenario.

The HCMC tsunami rescue plan process model (Figure 5.9) is used as input to build the static visualization (Figure 5.17) of organizations and their relations. An organization or element (represented by a rectangle) is evaluated based on *flow complexity*, *flow absolute complexity* and *number of relations* corresponding to *width* (CFC metric), *height* (CFCAbs metric) and *colour*, respectively.

⁷Assessment for Business Process <http://www.a4bp.com/>

⁸Pharo <http://pharo.org/>

⁹Roassal <http://agilevisualization.com/>

¹⁰BPSim <http://www.bpsim.org/>

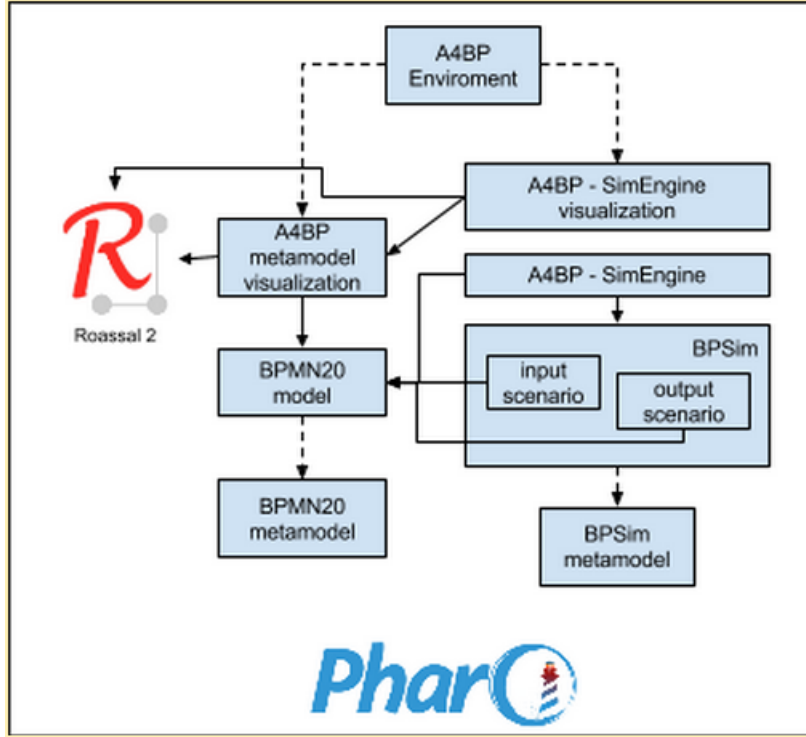


Figure 5.15: The A4BP architecture

Table 5.15: A4BP metrics

Metric	Description
Numbers of elements	Counting the number of element defined in the formal meta-model description
Control Flow Complexity (CFC)	Using Cardoso proposal for control flow complexity in business process
Control Flow Complexity Absolute (CFCAbs)	A variant of CFC used to find the complexity when the elements have more related split elements. The basic idea is to sum all CFC in order to have the absolute value
Process Length	$N = n_1 \times \log_2(n_1) + n_2 \times \log_2(n_2)$
Process Volume	$V = (N_1 + N_2) \times \log_2(n_1 + n_2)$
Process Difficulty	$D = (n_1/2) \times (N_2/n_2)$

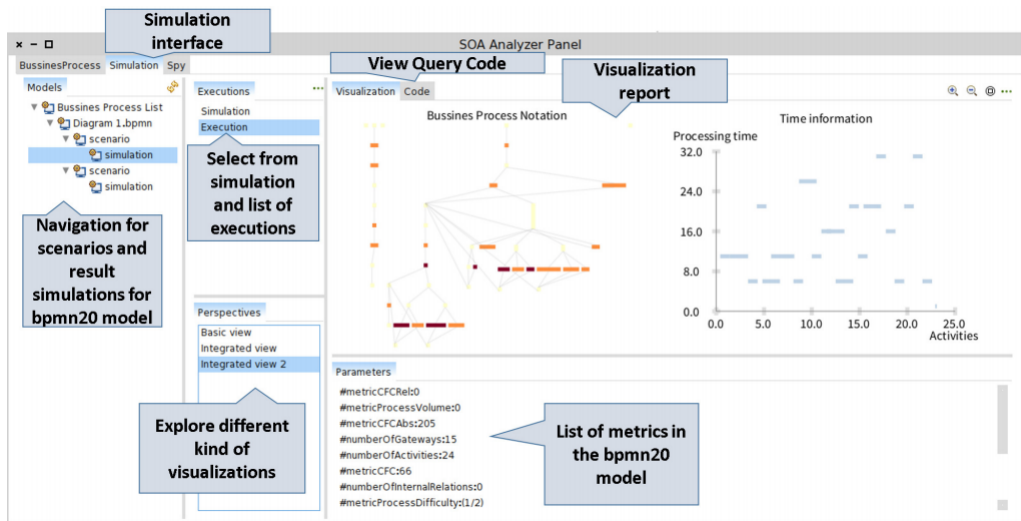


Figure 5.16: The A4BP interface

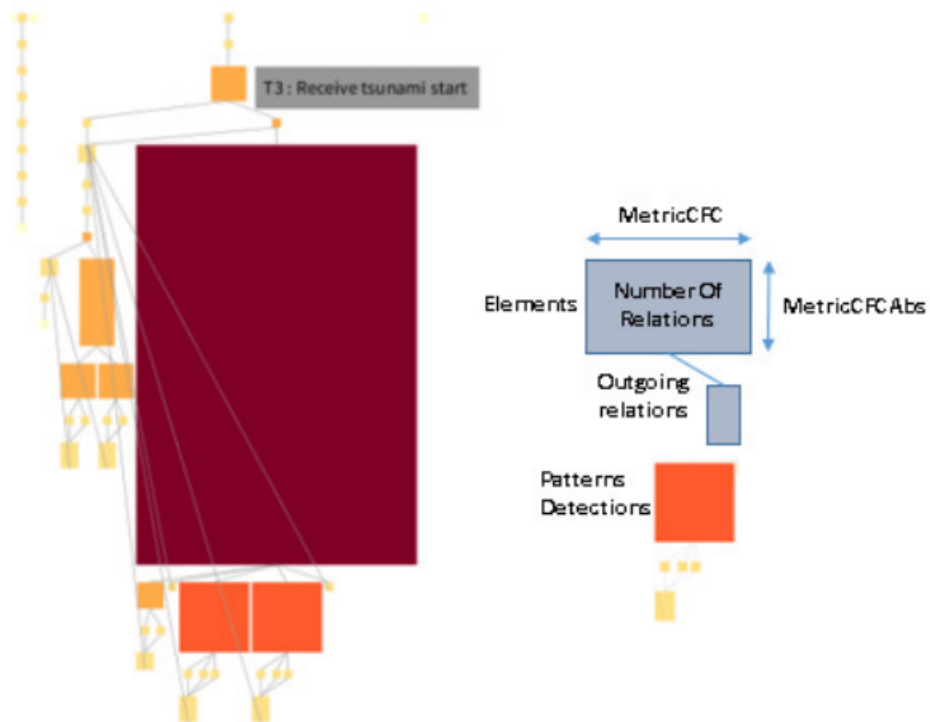


Figure 5.17: Visualization of process complexity in the HCMC tsunami rescue plan

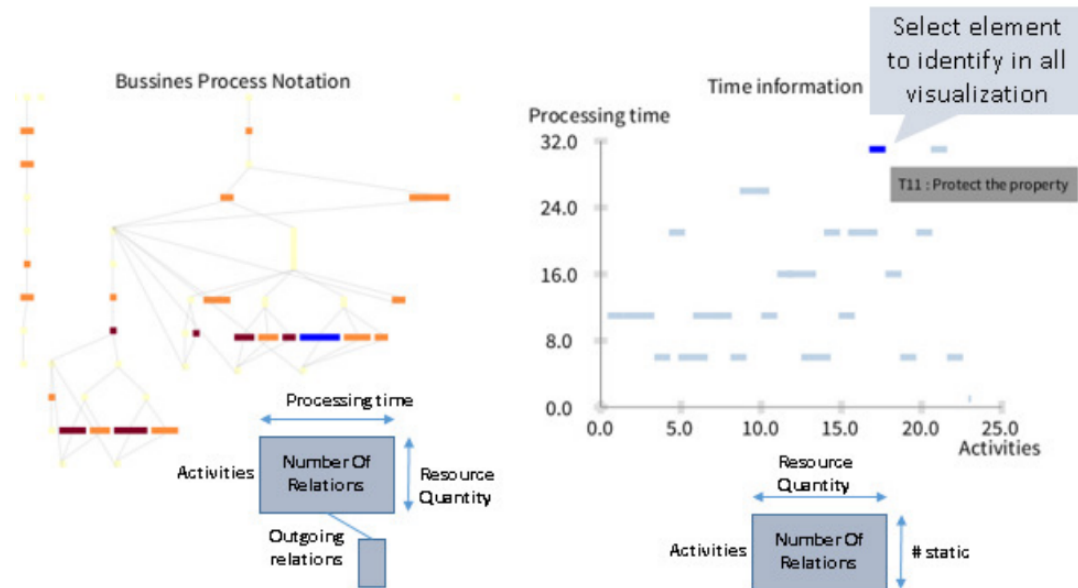


Figure 5.18: Visualization of process activities based on time and resources following the simulation of the post-HCMC tsunami rescue plan

This first visualization provides a more thorough analysis than the BPMN diagram (Figure 5.9) for each element, and is therefore more valuable for end-users and/or the authority. It is apparent that the parallel gateway element (the biggest rectangle) has not only the highest complexity (*width & height*), but also the highest relation (*colour*). It is likely that this element corresponds to a critical task in the rescue plan. The visualization also highlights recurrent patterns in terms of both structure and complexity. Based on these assessments, the rescue plan analyst might decide to enrich the initial visualization with information provided by other perspectives, for example the simulation.

Figure 5.18 shows the second visualization after executing BPSim engine with two input parameters (*time* and *resources*) to simulate the HCMC tsunami rescue plan. This visualization uses a static meta-model to show the elements, and a dynamic BPSim meta-model to capture time. Width and height values are used to represent rectangles. The view combines a static visualization of the BPMN diagram (on the left) with a visualization of the dynamic BPMN simulation (time chronograph) on the right. The combination of these two perspectives in the rescue plan analysis can help to understand the relationships between complex activities, and estimate the time needed to process each element.

5.7 Conclusion

This chapter has presented an approach for coordination from a process-oriented perspective. This approach consists of several components: a lifecycle, a crisis meta-model, recommendations in terms of combined notation and formalisms, model transformation rules, and the design and use of tools. The lifecycle includes four steps:

1. Identifying high-level objects from text with the support of a crisis meta-model;
2. Building scenarios;
3. Deriving the crisis resolution plan as a collaborative process. A process mining technique (the α -algorithm) is used to derive a Petri net-based process from scenarios; we then derive a BPMN representation of this process;
4. Simulating and analysing the obtained processes. We show how to analyse, check or validate properties and perform simulations. In particular, we present the A4BP tool that is used to measure process complexity as a complement to metrics proposed by [Cardoso 2008]

We believe that this approach can help authorities to have a more accurate view of the level of the tsunami response, and make more informed decisions at different stages of the crisis life-cycle (to adapt the plan, determine how many people are needed, improve stakeholder communication, etc.).

In the next chapter, we will present multi-agent-based coordination models for crisis management. Two types of these models, based on agents or organizations, will be used. Also, algorithms to convert process-based coordination models into multi-agent ones will be provided. Finally, the organizational model will be assessed through different metrics (flexibility, robustness and efficiency).

Organization-based Coordination Models

Contents

6.1	Introduction	115
6.2	OCMAS vs ACMAS Models	116
6.2.1	The AGR Model	117
6.2.2	The BDI Agent Model	119
6.3	The Role Graph	120
6.3.1	Grossi's Definition	120
6.3.2	Extracting the Role Graph from Textual Rescue Plans . .	125
6.3.3	Extracting the Role Graph from AGR models	126
6.3.4	New Structural Metrics for dynamic situations	127
6.4	Extracting Organizational Models from Business Processes	129
6.4.1	Mapping a Process Model onto an AGR Model	131
6.4.2	Mapping a Process Model onto a Role Graph	132
6.4.3	Mapping a Process Model onto a BDI Agent Model . . .	135
6.5	Case Study: the HCMC Tsunami Response Plan	137
6.5.1	Extracting the AGR Model from the Process Model . . .	137
6.5.2	Extracting the Role Graph Model from the Process Model	139
6.5.3	Extracting the BDI Agent Model from the Process Model	141
6.6	Assessment of the HCMC Tsunami Response Plan . . .	143
6.7	Conclusion	148

6.1 Introduction

In the previous chapter, we examined how to use a process-oriented view to model and analyse coordination in a crisis. Processes offer an aggregated view of the modelled phenomena with a focus on providing a representation of the response plan. However, in a crisis context that involves a great number of actors who need to be coordinated, there is another interesting perspective. The multi-agent point of view is based on the idea of capturing the behaviour of coordinating entities and other organizational aspects that are not usually taken directly and/or accurately into account by processes.

A Multi-Agent System (MAS) is composed of several agents that interact with each other and their environment in order to achieve their long-term goals. Among the different approaches, there are two principal types of model: Agent-centered Multi-Agent Systems (*ACMAS*) that include, for example, the Belief–Desire–Intention (BDI) paradigm, in which agents are considered as first-class entities; and Organization-centered Multi-Agent Systems (*OCMAS*), where a special focus is put on the organizational dimension, for example the Agent-Group-Role (AGR) model, GAIA, TROPOS and MOISE [Ferber 2004].

An *ACMAS* model assumes that agents have high-level reasoning skills, which can be used to describe the decision-making powers of stakeholders in a crisis unit, or to model autonomous, realistic actor behaviour. One of the main problems with the *ACMAS* approach, however, is that an agent can interact with all other agents with no barriers or restrictions, which can lead to chaotic (or at least inefficient) collective behaviour. Moreover, an agent is able to respond freely to requests from other agents, while in reality they are linked by their commitments to other agents, and should instead respect interaction rules defined by their role within the organization.

The majority of *ACMAS* models do not contain an explicit notion of the organization. In contrast, *OCMAS* is a particular type of MAS in which the notion of the organization is defined explicitly. Generally, an organization is split into sub-organizations called **groups** that may overlap. Agents occupy specific **roles** in these groups [Ferber 2004]. The organizational dimension of a MAS does not specify how the system works, but rather how it should evolve according to rules and norms that agents are expected to follow. The organization, therefore, does not define the behaviour of each agent, but simply collective constraints: every agent action that is not forbidden by collective constraints is allowed. Moreover, interaction in an *OCMAS* becomes contextual, because an agent can only interact with other agents belonging to its group.

This chapter describes algorithms that can be used to map BPMN diagrams onto agent models, both ACMAS (e.g. BDI Agent) and OCMAS (e.g. AGR). We extend the metrics of Grossi et al. to evaluate the quality of actual organizations deployed in real-life crises, which is an enhancement to the evaluation of abstract organizations proposed in [Grossi 2007].

We distinguish two types of organizations: at the design level, there is an abstract organization, which may correspond to different, concrete organizations at the real-life level. Here, a concrete organization is an instance of an abstract organization. In an abstract organization, it is not possible to know, or therefore to specify how many agents or groups of a certain type there are, while in a concrete organization we can launch and observe one or many groups or agents of the same type, within the constraints of the organizational model.

The remainder of this chapter is organized as follows: in the next section, we introduce two kinds of MAM: an OCMAS (AGR) model, and an ACMAS (BDI Agent) model. We then present a lifecycle, showing how a BPMN diagram, which describes the crisis management resolution plan, can be transformed into these MAM. In Section 6.3, another organizational model (role graph) is introduced. This model allows us to add metrics to evaluate the structure of abstract and concrete organizations. Finally, we validate our transformation algorithms based on a case study of a crisis management scenario in Ho Chi Minh City (HCMC) in Vietnam.

6.2 OCMAS vs ACMAS Models

Modelling a complex system (e.g. crisis management) as a MAS is an approach that has already been proposed by several researchers [Dugdale 2010].

Despite the multitude of MAS platforms, based on various agent architectures (cognitive, rational, reactive, cooperative, intentional, adaptive, communicative, ambient agent, etc.) there is no widely accepted definition of the term agent. Generally, agents are found in an environment where they can communicate and cooperate with each other.

The ACMAS approach focuses on individuals (agent behaviour and simple mutual interactions) and supports micro-simulations. However, the lack of social aspects leads to various weaknesses, including the difficulty of predicting the behaviour of the overall system, unpredictable interactions, security problems, etc. [Ferber 2004].

On the other hand, OCMAS supports macro-simulations that can be used

to evaluate the quality of organizations. In the context of crisis management, this aspect is of paramount importance given the fact that various stakeholders, represented by different actors, are involved in crisis resolution. Moreover, the model of the organization should include qualities such as robustness, efficiency and flexibility. Combining the strengths of the two approaches could lead to a complete simulation framework at both micro and macro levels.

Consequently, both approaches are illustrated in the following sub-sections using two representations: *AGR* and *BDI Agent*.

6.2.1 The AGR Model

The AGR model, as defined by Ferber [Ferber 2004], is a very minimal OCMAS model that only draws upon three concepts: agents, groups, and roles.

An *Agent* is an active entity with communication abilities. It can belong to one or more groups, and play several roles. Agents may communicate with each other only if they are in the same group. A *Group* is a collection of agents (roles) containing at least one agent (and one role). A *Role* is an abstract representation of a function, service or position of a set of agents within a group [Ferber 2004]. Each agent can hold one or many roles, the same role can be held by different agents, and roles always belong to groups. The attributes of a role include: skills/abilities, constraints (requirements, obligations), benefits (profits, authorizations) and responsibilities.

The behaviour of the overall system is determined by group structures and the roles authorized for agents. No particular internal structure is imposed on agents. An AGR model can be represented as a triple $AGR = (A, G, R)$ where:

- A is a collection of agents. Each agent is defined as a tuple $(NameA, T, Rs, Gs)$ in which $NameA$ is its identifier; T is its type (e.g. reactive or intentional); Rs is the list of roles that this agent can hold; and Gs is the list of groups to which this agent may belong.
- G is a collection of groups. Each group is defined as a couple $(NameG, Rs)$ in which $NameG$ is its identifier; and Rs is the list of roles for this group.
- R is a set of roles defined as a tuple $(NameR, C, B, D, Pc, I)$ where $NameR$ is its identifier; C is the list of constraints (e.g. obligations, requirements, skills); B is the list of benefits (e.g. abilities, authorization, profits); D is the list of duties or responsibilities; Pc is the pattern of communication or interaction, and I is useful information.

Although AGR is an abstract organizational model, [Ferber 2004] provided several primitives in order to implement organizations within a MAS. One example is the MADKIT¹ toolkit. In our transformation algorithms, we use a subset of these primitives, namely statements and actions.

- Statements
 - $roleIn(r, g)$: role r belongs to group g ;
 - $plays(a, r, g)$: agent a plays the role r in group g ;
 - $GStruct(g, gs)$: group g is an instance of group structure gs ;
 - $member(a, g)$: agent a is member of group g ;
- Actions
 - $x.send(y, m)$: denotes the action of an agent x sending a message m to agent y ;
 - $x.start$: this action means agent x initiates his state and/or work;
 - $x.do(act)$: this action means agent x performs the activity act ;
 - $x.wait(time)$: this action means agent x has to wait for a $time$;
 - $x.end$: this action means agent x terminates his work;

Figure 6.1 shows the AGR meta-model proposed by [Ferber 2004]. The relationship between two roles can be described by the *interaction protocol* or *structural constraint* (*correspondence* or *dependence*). A correspondence constraint between two roles A and B means that an agent with role A can automatically take the role B . A dependence constraint means that an agent can only take role A if it already has the role B .

The AGR meta-model has many advantages in our crisis management context:

- The key concepts of ACMAS models are represented, notably groups and roles. Transformations and results obtained with AGR can be very easily generalized to other ACMAS meta-models such as GAIA or TROPOS.
- The autonomy of agents is preserved because only the role is fixed by the group. There are no constraints on the internal structure of agents.

¹A lightweight Java library for designing and simulating Multi-Agent Systems <http://www.madkit.net/madkit/>

- Out is the list of its output variables;
- S is its script (a sequence of control flow elements) that may contain some actions:
 - $invoke(name)$ for invoking a plan for execution;
 - $addGoal(g)$ for adding a goal g ;
 - $send(m)$ for sending a message m ;
 - $receive(m)$ for receiving a message m ;
- 3. Γ is a set of goals that an agent tries to achieve. A goal γ is represented by a triple $\gamma = (Name, In, Out)$;
- 4. I is a set of intentions (i.e. selected plans) of the agent;
- 5. Θ is a set of known beliefs of the agent (i.e. a set of facts that the agent assumes to be true). A belief is represented by a triple $\theta = (Name, Value, Type)$.

In addition to the original model of [Michael 1987], [Endert 2007] introduces message exchanges between agents. A tuple $M = (Name, Sender, Receiver, Content)$ defines a message, where $Name$ is the id of the message, $Sender$ is the id of the sender of message, $Receiver$ is the id of the receiver of message, and $Content$ is the content of message.

Based on this definition, a BDI Agent interpreter operates as follows: in each cycle, according to the agent's perceptions, their beliefs are updated, and desires are revised. An intention is chosen or confirmed, and finally the corresponding plan is executed that potentially modifies the agent's environment.

6.3 The Role Graph

6.3.1 Grossi's Definition

AGR is a powerful tool for macro-simulations, but it is difficult to exploit individual executions in order to evaluate dynamic organizations. In order to overcome this issue, we introduce the simplified OCMAS model defined by Grossi et al. [Grossi 2007] called the *role graph* that can help to define metrics and compare crisis management organizations.

A role graph is a directed graph based on the definition of *Organizational Structure* (OS) proposed by Grossi et al. [Grossi 2007]. An *OS* is a tuple: $<$

$Roles, R_{Pow}, R_{Coord}, R_{Contr} >$ where $Roles$ are a set of roles held by a MAS, and $R_{Pow}, R_{Coord}, R_{Contr}$ are binary relations between roles that represent respectively Power, Coordination and Control structures.

The thinking behind these relationships is as follows: the *Power* relationship defines the pattern of task delegation; the *Coordination* dimension concerns the flow of knowledge within the organization; and the *Control* dimension operates between agents A and B and indicates that agent A must monitor agent B 's activities and possibly take over any incomplete tasks. These dimensions express the fact that an organization does not consist of only one dimension, but is a combination of several dimensions.

[Grossi 2007] defined $Roles_k$ with $k \in \{Pow, Coord, Contr\}$ as the smallest subset of $Roles$ that can cover R_k , such that if $(x, y) \in R_k$ then $x, y \in Roles_k$, i.e. $Roles_k$ is the set of roles involved in the structural dimension k (*Power*, *Coordination* or *Control*).

Furthermore, Grossi et al. [Grossi 2007] defined a set of metrics to assess the *Robustness*, *Flexibility* and *Efficiency* of an organization based on the relations between these roles. *Robustness* indicates the stability of an organization faced with anticipated risks. *Flexibility* is its capacity to adapt to environmental changes. *Efficiency* refers to the amount of resources it requires to execute its tasks.

In the following, we present Grossi et al.'s formal definition of these metrics [Grossi 2007]. We take as our example the *Completeness* and *Connectedness* of an *Organization Structure* (OS), to show how tightly roles are linked with one another in the context of the structural dimension k :

$$Completeness_k(OS) = \frac{|R_k|}{|Roles_k| * (|Roles_k| - 1)}$$

$$Connectedness_k(OS) = 1 - \frac{|DISCON_k|}{|Roles_k| * (|Roles_k| - 1)}$$

where $|R_k| \geq 0$ (i.e. roles in dimension k) and $DISCON_k$ is the set of ordered pairs (x, y) of $Roles_k$ (in order to not double count the same path from two roles). Intuitively, $Completeness_k(OS) \geq 0$ and measures the percentage of actual links in dimension k out of all of those available. The *Connectedness* metric measures how connected the dimension is. For example, some organizations may have a key role that, when removed, decreases the connectedness of the whole structure.

