



HAL
open science

System reconfiguration: A Model based approach; From an ontology to the methodology bridging engineering and operations

Lara Qasim

► **To cite this version:**

Lara Qasim. System reconfiguration : A Model based approach; From an ontology to the methodology bridging engineering and operations. Mechanical engineering [physics.class-ph]. Université Paris-Saclay, 2020. English. NNT : 2020UPAST031 . tel-03242294

HAL Id: tel-03242294

<https://theses.hal.science/tel-03242294>

Submitted on 31 May 2021

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.

System reconfiguration: A Model based approach

From an ontology to the methodology bridging engineering and operations

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 573, Approches interdisciplinaires, fondements,
applications et innovation (Interfaces)
Spécialité de doctorat : Génie Industriel
Unité de recherche : Université Paris-Saclay, CentraleSupélec, Laboratoire Génie
Industriel, 91190, Gif-sur-Yvette, France
Réfèrent : CentraleSupélec

Thèse présentée et soutenue à Gif-sur-Yvettes, le 02 décembre 2020, par

Lara QASIM

Composition du Jury

Vincent Chapurlat

Professeur, Ecole des Mines d'Alès

Président

Afreen Siddiqi

Associate professor, Massachusetts Institute of
Technology & Harvard Kennedy School

Rapporteur &
Examineur

Michel Aldanondo

Professeur, Toulouse University - Mines Albi

Rapporteur &
Examineur

Guy André Boy

Professeur, CentraleSupélec

Examineur

Didier Theilliol

Professeur, Université de Lorraine

Examineur

Marija Jankovic

Professeur, CentraleSupélec

Directrice de
thèse

Sorin Olaru

Professeur, CentraleSupélec

Co-encadrant

Andreas Makoto Hein

Chercheur, CentraleSupélec

Co-encadrant

Jean-Luc Garnier

Directeur Architecture et Ingénierie systèmes, Thales

Invité

Titre : Reconfiguration du système : Une approche basée sur les modèles ; d'une ontologie à la méthodologie faisant le lien entre l'ingénierie et les opérations

Mots clés : Reconfiguration de système, management de système, Model-based systems engineering, développement d'ontologie, génération des configurations, évaluation and sélection des configurations, opération

Résumé : Les évolutions des systèmes doivent être gérées de manière à garantir l'efficacité et l'efficacité du système tout au long de son cycle de vie, en particulier lorsqu'il s'agit de systèmes complexes qui nécessitent des années de développement et des dizaines d'années d'utilisation. La reconfiguration des systèmes est primordiale pour la gestion des systèmes complexes, car elle permet d'assurer la flexibilité et l'adaptabilité des systèmes en ce qui concerne leur évolution. La reconfiguration des systèmes assure l'efficacité opérationnelle et augmente les qualités des systèmes (par exemple, la fiabilité, la disponibilité, la sécurité, etc.).

Cette thèse a été effectuée en partenariat avec une entreprise évoluant dans les domaines de l'aérospatial, de l'espace, du transport, de la défense et de la sécurité. Les entreprises portent un intérêt croissant sur la reconfiguration des systèmes afin de garantir leurs efficacités opérationnelles. L'objectif de cette thèse est de proposer une approche basée sur les modèles pour soutenir la reconfiguration de système.

En effectuant une étude descriptive, basée sur une étude de terrain et l'analyse de l'état de l'art, le développement d'un support lié à la reconfiguration de système a été identifié comme enjeu industriel majeur.

Le défi principal consiste à identifier les données relatives à la reconfiguration des systèmes et leurs mécanismes d'intégration afin d'atteindre cet objectif.

Dans cette thèse, nous présentons une ontologie, que nous avons nommé OSysRec, qui intègre les données nécessaires pour la reconfiguration et gestion des systèmes. De plus, OSysRec agrège les trois aspects indispensables à la gestion des processus de la reconfiguration de système: la structure, la dynamique, et la gestion.