In total, eleven metrics are provided by Grossi et al. [Grossi 2007] which can be divided into three main groups:

1. *Completeness, Connectedness and Economy.* The first two metrics measure how roles are linked to each other in dimension k . *Economy* is a compromise between *Completeness* and *Connectedness*.
2. *Unilaterality and Univocity.* These metrics measure the level of subordination in a structure by examining the orientation of the link (*Unilaterality*), conflicts or redundancy (*Univocity*):
 - SIM_k is the set of symmetric links (x, y) in k , meaning that if $(x, y) \in R_k$ then $(y, x) \in R_k$;
 - IN_k is the set of roles $x \in Roles_x$ whose in-degree in k is equal to 1 ($id_k(x) = 1$) or that are a source of k ($id_k(x) = 0$); and
 - CUT_k is the set of roles x whose out-degree and in-degree in k are greater than or equal to 1 ($od_k(x) \geq 1$ and $id_k(x) \geq 1$).
3. *InCover, OutCover, Chain, Detour and Overlap.* This group of metrics correlates two structural dimensions. *InCover* and *OutCover* measure the number of ingoing and outgoing links for each role in a dimension compared to the same role in another dimension. *Chain* measures the number of roles that are ingoing links in one dimension and outgoing links in another. Finally, *Detour* and *Overlap* (*Overlap* being a special case of *Detour*) measure the degree of the structural dimension j that “follows” a structural dimension k :
 - $PATH_{jk}$ is the set of ordered pairs (x, y) where there is a path in dimension j from x to y and x to y are also in dimension k . $LINK_{jk}$ is the special case of $PATH_{jk}$ with a path length equal to 1;
 - IN_i^+ is the set of elements $x \in Roles_k$ whose in-degree in dimension i is smaller than or equal to 1 ($id_i(x) \leq 1$); and
 - OUT_i^+ is the set of elements $x \in Roles_i$ whose out-degree in dimension i is smaller than or equal to 1 ($od_i(x) \leq 1$).

$$Completeness_k(OS) = \frac{|R_k|}{|Roles_k| \times (|Roles_k| - 1)} \quad (6.1)$$

$$Connectedness_k(OS) = 1 - \frac{|DISCON_k|}{|Roles_k| \times (|Roles_k| - 1)} \quad (6.2)$$

$$Economy_k(OS) = 1 - \frac{|R_k| - (|Roles_k| - 1)}{|Roles_k| \times (|Roles_k| - 1) - (|Roles_k| - 1)} \quad (6.3)$$

$$Unilaterality_k(OS) = 1 - \frac{|SIM_k|}{|R_k|} \quad (6.4)$$

$$Univocity_k(OS) = \frac{|IN_k|}{|Roles_k|} \quad (6.5)$$

$$Flatness_k(OS) = 1 - \frac{|CUT_k|}{|Roles_k|} \quad (6.6)$$

$$Detour_{jk}(OS) = \frac{|PATH_{jk}|}{|R_k|} \quad (6.7)$$

$$Overlap_{jk}(OS) = \frac{|LINK_{jk}|}{|R_k|} \quad (6.8)$$

$$InCover_{jk}(OS) = \frac{|IN_j^+ \cap IN_k^+|}{|IN_k^+|} \quad (6.9)$$

$$OutCover_{jk}(OS) = \frac{|OUT_j^+ \cap OUT_k^+|}{|OUT_k^+|} \quad (6.10)$$

Robustness		Flexibility		Efficiency	
$Overlap_{Coord-Pow}$	1	$Completeness_{Pow}$	0	$Connectedness_{Pow}$	1
$Chain_{Contr-Pow}$	1	$Connectedness_{Pow}$	0	$Economy_{Pow}$	1
$Chain_{Contr-Coord}$	1	$Chain_{Contr-Pow}$	1	$Economy_{Coord}$	1
$InCover_{Contr-Coord}$	1	$Completeness_{Coord}$	1	$Overlap_{Coord-Pow}$	1
$OutCover_{Pow-Contr}$	1	$Connectedness_{Coord}$	1	$Overlap_{Pow-Coord}$	1
$OutCover_{Pow-Coord}$	1	$OutCover_{Pow-Contr}$	1	$Unilaterality_{Pow}$	1
$Completeness_{Coord}$	1			$Univocity_{Pow}$	1
$Connectedness_{Coord}$	1			$Economy_{Contr}$	1
$Univocity_{Pow}$	0			$Overlap_{Contr-Pow}$	1
$Unilaterality_{Coord}$	0			$Overlap_{Pow-Contr}$	1
$Univocity_{Contr}$	0				
$Flatness_{Contr}$	0				

Figure 6.2: Optimal values for the maximization of robustness, flexibility and efficiency as defined by Penserini et al. [Penserini 2009], and Grossi et al. [Grossi 2007]

$$Chain_{jk}(OS) = \frac{|IN_j^+ \cap OUT_k^+|}{|OUT_k^+|} \quad (6.11)$$

Given these three dimensions and eleven metrics defined for dimension k , we can define 33^2 full metrics. All metrics have a value of between 0 and 1. Penserini et al. [Penserini 2009] apply the Grossi metrics to disaster management, with the addition of two more metrics that solve issues related to star-like organizations.

In addition to these structural metrics, Grossi et al. defined other criteria to characterize high-level properties: *Robustness*, *Flexibility* and *Efficiency*. The results of the metrics can be compared with optimum values proposed by [Grossi 2007] in order to evaluate these criteria (see Figure 6.2).

In order to apply the model to our scenario, we defined a role graph corresponding to the tsunami response plan (Figure 5.9, Chapter 5), as shown in Figure 6.3. This graph contains seven roles (actors) represented by nodes (labelled IG , LA , P , M , $LCDF$, HR , and CU). An arc corresponds to a relationship between two nodes. We consider three types of arcs: i) red arcs are labelled p for the *Power* relation; ii) green arcs are labelled c for the *Coordination* relation;

²There are 3 dimensions. 6 metrics deal with one dimension only (this gives 18 metrics for all 3 dimensions). And 5 metrics that are between two dimensions (15 for all 3 dimensions). So a total of $18 + 15 = 33$ metrics

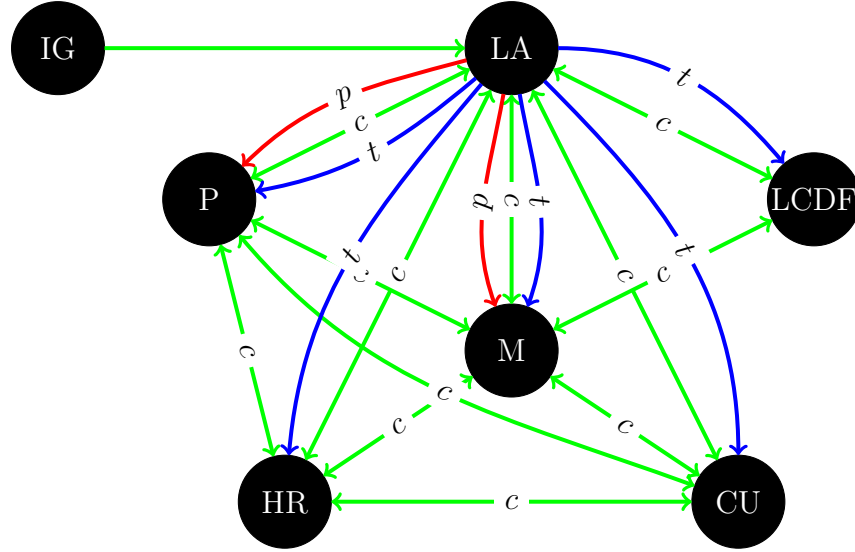


Figure 6.3: The role graph built by stakeholders based on the tsunami response plan

and iii) blue arcs are labelled t for the *Control* relation.

6.3.2 Extracting the Role Graph from Textual Rescue Plans

A Role Graph derived from a response plan can be built from different sources (textual plans, BPMN diagrams, etc.). Textual plans are the most common format in crisis management. Therefore, here we propose some guidelines for identifying roles and their relations from this type of document. While an actor can be considered as a role, the relations (*Power*, *Coordination*, and *Control*) between two roles can be identified by applying the following three rules:

- If actor A can transfer tasks to actor B , we create a *Power* relation between them;
- If two actors can communicate or interact with each other (sending and receiving messages), we create a *Coordination* relation between them;
- If actor A can monitor the activities of actor B , we create a *Control* relation between them.

Algorithm 2 is designed to detect the roles and their relationships in textual plans. Lines 1 to 2 identify roles. Lines 3 to 9 detect relations between two roles, specifically: i) lines 4 to 5 identify the *Power* relationship; ii) lines 6 to 7 identify the *Coordination* relationship; and iii) lines 8 to 9 identify the *Control* relationship.

Algorithm 2 Algorithm to identify the role graph from textual plans

Input: a textual plan

Output: a role graph defined by set of roles R and binary relations: R_{Pow} , R_{Coord} and R_{Contr}

% {Step 1: Identify roles}

1: **for each** actors in *textualPlan* **do**

2: Create new Role $R[i]$

% {Step 2: Identify relations between two roles}

3: **for each** pair $R[i]$ and $R[j]$ **do**

4: **if** $R[i]$ can transfer tasks to $R[j]$ **then**

5: Create $R_{pow}[k]$ to connect $R[i]$ and $R[j]$

6: **else if** $R[i]$ can communicate/interacts with $R[j]$ **then**

7: Create $R_{coord}[l]$ to connect $R[i]$ and $R[j]$

8: **else if** $R[i]$ can monitor the activities of $R[j]$ **then**

9: Create $R_{contr}[m]$ to connect $R[i]$ and $R[j]$

6.3.3 Extracting the Role Graph from AGR models

The Role Graph expressing the relationship between stakeholders can be derived not only from textual plans but also from AGR models thanks to the similarities between the two representations, and by drawing upon the following three principles:

- If there is an *interaction* between Role A and Role B (in the same group) in AGR, we create a *Coordination* relation between them in the role graph;
- If there is a *correspondence constraint* between Role A and Role B , we create a *Power* relation from A to B in the role graph;
- If there is a *dependence constraint* between Role A and Role B , we create a *Control* relation from A to B in the role graph;

6.3.4 New Structural Metrics for dynamic situations

The role graph model developed by Grossi et al. [Grossi 2007] is a directed graph without weighted edges. The assumption is that all edges have the same degree (i.e. equal to 1). This assumption is only acceptable in static situations, i.e. pre-defined crisis plans that are mostly used in the mitigation or preparedness phases. In reality (and especially in the response phase) due to the dynamic situation, the intensity of relationships between a pair of roles evolves and can change from one crisis to another. During a crisis response, the number of communications or interactions between two stakeholders increases; consequently the degree of their relationship (corresponding to the edge's degree) also increases. Moreover, a relation between two roles that exists in a normal situation can disappear in a crisis. Here, it is necessary to distinguish between the structure of a relation and its actual use at runtime.

Therefore, we suggest improving Grossi et al.'s framework to take account of dynamic situations, based on a weighted graph. This new framework can be used in both static and dynamic situations.

The metrics proposed by [Grossi 2007] are reconsidered in this new context and two new notations are added: i) $Degree_{k,i}$ is the degree of edge i in dimension k ; and ii) $Max_{Degree.k}$ is the maximum degree in dimension k . The metrics *Completeness*, *Connectedness*, *Economy*, *Unilaterality*, *Univocity*, *Flatness*, *Detour*, *Overlap*, *InCover*, *OutCover* and *Chain* are redefined as follows:

$$Completeness_k(OS) = \frac{\sum_{i=1}^{|R_k|} Degree_{k,i}}{|Roles_k| \times (|Roles_k| - 1) \times Max_{Degree.k}} \quad (6.12)$$

$$Connectedness_k(OS) = 1 - \frac{\sum_{i=1}^{|DISCON_k|} Degree_{k,i}}{|Roles_k| \times (|Roles_k| - 1) \times Max_{Degree.k}} \quad (6.13)$$

$$Economy_k(OS) = 1 - \frac{\sum_{i=1}^{|R_k|} Degree_{k,i} - (|Roles_k| - 1)}{|Roles_k| \times (|Roles_k| - 1) \times Max_{Degree.k} - (|Roles_k| - 1)} \quad (6.14)$$

$$Unilaterality_k(OS) = 1 - \frac{\sum_{i=1}^{|SIM_k|} Degree_{k,i}}{|R_k| \times Max_{Degree.k}} \quad (6.15)$$

$$Univocity_k(OS) = \frac{\sum_{i=1}^{|IN_k|} Degree_{k,i}}{|Roles_k| \times Max_{Degree.k}} \quad (6.16)$$

$$Flatness_k(OS) = 1 - \frac{\sum_{i=1}^{|CUT_k|} Degree_{k,i}}{|Roles_k| \times Max_{Degree.k}} \quad (6.17)$$

$$Detour_{jk}(OS) = \frac{\sum_{i=1}^{|PATH_{jk}|} Degree_{jk,i}}{\sum_{i=1}^{|R_k|} Degree_{k,i}} \quad (6.18)$$

$$Overlap_{jk}(OS) = \frac{\sum_{i=1}^{|LINK_{jk}|} Degree_{jk,i}}{\sum_{i=1}^{|R_k|} Degree_{k,i}} \quad (6.19)$$

$$InCover_{jk}(OS) = \frac{\sum_{i=1}^{|IN_j^+ \cap IN_k^+|} Degree_{jk,i}}{\sum_{i=1}^{|IN_k^+|} Degree_{k,i}} \quad (6.20)$$

$$OutCover_{jk}(OS) = \frac{\sum_{i=1}^{|OUT_j^+ \cap OUT_k^+|} Degree_{jk.i}}{\sum_{i=1}^{|OUT_k^+|} Degree_{k.i}} \quad (6.21)$$

$$Chain_{jk}(OS) = \frac{\sum_{i=1}^{|IN_j^+ \cap OUT_k^+|} Degree_{jk.i}}{\sum_{i=1}^{|OUT_k^+|} Degree_{k.i}} \quad (6.22)$$

In static situations, where all $Degree_{k.i}$ are equal to 1 (and thus $Max_{Degree.k}$ also equals 1), these equations have the same values as the original metrics proposed by [Grossi 2007].

6.4 Extracting Organizational Models from Business Processes

The idea of combining business process and MAS models has already been proposed, with the aim of improving agent-based design [Küster 2012] and perform analysis that draw upon the strengths of two paradigms (e.g. control flow complexity metrics for process models [Cardoso 2008], and organizational structure metrics for OCMAS [Grossi 2007]). Process models could be considered as complementary to agent models, as they can represent an aggregate view of a MAS behaviour. Process models also share several MAS concepts. Therefore, we argue that a marriage of process and agent models is a good way to design an efficient coordination framework for complex systems such as crisis management systems [Küster 2014]. While stakeholders and their behaviour can be described by agent models, the crisis resolution plan is better-suited to a process representation.

Figure 6.4 shows the lifecycle of extracting organizational models (*Role Graph*, *ACMAS*, *OCMAS*) from a process model.

While BPMN provides an understandable and aggregate representation of stakeholder behaviour and offers the ability to analyse and simulate processes, the role graph focuses on dependencies between roles and enables an analysis of

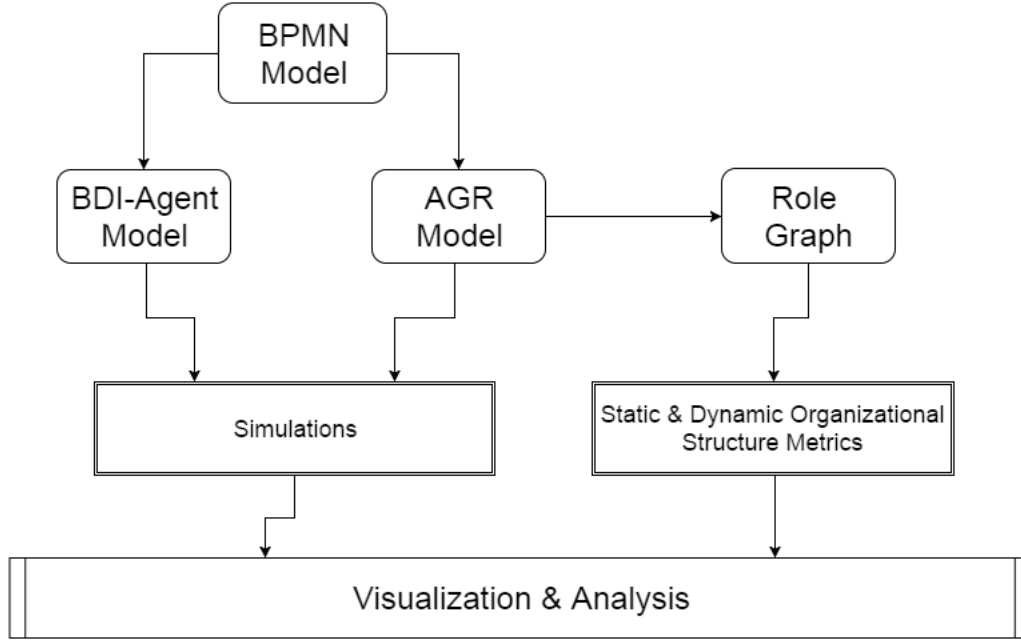


Figure 6.4: Lifecycle of mapping the BPMN model to organizational models (role graph, ACMAS, OCMAS)

the robustness, flexibility and efficiency of organizational structures. OCMAS supports macro-simulations, while ACMAS supports micro-simulations.

In order to simplify the extraction from a business process, we consider, following Endert [Endert 2007], a simplified BPMN diagram. This diagram can be seen as a graph (O, F) where O is the set of nodes (objects, swimlanes, or artifacts), and F is the set of edges (message flows, sequence flows), each edge being a function $O \rightarrow O$ from a source node to a target node [Endert 2007].

In order to represent BPMN elements, we use the notation X_I^J , where X is O or F , I is a type and J a subtype:

1. *Node objects*

- Event nodes (O^E): like Start Event (O_S^E), Intermediate Event (O_I^E) or End Event (O_E^E)
- Activity nodes (O^A): like Atomic Activity (O_{At}^A) or Sub-Process (O_{Sub}^A)
- Gateway nodes (O^G): like Exclusive (O_X^G), Inclusive (O_O^G), Parallel (O_A^G)
- Pool nodes (O^P)

- Lane nodes (O^L)
- Artifacts (O^{At}): like Group (O_G^{At}), Annotation (O_A^{At}), Image (O_I^{At}), Header (O_H^{At}), Formatted Text (O_T^{At}) etc.

2. Connectors

- Sequence Flow (F^S)
- Message Flow (F^M)
- Association (F^A)

6.4.1 Mapping a Process Model onto an AGR Model

To map a process model onto an AGR model we consider a lane, or pool without a lane as a role. A group constitutes a context for interaction between agents. Hence, we consider two cases: i) each pool with more than one lane becomes a group; and ii) for each message flow between two pools A and B, we create a new group in which the roles of A and B can be played. Agents are not directly incorporated into BPMN diagrams, but can be represented by additional data that indicates the number of occurrences of agents for each role.

Algorithm 3 shows how the AGR model is derived from a process model. We implemented this algorithm using ATL (see Section D.3 of Appendix D). The mapping consists of five steps, as follows.

1. **Step 1** (lines 1 to 6): The roles and groups are extracted from BPMN diagrams. Let us illustrate this step through our tsunami response case study. We have the first pool $O^P(1)$ where $O^P(1).name = \text{"Institute of Geophysics"}$ and it has no lane, therefore we consider it as a role $R(1)$. In contrary, for the second pool $O^P(2)$ where $O^P(2).name = \text{"Ho Chi Minh City"}$, it has six lanes: $O^L(1)$ ($O^L(1).name = \text{"Local Administration"}$), $O^L(2)$ ($O^L(2).name = \text{"Local Civil Defence Forces"}$), $O^L(3)$ ($O^L(3).name = \text{"Communication Unit"}$), $O^L(4)$ ($O^L(4).name = \text{"Military"}$), $O^L(5)$ ($O^L(5).name = \text{"Police"}$) and $O^L(6)$ ($O^L(6).name = \text{"Health \& Red Cross"}$). Thus we transfer them respectively to six roles $R(2)$, $R(3)$, $R(4)$, $R(5)$, $R(6)$ and $R(7)$ which belong to group $G(1)$;
2. **Step 2** (lines 7 to 9): The properties of role are identified by obtaining information extracted from the artifact elements;

3. **Step 3** (lines 10 to 13): The attributes of agent (e.g. type, number of agents playing a role, number of agents belonging to a group, etc.) are identified by analysing user's additional data;
4. **Step 4** (lines 14 to 18): The communication or interaction protocol between groups and new possible groups are identified by analysing message flows. In our case study, we create a new group $G(2)$ based on the message flows (F^M) between two roles $R(1)$ and $R(2)$;
5. **Step 5** (lines 19 to 20): The activities of roles are identified by analysing sequence flows (F^S).

6.4.2 Mapping a Process Model onto a Role Graph

The role graph aims to analyse the properties of an organization involved in a crisis plan, notably its robustness, flexibility and efficiency as presented in [Le 2015]. They can be derived from a (business) process model using mapping rules. Roles correspond to lanes (or pools without lanes) in a process. The relationships between roles are not defined clearly in a BPMN diagram. Therefore, we propose three patterns to represent three types of relation, by analysing the semantics of *Connector* elements found in the BPMN model (*Sequence Flow*, *Message Flow*, *Association*), as follows.

- *Power* relation: if lane/pool A has a *unidirectional* message or sequence flow with another lane/pool B , we assume that there is a power relation from A to B . For example, as shown in Figure 6.5, two sequence flows connect lane A to lane B , while there is no flow in the opposite direction. Thus we conclude that role A has a power relation with role B .
- *Coordination* relation: if lane/pool A has *bidirectional* message or sequence flows with another pool/lane B , we assume that there is a coordination relation between A and B , as illustrated in Figure 6.6.
- *Control* relation: if lane/pool A has *bidirectional* message or sequence flows for all tasks with another lane/pool B , we assume that A controls B , as illustrated in Figure 6.7.