Nous présentons également une méthode basée sur les modèles (MBSysRec) qui intègre les données de reconfiguration et fait le lien entre les phases d'ingénierie et d'opération. Cette méthode est multidisciplinaire qui implique des générations combinatoires de configurations et des décisions multicritères pour leurs évaluations et sélections. Nous avons pu démontrer sur deux cas d'étude la validité de cette méthode pour trouver des solutions performantes et pertinentes.

Cette thèse est un premier étape pour la mise en œuvre d'une approche basée sur les modèles pour la reconfiguration de système permettant leur flexibilité et leur adaptabilité

Title : System reconfiguration: A Model based approach; From an ontology to the methodology bridging engineering and operations

Keywords : System reconfiguration, system management, Model-based systems engineering, ontology development, configuration generation, configuration evaluation and selection, operations.

Abstract : System evolutions have to be managed to ensure system effectiveness and efficiency through its whole lifecycle, particularly when it comes to complex systems that take years of development and dozens of years of usage. System Reconfiguration is key in complex systems management, as it is an enabler of system flexibility and adaptability regarding system evolutions. System reconfiguration ensures operational effectiveness and increases system qualities (e.g., reliability, availability, safety, and usability).

This research has been conducted in the context of a large international aerospace, space, ground transportation, defense, and security company. This research aims at supporting system reconfiguration during operations.

First, we conducted a descriptive study based on a field study and a literature review to identify the industrial challenges related to system reconfiguration.

The main issue lies in the development of reconfiguration support. More specifically, challenges related to data identification and integration were identified. In this thesis, we present the OSysRec ontology, which captures and formalizes the reconfiguration data. The ontology synthesizes the structure, dynamics, and management aspects necessary to support the system reconfiguration process in an overall manner.

Furthermore, we present a model-based method (MBSysRec) that integrates system reconfiguration data and bridges both the engineering and the operational phases. MBSysRec is a multidisciplinary method that involves combinatorial configuration generation and a multi-criteria decision-making method for configuration evaluation and selection.

This thesis is a step towards a model-based approach for system reconfiguration of evolving systems, ensuring their flexibility and adaptability

Abstract

System Reconfiguration is essential in complex systems management, as it is an enabler of system flexibility and adaptability with regard to system evolutions. System evolutions have to be managed to ensure system effectiveness and efficiency through its whole life cycle, particularly when it comes to complex systems that take years of development and dozens of years of usage. In this context, system reconfiguration ensures system operation and increases reliability, availability, maintainability, testability, safety, and reuse of system entities and technologies.

This research has been conducted in the context of a large international aerospace, space, ground transportation, defense, and security company. Large industrial companies developing products, systems, and solutions in critical domains (e.g., aerospace) are often concerned with safety and other operational effectiveness measures. Such companies are increasingly interested in system reconfiguration as it is an enabler for managing systems, ensuring their operational effectiveness. The objective of this research is to propose a model-based approach for system reconfiguration.

To identify the difficulties and challenges related to system reconfiguration, we conducted a descriptive study based on a field study and a literature review. The industrial challenges have been identified based on the analysis of the reconfiguration process throughout the different life cycle phases, particularly during operations. The development of model-based support for system reconfiguration is a major industrial challenge. More specifically, the main issues are linked to identifying data related to system reconfiguration as well as mechanisms that allow data integration.

The system reconfiguration process relies on data related to structural, dynamics, and management aspects. These aspects have been considered independently up to now. The existing body of research lacks an ontology or a data model that considers the system reconfiguration process in an overall manner. Furthermore, developing reconfiguration supports is challenging as it requires integrating data related to observations (from operations) and system design (from engineering). The existing reconfiguration methods do not integrate this data, which is essential for system reconfiguration.

In this thesis, we present the system reconfiguration ontology, which we call OSysRec. The ontology synthesizes the structure, dynamics, and management aspects necessary to support the system reconfiguration process. The presented ontology provides a comprehensive conceptual framework allowing engineers to study systematically various problems and solutions related to dynamic evolution.