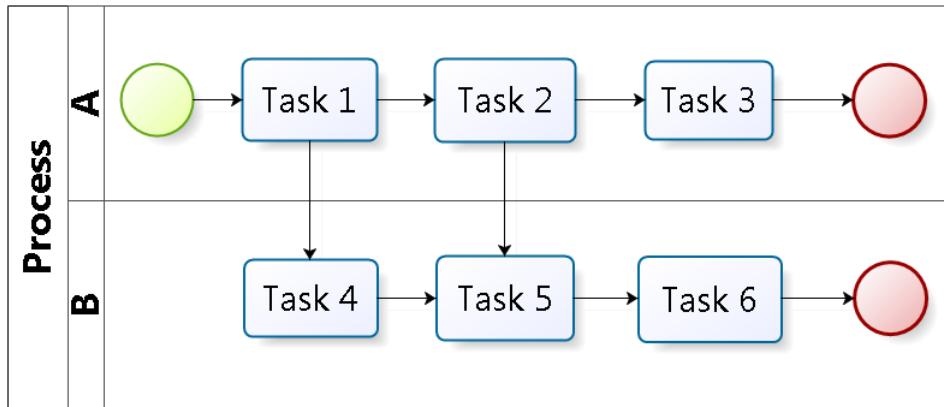
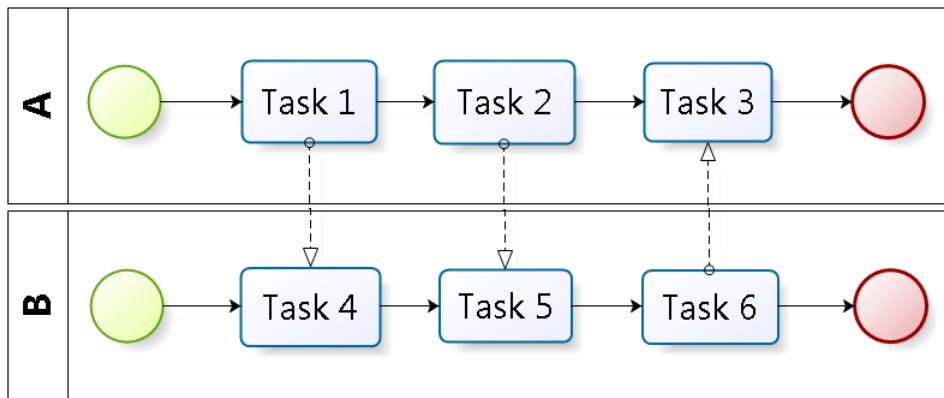
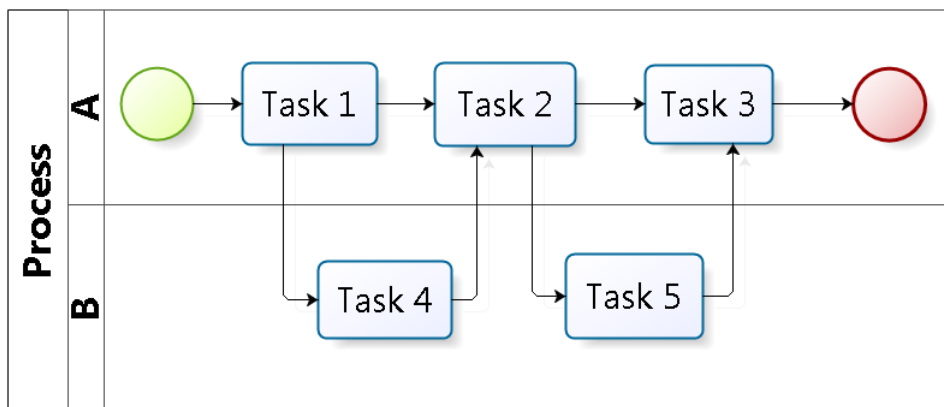
Our role graph is derived using these three heuristics. An alternate approach is to use *Artifact* elements to express directly (by text) the *Power*, *Coordination*, or

Algorithm 3 Mapping from the business process model to the AGR model**Input:** a BPMNModel**Output:** an AGRModel defined as a tuple (A, G, R)

```

% {Step 1: Extract the roles and groups from process model}
% {Create only one role if there is no lanes in a pool, a role for each lanes
otherwise}
1: for each pool  $O^P[i]$  do
2:   if  $O^P[i]$  does not contain lanes then
3:      $R[i] \leftarrow (O^P[i].name, C\{\}, B\{\}, D\{\}, Pc\{\}, I\{\})$ 
4:   else
5:     Create new group  $G[j]$ 
6:     for each lane  $O^L[k]$  that belongs to  $O^P[i]$  do
7:        $R[k] \leftarrow (O^L[k].name, C\{\}, B\{\}, D\{\}, Pc\{\}, I\{\})$ 
8:        $G[j] \leftarrow R[k]$ 
% {Step 2: Obtain informations from artifact elements to identify roles' prop-
erties}
9: for each artifact  $O^{At}[j]$  do
10:  if  $O^{At}[j]$  contains roles properties (C, B, D or I) then
11:     $R[i].C \leftarrow R[i].C \cup O^{At}[j]$ 
12:     $R[i].B \leftarrow R[i].B \cup O^{At}[j]$ 
13:     $R[i].D \leftarrow R[i].D \cup O^{At}[j]$ 
14:     $R[i].I \leftarrow R[i].I \cup O^{At}[j]$ 
% {Step 3: Use additional data for creating agents. Each agent is defined by
his name, type, groups and roles endorsed}
15: for each roles, groups and agents to be defined do
16:    $Rs[j] \leftarrow \{R_{j1} : Nb_{j1}, R_{j1+1} : Nb_{j1+1}, \dots\}$ 
17:    $Gs[k] \leftarrow \{G_{k1}, G_{k1+1}, \dots\}$ 
18:    $A[i] \leftarrow (Name, Type, Rs[j], Gs[k])$ 
% {Step 4: Identify communication between Roles & Create new Groups
based on Message Flows}
19: for each message flow  $F^M[i]$  do
20:    $R[i1].Pc \leftarrow \{send(R[i2], F^M[i].msg)\}$ 
21:    $R[i2].Pc \leftarrow \{receive(R[i1], F^M[i].msg)\}$ 
22:   if Group  $(F^M[i].msg, \{R[i1], R[i2]\})$  does not exist then
23:      $G[k] \leftarrow (F^M[i].msg, \{R[i1], R[i2]\})$ 
% {Step 5: Identify activities of roles based on Sequence Flows}
24: for each sequence flow  $F^S[i]$  do
25:    $R[j].D \leftarrow R[j].D \cup F^S[i].sourceTask$ 
26:    $R[j].D \leftarrow R[j].D \cup F^S[i].targetTask$ 

```

Figure 6.5: Pattern to detect the *Power relation* between two actorsFigure 6.6: Pattern to detect the *Coordination relation* between two actorsFigure 6.7: Pattern to detect the *Control relation* between two actors

Control relations. However, it is difficult to guarantee, when designing a process model, that users will supply relationship information as it represents additional work for them.

Algorithm 4 shows how to derive a role graph from a process model. It consists of the following two steps.

1. **Step 1** (lines 1 to 8): Roles are identified from lanes or pools;
2. **Step 2** (lines 9 to 26): The relationships between roles are built based on the three proposed patterns.

6.4.3 Mapping a Process Model onto a BDI Agent Model

A BDI-Agent is defined as a tuple $\Delta = (id, P, G, I, B)$ (as defined in Section 6.2.2). We propose Algorithm 5 to derive a BDI-agent model from a process model (i, j, k, z are used as arbitrary variables). We implemented this algorithm using ATL (see Section D.4 of Appendix D). It contains nine steps, corresponding to the rules presented in [Endert 2007]:

1. **Step 1:** Each pool gives rise to an agent Δ with the same name as its corresponding pool. Other parameters (P, G, I, B) are empty;
2. **Step 2:** The plan (P) of an agent is initiated. Line 3 of Algorithm 5 means *creating the plan of an agent where its ID is the process ID of the pool and the other elements ($In, Out, Script$) are empty*;
3. **Step 3:** The input list of a plan ($P.In$) is completed with start events (lines 4 to 5 of Algorithm 5);
4. **Step 4:** The output list of a plan ($P.Out$) is completed with end events (lines 6 to 7 of Algorithm 5);
5. **Step 5:** The embedded sub-process activities are transferred to another plans of agent (P') corresponding to lines 9 to 11 of Algorithm 5;
6. **Step 6:** The independent sub-process activities are mapped to goals of agents (G) corresponding to lines 12 to 14 of Algorithm 5;
7. **Step 7:** The elements with *Send* and *Receive* messages are appended to plan's script ($P.Script$) corresponding to lines 15 to 20 of Algorithm 5;

Algorithm 4 Mapping the from business process to the role graph

Input: a BPMN model**Output:** a role graph RG

```

    % {Step 1: Identify the roles}
  1: for all  $O^P[i]$  do
  2:   if  $O^P[i]$  does not contain lane then
  3:      $R[j].name = \text{name of } O^P[i]$ 
  4:      $RG \leftarrow RG \cup R[j]$ 
  5:   else
  6:     for all  $O^L[k]$  belongs to  $O^P[i]$  do
  7:        $R[j].name \leftarrow \text{name of } O^L[k]$ 
  8:        $RG \leftarrow RG \cup R[j]$ 
    % {Step 2: Build the relationship between roles based on three patterns}
  9: for all  $F^M[i]$  do
 10:    $R[s] \leftarrow F^M[i].src$ 
 11:    $R[t] \leftarrow F^M[i].tar$ 
 12:   if  $(R[s], R[t]) \notin R_{coord}$  then
 13:      $R_{coord} \leftarrow R_{coord} \cup (R[s], R[t])$ 
 14: for all  $F^S[i]$  do
 15:    $R[s] \leftarrow F^S[i].src$ 
 16:    $R[t] \leftarrow F^S[i].tar$ 
 17:   if  $R[s] \neq R[t]$  then
 18:     if  $(R[t], R[s]) \in A^{pow}$  then
 19:        $R_{pow} = R_{Pow} - (R[t], R[s])$ 
 20:        $R_{coord} \leftarrow R_{coord} \cup (R[t], R[s])$ 
 21:     else
 22:        $R_{pow} \leftarrow R_{pow} \cup (R[s], R[t])$ 
 23: for all  $(R[i], R[j]) \in R_{coord}$  do
 24:   if  $\sum((R[i], R[j]) \in R_{coord}) = \sum(O^A \in R[j])$  then
 25:      $R_{coord} \leftarrow R_{coord} - (R[i], R[j])$ 
 26:      $R_{contrl} \leftarrow R_{contrl} \cup (R[i], R[j])$ 

```

8. **Step 8:** The data flows (additional information of pools) are mapped to the belief of agents (B) corresponding to line 21 of Algorithm 5;
9. **Step 9:** The control flows (gateways) are considered to orchestrate the structure of agents' plan with AND, OR, XOR structure (lines 22 to 23 of Algorithm 5).

6.5 Case Study: the HCMC Tsunami Response Plan

In this section we apply transformation algorithms 3, 4, and 5 to the HCMC tsunami response plan used as a case study. The BPMN diagram of the tsunami response plan (Figure 5.9) is the source of the mapping.

6.5.1 Extracting the AGR Model from the Process Model

Table 6.1 shows the result of the mapping from the business process to the AGR model. It shows that two groups and seven roles, together with their attributes, interactions and responsibilities are identified. $R(2)$ is the only role that belongs to both groups.

Table 6.1: Mapping the business process to the AGR model for the HCMC tsunami response plan

	BPMN	AGR	Applied to HCMC Plan
Step 1 Identify Roles & Groups	$O^P(1)$ has no lane	$R(1) = (O^P(1).name, C\{\}, B\{\}, D\{\}, Pc\{\}, I\{\})$	Institute of Geophysics.
	$O^P(2)$ has 6 lanes	$R(2) = (O^L(1).name, C\{\}, B\{\}, D\{\}, Pc\{\}, I\{\})$	Local Administration.
		$R(3) = (O^L(2).name, C\{\}, B\{\}, D\{\}, Pc\{\}, I\{\})$	Local Civil Defence Forces.
		$R(4) = (O^L(3).name, C\{\}, B\{\}, D\{\}, Pc\{\}, I\{\})$	Communication Unit.
		$R(5) = (O^L(4).name, C\{\},$	Military.
Continued on next page			

Table 6.1 – Continued from previous page

	BPMN	AGR	Applied to HCMC Plan
		$B\{\}, D\{\}, Pc\{\}, I\{\}$ $R(6) = (O^L(5).name, C\{\},$ $B\{\}, D\{\}, Pc\{\}, I\{\})$ $R(7) = (O^L(6).name, C\{\},$ $B\{\}, D\{\}, Pc\{\}, I\{\})$ $G(1) = (O^P(2).name, \{R(2),$ $R(3), R(4), R(5),$ $R(6), R(7)\})$	Police. Health & Red Cross. Ho Chi Minh City.
Step 2 Identify Roles' Properties by examining Artifacts	N/A ³	N/A	N/A
Step 3 Identify Agents by reading additional data		$A(i) = (Name, Type,$ $Rs(i), Gs(i))$ $Rs(i) = \{R_k : Nb_k,$ $R_{k+1} : Nb_{k+1}, \dots\}$ $Gs(i) = \{G_j, G_{j+1}, \dots\}$	
Step 4 Identify communication between Groups & Create new group based on Message Flow	$F^M(1)$ $F^M(2)$	$R(1).Pc \leftarrow \{send(R(2),$ $F^M(1).msg)\}$ $R(2).Pc \leftarrow \{receive(R(1),$ $F^M(1).msg)\}$ $R(1).Pc \leftarrow \{send(R(2),$ $F^M(2).msg)\}$ $R(2).Pc \leftarrow \{receive(R(1),$ $F^M(2).msg)\}$ $G(2) = (F^M(1, 2).msg,$ $\{R(1), R(2)\})$	Message: <i>Tsunami Start</i> Message: <i>Tsunami End</i>
Step 5 Identify Roles' activities based on	$F^S(1),$ $F^S(2)$ $F^S(4),$ $F^S(5)$ $F^S(6),$	$R(1).D \leftarrow \{Start, Do(O_{At}^A(1))\},$ $\{Do(O_{At}^A(2))\}$ $R(1).D \leftarrow \{Wait(O_{I,T}^E(1))\},$ $\{Do(O_{At}^A(3))\}$ $R(1).D \leftarrow \{Do(O_{At}^A(4))\},$	<i>T1: Detect Tsunami risk</i> <i>T2: Inform Tsunami start</i> Timer Event <i>T14: Detect Tsunami end</i> <i>T15: Inform Tsunami end</i>
Continued on next page			

³There are no artifacts in this example

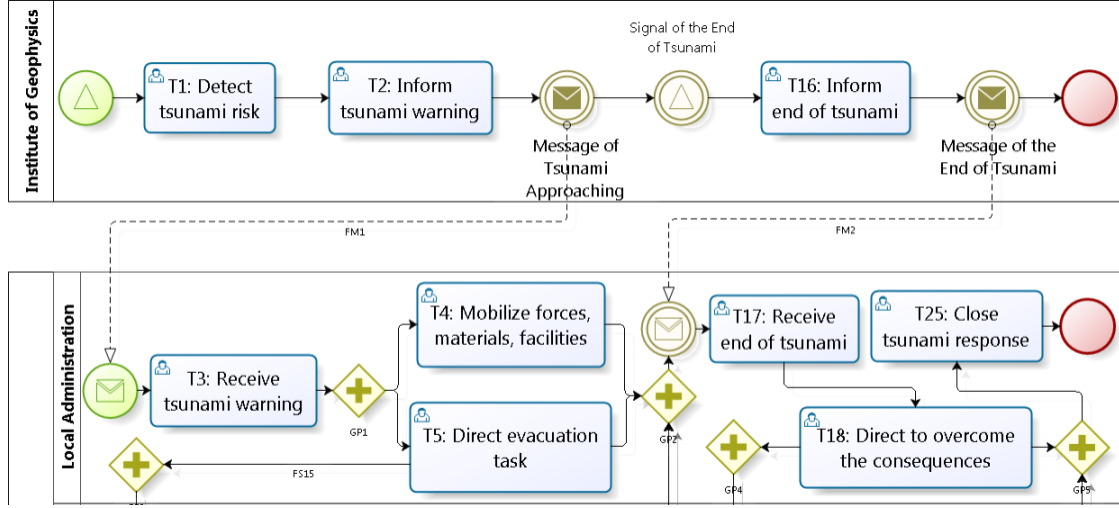
Figure 6.8: Relation between the pool *IG* and the lane *LA*

Table 6.1 – Continued from previous page

	BPMN	AGR	Applied to HCMC Plan
Sequence	$F^S(8)$	$\{End\}$	End Event
Flow	

6.5.2 Extracting the Role Graph Model from the Process Model

Based on the three proposed patterns, we analyse the BPMN diagram to build the corresponding role graph. Seven roles are detected: *IG* for the pool *Institute of Geophysics*, *LA* for the lane *Local Administration*, *LCDF* for the lane *Local Civil Defense Forces*, *CU* for the lane *Communication Unit*, *M* for the lane *Military*, *P* for the lane *Police* and *HR* for the lane *Health & Red Cross*.

The pool *IG* sends two message flows to the lane *LA* and there is no flow in opposite direction (Figure 6.8). Therefore, we create a *Power* relation from *IG* to *LA*.

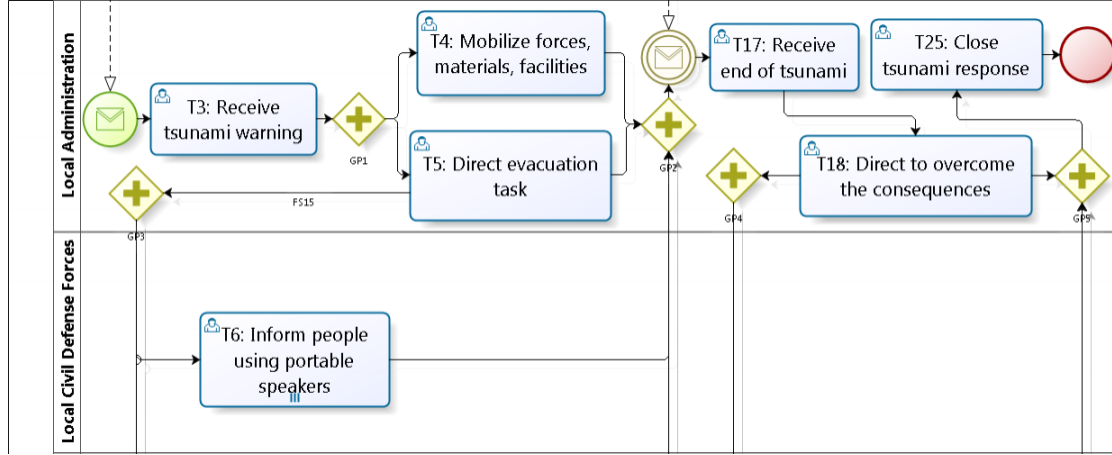
The lane *LA* has bidirectional sequence flows for all tasks with the lanes *LCDF* (Figure 6.9), *CU*, *M*, *P* and *HR*. Therefore, we create *Control* relations between them.

Algorithm 5 Mapping the business process to the BDI Agent model

Input: *BPMNmodel***Output:** *BDI – Agent*

```

    % {Step 1: Identify agents corresponding to pools}
1: for all pool  $O^P[i]$  do
2:    $\Delta[i] \leftarrow (O^P[i].name, P\{\}, G\{\}, I\{\}, B\{\})$ 
    % {Step 2: Initiate plan of agents}
3:    $\Delta[i].P \leftarrow (O^P[i].processID, In\{\}, Out\{\}, Script\{\})$ 
    % {Step 3: Complete input list of plan with start events}
4:   for all  $O_S^E[k] \in O^P[i]$  do
5:      $\Delta[i].P[x].In \leftarrow (O_S^E[k].name, O_S^E[k].type)$ 
    % {Step 4: Complete output list of plan with end events}
6:   for all  $O_E^E[k] \in O^P[i]$  do
7:      $\Delta[i].P[x].Out \leftarrow (O_E^E[k].name, O_E^E[k].type)$ 
    % {Step 5: Transfer embedded sub-process activities to another plans}
    % {Step 6: Transfer independent sub-process activities to goals}
8:   for all  $O_{Sub}^A[k] \in O^P[i]$  do
9:     if  $O_{Sub}^A[k].type = Embedded$  then
10:       $\Delta[i].P \leftarrow (O_{Sub}^A[k].name, In\{\}, Out\{\}, Script\{\})$ 
11:       $\Delta[i].P[x].Script \leftarrow \{invoke(O_{Sub}^A[k].name)\}$ 
12:     else if  $O_{Sub}^A[k].type = Independent$  then
13:       $\Delta[i].G \leftarrow (O_{Sub}^A[k].name, In\{\}, Out\{\})$ 
14:       $\Delta[i].P[x].Script \leftarrow \{addGoal(O_{Sub}^A[k].name)\}$ 
    % {Step 7: Transfer elements having send or receive message to plan's script}
15:   for all  $O^A|O^E[k] \in O^P[i]$  do
16:      $M[j] \leftarrow ("msg - j", O^A|O^E[k].src, O^A|O^E[k].tar, O^A|O^E[k].content)$ 
17:     if  $O^A|O^E[k].type = SendMessage$  then
18:        $\Delta[i].P[y].Script \leftarrow \{send(M[j])\}$ 
19:     if  $O^A|O^E[k].type = ReceiveMessage$  then
20:        $\Delta[i].P[y].Script \leftarrow \{receive(M[j])\}$ 
    % {Step 8: Transfer data flow of pool to the belief of agent}
21:    $\Delta[i].B \leftarrow \{O^P[i].id, O^P[i].req, O^P[i].ans\}$ 
    % {Step 9: Consider control flow (gateways) to orchestrate agents' plan}
22:   for all  $O^G[j] \in O^P[i]$  do
23:     Orchestrate agents' plan with AND, OR, XOR structure
  
```

Figure 6.9: Relation between the lane *LA* and lane *LCDF*

The resulting role graph, based on the three patterns, is shown in Figure 6.10, where each circle corresponds to a role. Arcs correspond to relationships: i) red arcs (labelled *p*) designate *Power* relations; ii) green arcs (labelled *c*) designate *Coordination*; and iii) blue arcs (labelled *t*) are *Control* relations. *IG* has a *Power* relation with *LA*, while *LA* has *Control* relations with *P*, *M*, *HR*, *CU*, and *LCDF*. Compared to the role graph built by stakeholders (see Figure 6.3), this version is simpler and identifies no *Coordination* relations.

6.5.3 Extracting the BDI Agent Model from the Process Model

Table 6.2 shows the result of executing the transformation algorithm described above on the HCMC tsunami response plan scenario. Some information, such as agent roles and groups, must be provided by the user. The transformation results in different BDI Agents and attributes (plans and beliefs). These two types of agents correspond to the two pools in the original scenario.

Table 6.2: Mapping the business process to the BDI Agent model for the HCMC tsunami response plan

	BPMN Diagram	BDI Agent
Step 1	$O^P(1)$ $O^P(2)$	$\Delta(1) = (O^P(1).name, P\{, G\{, I\{, B\{)$ $\Delta(2) = (O^P(2).name, P\{, G\{, I\{, B\{)$
Step 2	$Pr(1) = O^P(1).process$ $Pr(2) = O^P(2).process$	$\Delta(1).P = (Pr(1).id, In\{, Out\{, Script\{)$ $\Delta(2).P = (Pr(2).id, In\{, Out\{, Script\{)$
Step 3	Start Event \Downarrow $P.In$	$\Delta(1).P.In+ = (O_S^E(1).name, O_S^E(1).type)$ $\Delta(2).P.In+ = (O_S^E(2).name, O_S^E(2).type)$
Step 4	End Event \Downarrow $P.Out$	$\Delta(1).P.Out+ = (O_E^E(1).name, O_E^E(1).type)$ $\Delta(2).P.Out+ = (O_E^E(2).name, O_E^E(2).type)$
Step 5	Embedded Activity $O_{Sub}^A \rightarrow$ $P.Script\ invoke$	
Step 6	Independent Activity $O_{Sub}^A \rightarrow$ $P.Script\ addGoal$	
Step 7	O_{At}^A send message $\rightarrow P.Script\ send$ $O_{I,M}^E(1)$ $O_{I,M}^E(2)$ $O_{S,M}^E(1)$ $O_{I,M}^E(3)$	$M(1) = ("msg_1", O^P(1), O^L(1), [msg_reg1])$ $\Delta(1).P.Script \leftarrow \{send(M(1))\}$ $M(2) = ("msg_2", O^P(1), O^L(1), [msg_reg2])$ $\Delta(1).P.Script \leftarrow \{send(M(2))\}$ $\Delta(2).P.Script \leftarrow \{receive(M(1))\}$ $\Delta(2).P.Script \leftarrow \{receive(M(2))\}$
Step 8	O properties $\rightarrow B$	
Data Flow	$O^P(1)$ properties $O^P(2)$ properties	$\Delta(1).B \leftarrow \{O^P(1).reg, O^P(1).ans\}$ $\Delta(2).B \leftarrow \{O^P(2).reg, O^P(2).ans\}$
Step 9	Control Flow	
Element	Properties	Assignment
$O^P(1)$	reg: String, ans: String	-
$O_{S,M}^E(1)$	msg_reg: String	-
$O_{S,M}^E(2)$	msg_reg: String	-
$O^P(2)$	reg: String, ans: String	-
$O_{S,M}^E(1)$	msg_ans: String	-
$O_{S,M}^E(3)$	msg_ans: String	-

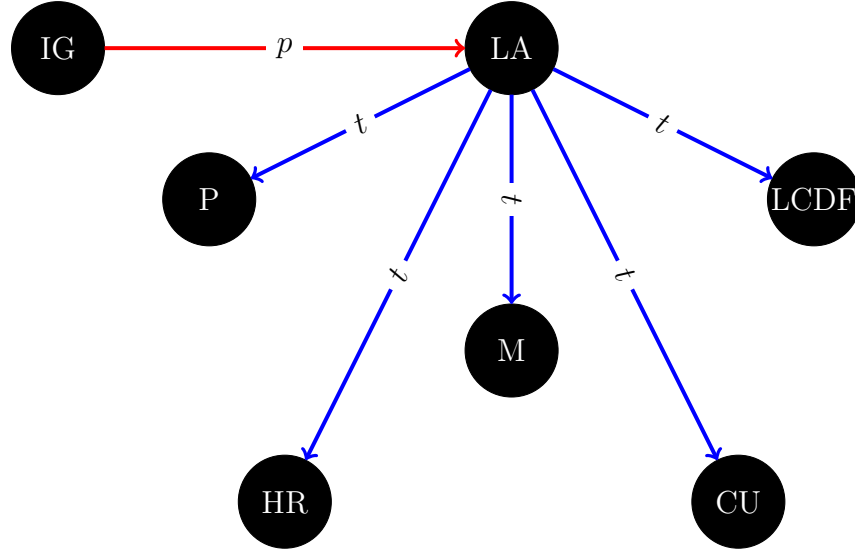


Figure 6.10: Role graph derived from the BPMN diagram for the HCMC tsunami response plan

6.6 Assessment of the HCMC Tsunami Response Plan

Following Grossi et al. [Grossi 2007], three steps are needed to assess the organizational structure: i) a role graph is built based on the three dimensions (*Power*, *Coordination* and *Control*); ii) the organizational metrics proposed by [Grossi 2007] are calculated; and iii) results are compared with standard values to assess the *Robustness*, *Flexibility* and *Efficiency* of the organization.