Furthermore, we present a model-based method (MBSysRec) that integrates system reconfiguration data and bridges both the engineering and the operational phases. MBSysRec is a multidisciplinary method that involves combinatorial configuration generation and a multi-criteria decision-making method for configuration evaluation and selection. The method has been assessed with two case studies from the industry; the Search and Rescue (SAR) case study, and the Short-Term Conflict Alert from the Air Traffic Management case study. The method is proven effective for finding relevant system configurations and, therefore, supporting system reconfiguration. MBSysRec assists command and control operators with decision support for system reconfiguration during operation.

This thesis is a step towards a model-based approach for system reconfiguration of evolving systems, ensuring their flexibility and adaptability.

Acknowledgements

My sincere acknowledgement and gratitude go to:

The jury members who accepted to review and evaluate my work.

My supervisors for their support during this thesis.

My managers and colleagues at Thales for offering me this great opportunity and the very enriching experience.

My colleagues and friends at LGI for the great moments I shared with them.

My manager at EDF for giving me all the time I needed to prepare my thesis defense.

Andreas for being a real friend that I want to keep for the rest of my life.

Naouress with whom I shared this journey with all its ups and downs.

My friends Kafa, Laith, Radia and Tony for everything.

My French family for their love and support.

My Vincent for being “my Vincent” and for loving the unbearable version of me.

My family for their unconditional love, help and support.

My brothers Faheem, Rami and Fadi for all the happiness they bring to my life.

My sister Rula for encouraging me and giving me the opportunities and experiences that have made me who I am.

The absolute love, my mother.

The only person that is always in my mind and my heart, my father...

Contents

Abstract	I
Acknowledgements	III
Contents.....	V
List of Figures	IX
List of Tables.....	XI
List of Abbreviations.....	XIII
Foreword	XV
1. Introduction.....	1
1.1. Research Motivation	1
1.1.1. Role of System Reconfiguration in System Management	1
1.1.2. Systems reconfiguration at Thales	3
1.2. Research Objective.....	4
1.3. Thesis Outline	5
2. Literature Review	7
2.1. Understanding Reconfiguration	7
2.1.1. Design-time Related Domains	8
2.1.2. Run-time Related Domains.....	13
2.2. Systems Engineering (SE) and Model-Based Systems Engineering (MBSE).....	18
2.2.1. Systems Engineering (SE)	18
2.2.2. Model-Based Systems Engineering (MBSE).....	19
2.3. Synthesis	20
3. Research Overview and Thesis Contributions	23
3.1. Research Overview	23
3.1.1. Synthesis of main issues from the industrial audit.....	23
3.1.2. Scientific State-of-the Art and Research Gaps	24
3.1.3. Gaps between industrial practice and research	25
3.1.4. Research questions.....	26
3.2. Thesis Contributions	29
3.2.1. A Descriptive Study of the System Reconfiguration Process Over System Life Cycle Stages ...	29
3.2.2. An overall Ontology for System Reconfiguration using Model-Based System Engineering.....	31
3.2.3. A Model-based Method for System Reconfiguration	34
4. Research Methodology.....	37
4.1. The Design Research methodology.....	38
4.2. Research Quality	40

4.3. The Research Methodology used in the Present Thesis	40
5. A Descriptive Study of the System Reconfiguration Process Over System Life Cycle Stages	43
5.1. Introduction	44
5.2. Literature Review	45
5.3. Industrial Context	46
5.4. Methodology	47
5.4.1. Observations	48
5.4.2. Documentation Exploration	48
5.4.3. Experts interviewing	49
5.5. Results	51
5.5.1. The engineering (re)configuration Process	51
5.5.2. Difficulties within the in-use engineering (re)configuration process	52
5.5.3. Reconfiguration as an enabler for system management during operations	54
5.5.4. Challenges related to system reconfiguration during operations	56
5.6. Discussion and Future Work	58
5.7. Conclusion	60
6. An overall Ontology for System Reconfiguration using Model-Based System Engineering	63
6.1. Introduction	64
6.2. Literature review	66
6.2.1. Reconfiguration in system's engineering	66
6.2.2. Model-Based Systems Engineering (MBSE) and ontologies	68
6.2.3. System reconfiguration ontologies and meta-models	69
6.2.4. Synthesis and research gap	70
6.3. Methodology	73
6.3.1. Ontology specification	73
6.3.2. Ontology conceptualization, formalization, and implementation	74
6.3.3. Ontology evaluation	76
6.4. <i>OSYSREC</i> : Ontology for systems reconfiguration	77
6.4.1. <i>OSysRec</i> conceptualization	77
6.4.2. <i>OSysRec</i> description with SysML formalism	80
6.5. Scenario-based validation: Connected Battle group	84
6.5.1. Battlegroup system organization	85
6.5.2. <i>OSysRec</i> : illustration scenarios	87
6.6. Scenario-Based Validation: Integrated Modular Avionics (IMA)	92
6.7. Discussion	95
6.8. Conclusion and future work	96
7. A Model-based Method for System Reconfiguration	99

7.1. Introduction.....	100
7.2. Literature Review.....	101
7.2.1. System Architecture Methods.....	101
7.2.2. Product Configuration and Reconfiguration Methods.....	103
7.2.3. Synthesis and Research Gap.....	105
7.3. MBSYSREC Method.....	106
7.3.1. MBSysRec: Objective and Requirements.....	106
7.3.2. MBSysRec: Process Steps.....	109
7.3.3. MBSysRec: Information Sources for the Matrix System.....	115
7.4. Case Study: Search And Rescue.....	117
7.4.1. Case Study Description.....	117
7.4.2. Further Scenarios from the SAR Case Study.....	132
7.5. Industrial Case Study: Short-Term Conflict Alert in Air Traffic Managemnt System.....	134
7.5.1. Problem Definition.....	135
7.5.2. Reconfiguration Scenario from the STCA Case Study.....	137
7.6. Evaluation and Discussion.....	138
7.7. Conclusions.....	140
8. Conclusions.....	143
8.1. Research Key Results.....	143
8.2. Generalizability and Applicability.....	148
8.2.1. Generalizability through case studies.....	148
8.2.2. Applicability based on Thales References.....	149
8.3. Future Work.....	150
References.....	153
Appendix.....	165
Résumé en Français.....	175

List of Figures

Figure 1-1: “Charles de Gaulle” carrier.....	2
Figure 3-1: Summary of the thesis steps, used approaches, and their link to research questions and contributions.....	28
Figure 3-2: Management of functions and resources jointly.....	30
Figure 3-3: OSysRec macroscopic view and main packages.....	32
Figure 3-4: MBSysRec synthesis, inputs, outputs and steps.....	35
Figure 4-1: DRM methodological framework, from (Blessing and Chakrabarti, 2009).....	38
Figure 4-2: Types of design research in DRM, from (Blessing and Chakrabarti, 2009).....	39
Figure 4-3: The adapted research methodology inspired by DRM (whose main stages are shown on the left side of the figure).....	41
Figure 4-4: Research progress in terms of RQs and thesis chapters presented across DRM stages. Crosses indicate that the works included the chapters (column) address the specified DRM stage (row).....	42
Figure 5-1: Empirical study methodology.....	47
Figure 5-2: (re)configuration process with data capitalization shortage during design.....	53
Figure 5-3: Management of functions and resources jointly to achieve missions during operations.....	55
Figure 5-4: Reconfiguration support desired for systems operations.....	55
Figure 6-1: OSysRec Macroscopic view.....	80
Figure 6-2: OSysRec ontology.....	82
Figure 6-3: Battlegroup illustration scenarios.....	90
Figure 6-4: OSysRec instance for scenario 1.....	91
Figure 6-5: (a): Mapping of applications on available modules of IMA (b): Reconfiguration of IMA based on different events.....	93
Figure 6-6: OSysRec instance for IMA scenario 1.....	95
Figure 7-1: MBSysRec concept diagram.....	107
Figure 7-2: Overview of MBSysRec method.....	108
Figure 7-3: The matrix system used in MBSysRec.....	109
Figure 7-4: MBSysRec steps and link to the matrix system.....	111
Figure 7-5: Instantiation information input to MBSysRec and MBSysRec-Observe flowchart.....	120
Figure 7-6: Objective-Functional chain relation and Functional chain - function relation matrices.....	122
Figure 7-7: Context-Component relation, Function-Component relation, and availability information... ..	122
Figure 7-8: Updated Function - component relation M'3.....	123
Figure 7-9: Component - Component relation for the SAR system.....	124