We implemented these three steps and visualized the results with the help of A4BP⁴ [Peralta 2015], a platform that aims to assess business processes (described in Chapter 5).

The implemented metrics include *isolation* (*completeness*, *connectedness*, *economy*, *unilaterality*, *univocity*, *flatness*) and *interaction* (*detour*, *overlap*, *incover*, *outcover* and *chain*) metrics [Grossi 2007].

For each characteristic (*Robustness*, *Flexibility* or *Efficiency*), our results are compared with the optimum values proposed by Grossi et al., in order to evaluate the organizational structure.

⁴Assessment for Business Process Platform <http://www.a4bp.com/> or <https://ci.inria.fr/pharo-contribution/job/A4BP/>

Table 6.3: Robustness (on the right) compared to standard values (on the left)

$Completeness_{Coord}$	1	25/42	$Overlap_{Coord-Pow}$	1	1
$Connectedness_{Coord}$	1	1	$Chain_{Contr-Pow}$	1	1
$Univocity_{Pow}$	0	1	$Chain_{Contr-Coord}$	1	0
$Unilaterality_{Coord}$	0	1/25	$InCover_{Contr-Coord}$	1	0
$Univocity_{Contr}$	0	1	$OutCover_{Pow-Contr}$	1	2/5
$Flatness_{Contr}$	0	1	$OutCover_{Pow-Coord}$	1	0

Table 6.4: Flexibility (on the right) compared to standard values (on the left)

$Completeness_{Pow}$	0	1/3	$Completeness_{Coord}$	1	25/42
$Connectedness_{Pow}$	0	1	$Connectedness_{Coord}$	1	1
$Chain_{Contr-Pow}$	1	1	$OutCover_{Pow-Contr}$	1	2/5

Table 6.3 shows the characteristic *Robustness*. Our results find three over twelve optimum metrics: $Connectedness_{Coord}$, $Overlap_{Coord-Pow}$ and $Chain_{Contr-Pow}$. Variation is above average compared to standard values (0.54), which leads to the conclusion that the organization is not robust enough.

Table 6.4 shows how flexible the organizational structure is, and highlights two over six optimum metrics: $Chain_{contr-pow}$ and $Connectedness_{coord}$. Here, variation is below average (0.33), thus the organization is sufficiently flexible.

Table 6.5 shows the Efficiency of the organizational structure, indicating seven out of ten optimum metrics: $Completeness_{Pow}$, $Economy_{Pow}$, $Overlap_{Coord-Pow}$, $Unilaterality_{Pow}$, $Univocity_{Pow}$, $Economy_{Contr}$ and $Overlap_{Contr-Pow}$. As the variation compared to standard values is small (0.193), the organization is fairly efficient.

In addition, our results and their corresponding standard values are shown using a radar chart visualization provided by the Roassal⁵ platform. Figures 6.11, 6.12 and 6.13 show *Robustness*, *Flexibility* and *Efficiency* respectively. Yellow lines represent standard values proposed by Grossi et al., while blue lines show the results for our organizational structure.

Based on this comparison, we can conclude that our organization is efficient and sufficiently flexible, but not robust enough. Flexibility and efficiency are due

⁵Roassal <http://agilevisualization.com/>

Table 6.5: Efficiency (on the right) compared to standard values (on the left)

$Completeness_{Pow}$	1	1	$Unilaterality_{Pow}$	1	1
$Economy_{Pow}$	1	1	$Univocity_{Pow}$	1	1
$Economy_{Coord}$	1	17/36	$Economy_{Contr}$	1	1
$Overlap_{Coord-Pow}$	1	1	$Overlap_{Contr-Pow}$	1	1
$Overlap_{Pow-Coord}$	1	2/25	$Overlap_{Pow-Contr}$	1	2/5

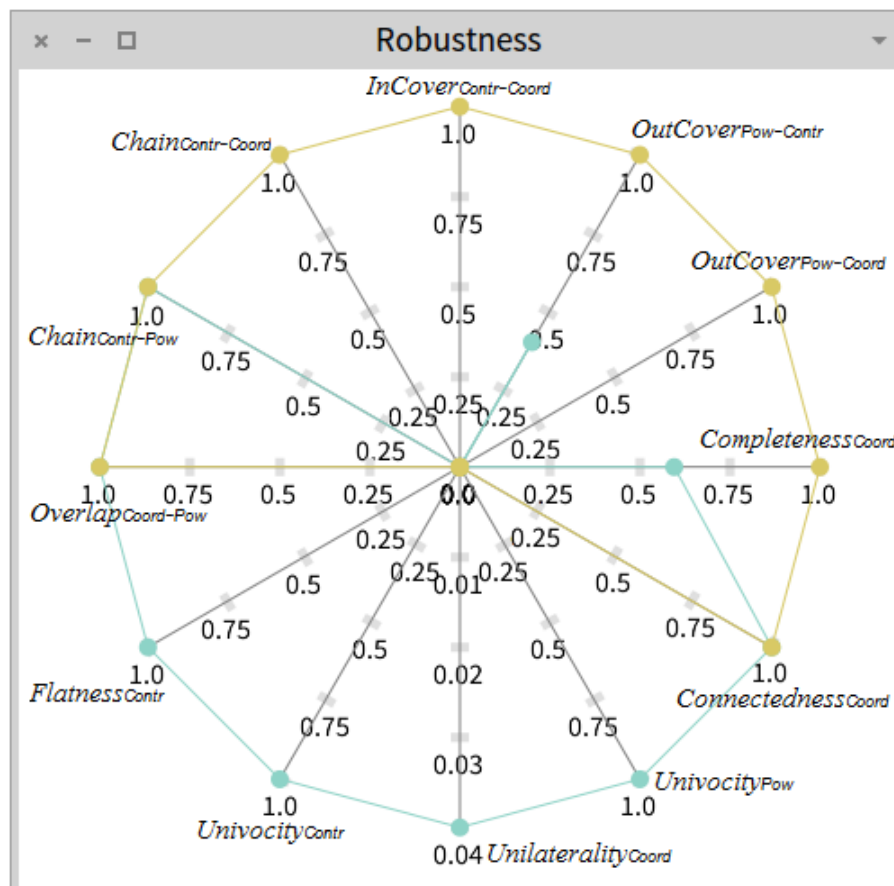


Figure 6.11: Robustness of our organization compared to standard values

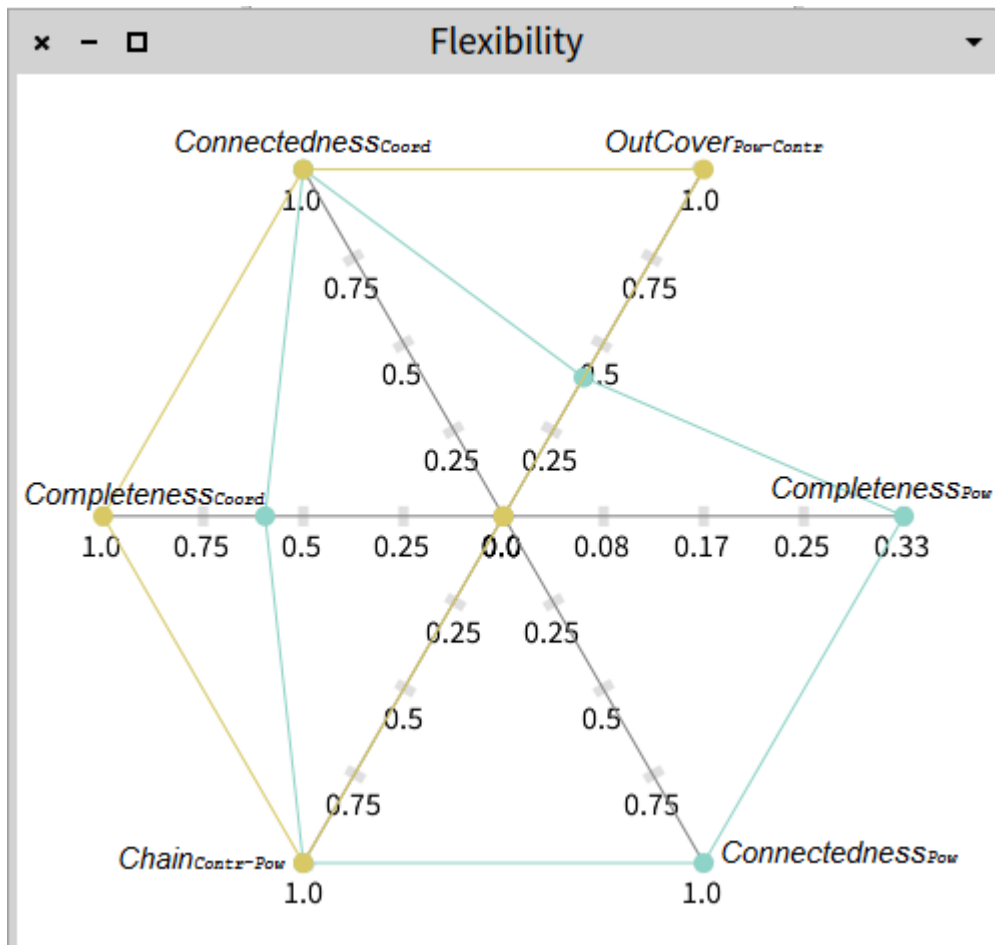


Figure 6.12: Flexibility of our organization compared to standard values

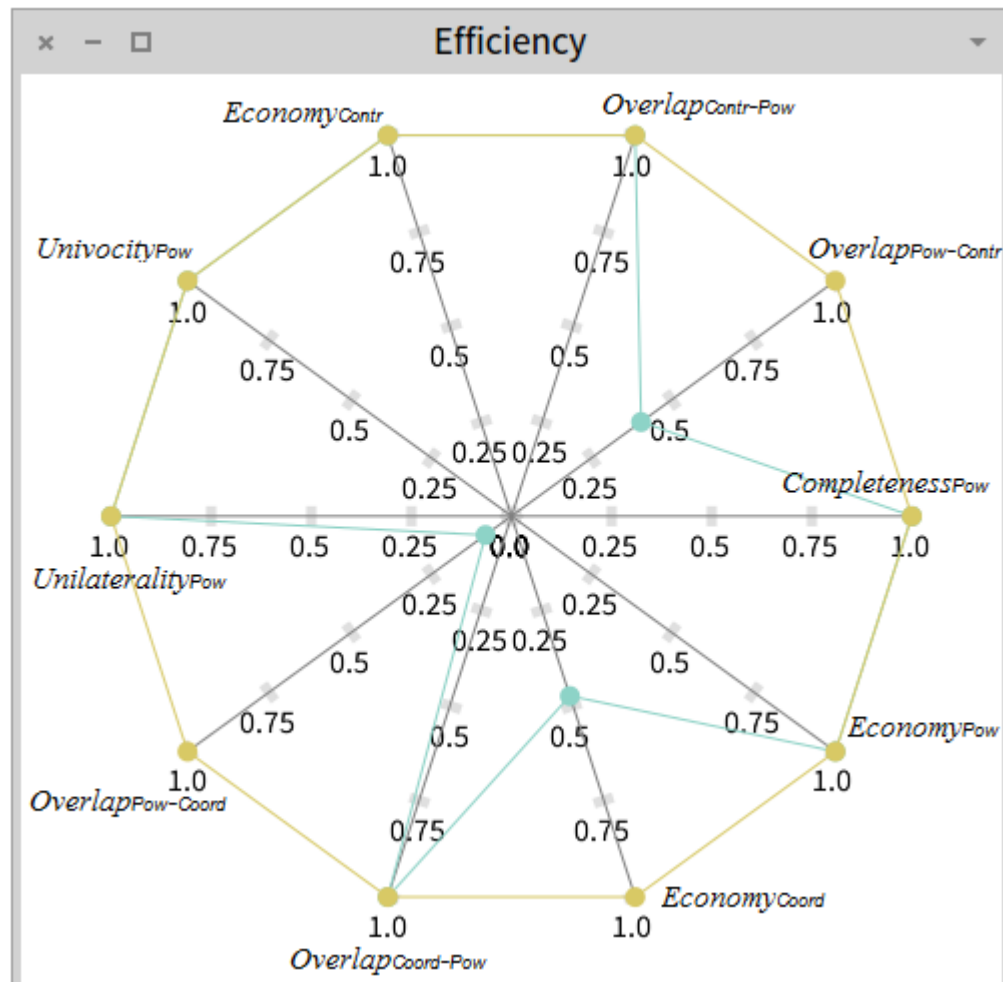


Figure 6.13: Efficiency of our organization compared to standard values

to the fact that roles are well connected, while at the same time there is a minimal number of symmetric and redundant links. Optimal robustness would require complete connectivity between all nodes. This property is useful to guarantee that the plan can continue to operate if certain resources are destroyed and a role disappears, for example. However, it is not possible to simultaneously maximize all criteria [Grossi 2007].

6.7 Conclusion

This chapter examined the organizational aspect of crisis management coordination models, which was abstracted using the Multi-Agent paradigm (Section 6.2). The abstraction took into account two complementary perspectives: one focused on a social/organizational view (OCMAS), the other on an individual view (ACMAS). The OCMAS view was abstracted using the AGR model [Ferber 2004], while the ACMAS view was represented using the BDI Agent architecture.

In order to assess the quality of an organization, we used a simplified AGR representation: the role graph, which can be derived from various sources (textual plans, BPMN diagrams, AGR, etc.). We proposed guidelines to detect roles and their relationships from textual plans. The framework proposed by Grossi et al. [Grossi 2007] to evaluate a role graph was extended to dynamic situations to evaluate concrete (rather than abstract) organizations.

Furthermore, we proposed several mapping algorithms to obtain different agent representations (BDI Agent, AGR, and role graph) from process models. These algorithms were applied to the HCMC tsunami response plan.

However, the mapping from the BPMN process model to the BDI Agent remains incomplete because:

1. It only focuses on the plans and beliefs of agents (i.e. generated agents have no goals and intentions as BDI Agents);
2. The mapping does not take into account the lane concept.

Regarding the mapping from the process model to the AGR model, the outcome is more satisfactory because AGR is compliant with BPMN. In addition, the three patterns used to derive the role graph from BPMN diagrams are relatively straightforward.

These transformation mapping algorithms highlight the lack of BPMN concepts that would enable a straightforward derivation based on the operational MAS: we argue that BPMN should be extended for this purpose.

Despite these drawbacks, we believe that the combination of process and organization views of the same plan provides authorities with a fairly complete overview of the crisis management response, although the ACMAS view is more useful for micro-simulation purposes.

The organizational structure of the HCMC tsunami response plan was evaluated and found to be efficient and sufficiently flexible, but not robust enough. This evaluation could help authorities to adapt the organization's robustness, efficiency or flexibility to the crisis context and their objectives.

Conclusion, limits and Future Works

Contents

7.1 Conclusion	150
7.2 Limitations and Perspectives for Future Research	152
7.2.1 Involving Citizens in Crisis Resolution through Socio-media	152
7.2.2 Improving the Derivation of Standalone Agents from BPMN	153

7.1 Conclusion

The objective of my PhD thesis was to contribute to coordination engineering in crisis domains by providing a comprehensive approach that takes into account both task and organizational aspects in a coherent conceptual framework. Indeed, most of the time, crisis resolution plans are available in a textual format defining the actors, their roles and coordination recommendations at the different steps of crisis life-cycle (mitigation, preparedness, response and recovery). As discussed, in this textual format, plans may be understood differently and do not provide direct means to be analyzed, simulated, adapted or improved. Therefore, they are difficult to manage in real time and in a distributed setting. On the basis of these observations, we proved the need for these textual plans to be modelled in order to have an accurate representation of them, to reduce ambiguity, support coordination between stakeholders, and ease an efficient control and crisis resolution.

In this perspective, the approach proposed in my PhD thesis combines Business Process and Multi-Agent paradigms and provides a mapping algorithm between their concepts. BPM (Business Process Modelling) provides an aggregate

view of the coordination through the task aspect and so doing eases the validation, simulation and intelligibility of crisis resolution plans at design time and its monitoring at run time. The Multi-Agent paradigm provides social abstractions (high-level interactions and organization structures) to model, analyze and simulate an organizational view of the coordination by representing the structure and the behavior of the system being developed at a macro level, whatever the internal structure of agents (micro level) is. The macro structure can then be refined to get a more realistic simulation model.

To summarize, the contribution of this thesis is a coordination framework, which consists of three main and related components, as illustrated in Figure 4.5 (Chapter 4):

- A design and development approach (design/discovery, analysis, simulation) that provides means (recommendations, formalisms, life-cycle, algorithms) to produce process-based coordination models from a textual plan (see Chapter 5). We used several language formats (BPMN, Petri Nets, ...) on top of which simulation and analysis have been carried out using existing tools (YAWL, PROM). In addition, we have developed a tool, called A4BP¹ [Peralta 2015], to measure process quality and complexity;
- A mapping algorithm, and its implementation using ATL², deriving BPMN process schemas onto multi-agent structures (see Chapter 6);
- Coordination evaluation metrics. We have extended the works of Grossi (see Chapter 6) and defined formal metrics that allow the evaluation of the quality (efficiency, robustness and flexibility) of multi-agent system organizations. We have also implemented metrics for organizational structure on top of the Pharo³ environment.

Finally, we have illustrated the use of this framework with the Ho Chi Minh City tsunami resolution plan.

In conclusion, the combined uses of MAS and BPM appear complementary and interesting to represent and validate explicit and interconnected coordination models.

¹Assessment for Business Process <http://www.a4bp.com/>

²ATL Transformation Language <https://eclipse.org/atl/>

³Pharo environment <http://pharo.org/>

7.2 Limitations and Perspectives for Future Research

Our approach and contributions face some limitations. First, we must acknowledge that our framework does not take into account, in the coordination models, external unpredictable information, knowledge and skills provided by citizens who could play a crucial and efficient role in crisis resolution (as proved in several recent crisis such as *Paris Terror Attacks* in 2015). We will discuss this point in Section 7.2.1 as a future work. Moreover, the usability and generalization of our findings require a more integrated human-computer interface distinguishing clearly the components devoted to the designer from those provided to the end-user, notably stakeholders during crisis resolution. One other limitation, from a software engineering point of view, deals with the mapping from BPMN to BDI. This transformation is obviously incomplete, as BPMN does not provide enough information to build standalone agent. We will discuss how to improve this in Section 7.2.2.

Furthermore, we must admit that this study is not based on a variety of examples. We focus on the Ho Chi Minh City tsunami resolution plan whereas other plans could have been studied to capture other specificities due to the type of crisis (e.g. humanitarian disasters) or countries (we limit ourselves to Vietnam). However, we strongly believe that our approach remains general enough to be used in other contexts. The concepts used (process, roles, agent, group, organizations, ...) remain case-independent. Only their extensions could differ from one case to another.

Moreover, the deduction of the resolution process assumes a human intervention to extract the tasks, their interdependencies and the actors. The deduction could be improved by using natural language processing in order to parse texts and fill in a crisis ontology on top of which deduction could be performed (see the work of [Viorica Epure 2015] in this direction).

7.2.1 Involving Citizens in Crisis Resolution through Socio-media

Several recent crises have shown the role of socio-media as an enabling technology to improve crisis resolution efficiency. Indeed, dedicated socio-media applications (Ushahidi, OneResponse, Tweet4Act, Google CrisisResponse, NYPA, etc.) have become a very effective means at citizens' disposal to self-organize and take part

in various crisis situations (see Section 3.4 of Chapter 3 for a review). Citizens use socio-media to make helpful information and knowledge available (events, video, expertise, ...), to accept or distribute tasks to volunteers and also to give their opinions on the way official responders manage crisis. Consequently, cooperation between official responders and citizens has become indispensable, not only for the efficiency of crisis resolution but also because the credibility, reputation and control of the crisis situation by official responders are heavily influenced by the way citizens perceive official responder reactions to the crisis. We have deployed an instance of Ushahidi to collect the crisis-related reports from citizen, call VCERS⁴. The existing of conventional tools (Facebook, Twitter, ...) and dedicated ones (mentioned above) does not guarantee the improvement of the efficiency of the crisis resolution plan. Also coordination mechanisms are required to exploit efficiently all the data generated by the socio-media and to rule the interactions between citizens and officials stakeholders. The definition of these coordination mechanisms has to deal with information trust, situation awareness, and take into account critical properties: reliability, pro-activity, security and heterogeneity. I plan to investigate these new coordination mechanisms in my future works that bring us towards social-aware coordination integrating a human and collective dimension.

From a practical point of view, we intend to build a crowdsourcing prototype enabling authorized and reliable citizens to interact and self-organize in order to perform collaborative tasks efficiently, considering citizens' skills, availability, willingness and commitments. Also, citizens' reliability should be managed and its value depending on how they have met their commitments or not. Commitments in collaboration require the introduction of deontic aspects (obligations, permissions, prohibitions, penalties, ...) and this could lead to the introduction or proposition of more sophisticated organizational and/or process models.

7.2.2 Improving the Derivation of Standalone Agents from BPMN

The automatic derivation of standalone agents from BPMN could be useful for speeding the engineering life-cycle of coordination. However, it is a hard task if we do not provide more information about the type of agents to generate and their interactions. We can imagine two means to improve the derivation process

⁴Vietnam's Crisis Early-Warning and Response System <http://vcers.byethost7.com/ushahidi-2.7.4/>

and ease the designer work. First an extension of BPMN could be considered to associate a behavioral type to each role involved in a schema. Also, BPMN should include specific multi-agent interaction protocols available as patterns at design time. The idea would be to provide the designers with such high level patterns to specify as accurately as possible the BPMN schema. On the agent side, once the multi-agent derived from BPMN, we need the definition of parameters to configure and build population of agents compliant with the previous roles and statistically compliant with real situations or the situation to be simulated. If we connect such a simulation environment to socio-media tools (as discussed in the previous section), we fall within the more general context of system-of-systems raising key issues: model interoperability, task delegation to emergent roles/organizations, their control, etc.

Bibliography

- [Ahmad 2012] Alexandre Ahmad, Olivier Balet, Jesse Himmelstein, Arjen Boin, Maaïke Schaap, Paolo Brivio, Fabio Ganovelli, Enrico Gobetti, Giovanni Pintore and Jean-Baptiste de la Rivière. *Interactive Simulation Technology for Crisis Management and Training: The INDIGO Project. ...*, April 2012. (Cited on pages [17](#), [45](#) and [199](#).)
- [Ahuja 1986] Sudhir Ahuja, N Curriero and David Gelernter. *Linda and friends*. Computer;(United States), vol. 19, no. 8, 1986. (Cited on page [26](#).)
- [Anicic 2010] Darko Anicic, Paul Fodor, Sebastian Rudolph, Roland Stühmer, Nenad Stojanovic and Rudi Studer. *A rule-based language for complex event processing and reasoning*. In Web Reasoning and Rule Systems, pages 42–57. Springer, 2010. (Cited on page [31](#).)
- [Babitski 2011] Grigori Babitski, Simon Bergweiler, Olaf Grebner, Daniel Oberle, Heiko Paulheim and Florian Probst. *SoKNOS - Using Semantic Technologies in Disaster Management Software*. In The Semantic Web: Research and Applications, pages 183–197. Springer, 2011. (Cited on page [44](#).)
- [Bénaben 2008] Frédéric Bénaben, Chihab Hanachi, Matthieu Lauras, Pierre Couget and Vincent Chapurlat. *A Metamodel and its Ontology to Guide Crisis Characterization and its Collaborative Management*. In ISCRAM 2008, editeur, Proceedings of the 5th International Conference ISCRAM, 2008. (Cited on pages [17](#), [39](#), [73](#) and [199](#).)
- [Bhattacharjee 2011] Abhijit Bhattacharjee and Roberta Lossio. *Evaluation of OCHA response to the Haiti earthquake*. Final Report January, 2011. (Cited on page [23](#).)
- [Ca 2008] Vu Thanh Ca and Nguyen Dinh Xuyen. *Tsunami risk along Vietnamese coast*. Journal of Water Resources and Environmental Engineering, pages 24–33, 2008. (Cited on pages [14](#) and [196](#).)
- [Cardoso 2006] Jorge Cardoso, Jan Mendling, Gustaf Neumann and Hajo A Reijers. *A discourse on complexity of process models*. In Business process management workshops, pages 117–128. Springer, 2006. (Cited on page [108](#).)