Figure 7-10: Components levels of functioning (H2) and configuration X matrices	125
Figure 7-11: Configuration X functional level, Functions weights(M6) , and Function interaction indices (M11) matrices	126
Figure 7-12: Score on each criterion of all possible configurations	126
Figure 7-13: Impact of change matrix	127
Figure 7-14: Component performance matrix	128
Figure 7-15: Configuration X score, criteria weights(M9), and criteria interaction indices (M11)	129
Figure 7-16: Overall satisfaction level and total number of configurations in a "Small object day" context and for a "Search" objective	130
Figure 7-17: Configuration 53 analysis per function.....	131
Figure 7-18: Summary of SAR scenarios results	133
Figure 7-19: MBSysRec proposal of a better configuration.....	134
Figure 7-20: STCA simplified functional chain	135
Figure 7-21: Overall satisfaction level for the STCA reconfiguration scenario.....	137
Figure 8-1: Thesis results applicability mapping	149
Figure 8-2: Simplified HUMS architecture from Thales.....	150
Figure A- 1: OSysRec "Zoom-in" part 1.....	166
Figure A- 2: OSysRec "Zoom-in" part 2.....	167
Figure A- 3:OSysRec instance for scenario 2	168
Figure A- 4:OSysRec instance for scenario 3	169
Figure A- 5: OSysRec instance for scenario 4	170
Figure A- 6: OSysRec instance for scenario 5	171
Figure A- 7: OSysRec instance for scenario 6	172
Figure A- 8: OSysRec instance for scenario 7	173

List of Tables

Table 2-1: Configuration as a product definitions.....	9
Table 2-2: Configuration task definitions.....	10
Table 5-1: Experts interviewed	49
Table 5-2: Questions list used for interviewing domain experts	50
Table 6-1: Meta-models/ontologies for reconfiguration classification	72
Table 6-2: Key concepts in the OSysRec Ontology	78
Table 7-1: Matrices description.....	110
Table 7-2: Linguistic terms for performance estimation	115
Table 7-3: Linguistic terms for the impact of change estimation	116
Table 7-4: SAR problem data.....	118
Table 7-5: Summary of configurations 53 and 55 scores and overall satisfaction levels.....	132
Table 7-6: STCA problem data	136
Table 7-7: Summary of configurations 1 and 2 scores for the STCA reconfiguration scenario	138

List of Abbreviations

HUMS	Health and Usage Monitoring System
SAR	Search And Rescue
RQ	Research Question
DRM	Design Research Methodology
PAR	Participation Action Research
OSysRec	Ontology for System Reconfiguration
MBSysRec	Model-Based System Reconfiguration method
SE	Systems Engineering
MBSE	Model-Based Systems Engineering
RTES	Real-Time Embedded System
RMS	Reconfigurable Manufacturing System
FTC	Fault Tolerant Control
FDI	Fault Detection and Isolation
RC	Reconfiguration Control
KTD	Thales Technical Directorate
IMA	Integrated Modular Avionics
CBG	Connected Battle Group
ATM	Air Traffic Management
STCA	Short Term Conflict Alert

Foreword

This PhD thesis dissertation results from a collaboration between Ecole CentraleSupélec and Thales Technical Directorate under a CIFRE (Conventions Industrielles de Formation par la REcherche) contract between November 2017 and October 2020.

The following papers, used to constitute this dissertation, have been either published or submitted to international conferences and journals:

- L. Qasim, M. Jankovic, S. Olaru, and J.-L. Garnier, “Model-Based System Reconfiguration: A Descriptive Study of Current Industrial Challenges,” in *Complex Systems Design & Management*, Cham: Springer International Publishing, 2019, pp. 97–108. (Chapter 5)
- L. Qasim, A. M. Hein, M. Jankovic, S. Olaru, and J.-L. Garnier, “Towards a reconfiguration framework for systems engineering integrating use phase data,” in *Proceedings of ICED 2019, the 22nd International Conference on Engineering Design: Responsible Design for Our Future*, Delft, Netherlands, 05.-08.08. 2019, 2019. (Chapter 5)
- L. Qasim, A. M. Hein, S. Olaru, M. Jankovic, and J.-L. Garnier, “An overall Ontology for System Reconfiguration using Model-Based System Engineering,” *Submitt. to IEEE Trans. Syst. Man, Cybern. Syst.*, 2020. (**Under minor revision**), (Chapter 6)
- L. Qasim, A. M. Hein, S. Olaru, M. Jankovic, and J.-L. Garnier, “An Ontology for System Reconfiguration: Integrated Modular Avionics IMA Case Study,” in the *Proceedings of the Conference on Systems Engineering Research 2020*, 2020. (**Accepted**) (Chapter 6)
- L. Qasim, A. M. Hein, S. Olaru, J.-L. Garnier, and M. Jankovic, “A model-based method for System Reconfiguration” *Submitt. to J. Mech. Des.*, 2020.(Chapter 7)

Some repetitions between the different chapters can be witnessed, which is inherent to the dissertation by papers style. The chapter “References” includes overall references.

1. Introduction

System reconfiguration is key for managing evolving systems as it enables systems to deal with various operating conditions. However, adequate system reconfiguration supports in terms of models, methods and processes are still lacking, specifically, for complex systems with various interacting elements. This research work aims at supporting system reconfiguration during operations towards an agile and optimized behavior. This chapter presents the research motivation and objectives. An outline of this dissertation is also given in this chapter.

1.1. Research Motivation

1.1.1. Role of System Reconfiguration in System Management

The "*Charles de Gaulle*" carrier has been ordered in 1986, has been put in operations in 1994, and is still in use. This, what one can consider as a relatively old system (26 years old), is planned to operate for several generations. However, keeping this carrier in use is a real challenge due to possible evolutions during its lifecycle (for instance, updating computer systems to collaborate with new and more modern planes). Life cycle evolutions are present in diverse forms and are often due to 1) changing operational contexts and conditions, 2) technology emergence or obsolescence, or 3) mission evolution. Any of these evolutions may actually disrupt the functioning of this system. Hence, these evolutions may lead to changes in the system's ability to perform its intended functions, which are called capabilities (ISO/IEC/IEEE:15288, 2015). For instance, the emergence of a new Rafale fighter increases the fighting capabilities of the Charles de Gaulle carrier.



Figure 1-1: “Charles de Gaulle” carrier

As the cost related to the development and acquisition of complex systems is considerable (€3 billion for the Charles de Gaulle carrier), there is an increased need to adapt such systems to accommodate the lifecycle evolutions and ensure the operational effectiveness that is defined as the degree to which a system satisfies its intended operational uses and delivers its defined capabilities (ISO/IEC/IEEE:24765, 2017).

System reconfiguration can be viewed as the adaptation of the system by changing its arrangement to meet a capability. System configuration is a key concept related to system reconfiguration. In Systems Engineering domain, the system configuration is defined as the set of elements that compose the system in terms of hardware devices, software, interfaces, human profiles, and processes (INCOSE Systems Engineering Handbook V4, 2015; ISO/IEC/IEEE:15288, 2015; NASA, 2007). For instance, to ensure the navigation capability of the Charles de Gaulle carrier, a configuration is composed of radars and thrusters. In general, the choice of one configuration depends on different aspects related to economy, environment, regulations, operation, system behavior and structure, and society (ISO/IEC/IEEE:15288, 2015). Any change in these aspects can impact the ability of the configuration to demonstrate the provided capability and may lead to system reconfiguration. Therefore, one can see that a capability is usually demonstrated through a configuration (NAF v4, 2018).