- [Cardoso 2008] Jorge Cardoso. *Business process control-flow complexity: Metric, evaluation, and validation*. International Journal of Web Services Research (IJWSR), vol. 5, no. 2, pages 49–76, 2008. (Cited on pages 63, 93, 108, 112 and 129.)
- [Catarci 2010] Tiziana Catarci, Massimiliano de Leoni, Andrea Marrella, Massimo Mecella, Manfred Bortenschlager and Renate Steinmann. *The WORKPAD project experience: Improving the disaster response through process management and geo collaboration*. Proceedings of the Information Systems for Crisis Response and Management (ISCRAM’10), 2010. (Cited on pages 17, 44 and 199.)
- [Catarci 2011] Tiziana Catarci, Massimiliano de Leoni, Andrea Marrella, Massimo Mecella, Alessandro Russo, Renate Steinmann and Manfred Bortenschlager. *Workpad: process management and geo-collaboration help disaster response*. International Journal of Information Systems for Crisis Response and Management (IJISCRAM), vol. 3, no. 1, pages 32–49, 2011. (Cited on pages 17, 42 and 199.)
- [Chan 2012] Jason Christopher Chan. *The Role of Social Media in Crisis Preparedness, Response and Recovery*, 2012. (Cited on page 45.)
- [Chen 2008] Rui Chen, Raj Sharman, H Raghav Rao and Shambhu J Upadhyaya. *An Exploration of Coordination in Emergency Response Management*. Communications of the ACM, vol. 51, no. 5, pages 66–73, 2008. (Cited on page 41.)
- [Cranefield 2005] Stephen Cranefield, Martin Purvis, Mariusz Nowostawski and Peter Hwang. *Ontologies for interaction protocols*. In Ontologies for agents: theory and experiences, pages 1–17. Springer, 2005. (Cited on page 32.)
- [Divitini 2001] Monica Divitini, Chihab Hanachi and Christophe Sibertin-Blanc. *Inter—Organizational Workflows for Enterprise Coordination*. In Coordination of Internet agents, pages 369–398. Springer, 2001. (Cited on pages 16 and 198.)
- [Dugdale 2010] J. Dugdale, N. Bellamine-Ben Saoud, B. Pavard and N. Pallamin. *Simulation and emergency management*, chapitre 10. Routledge, 2010. (Cited on pages 37 and 116.)

- [Dugdale 2013] Julie Dugdale. *Human behaviour modelling in complex socio-technical systems: an agent based approach*. Habilitation à diriger des recherches (HDR), Université Joseph-Fourier-Grenoble I, 2013. (Cited on pages 17 and 199.)
- [Durfee 1988] Edmund H Durfee and Victor R Lesser. *Using partial global plans to coordinate distributed problem solvers*. Readings in distributed artificial intelligence, pages 285–293, 1988. (Cited on page 33.)
- [El Fallah-Seghrouchni 2001] Amal El Fallah-Seghrouchni. *Modèles de coordination d’agents cognitifs*. Principes et architecture des systèmes multi-agent, pages 139–176, 2001. (Cited on pages 22, 26 and 27.)
- [Endert 2007] Holger Endert, Tobias Küster, Benjamin Hirsch and Sahin Albayrak. *Mapping BPMN to Agents: An Analysis*. In Matteo Baldoni, Christina Baraglio and Viviana Mascardi, editeurs, Proceedings of Int. Workshop MALLOW-AWESOME’007. Agent, Web Services, and Ontologies. Integrated Methodologies., pages 43–58, 2007. (Cited on pages 9, 61, 120, 130, 135 and 192.)
- [Engelmore 1988] Robert Engelmore and Tony Morgan. Blackboard systems: edited by robert engelmore, tony morgan. Addison Wesley Publishing Company, 1988. (Cited on page 26.)
- [Ferber 1995] Jacques Ferber and Jean-François Perrot. Les systèmes multi-agents: vers une intelligence collective. InterEditions, 1995. (Cited on pages 23, 26, 27 and 32.)
- [Ferber 1998] Jacques Ferber and Olivier Gutknecht. *A meta-model for the analysis and design of organizations in multi-agent systems*. In Multi Agent Systems, 1998. Proceedings. International Conference on, pages 128–135. IEEE, 1998. (Cited on page 30.)
- [Ferber 2004] Jacques Ferber, Olivier Gutknecht and Fabien Michel. *From agents to organizations: An organizational view of multi-agent systems*. In Agent-Oriented Software Engineering IV, pages 214–230. Springer, 2004. (Cited on pages 8, 9, 61, 62, 115, 116, 117, 118, 119, 148 and 187.)
- [Finin 1994] Tim Finin, Richard Fritzson, Don McKay and Robin McEntire. *KQML as an agent communication language*. In Proceedings of the

- third international conference on Information and knowledge management, pages 456–463. ACM, 1994. (Cited on page 26.)
- [FIPA 2002] CA FIPA. *FIPA Communicative Act Library Specification*. The Foundation for Intelligent Physical Agents, 2002. (Cited on page 26.)
- [Franke 2011a] Jörn Franke. *Coordination of Distributed Activities in Dynamic Situations. The Case of Inter organizational Crisis Management*. Ph.d. dissertation, University of Nancy, France, 2011. (Cited on pages 15, 38, 39, 42, 48 and 197.)
- [Franke 2011b] Jörn Franke, François Charoy and Cédric Ulmer. *Handling conflicts in autonomous coordination of distributed collaborative activities*. In *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, 2011 20th IEEE International Workshops, pages 319–326. IEEE, 2011. (Cited on page 43.)
- [Franke 2013] Jörn Franke, François Charoy and Paul El Khoury. *Framework for coordination of activities in dynamic situations*. *Enterprise Information Systems - Information Systems for Enterprise Integration, Interoperability and Networking: Theory and Applications*, vol. 7, no. 1, pages 33–60, February 2013. (Cited on pages 42 and 48.)
- [Gasser 2001] Les Gasser. *Organizations in multi-agent systems*. In 10th European Workshop on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW-2001). Citeseer, 2001. (Cited on page 30.)
- [Grossi 2007] Davide Grossi, Frank Dignum, Virginia Dignum, Mehdi Dastani and Lamber Royakkers. *Structural aspects of the evaluation of agent organizations*. In *Coordination, Organizations, Institutions, and Norms in Agent Systems II*, pages 3–18. Springer, 2007. (Cited on pages 8, 18, 56, 61, 63, 65, 116, 120, 121, 124, 127, 129, 143, 148, 173 and 200.)
- [Hanachi 2004] Chihab Hanachi and Christophe Sibertin-Blanc. *Protocol moderators as active middle-agents in multi-agent systems*. *Autonomous Agents and Multi-Agent Systems*, vol. 8, no. 2, pages 131–164, 2004. (Cited on page 32.)
- [Hanachi 2012] Chihab Hanachi, François Charoy and Serge Stinckwich. *CT2CM Track Report: 2nd Track on Collaborative Technology for Coordinating*

- Crisis Management*. In Proceedings of the 2012 IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE '12, pages 349–351, Washington, DC, USA, 2012. IEEE Computer Society. (Cited on page 41.)
- [Imran 2015] Muhammad Imran, Carlos Castillo, Fernando Diaz and Sarah Vieweg. *Processing Social Media Messages in Mass Emergency: A Survey*. ACM Computing Surveys (CSUR), vol. 47, no. 4, page 67, 2015. (Cited on pages 15 and 197.)
- [Jennings 1996] Nick R Jennings. *Coordination techniques for distributed artificial intelligence*. Foundations of distributed artificial intelligence, pages 187–210, 1996. (Cited on page 23.)
- [Küster 2012] Tobias Küster, Marco Lützenberger, Axel Heßler and Benjamin Hirsch. *Integrating process modelling into multi-agent system engineering*. Multiagent and Grid Systems, vol. 8, no. 1, pages 105–124, 2012. (Cited on page 129.)
- [Küster 2014] Tobias Küster, Axel Heßler and Sahin Albayrak. *Towards Process-Oriented Modelling and Creation of Multi-Agent Systems*. ..., pages 163–180, 2014. (Cited on page 129.)
- [La Rosa 2009] Marcello La Rosa and Jan Mendling. *Domain-driven process adaptation in emergency scenarios*. In Business Process Management Workshops, pages 290–297. Springer, 2009. (Cited on page 41.)
- [Le 2013] Nguyen-Tuan-Thanh Le, Chihab Hanachi, Serge Stinckwich and Tuong-Vinh Ho. *Representing, Simulating and Analysing Ho Chi Minh City Tsunami Plan by Means of Process Models*. arXiv preprint arXiv:1312.4851 for ISCRAM VN Conference, 2013. (Cited on pages 7, 28, 42, 60 and 61.)
- [Le 2015] Nguyen-Tuan-Thanh Le, Chihab Hanachi, Serge Stinckwich and Tuong-Vinh Ho. *Combining Process Simulation and Agent Organizational Structure Evaluation in order to Analyze Disaster Response Plans*. In 9th International KES Conference on Agents and Multi-Agent Systems: Technologies and Applications. Springer, 2015. (Cited on pages 42, 43, 60, 61 and 132.)

- [Lesser 2014] Victor Lesser and Daniel Corkill. *Challenges for multi-agent co-ordination theory based on empirical observations*. In Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems, pages 1157–1160. International Foundation for Autonomous Agents and Multiagent Systems, 2014. (Cited on pages 22 and 43.)
- [Ma 2012] Yu Ma, Fuzhou Duan, Hao Liu, Jinyan Tian and Guangyao Duan. *Research on the management and expression of disaster data*. In Geoinformatics (GEOINFORMATICS), 2012 20th International Conference on, pages 1–5. IEEE, 2012. (Cited on page 42.)
- [Michael 1987] Bratman Michael. Intentions, plans and practical reason. Harvard University Press, 1987. (Cited on pages 9, 119, 120 and 192.)
- [Murata 1989] Tadao Murata. *Petri nets: Properties, analysis and applications*. Proceedings of the IEEE, vol. 77, no. 4, pages 541–580, 1989. (Cited on page 71.)
- [Nguyen 2014] Van Tho Nguyen, Dominique Longin, Tuong Vinh Ho and Benoit Gaudou. *Integration of Emotion in Evacuation Simulation*. In Information Systems for Crisis Response and Management in Mediterranean Countries, pages 192–205. Springer, 2014. (Cited on pages 17 and 199.)
- [Noriega 1999] Pablo Noriega and Carles Sierra. *Auctions and multi-agent systems*. In Intelligent Information Agents, pages 153–175. Springer, 1999. (Cited on page 32.)
- [Nwana 1996] Hyacinth S Nwana, Lyndon C Lee and Nicholas R Jennings. *Co-ordination in software agent systems*. British Telecom Technical Journal, vol. 14, no. 4, pages 79–88, 1996. (Cited on page 43.)
- [Nwana 1997] Hyacinth S Nwana, L Lee and Nicholas R Jennings. *Co-ordination in multi-agent systems*. In Software Agents and Soft Computing Towards Enhancing Machine Intelligence, pages 42–58. Springer, 1997. (Cited on pages 23 and 27.)
- [Okolloh 2009] Ory Okolloh. *Ushahidi, or ‘testimony’: Web 2.0 tools for crowd-sourcing crisis information*. Participatory learning and action, vol. 59, no. 1, pages 65–70, 2009. (Cited on pages 17, 46 and 199.)

- [Omicini 2001] Andrea Omicini, Franco Zambonelli, Matthias Klusch and Robert Tolksdorf, editors. *Coordination of internet agents: Models, technologies, and applications*. Springer, 2001. (Cited on pages 23, 33 and 43.)
- [Omicini 2008] Andrea Omicini, Alessandro Ricci and Mirko Viroli. *Artifacts in the AEA meta-model for multi-agent systems*. *Autonomous agents and multi-agent systems*, vol. 17, no. 3, pages 432–456, 2008. (Cited on page 29.)
- [Papadopoulos 2001] George A Papadopoulos. *Models and technologies for the coordination of Internet agents: A survey*. In *Coordination of Internet agents*, pages 25–56. Springer, 2001. (Cited on page 26.)
- [Paulheim 2009] Heiko Paulheim, Sebastian Döweling, Karen HL Tso-Sutter, Florian Probst and Thomas Ziegert. *Improving Usability of Integrated Emergency Response Systems: The SoKNOS Approach*. In *GI Jahrestagung*, pages 1435–1349, 2009. (Cited on pages 17, 39, 44 and 199.)
- [Penserini 2009] L. Penserini, F. Dignum, V. Dignum, H. Aldewereld and D. Grossi. *Evaluating Organizational Configurations*. In *Web Intelligence and Intelligent Agent Technologies*, 2009. WI-IAT '09. IEEE/WIC/ACM International Joint Conferences, volume 2, pages 153–160, Sept 2009. (Cited on pages 8 and 124.)
- [Peralta 2015] Alvaro Jose Peralta, Le Nguyen Tuan Thanh, Chihab Hanachi, Serge Stinckwich, Ho Tuong Vinh and Alexandre Bergel. *A Tool for Assessing Quality of Rescue Plans by Combining Visualizations of Different Business Process Perspectives*. In *Second International Conference on Information Systems for Crisis Response And Management - Mediterranean Countries (ISCRAM-med)*, Tunisia. Springer, 2015. (Cited on pages 93, 108, 143, 151 and 203.)
- [Pitt 2000] Jeremy Pitt and Abe Mamdani. *Communication protocols in multi-agent systems: a development method and reference architecture*. In *Issues in agent communication*, pages 160–177. Springer, 2000. (Cited on page 32.)
- [Prizzia 2008] Ross Prizzia. *The Role of Coordination in Disaster Management*. *PUBLIC ADMINISTRATION AND PUBLIC POLICY-NEW YORK*, vol. 138, page 75, 2008. (Cited on pages 15, 41 and 197.)

- [Ramchurn 2015] Sarvapali D Ramchurn, Trung Dong Huynh, Yuki Ikuno, Jack Flann, Feng Wu, Luc Moreau, Nicholas R Jennings, Joel E Fischer, Wen-chao Jiang, Tom Rodden *et al.* *Hac-er: a disaster response system based on human-agent collectives*. In Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, pages 533–541. International Foundation for Autonomous Agents and Multiagent Systems, 2015. (Cited on pages 17, 46 and 199.)
- [Rolland 1998] Colette Rolland, C Ben Achour, Corine Cauvet, Jolita Ralyté, Alistair Sutcliffe, Neil Maiden, Matthias Jarke, Peter Haumer, Klaus Pohl, Eric Dubois *et al.* *A proposal for a scenario classification framework*. Requirements Engineering, vol. 3, no. 1, pages 23–47, 1998. (Cited on pages 7, 56, 71, 76, 78 and 100.)
- [Rozinat 2009] Anne Rozinat, RS Mans, Minseok Song and Wil MP van der Aalst. *Discovering simulation models*. Information Systems, vol. 34, no. 3, pages 305–327, 2009. (Cited on page 59.)
- [Sarcevic 2012] Aleksandra Sarcevic, Leysia Palen, Joanne White, Kate Starbird, Mossaab Bagdouri and Kenneth Anderson. *Beacons of hope in decentralized coordination: Learning from on-the-ground medical twitterers during the 2010 Haiti earthquake*. In Proceedings of the ACM 2012 conference on computer supported cooperative work, pages 47–56. ACM, 2012. (Cited on page 29.)
- [Sell 2009] Christian Sell and Iris Braun. *Using a workflow management system to manage emergency plans*. In Proceedings of the 6th International IS-CRAM Conference, volume 41, Sweden, May 2009. (Cited on pages 43 and 59.)
- [Smith 1980] Reid G Smith. *The contract net protocol: High-level communication and control in a distributed problem solver*. IEEE Transactions on computers, no. 12, pages 1104–1113, 1980. (Cited on pages 29 and 32.)
- [Smith 2011] Ben Lithgow Smith, Valentina Tamma and Michael Wooldridge. *An Ontology for Coordination*. Applied Artificial Intelligence, vol. 25, no. 3, pages 235–265, March 2011. (Cited on pages 7, 25, 43 and 65.)
- [Sycara 1999] Katia Sycara, Matthias Klusch, Seth Wido and Jianguo Lu. *Dynamic Service Matchmaking Among Agents in Open Information Environments*. SIGMOD Record, vol. 28, pages 47–53, 1999. (Cited on page 26.)

- [Turnbull 2010] J. Turnbull, D. Lea, D. Parkinson, P. Phillips, B. Francis, S. Webb, V. Bull and M. Ashby. Oxford advanced learner's dictionary, 8th edition: Paperback. Oxford Advanced Learner's Dictionary, 8th Edition. OUP Oxford, 2010. (Cited on page [39](#).)
- [Van der Aalst 1998] Wil M.P. Van der Aalst. *The application of Petri nets to workflow management*. Journal of circuits, systems, and computers, vol. 8, no. 01, pages 21–66, 1998. (Cited on page [31](#).)
- [Van der Aalst 2003a] Wil M.P. Van der Aalst, Arthur HM Ter Hofstede, Bartek Kiepuszewski and Alistair P Barros. *Workflow patterns*. Distributed and parallel databases, vol. 14, no. 1, pages 5–51, 2003. (Cited on page [30](#).)
- [Van der Aalst 2003b] Wil M.P. Van der Aalst, Arthur HM Ter Hofstede and Mathias Weske. *Business Process Management: A Survey*. In Wil M.P. Aalst and Mathias Weske, editeurs, Business Process Management, volume 2678 of *Lecture Notes in Computer Science*, pages 1–12. Springer Berlin Heidelberg, 2003. (Cited on page [42](#).)
- [Van der Aalst 2004] Wil M.P. Van der Aalst and Kees Max Van Hee. Workflow management: models, methods, and systems. MIT press, 2004. (Cited on page [30](#).)
- [Van der Aalst 2005] Wil M.P. Van der Aalst, Hajo A. Reijers and Minseok Song. *Discovering Social Networks from Event Logs*. Computer Supported Cooperative Work, vol. 14, no. 6, pages 549–593, December 2005. (Cited on page [61](#).)
- [Van der Aalst 2008] Wil M.P. Van der Aalst, J Nakatumba, A Rozinat and N Russell. *Business Process Simulation: How to get it right*. BPM Center Report BPM-08-07, BPMcenter. org, 2008. (Cited on page [96](#).)
- [Van der Aalst 2011] Wil M.P. Van der Aalst. Process mining: Discovery, conformance and enhancement of business processes. Springer, 2011. (Cited on pages [11](#), [72](#), [83](#), [86](#) and [105](#).)
- [Van der Aalst 2012] Wil M.P. Van der Aalst, Arya Adriansyah, Ana Karla Alves de Medeiros, Franco Arcieri, Thomas Baier, Tobias Blickle, Jagadeesh Chandra Bose, Peter van den Brand, Ronald Brandtjen, Joos Buijsset *al.* *Process mining manifesto*. In Business process management workshops, pages 169–194. Springer, 2012. (Cited on pages [16](#) and [198](#).)

- [Viorica Epure 2015] E. Viorica Epure, P. Martin-Rodilla, C. Hug, R. Deneckere and C. Salinesi. *Automatic process model discovery from textual methodologies*. In Research Challenges in Information Science (RCIS), 2015 IEEE 9th International Conference, pages 19–30, May 2015. (Cited on pages 73, 152 and 204.)
- [Wallace 1985] William A Wallace and Frank De Balogh. *Decision support systems for disaster management*. Public Administration Review, pages 134–146, 1985. (Cited on pages 7, 39 and 40.)
- [Wooldridge 2000] Michael Wooldridge, Nicholas R Jennings and David Kinny. *The Gaia methodology for agent-oriented analysis and design*. Autonomous Agents and multi-agent systems, vol. 3, no. 3, pages 285–312, 2000. (Cited on page 30.)

Part III

Appendix

APPENDIX A

Personal Publications

Five papers related to my PhD work have been accepted for publication in international conferences. They are described mainly in the Chapters 5 and 6. We also have an accepted article for publication in *Vietnam Journal of Computer Science (VJCS)* in August 2016, named “*Discovering Crisis Models to Help Assess Coordination Plans*” (VJCS-D-16-00068R1).

[1] - Le, N.T.T., Hanachi, C., Stinckwich, S., Ho, T.V.: *Representing, Simulating and Analyzing Ho Chi Minh City Tsunami Plan by Means of Process Models*. In: ISCRAM-VN 2013 (Information Systems for Crisis Response and Management), Vietnam (2013)

[2] - Le, N.T.T., Hanachi, C., Stinckwich, S., Ho, T.V.: *Combining Process Simulation and Agent Organizational Structure Evaluation in order to Analyze Disaster Response Plans*. In: 9th International KES Conference on Agents and Multi-Agent Systems: Technologies and Applications, Italy (2015)

[3] - Le, N.T.T., Hanachi, C., Stinckwich, S., Ho, T.V.: *Mapping BPMN Processes to Organization Centered Multi-Agent Systems to Help Assess Crisis Models*. In: 7th International Conference on Computational Collective Intelligence Technologies and Applications, Spain (2015)

[4] - Peralta, A.J., Le, N.T.T., Stinckwich, S., Hanachi, C., Ho, T.V., Bergel, A.: *A Tool for Assessing Quality of Rescue Plans by Combining Visualizations of Different Business Process Perspectives*. In: Second International Conference on Information Systems for Crisis Response And Management - Mediterranean Countries (ISCRAM-med), Tunisia (2015)

[5] - Andonoff, E., Hanachi, C., Le, N.T.T., Sibertin-Blanc, C.: *Interaction Protocols for Human-Driven Crisis Resolution Processes*. In: IFIP AICT series, Springer, 16th IFIP Working Conference on VIRTUAL ENTERPRISES, Albi, France (2015)

[6] - Le, N.T.T., Hanachi, C., Stinckwich, S., Ho, T.V.: *Discovering crisis models to help assess coordination plans*. In: Vietnam Journal of Computer Science, pp. 1–14, ISSN 2196-8896, DOI 10.1007/s40595-016-0078-9 (2016)

APPENDIX B

Glossary

B.1 Glossary

Acronyms

ACMAS Agent Centered Multi-agent System

BPM Business Process Management

BPMN Business Process Model and Notation

HCMC Ho Chi Minh City

MAS Multi-agent System

NGO Non Government Organizations

OCMAS Organization Centered Multi-agent System

UAV Unmanned aerial vehicles

Ontology for Crisis Management

C.1 Ontology for Crisis Management

We created an ontology for crisis management using Protégé framework. Figures [C.1](#) and [C.2](#) show the classes, object properties of our proposed ontology.

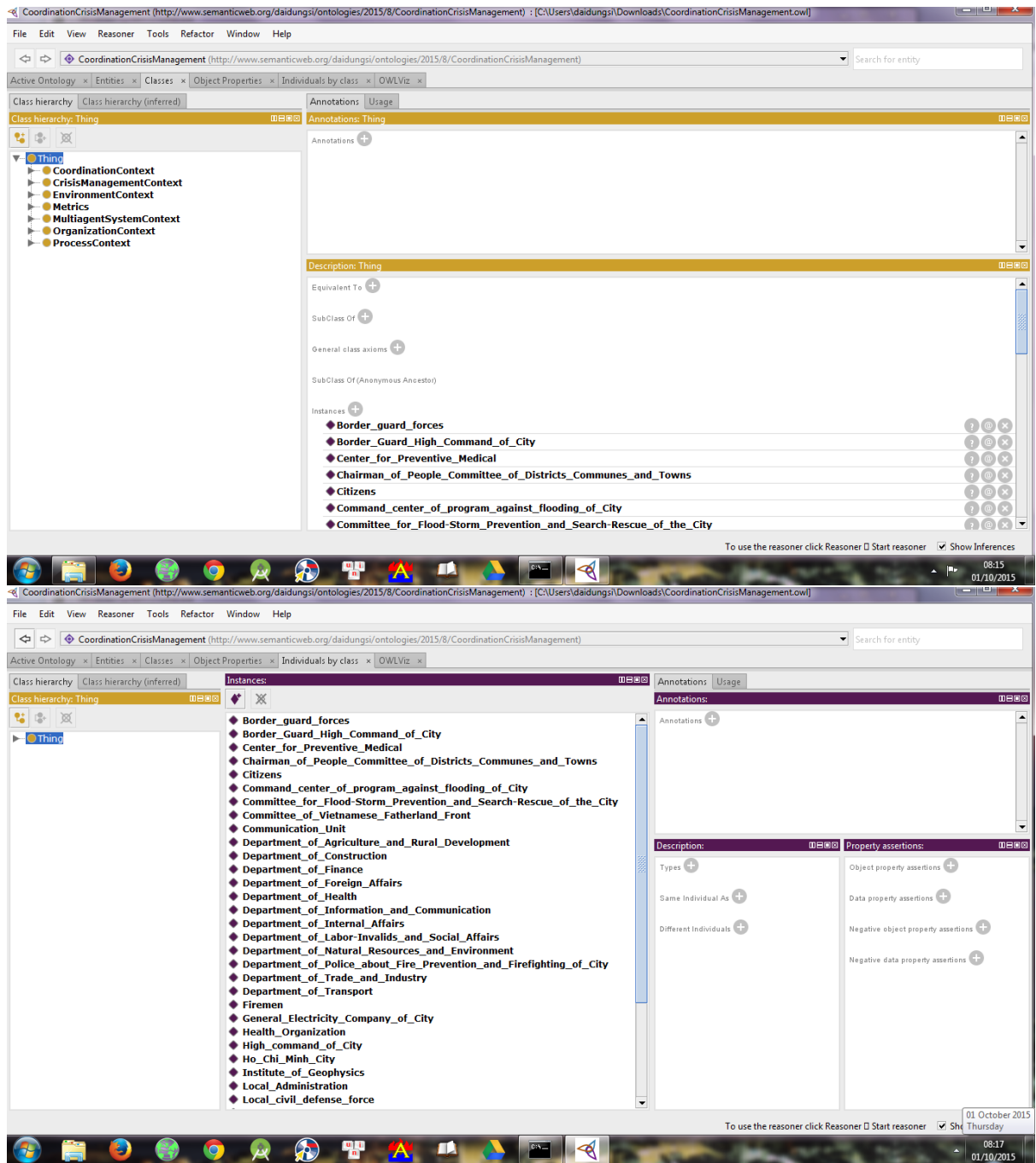


Figure C.1: Classes of our ontology for crisis management

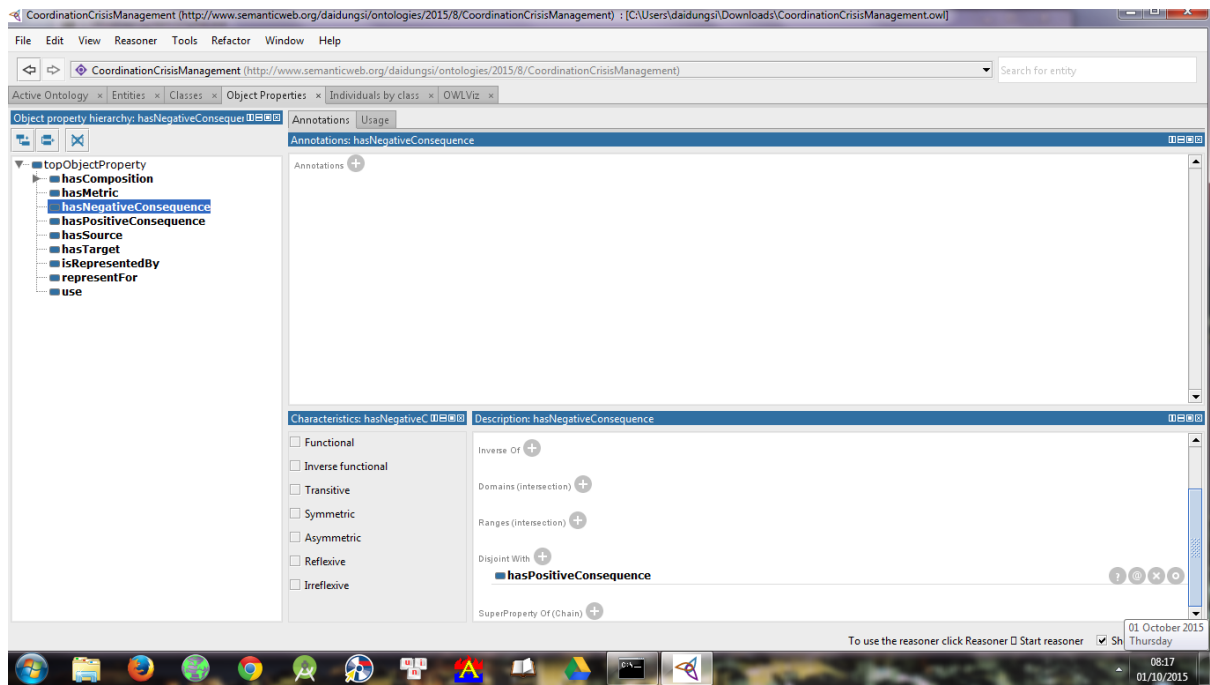


Figure C.2: Object properties of our ontology for crisis management

APPENDIX D

Source Code

D.1 Implementing the Framework for Assessing Organizational Structure based on Role Graph in Moose

D.1.1 Introduction

We have developed a Smalltalk application, named **AgentOrganizationEvaluation-Model**, implemented all metrics for assessing organizational structure proposed by [Grossi 2007]. This tool is developed on top of Moose technology.

To load the code, we can add this following repository, using *Monticello Browser*:

MCHttpRepository

location: 'http://smalltalkhub.com/mc/SergeStinckwich/AgentOrganizationEvaluation/main'

user: "

password: "

D.1.2 AgentOrganizationEvaluation Code in Moose

Our application includes two main classes: *AOERole* and *AOEOrganization*. Class *AOEOrganizationMetricsTest* is used for testing.

D.1.2.1 AOERole Class

This following script is *AOERole* class.

Listing D.1: AOERole Class

```
1 "AOERole Class"
2 MooseEntity subclass: #AOERole
```

```

3 instanceVariableNames: 'roleName roleColor coordinationRelations controlRelations
  powerRelations'
4 classVariableNames: ''
5 category: 'AgentOrganizationEvaluation-Model'

```

D.1.2.2 AOEOrganization Class

This following script is *AOEOrganization* class.

Listing D.2: AOEOrganization Class

```

1 "AOEOrganization Class"
2 MooseEntity subclass: #AOEOrganization
3 instanceVariableNames: 'roles rbnGoodComplCoord rbnGoodConnCoord rbnGoodUnivPow
  rbnGoodUnilCoord rbnGoodUnivContr rbnGoodFlatContr rbnGoodOverlapCoordPow
  rbnGoodChainContrPow rbnGoodChainContrCoord rbnGoodInCoverContrCoord
  rbnGoodOutCoverPowContr rbnGoodOutCoverPowCoord lfxGoodComplPow
  lfxGoodComplCoord lfxGoodConnPow lfxGoodConnCoord lfxGoodChainContrPow
  lfxGoodOutCoverPowContr effGoodComplPow effGoodEconPow effGoodEconCoord
  effGoodOverlapCoordPow effGoodOverlapPowCoord effGoodUnilPow effGoodUnivPow
  effGoodEconContr effGoodOverlapContrPow effGoodOverlapPowContr'
4 classVariableNames: ''
5 category: 'AgentOrganizationEvaluation-Model'

```

D.2 Implementing the Mapping from Petri nets to BPMN

We used ATL¹ to perform the mapping from Petri nets to BPMN models. The transformation code is represented as follows:

Listing D.3: Mapping from Petri nets to BPMN using ATL

```

1
2 -- @title: Petrinet2Bpmn.atl
3 --
4 -- @author: Ariouat HANANE and Nguyen-Tuan-Thanh LE, PhD Students, IRIT - Toulouse,
  France
5 --
6 --

```

¹ATL Transformation Language <https://wiki.eclipse.org/ATL>

D.2. IMPLEMENTING THE MAPPING FROM PETRI NETS TO BPMN175

```
7  -- @description: The example describes the transformation from Petri net diagrams to BPMN
   diagrams
8
9
10 module Petrinet2Bpmn;
11
12
13 -- @path Petrinet =/Petrinet2Bpmn/Metamodels/PetriNet/PNMLCoreModel.ecore
14 -- @path Bpmn2    =/Petrinet2Bpmn/Metamodels/BPMN2/BPMN20.ecore
15
16
17 create OUT: bpmn2 from IN: pnmlcoremodel;
18
19
20 -- HELPERS
21
22
23
24 --Helper to check if the Place has no arc input start event
25 helper context pnmlcoremodel!Place def: isHasNoArcInput(): Boolean =
26     if self.in.size() = 0 then
27         true
28     else
29         false
30     endif;
31
32
33 --Helper to check if the Place has no arc output end event
34 helper context pnmlcoremodel!Place def: isHasNoArcOutput(): Boolean =
35     if self.out.size() = 0 then
36         true
37     else
38         false
39     endif;
40
41
42 --Helper to check if the arc do not belongs to the sequence flow predefined
43 helper context pnmlcoremodel!Arc def: isARCdefinedBefore(): Boolean =
44     if ( self.source.ocllsKindOf(pnmlcoremodel!Transition)
45         and self.target.ocllsKindOf(pnmlcoremodel!Place)
46         and self.target.in.size() = 1
47         and self.target.out.size() = 1)
48     or (self.source.ocllsKindOf(pnmlcoremodel!Place)
49         and self.source.in.size() = 1
50         and self.source.out.size() = 1)
```

```

51         and self.target.ocllsKindOf(pnmlcoremodel!Transition)
52     ) then
53         true
54     else
55         false
56     endif;
57
58
59 -- Helper to check if the Place is an exclusive gateway input
60 helper context pnmlcoremodel!Place def: isHasExclusiveGatewayInput(): Boolean =
61     if self.in.size() >= 2 then
62         true
63     else
64         false
65     endif;
66
67
68 -- Helper to check if the Place is an exclusive gateway output
69 helper context pnmlcoremodel!Place def: isHasExclusiveGatewayOutput(): Boolean =
70     if self.out.size() >= 2 then
71         true
72     else
73         false
74     endif;
75
76
77
78 -- RULES --
79 rule PetriNet2Participant {
80     from
81         Source: pnmlcoremodel!PetriNet
82     to
83         Target: bpmn2!Participant (
84             id <- Source.id,
85             name <- Source.name
86         )
87 }
88
89
90 rule Page2Lane {
91     from
92         Source: pnmlcoremodel!Page
93     to
94         Target: bpmn2!Participant (
95             id <- Source.id,

```

D.2. IMPLEMENTING THE MAPPING FROM PETRI NETS TO BPMN177

```

96     name <- Source.name
97   )
98 }
99
100
101 rule Arc2SequenceFlow2 {
102   from
103     Arc2: pnmlcoremodel!Arc (
104       not Arc2.isARCdefinedBefore()
105     )
106   to
107     SeqFlow: bpmn2!SequenceFlow (
108       id <- 'SF-' + Arc2.id + '-5',
109       targetRef <- Arc2.target,
110       sourceRef <- Arc2.source
111     )
112 }
113
114
115
116 rule Place2StartEvent {
117   from
118     PlaceAsStartEvent: pnmlcoremodel!Place (
119       PlaceAsStartEvent.isHasNoArcInput()
120       and PlaceAsStartEvent.out.size() = 1
121     )
122   to
123     StartEvent: bpmn2!StartEvent (
124       id <- 'SE-' + PlaceAsStartEvent.id + '-1'
125     )
126 }
127
128
129 -- if Start Event becomes ExclusiveGateway
130 rule PlaceStart2ExclusiveGateway {
131   from
132     PlaceAsStartEvent: pnmlcoremodel!Place (
133       PlaceAsStartEvent.isHasNoArcInput()
134       and PlaceAsStartEvent.out.size() >= 2
135     )
136   to
137     StartEvent: bpmn2!StartEvent (
138       id <- 'SE-' + PlaceAsStartEvent.id + '-2'
139     ),
140     ExcluGat: bpmn2!ExclusiveGateway(
```

```

141     id <- 'EG-' + PlaceAsStartEvent.id + '-1',
142     outgoing <- PlaceAsStartEvent.out
143   ),
144   Seq: bpmn2!SequenceFlow (
145     id <- 'SF-' + PlaceAsStartEvent.id + '-1',
146     sourceRef <- StartEvent,
147     targetRef <- ExcluGat
148   )
149 }
150
151
152 -- ARC sequence flow
153 rule Arc2SequenceFlow {
154   from
155     Arc1: pnmlcoremodel!Arc (
156       Arc1.source.ocllsKindOf(pnmlcoremodel!Transition)
157       and Arc1.target.ocllsKindOf(pnmlcoremodel!Place)
158       and Arc1.target.in.size() = 1
159       and Arc1.target.out.size() = 1
160     )
161   using{
162     arcsSet : Sequence(pnmlcoremodel!Arc ) = pnmlcoremodel!Arc.allInstances()->select(i | i.
163       source.ocllsKindOf (pnmlcoremodel!Place) and i.source = Arc1.target);
164   }
165   to
166     SeqFlow: bpmn2!SequenceFlow (
167       id <- 'SF-' + arcsSet.first().id + '-4',
168       targetRef <- arcsSet.first().target,
169       sourceRef <- Arc1.source
170     )
171   }
172
173 -- Task in
174 rule Transition2Task {
175   from
176     Tran: pnmlcoremodel!Transition (
177       Tran.in.size()= 1
178       and Tran.out.size()=1
179     )
180   to
181     target: bpmn2!Task(
182       id <- 'Ta-' + Tran.id + '-1'
183     )
184 }

```

D.2. IMPLEMENTING THE MAPPING FROM PETRI NETS TO BPMN179

```
185
186
187 -- Task out
188 rule Transition2Task4 {
189   from
190     Tran: pnmlcoremodel!Transition (
191       Tran.out.size() >= 2
192     )
193   to
194     Target: bpmn2!Task (
195       id <- Tran.id+'-2'
196     ),
197     Seq: bpmn2!SequenceFlow(
198       id <- 'SF-'+Tran.id+'-2',
199       sourceRef <- Target,
200       targetRef <- ParaGat
201     ),
202     ParaGat: bpmn2!ParallelGateway(
203       id <- Tran.id+'-2',
204       outgoing <- Tran.out
205     )
206 }
207
208
209 -- Task in
210 rule Transition2Task1 {
211   from
212     Tran: pnmlcoremodel!Transition (
213       Tran.in.size() >= 2
214     )
215   to
216     ParaGat: bpmn2!ParallelGateway(
217       id <- 'PG-'+Tran.id+'-2'
218     ),
219     Seq: bpmn2!SequenceFlow(
220       id <- 'SF-'+Tran.id+'-3',
221       sourceRef <- ParaGat,
222       targetRef <- Target
223     ),
224     Target: bpmn2!Task (
225       id <- 'Ta-'+Tran.id+'-3',
226       outgoing <- Tran.out
227     )
228 }
229
```

```

230
231 -- Exclusive
232 rule Place2ExclusiveGateway {
233   from
234     GatExclu: pnmlcoremodel!Place (
235       not GatExclu.isHasNoArcInput()
236       and not GatExclu.isHasNoArcOutput()
237       and ( GatExclu.isHasExclusiveGatewayOutput()
238         or GatExclu.isHasExclusiveGatewayInput())
239     )
240   to
241     EndEvent: bpmn2!ExclusiveGateway (
242       id <- 'EG-' + GatExclu.id + '-2'
243     )
244 }
245
246
247 -- if Place has no arc output then map to End Event
248 rule Place2EndEvent {
249   from
250     PlaceAsEndEvent: pnmlcoremodel!Place (
251       PlaceAsEndEvent.isHasNoArcOutput()
252       and PlaceAsEndEvent.in.size() = 1
253     )
254   to
255     EndEvent: bpmn2!EndEvent (
256       id <- 'EE-' + PlaceAsEndEvent.id + '-1',
257       name <- PlaceAsEndEvent.name
258     )
259 }
260
261
262 -- if end Event Becomes ExclusiveGateway
263 rule Placeend2ExclusiveGateway {
264   from
265     PlaceAsEndEvent: pnmlcoremodel!Place (
266       PlaceAsEndEvent.isHasNoArcOutput()
267       and PlaceAsEndEvent.in.size() >= 2
268     )
269   to
270     EndEvent: bpmn2!EndEvent (
271       id <- 'EE-' + PlaceAsEndEvent.id + '-2'
272     ),
273
274     ExcluGat: bpmn2!ExclusiveGateway (

```

D.2. IMPLEMENTING THE MAPPING FROM PETRI NETS TO BPMN181

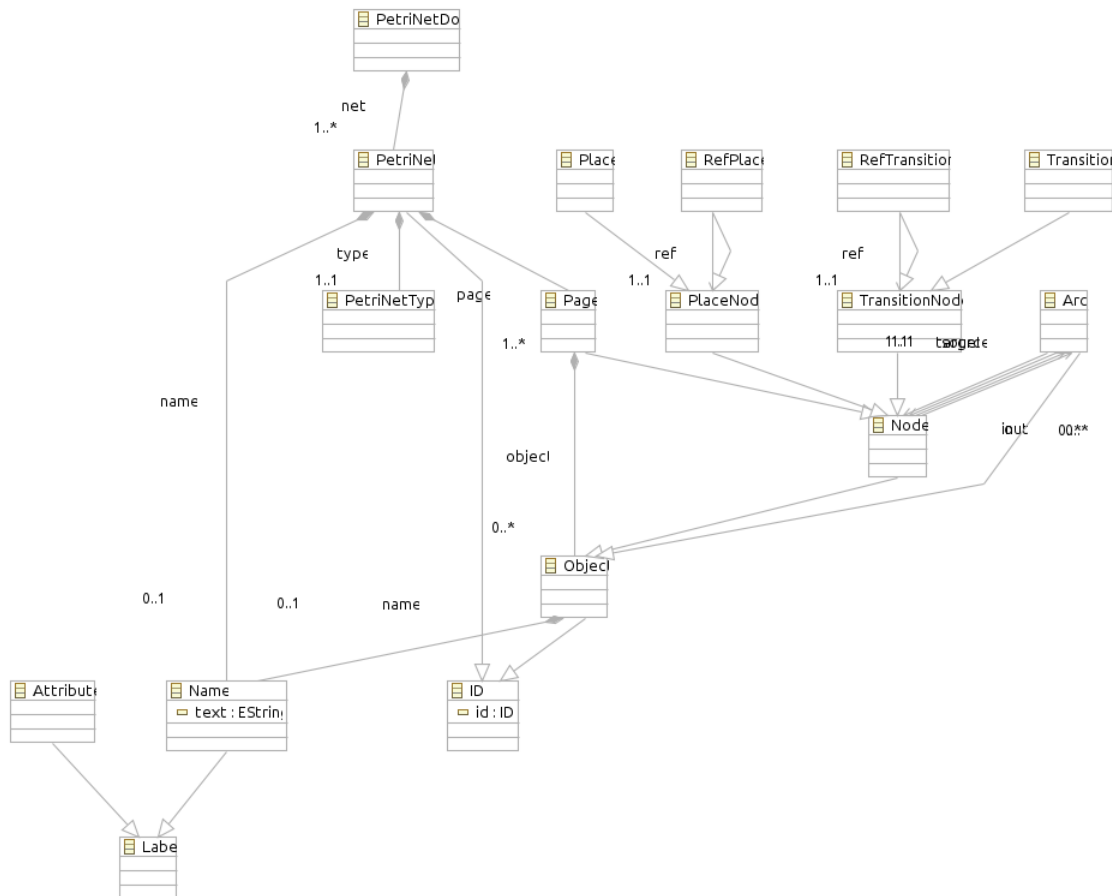


Figure D.1: The Meta-model of Petri nets

```

275   id <- 'EG-' + PlaceAsEndEvent.id + '-3'
276   ), Seq: bpmn2!SequenceFlow (
277     id <- 'SF-' + PlaceAsEndEvent.id + '-6',
278     targetRef <- EndEvent,
279     sourceRef <- ExcluGat
280   )
281 }

```

Our mapping program uses two meta-models: the source, meta-model of Petri nets (see Figure D.1) and the target, meta-model of BPMN 2.0 (see Figure D.2).

Then a Petri net model, named *PetriNet-Cas4.pnx*, conforming to the meta-model of Petri nets (Figure D.1) is used as input of our ATL transformation. The

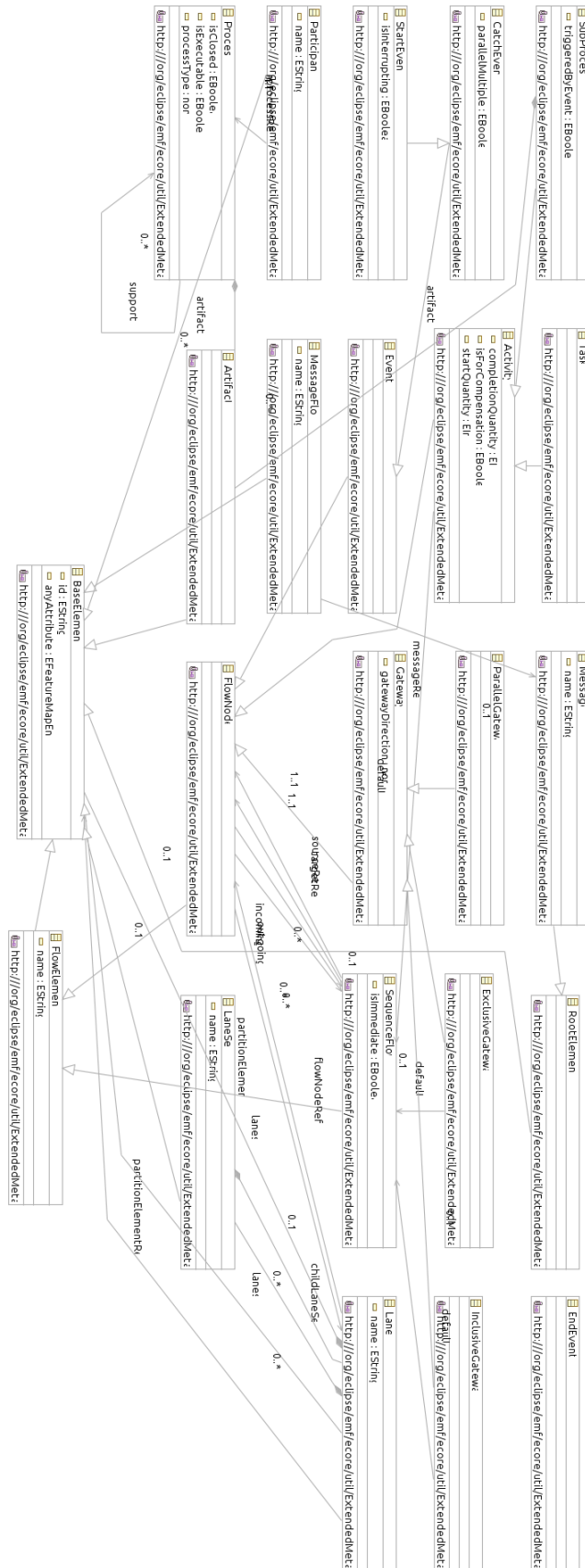


Figure D.2: The Meta-model of BPMN 2.0

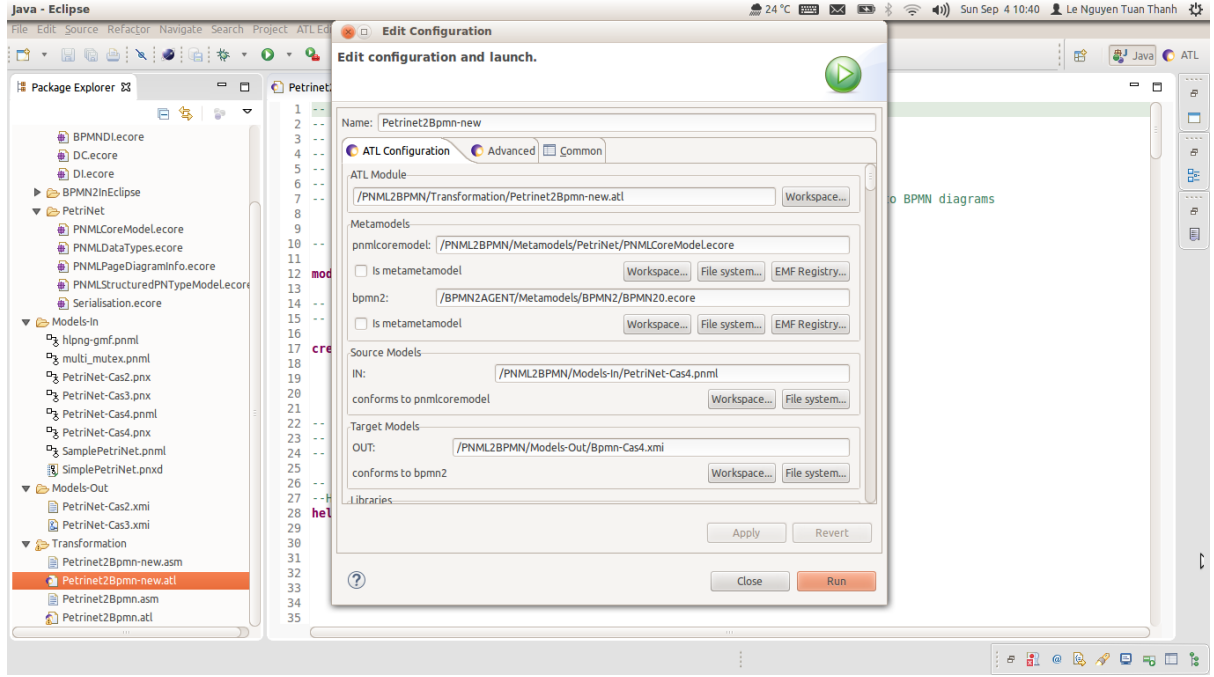


Figure D.3: The configuration of our transformation program

configuration is shown in Figure D.3.

The Petri net model contains two pages, corresponding two processes (see Figure D.4 and Figure D.5).

Finally, the result of our transformation program is depicted in Figure D.6, a BPMN model conforming to the meta-model of BPMN.

D.3 Implementing the Mapping from BPMN to AGR

We also used ATL to perform the mapping from BPMN models to AGR models. Our mapping program uses two meta-models: the source, meta-model of BPMN 2.0 (see Figure D.2) and the target, meta-model of AGR (shown in Figure D.7).

The transformation code is represented as follows:

Listing D.4: Mapping from BPMN to AGR using ATL

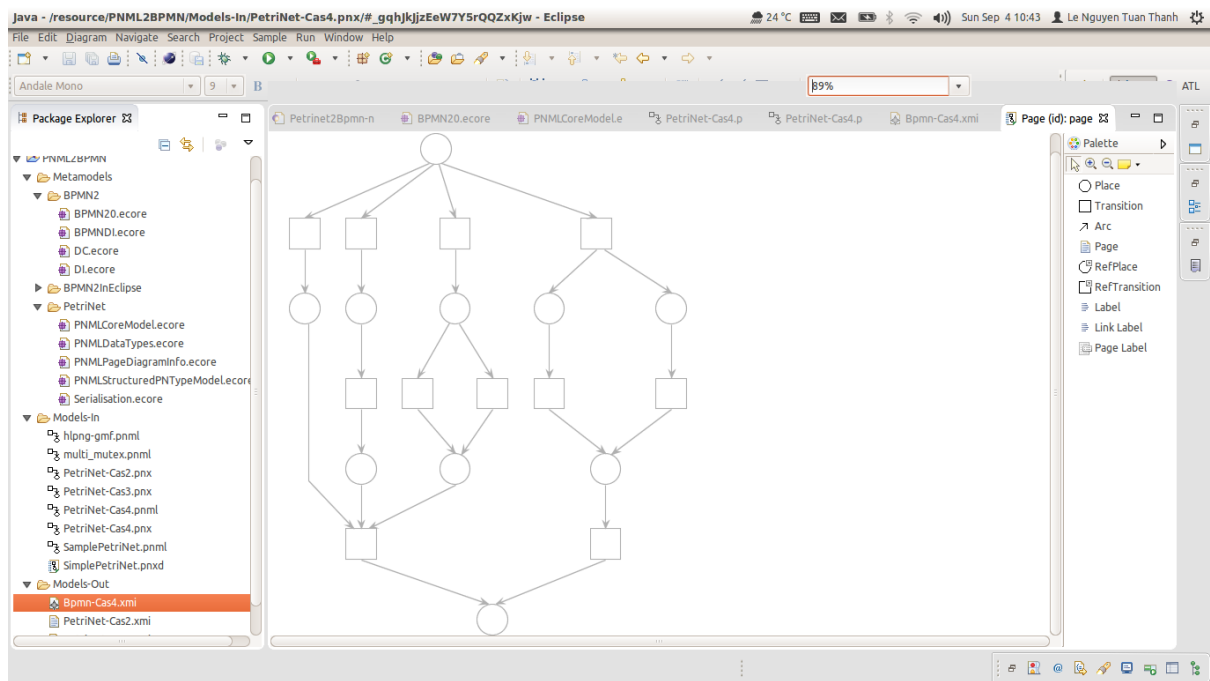


Figure D.4: The visualization of page 1 of our input Petri nets model

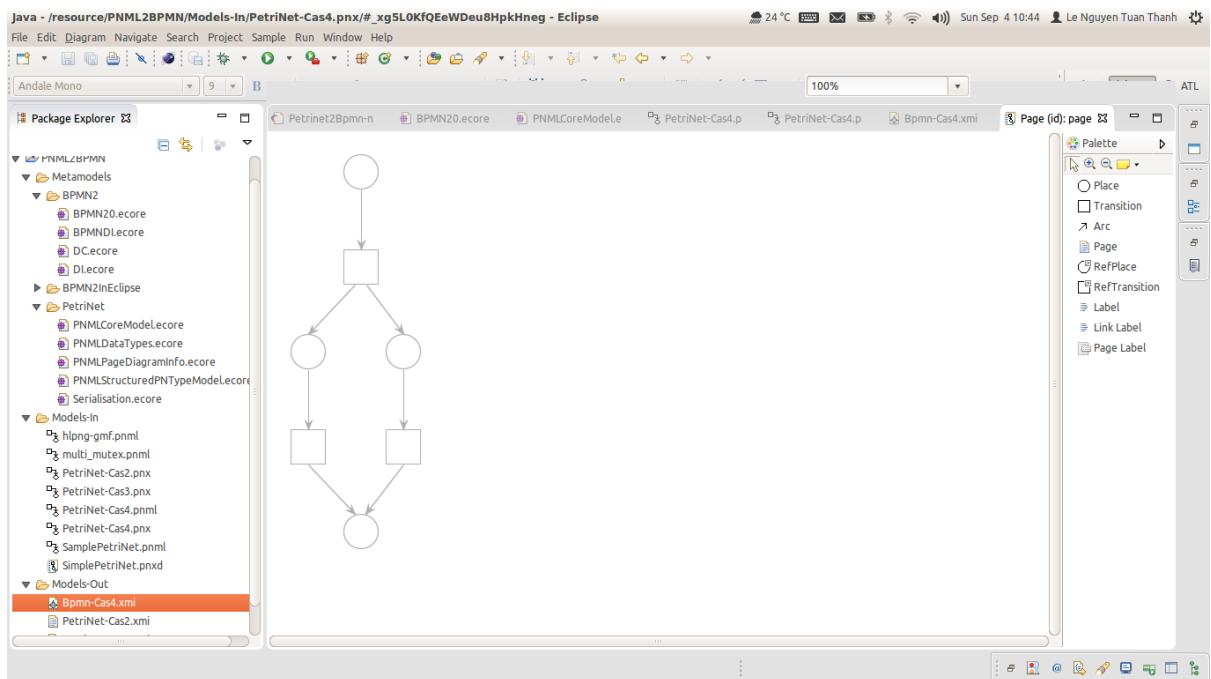


Figure D.5: The visualization of page 2 of our input Petri nets model

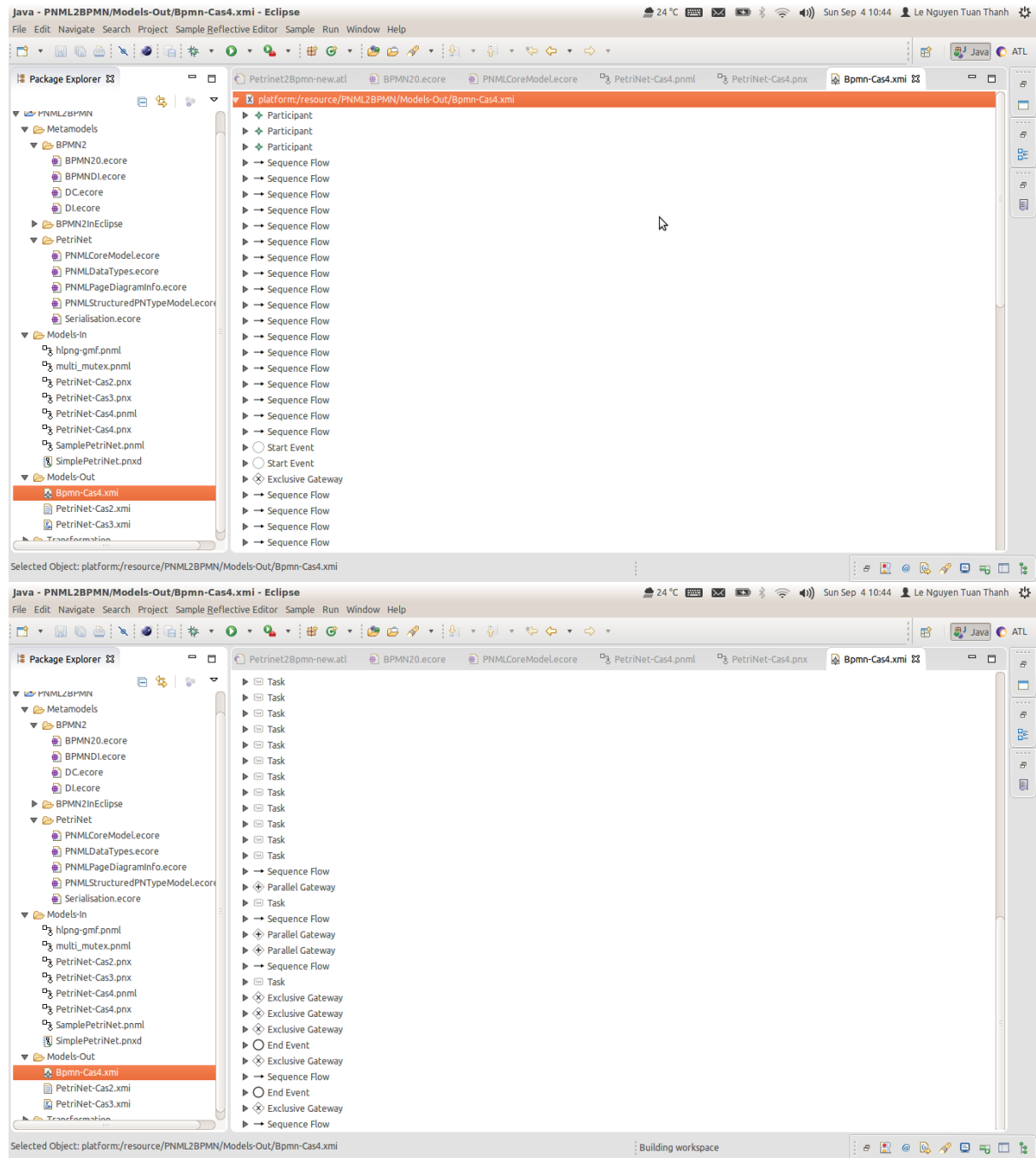


Figure D.6: The output BPMN model of our transformation

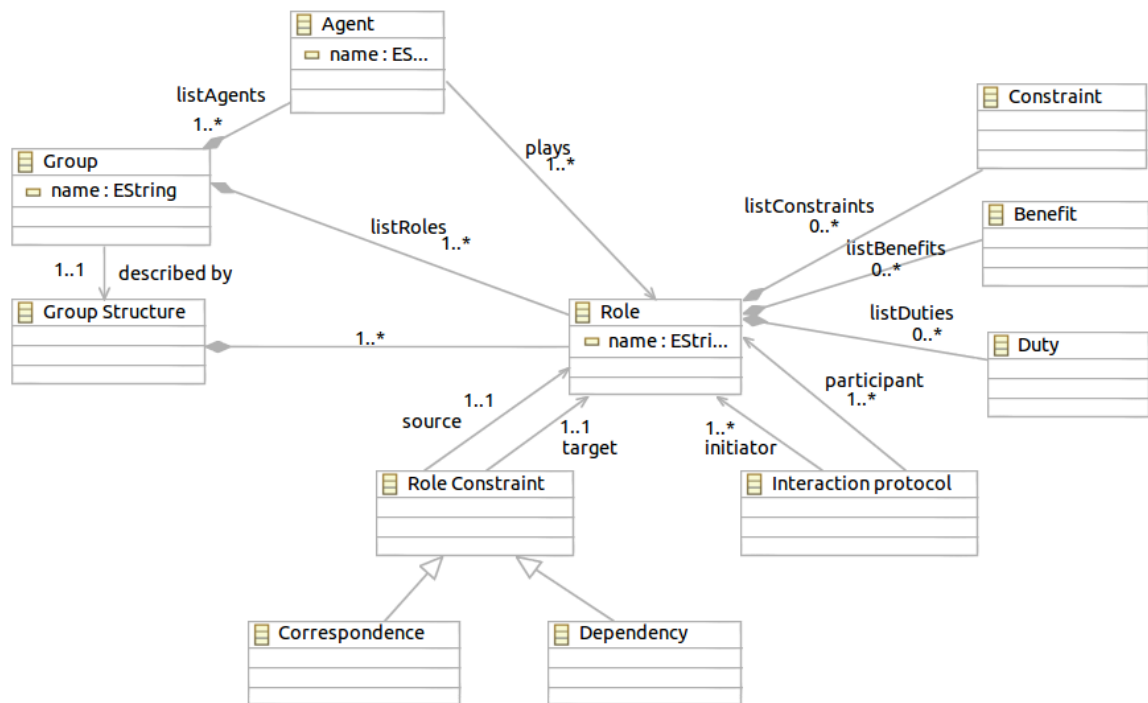


Figure D.7: The Meta-model of AGR according to [Ferber 2004]

```

2  --
3  -- @title: Bpmn2Agr.atl
4  --
5  -- @author: LE Nguyen Tuan Thanh
6  --      PhD Researcher, IRIT – Toulouse, France
7  --
8  -- @description: The program describes the transformation from BPMN diagrams to AGR
9  --      models following 5 steps
10
11
12 module Bpmn2Agr;
13
14
15 -- @path Bpmn2=/BPMN2AGR/Metamodels/BPMN2/BPMN20.ecore
16 -- @path Agr =/BPMN2AGR/Metamodels/AGR/AGR.ecore
17
18
19 create OUT: agr from IN: bpmn2;
20
21
22 --
23 -- HELPERS
24 --
25
26
27 -- Helper to check if an artifact contains information about Constraint
28 helper context bpmn2!Artifact def: isContainConstraintInfo(): Boolean =
29     if self.name.indexOf('Constraint') > -1 then
30         true
31     else
32         false
33     endif;
34
35 -- Helper to check if an artifact contains information about Benefit
36 helper context bpmn2!Artifact def: isContainBenefitInfo(): Boolean =
37     if self.name.indexOf('Benefit') > -1 then
38         true
39     else
40         false
41     endif;
42
43 -- Helper to check if an artifact contains information about Duty
44 helper context bpmn2!Artifact def: isContainDutyInfo(): Boolean =
45     if self.name.indexOf('Duty') > -1 then

```

```

46     true
47   else
48     false
49   endif;
50
51
52  --
53  -- RULES
54  --
55
56
57
58  -- Step 1: Extracting the roles and the groups from process model
59  --   Creating only one role if there is no lanes in a pool, a role for each lanes otherwise
60
61
62  rule ParticipantNoLane2Role{
63    from
64      source: bpmn2!Participant (source.participantMultiplicity.maximum = 1)
65    to
66      -- create a role for this pool/participant
67      target: agr!Role (
68        name <- source.name
69      )
70  }
71
72  rule ParticipantManyLane2Role{
73    from
74      source: bpmn2!Participant (source.participantMultiplicity.maximum = 1)
75    to
76      -- create new group
77      target: agr!Group (
78        name <- source.name
79      )
80  }
81
82  rule Lane2Role {
83    from
84      source: bpmn2!Lane
85
86    using {
87      groupSet: Sequence(agr!Group) = agr!Group.allInstances()->select(g | g.name = source.
        childLaneSet.name);
88    }
89

```

```

90  to
91      -- create a role for this lane
92      target: agr!Role (
93          name <- source.name
94      )
95      -- insert this role into group
96      do{
97          groupSet.first().listRoles.append(target);
98      }
99
100 }
101
102 -- Step 2: Obtain informations from artefact elements to identify roles's properties
103 rule Artifact2RolePropertiesConstraint{
104     from
105         source: bpmn2!Artifact (source.isContainConstraintInfo())
106     to
107         target: agr!Role(
108             listConstraints <- target.listConstraints.union(source.name)
109         )
110 }
111
112 rule Artifact2RolePropertiesBenefit{
113     from
114         source: bpmn2!Artifact (source.isContainBenefitInfo())
115     to
116         target: agr!Role(
117             listBenefits <- target.listBenefits.union(source.name)
118         )
119 }
120
121 rule Artifact2RolePropertiesDuty{
122     from
123         source: bpmn2!Artifact (source.isContainDutyInfo())
124     to
125         target: agr!Role(
126             listDuties <- target.listDuties.union(source.name)
127         )
128 }
129
130 -- Step 3: Use additional data (file) for agents. Each agent is defined by his name, type, groups
131 -- and roles
132 -- DO NOTHING IN THIS SITUATION: WE USE ONLY 2 META-MODELS, NOT
133 -- ANOTHER FILE

```



```

133
134
135 -- Step 4: Identify communication between Roles and Creating new Groups based on Message
    Flows
136 rule MessageFlow2NewGroup{
137   from
138     source: bpmn2!MessageFlow
139   using {
140     roleStart: Sequence(agr!Role) = agr!Role.allInstances()->select(r | r.name = source.
        SourceRef.name);
141     roleEnd: Sequence(agr!Role) = agr!Role.allInstances()->select(r | r.name = source.
        TargetRef.name);
142   }
143   to
144     target: agr!Group(
145       name <- source.name,
146       listRoles <- Sequence{roleStart, roleEnd}
147     )
148 }
149
150 -- Step 5: Identify activities of roles based on Sequence Flows
151 rule SequenceFlow2RoleActivity{
152   from
153     source: bpmn2!SequenceFlow
154   using {
155     lane: Sequence(bpmn2!Lane) = bpmn2!Lane.allInstances()->select(l | l.flowNodeRefs.
        includes(source));
156     role: Sequence(agr!Role) = agr!Role.allInstances()->select(r | r.name = lane.first().name);
157     taskStart: Sequence(bpmn2!Task) = bpmn2!Task.allInstances()->select(t | t.name =
        source.SourceRef.name);
158     taskEnd: Sequence(bpmn2!Task) = bpmn2!Task.allInstances()->select(t | t.name = source
        .TargetRef.name);
159   }
160   to
161
162     --target: agr!Role(
163     --)
164
165   do {
166     role.first().listDuties <- role.first().listDuties.union(taskStart.first());
167     role.first().listDuties <- role.first().listDuties.union(taskEnd.first());
168   }
169 }

```

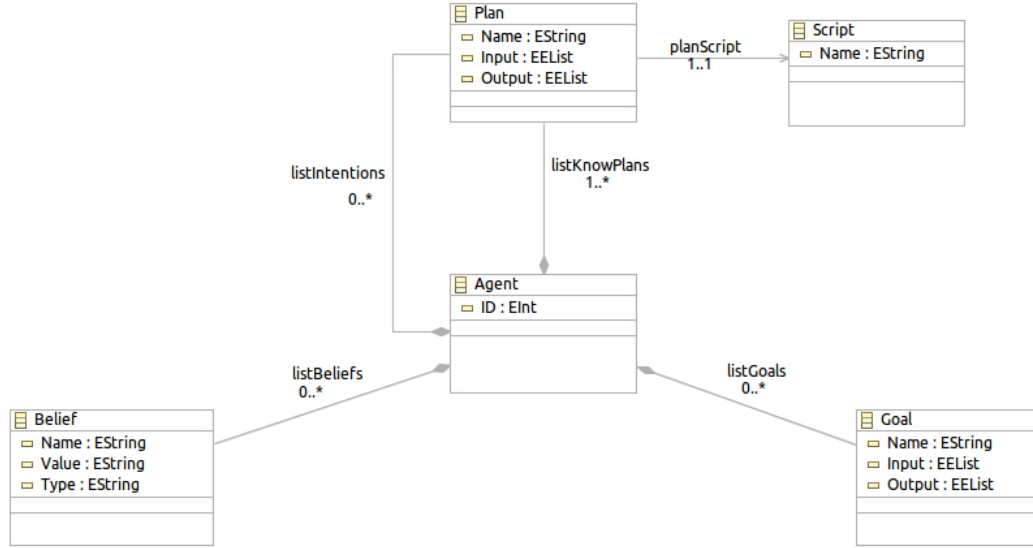


Figure D.8: The Meta-model of BDI-Agent according to [Michael 1987] and [Endert 2007]

D.4 Implementing the Mapping from BPMN to BDI-Agent

Our mapping program uses two meta-models: the source, meta-model of BPMN 2.0 (see Figure D.2) and the target, meta-model of BDI-Agent (shown in Figure D.8).

The transformation code is represented as follows:

Listing D.5: Mapping from BPMN to AGR using ATL

```

1
2  -- @title: Bpmn2Bdi.atl
3  --
4  -- @author: LE Nguyen Tuan Thanh
5  --      PhD Researcher, IRIT – Toulouse, France
6  --
7  -- @description: The program describes the transformation from BPMN diagrams to BDI-
8  --      Agent models following 9 steps
9
10 module Bpmn2Agr;
11
12
  
```

D.4. IMPLEMENTING THE MAPPING FROM BPMN TO BDI-AGENT 193

```
13 -- @path Bpmn2=/BPMN2BDI/Metamodels/BPMN2/BPMN20.ecore
14 -- @path Bdi =/BPMN2BDI/Metamodels/BDI/BDI.ecore
15
16
17 create OUT: bdi from IN: bpmn2;
18
19
20 --
21 -- HELPERS
22 --
23
24
25
26 --
27 -- RULES
28 --
29
30
31 -- Step 1: Identify agents corresponding to pools
32 rule Participant2Agent{
33   from
34     source: bpmn2!Participant
35   using{
36     startEventSet: Sequence(bpmn2!StartEvent) = source.processRef.flowElements->select(se |
37       se.ocllsKindOf(bpmn2!StartEvent));
38     endEventSet: Sequence(bpmn2!EndEvent) = source.processRef.flowElements->select(se | se
39       .ocllsKindOf(bpmn2!EndEvent));
40
41     embeddedSubProcess: Sequence(bpmn2!SubProcess) = source.processRef.flowElements->
42       select(se | se.ocllsKindOf(bpmn2!SubProcess) and se.triggeredByEvent = true);
43     independentSubProcess: Sequence(bpmn2!SubProcess) = source.processRef.flowElements->
44       select(se | se.ocllsKindOf(bpmn2!SubProcess) and se.triggeredByEvent = false);
45
46     sendTaskSet: Sequence(bpmn2!SendTask) = source.processRef.flowElements->select(ac | ac
47       .ocllsKindOf(bpmn2!SendTask));
48     receiveTaskSet: Sequence(bpmn2!ReceiveTask) = source.processRef.flowElements->select(
49       ac | ac.ocllsKindOf(bpmn2!ReceiveTask));
50
51     parallelGatewaySet: Sequence(bpmn2!ParallelGateway) = source.processRef.flowElements->
52       select(ac | ac.ocllsKindOf(bpmn2!ParallelGateway));
53   }
54   to
55     plan: bdi!Plan(
56       name <- source.processRef.name,
```

```

51      -- Step 3: Complete input list of plan with start event
52      input <- plan.input.union(startEventSet.first().name),
53
54      -- Step 4: Complete output list of plan with end event
55      output <- plan.output.union(endEventSet.first().name),
56
57      -- Step 7: Transfer elements having send- or receive- message to plan's script (activity
or event nodes)
58      planScript <- plan.planScript.union(sendTaskSet->collect(st | 'send(' + st.name + ')')),
59      planScript <- plan.planScript.union(receiveTaskSet->collect(rt | 'receive(' + rt.name +
')'))
60
61      -- Step 9: Consider control flows (gateways) to orchestrate agents' plan
62      ),
63
64      -- create a role for this pool/participant
65      target: bdi!Agent (
66          name <- source.name,
67
68          -- Step 2: Initiate plan of agents
69          listKnowPlans <- target.listKnowPlans.union(plan),
70
71          -- Step 5: Transfer embedded sub-process activities onto another plans
72          listKnowPlans <- target.listKnowPlans.union(embeddedSubProcess->collect(esp |
thisModule.EmbeddedSubProcess2Plan(esp))),
73
74          -- Step 6: Transfer independent sub-process activities onto goals
75          listGoals <- target.listGoals.union(independentSubProcess->collect(isp | thisModule.
IndependentSubProcess2Goal(isp))),
76
77          -- Step 8: Transfer data flow of pool to belief of agent, using additional data of pools
78          listBeliefs <- target.listBeliefs.union(source.documentation->collect(d | d.text))
79      )
80  }
81
82  lazy rule EmbeddedSubProcess2Plan{
83      from
84          esPro: bpmn2!SubProcess
85      to
86          script: bdi!Script(
87              name <- 'invoke(' + esPro.name + ')'
88          ),
89
90          plan: bdi!Plan(
91              name <- esPro.name,

```

```
92     planScript <- script
93   )
94 }
95
96 lazy rule IndependentSubProcess2Goal{
97   from
98     esPro: bpmn2!SubProcess
99   to
100     script: bdi!Script(
101       name <- 'addGoal(' + esPro.name + ')'
102     ),
103
104     goal: bdi!Goal(
105       name <- esPro.name
106     )
107 }
```

Résumé Long en Français

E.1 Introduction

E.1.1 Contexte

La dernière décennie a vu l'apparition d'un nombre grandissant de crises (printemps arabe, émeutes de Baltimore, ouragan de Katrina, accident nucléaire de Fukushima et plus récemment la crise des réfugiés en Europe, ...) de différentes natures (catastrophes naturelles ou industrielles, explosions de violence, ...).

Une des motivations importantes de ce travail est de contribuer à la définition d'un système d'informations qui pourraient aider à la gestion de crise et nous nous sommes intéressés plus particulièrement à l'ingénierie associée aux plans de réponse après une crise. Contrôler une crise, sauver des vies humaines, réduire les impacts et les dommages sont des challenges les plus importants que l'on puisse avoir.

En particulier dans mon pays, le Vietnam, à cause de sa position géographique en Asie du Sud-Est, peut être touché par des catastrophes naturelles comme les tsunamis ou les tremblements de terre, la menace la plus importante restant les inondations qui ont lieu chaque année principalement dans la région du centre. Le risque de tsunami ou de tremblements de terre n'est pas le plus important mais existe et il est nécessaire d'être suffisamment préparé pour les affronter [Ca 2008]. Une bonne préparation nécessitant une coordination entre les différences organisations et personnes impliquées devrait permettre d'éviter les dommages non nécessaires.

Dans de telles crises, les différents acteurs impliqués dans sa résolution doivent agir rapidement et simultanément. Afin de réaliser ce but commun de manière aussi rapide qu'efficace, les acteurs (police, forces militaires, organisations médicales) doivent joindre leurs ressources et compétences respectives afin de collaborer et de travailler de manière coordonnée, le plus souvent en suivant un plan qui spécifie le flux d'informations et de responsabilité entre eux.

E.1.2 Problème adressé

Cette thèse se focalise sur la coordination en univers multi-agents et plus particulièrement dans un contexte de crise. Par *coordination*, nous parlons du travail nécessaires pour mettre ensemble les compétences, les ressources et plans afin d'atteindre un but commun d'une manière efficace, notamment en réduisant les actions redondantes et les collisions éventuelles entre elles. L'inefficacité en terme de coordination peut produire des dommages et des pertes. Au moment, où j'écris cette thèse (Octobre 2015), la crise des réfugiés se déroule. La coordination inefficace entre les différents pays européens fait en sorte que des centaines de milliers de migrants ne peuvent pas rejoindre les zones qu'ils voudraient et ils sont contraint de stationner aux frontières avec peu de nourriture et de protection. Au USA, lorsque l'ouragan Katrina s'est déroulé en 2005, l'absence de coordination a contribué à tuer plus de 1200 personnes en plus, faire des dégâts supplémentaires pour plus de 10 milliards de dollars et ajouter plusieurs milliers de personnes sans abris [Prizzia 2008]. Même lorsque les personnes ont été sauvées, elles n'ont pas eu la protection suffisante ou bien ont manqué de nourriture [Franke 2011a].

La coordination de tel univers multi-agent posent plusieurs problèmes dus à la distribution des agents, l'hétérogénéité et l'autonomie des actions. De plus, ces univers peuvent être ouvert de nouvelles parties prenantes (e.g des organisations non-gouvernementales) peuvent se joindre et partir à tout moment. Des citoyens peuvent aussi influencer la résolution de la crise en agissant ou en propagant des informations sur les réseaux sociaux comme *Twitter* ou *Facebook* [Imran 2015]. Finalement, une crise est par nature un phénomène dynamique et les acteurs peuvent être amenés à reconsidérer leurs plans afin de s'adapter à des événements imprévus ou de nouvelle forme de crise (e.g. une catastrophe naturelle mal gérée peut ensuite provoquer une crise sociale).

La coordination est un problème complexe puisqu'il faut prendre en compte plusieurs phénomènes interdépendants (voir figure E.1) :

- L'aspect *informationnel*, qui décrit l'univers du discours sous la forme d'une représentation commune partageable par les partenaires. Il faut alors résoudre l'hétérogénéité sémantique dans l'échange d'informations qui peut apparaître entre les différents acteurs.
- L'aspect *organisationnel*, qui identifie les rôles, acteurs et groupes impliqués dans la gestion d'une crise mais également les relations (délégation, sous-contractance, coordination, contrôle) qui existe entre eux et aussi les protocoles qui régissent leurs interactions.

- L'aspect *tâche*, qui spécifie le flot de travail entre les acteurs. Le plus souvent, un plan est soumis à un processus qui définit les tâches et leurs synchronisations.

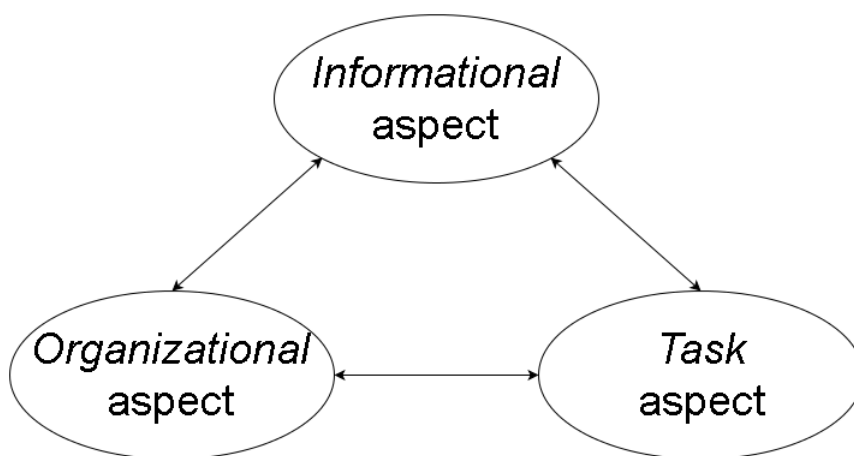


Figure E.1: Trois aspects interdépendants dus à la coordination

Dans le plus simple des cas, la coordination est explicite et des modèles concrets peuvent être fournis pour chacun des aspects. Dans certaines situations extrême, la coordination est implicite et déterminé par un but commun partagé par des parties prenantes émergentes et interagissants [Divitini 2001]. Dans ce cas, les modèles peuvent être retrouvé à posteriori en utilisant des technique d'extractions de connaissances à partir de processus (process mining) [Van der Aalst 2012] afin d'extraire les actions et les interactions des acteurs à partir de traces d'événements. Évidemmmment, il y a un continuum de scénarios de coordinations entre ces deux extrêmes.

Quelque soit le scénario de coordination afin de pouvoir traiter des univers dynamiques et complexes, les acteurs ont besoin d'outils informatiques et de systèmes d'informations afin de supporter leurs activités et faciliter la coordination. Ces technologies facilitantes permettent aux acteurs d'avoir une vision précise de l'état courant d'une crise, de connaître les actions prises dans le passé et de déterminer ce qui doit être fait dans le futur et par qui. D'un point de vue informatique, nous avons besoin de modèles et techniques pour exprimer la coordination d'une manière adaptée afin de prendre en compte l'ensemble du cycle de vie de l'ingénierie logicielle associée à la gestion de crise. En effet, il est non seulement nécessaire de spécifier et de simuler ces modèles dans la phase de prépa-

ration de la gestion de crise, mais également il faudrait supporter la coordination également dans la phase de résolution.

Plusieurs modèles et plateformes de coordination ont été construits pour la gestion de crise (e.g. WORKPAD [Catarci 2010] [Catarci 2011], SoKNOS [Paulheim 2009], INDIGO [Ahmad 2012] or USHAHIDI [Okolloh 2009], HACHER [Ramchurn 2015], ...) mais la plupart se focalise uniquement sur un aspect. L'aspect informationnel a été traité en détail en fournissant des ontologies de crises [Bénaben 2008] ou des méta-modèles, sur lesquels des artefacts partagées ont été construits (cartes, rapports, ...). L'aspect tâche a aussi été étudié avec un accent mis sur les problèmes d'allocations de tâches ou de ressources, mais la plupart des travaux existants se limite à la phase d'exécution dans le cycle de vie logiciel. Enfin en ce qui concerne l'aspect organisation, les principaux travaux suivent les approches de type simulation à base d'agents¹ [Dugdale 2013] qui se concentrent principalement sur la modélisation des interactions entre agents et sur la question de savoir comment le comportement collectif ou la prise en compte du contexte ou des émotions [Nguyen 2014] peut émerger à partir d'acteurs auto-organisés.

E.1.3 Approches suivies

L'objectif de cette thèse est de contribuer à l'ingénierie de la coordination dans le domaine de la crise en fournissant une approche intégrée qui considère à la fois les aspects organisationnels et tâches dans un cadre conceptuel cohérent. Dans cette perspective, notre approche combine les paradigmes de processus métier et multi-agents et fournisse des règles de transformations entre ces concepts.

La modélisation sous forme de processus métiers² fournit une vue agrégée de la coordination à travers l'aspect tâche et ainsi facilite i) la validation, l'analyse, la simulation et l'intelligibilité des plans de résolutions de crises (le plus souvent représenté sous une forme textuelle) au moment de leur conception, ii) la surveillance de la crise à l'exécution. Le paradigme multi-agent fournit des abstractions sociales (interactions de haut niveau et structures organisationnelles) afin de modéliser, analyser et simuler une vue organisationnelle de la coordination en représentant la structure et le comportement du système définie à un niveau macro, indépendamment de ces structures internes (micro-niveau). Une vue centrée agent d'un système multi-agents peut être également dérivée, par

¹Agent Based Social Simulation (ABSS)

²BPM (Business Process Modelling)

raffinement à partir d'une vue organisationnelle. Ceci peut être utile pour la simulation comme nous le discuterons plus tard.

E.1.4 Contributions

Les contributions de ce travail sont les suivantes :

- Une approche informelle (cycle de vie, recommandations, méta-modèles, algorithmes, ...) qui guide les concepteurs dans la modélisation d'une représentation à base de processus d'un plan de secours pour une crise. C'est une approche avec trois étapes. Nous donnons d'abord des lignes directrices pour dériver un premier processus à partir d'un document textuel décrivant les rôles et interactions entre les parties prenantes concernées. Ce document textuel considère généralement une vue idéalisée de la réalité. C'est pour cela que dans une deuxième étape, nous suggérons de définir plusieurs scénarios pour couvrir des simulations plus réalistes. Ces scénarios peuvent être dérivée du scénario idéal et différent en terme de performance ou de ressources disponibles, ... Puis ensuite une technique de fouille de processus (process mining) est utilisé pour un seul processus capable de jouer les différents scénarios incluant le scénario idéal. Ce modèle peut être ensuite simulé, analysé, déployé et transformé en système multi-agents. Nous avons suivi ces propositions afin de décrire la résolution d'un scénario de crise lors d'un tsunami dans la ville de Ho Chi Minh Ville.
- Un algorithme qui permet la transformation d'un modèle BPMN en système multi-agent.
- Une définition formelle des métriques qui permettent une évaluation de la structure organisationnelle d'un système multi-agent. Nous avons pour cela, étendu le travail de Grossi [Grossi 2007]. Alors que ce travail évalue uniquement les structures organisationnelles, notre extension évalue des organisations concrètes (abstraites), i.e celles qui sont effectivement déployées.
- Une implémentation de métriques organisationnelles et liés aux processus.

E.1.5 Vue d'ensemble de cette thèse

La thèse comprend deux parties intitulées *Etat de l'art* et *Contribution*. Elle contient 7 chapitres dont le contenu peut être résumé comme suit :

Chapitre 1: Introduction

Elle présente le contexte, le problème que nous adressons et notre approche. Puis ensuite elle résume notre contribution et donne la structure de la thèse.

Partie I: ETAT DE L'ART

Cette partie contient deux chapitres.

Chapitre 2: Coordination en univers multi-agent

Ce chapitre est consacré à l'étude de la coordination dans un univers multi-agent dans le contexte de cette thèse. Il définit précisément la notion de coordination et présente différents modèles et techniques de coordination. Des représentations formelles (ontologie, méta-modèles) sont également données.

Chapitre 3: Modèles de coordination pour la gestion de crise

Ce chapitre se consacre aux modèles de coordination utilisés pour la gestion de crise. Dans un premier temps, l'univers de la crise (terminologie, ontologie, méta-modèle, gestion du cycle de vie) est présenté. Plusieurs modèles de coordination (basé sur des processus ou des organisations ou bien des systèmes multi-agents) sont identifiés. Finalement, une comparaison des différentes plateformes de gestion de crise est fourni au dessus des ces modèles et les insuffisances de celles ci sont mis en évidence.

Partie II: CONTRIBUTION

Cette partie contient quatre chapitres.

Chapitre 4: Aperçu de l'approche

Ce chapitre donne un aperçu de l'approche qui consiste à combiner paradigme de type workflow et système multi-agent, puis dans un deuxième temps, la contribution de la thèse est mise en exergue.

Chapitre 5: Modèle de coordination à base de processus

Dans ce chapitre, nous décrivons l'aspect processus de notre approche (modèles de processus, simulation de processus, complexité et évaluation des processus). Ceci inclue des lignes directrices pour construire le modèle de processus à partir de la version textuelle. Une étude de cas concernant la ville de Ho Chi Minh Ville est donné pour illustrer notre approche.

Chapitre 6: Modèles de coordination organisationnels

Nous décrivons dans ce chapitre les aspects organisationnels de notre approche. Ceci inclue les algorithmes de transformation entre modèles à base de

processus et modèles organisationnels. Une évaluation des modèles organisationnels produits est aussi fournie.

Chapitre 7: Conclusion & Travaux futurs Ce chapitre conclue la thèse et discute des problèmes ouverts restants.

A la fin du document, on trouvera 4 annexes. La première contient la liste de nos publications. La deuxième est un glossaire des termes techniques, la troisième est a propos de l'ontologie de la gestion de gestion et enfin la dernière contient les codes des programmes développés dans notre travail.

E.2 Conclusion, limites et travaux futurs

E.2.1 Conclusion

L'objectif de ma thèse a été de contribuer à l'ingénierie de la coordination dans le domaine de la crise en fournissant une approche intégré qui prend en compte à la fois les aspects tâches et organisationnels dans un cadre conceptuel cohérent. En effet, la plupart du temps, les plans de résolutions de crises sont disponibles dans un format textuel définissant les acteurs, leurs rôles et les recommandations de coordination à différentes étapes du cycle de vie d'une crise (mitigation, préparation, réponse et rétablissement). Comme nous l'avons déjà discuté, les plans dans leurs formes textuelles peuvent être compris différemment et n'offre pas de moyens direct pour être analysé, être simulé, adapté ou amélioré.

De plus, ces plans sont difficiles à gérer en temps réel et dans un contexte distribué. Sur la base de ces observations, nous avons prouvé la nécessité pour ces plans textuels d'être modélisé de tel sorte à en avoir une représentation précise, afin de réduire les ambiguïtés, servir de support à la coordination entre les parties prenantes et faciliter le contrôle et la résolution d'une crise.

Dans cette perspective, l'approche proposé dans ma thèse est de combiner des modélisations à base de processus métier (Business Process) avec le paradigme multi-agent et de fournir des algorithmes pour passer de l'un à l'autre. BPM (Business Process Modelling) fournit une vue agrégée de la coordination par l'intermédiaire de l'aspect lié aux tâches et facilite ainsi la validation, simulation et la compréhension des plans de résolution de crises au moment de leur conception et leur monitoring à l'exécution. Le paradigme multi-agent fournit des abstractions sociales (interactions de haut niveau et structures hiérarchiques) afin de modéliser, analyser et simulation d'un point de vue organisationnel la coor-

dination en représentant la structure et le comportement d'un système à niveau macro, quelque soit la structure interne (niveau micro) des agents. La structure macro peut être alors raffinée afin d'obtenir des modèles de simulations plus réalistes.

En résumé, la contribution de cette thèse est un framework de coordination, qui consiste en trois composants comme illustré sur la figure 4.5 (Chapitre 4) :

- Une approche de conception et de développement (conception/découverte, analyse, simulation) qui fournit des moyens (recommandations, formalismes, cycle de vie, algorithmes) pour produire des modèles de coordination basé sur des processus à partir de plans textuels (voir chapitre 5). Nous avons utilisé plusieurs formalismes (BPMN, réseaux de Petri, ...) au dessus desquels des simulations et analyses ont pu être réalisées au moyen d'outils existants (YAWL, PROM). De plus, nous avons développé un outil, intitulé A4BP³ [Peralta 2015], afin de mesurer la qualité et la complexité des processus;
- Un algorithme de transformation et son implémentation en utilisant ATL⁴, permettant de dériver de processus BPMN, des structures multi-agents (voir chapitre 6);
- Des métriques pour évaluer la coordination. Nous avons étendu les travaux de Grossi (voir chapitre 6) et défini des métriques formelles afin d'évaluer la qualité (efficacité, robustesse and flexibilité) des organisations multi-agent. Nous avons également implémenté ces métriques de structures organisationnelles au dessus de Pharo⁵ environment.

Pour finir, nous avons illustré l'utilisation de ce framework en utilisant le plan de résolution d'une crise après un tsunami dans la ville de Ho Chi Minh Ville.

En conclusion, l'utilisation combinée de systèmes multi-agent et de processus apparaît complémentaire et intéressante pour représenter et valider des modèles explicites et interconnecté de coordination.

E.2.2 Limitations et perspectives pour de futurs travaux

Notre approche et nos contributions ont quelques limitations. Premièrement, nous devons consentir que notre framework ne prend pas en compte dans les

³Assessment for Business Process <http://www.a4bp.com/>

⁴ATL Transformation Language <https://eclipse.org/atl/>

⁵Pharo <http://pharo.org/>

modèles de coordination, des informations externes non prévisibles, des informations, connaissances ou compétences qui peuvent être fournis par des citoyens et qui peuvent jouer un rôle crucial et efficace dans la résolution d’une crise (comme cela a été montré sans les crises récentes comme les *attaques terroristes de Paris* de 2015). Nous discuterons de ce point dans la section E.2.2.1 comme un travail à faire dans le futur.

De plus, afin de pouvoir généraliser ce que nous proposons, il est nécessaire d’avoir une interface homme-machine plus intégrée qui permettent de distinguer clairement les composants utilisés par les concepteurs/experts, de ceux fournis par les utilisateurs finaux, notamment les parties prenantes pendant une résolution de crise. Une autre limitation, d’un point de vue génie logiciel est lié à la transformation des modèles BPMN en BDI. Cette transformation est bien entendu incomplète, car BPMN ne fournit pas suffisamment d’information pour construire des agents complets. Nous discuterons des moyens pour améliorer cette situation dans la section E.2.2.2.

Nous admettons de plus que cette étude est basée sur un nombre trop faible d’exemples. Nous nous sommes focalisé sur le plan de résolution d’un tsunami dans la ville Ho Chi Minh Ville alors que d’autres plans auraient pu être étudiés pour capturer une plus grande diversité de situation dans une situation de crises (e.g. catastrophes humanitaires) or de pays (nous nous sommes limité au Vietnam). Néanmoins nous croyons fortement que notre approche reste suffisamment général pour être utilisé dans d’autres contextes. Les concepts utilisés (processus, rôles, agents, groupes, organisations, ...) restent indépendant des cas étudiés. Seulement leurs extensions peuvent différents peuvent différer d’un cas à l’autre.

Jusqu’à présent, la déduction d’un processus de résolution de crise nécessite une intervention humaine pour extraire les tâches, leurs dépendances et les acteurs. Cette étape pourrait être amélioré grâce à des outils de traitement du langage naturel afin de traiter les textes et constituer une ontologie de la crise à partir de laquelle cette déduction pourrait être réalisée (voir le travail de [Viorica Epure 2015] dans cette direction).

E.2.2.1 Impliquer des citoyens dans la résolution de crises en utilisant des réseaux sociaux

Des crises récentes ont montré le rôle des réseaux sociaux comme technologies clés pour améliorer de manière efficace la résolution de la crise. En effet, des application de média-sociaux (Ushahidi, OneResponse, Tweet4Act, Google CrisisResponse, NYPA, etc.) sont devenus des moyens efficace mis à disposition des

citoyens pour s'auto-organiser et prendre part aux différentes situations de crises (voir section 3.4 du chapitre 3 pour une revue des travaux existants). Les citoyens utilisent les réseaux sociaux de telle sorte à rendre disponible des informations et des connaissances utiles (événements, vidéos, expertise, ...), afin d'accepter et distribuer des tâches à des volontaires et aussi pour donner leurs opinions sur la façon dont les autorités gèrent la crise. Par conséquent, la coopération entre les secouristes et les citoyens est devenue indispensable, non seulement pour avoir une résolution de crise efficace mais aussi parce que la crédibilité et la réputation des autorités officielles lors de la résolution d'une crise sont en grande partie influencées par la façon dont les citoyens perçoivent comment les autorités résolvent la crise. Nous avons déployé une instance de Ushahidi afin de collecter des rapports liés aux crises par des citoyens du Vietnam nommé VCERS⁶. Les réseaux sociaux conventionnels (Facebook, Twitter, ...) et ceux qui sont dédiés aux crises (mentionnés précédemment) ne garantissent pas une amélioration concernant la résolution d'une crise. De plus des mécanismes de coordinations sont requis pour exploiter de manière efficace toutes les données générées par ces médias et pour organiser les interactions entre citoyens et parties prenantes officielles. La définition de ces mécanismes de coordination doit prendre en compte la confiance que l'on a concernant les informations, la conscience de la situation et des propriétés critiques comme : la fiabilité, la pro-activité, la sécurité et l'hétérogénéité. Nous souhaitons étudier ces mécanismes de coordination dans des travaux futurs dans lesquels nous serons intégrés des mécanismes de coordination à ces médias sociaux suivant une dimension collective.

E.2.2.2 Améliorer la dérivation d'agents à partir de BPMN

La dérivation automatique d'agents complets à partir de BPMN peut être utile pour accélérer le cycle de vie d'ingénierie de la coordination. Néanmoins, ceci est une tâche difficile si on ne fournit pas d'informations supplémentaires concernant le type des agents à générer et leurs interactions. On peut imaginer deux façons d'améliorer le processus de dérivation et faciliter la tâche du concepteur. Premièrement une extension de BPMN peut être considérée pour associer un type comportemental à chaque rôle impliqué dans un schéma. De plus, BPMN devrait inclure des protocoles d'interactions multi-agent spécifiques disponibles comme des patrons au moment de la conception. L'idée serait alors de fournir aux concepteurs ces patrons de conception de haut niveau afin de spécifier de manière

⁶Vietnam's Crisis Early-Warning and Response System <http://vcers.byethost7.com/ushahidi-2.7.4/>

aussi précise que possible les schémas BPMN. Du côté agent, une fois qu'un modèle multi-agent a été dérivé de BPMN, nous avons besoin de la définition précises des paramètres afin de configurer et construire une population d'agents en accord avec les rôles et statistiquement en accord avec des situations réelles ou à simuler. Si nous connectons de tels environnement de simulation (comme discuté dans les sections précédentes), nous tombons dans le contexte général des systèmes de systèmes avec les problèmes suivants à résoudre: interopérabilité des systèmes, délégation des tâches à des rôles ou organisations émergentes, leurs contrôles, etc.