Even though system reconfiguration has many definitions in the literature, in this PhD thesis, we consider the system reconfiguration to be *the subsequent changes of the system configurations with the objective of maintaining or adapting (increasing or decreasing) the capabilities provided by the system* (Qasim, Hein, et al., 2019). For instance, in case that there is a problem or failure of a thruster on the Charles de

Gaule carrier there is a need to reconfigure (repair or change a thruster or find an equivalent solution) so that there is the continuity in its capacity to navigate.

Moreover, system reconfiguration is valuable for stakeholders not only as it is considered as a means for ensuring the operational effectiveness, but also as an enabler for adapting systems in order to ensure this operational effectiveness and increase system qualities such as safety and availability. System reconfiguration allows the control and optimization of system resources, services, and their interactions to accommodate lifecycle evolutions and manage the evolving systems (Dumond et al., 2009). Therefore, system reconfiguration is central to system management.

Given the strong interest in system management and reconfiguration, companies are becoming highly concerned with developing reconfiguration supports or functionalities, particularly when considering systems with increasing complexity, and that should last longer.

1.1.2. Systems reconfiguration at Thales

This research has been conducted in collaboration with the senior systems engineering experts of the Thales Group, a large international company developing various systems (aerospace, space, ground transportation, defense, and security) for both military and civil applications.

Thales is shifting towards building systems with improved survivability and adaptability. Future systems need to be built in a way to deal with unknown operational conditions and changing missions, making developing reconfiguration functionalities essential to ensure operational effectiveness. Increasing system operational effectiveness allows Thales to compete in different markets by offering its clients reconfiguration functionalities that ensure effective and efficient system management while reducing ownership and support expenses. Moreover, reconfiguration allows technology reuse leading to developing systems effectively while reducing cost and schedule overruns.

Future reconfiguration functionalities allow to adapt systems regarding evolutions in their missions and operational conditions. These functionalities can evaluate the operational situation based on the observed context and mission goals, find relevant configurations, and finally apply them by defining the corresponding reconfiguration actions. As Thales is developing different types of systems in various industrial sectors (e.g., radars for aerospace and navy sectors), the required reconfiguration functionalities in this context need to be generic. Moreover, generic system reconfiguration functionalities are known to improve system adaptability and survivability. Adaptability and survivability are achieved through

switching to another resource if the one in use fails, or through introducing a new resource if the provided capability is no more maintained or it does not fit the mission anymore. Adding or removing resources also allows for dealing with technology emergence or obsolescence. Emergent technologies can either be integrated into a resource and thereby introduced into the system. Obsolescent technologies can be removed with a resource. Therefore, system reconfiguration considers the lifecycle evolutions, increasing the adaptability and survivability of critical systems such as the ones developed within Thales (e.g., Air Traffic Management in the aerospace domain). Such functionalities rely on system engineering and operational data. As system reconfiguration considers the lifecycle evolutions, its related data is also subject to evolutions. Therefore, reconfiguration data needs to be captured in a way that will allow flexibility in the evolution of system information regarding resources addition or removal as well as its industrial deployment. However, these system reconfiguration functionalities are currently not standardized with patterns in Thales, and therefore, they have to be explicitly defined and modeled per system.

Approaches of Model-Based Systems Engineering (MBSE) get to be more and more accepted within industry as a foundation for better system engineering and management (Gausemeier et al., 2013). MBSE supports data modeling at a higher level of abstraction (Madni and Sievers, 2017). Therefore, we consider MBSE as a promising avenue to support system reconfiguration with modeling capabilities while ensuring the required abstraction and adaptability. This PhD focuses on supporting system reconfiguration by setting up adequate models, processes, methods, and approaches.

1.2. Research Objective

We formulated the research objective based on our preliminary industrial observation. The overall objective is to enhance system operational effectiveness by supporting system reconfiguration. As Thales is developing a variety of system types in various industrial domains, generic system reconfiguration functionalities are needed. Such functionalities should consider system lifecycle evolutions. Given that data is key for system reconfiguration, it is necessary to capture and use the data coming from system design and operational activities. However, our preliminary observation revealed that the standard definition of this data is lacking. To deal with the required abstraction of reconfiguration functionalities and its related data, models and model-based approaches were considered promising. We argue that using a model-based approach for system reconfiguration allows considering lifecycle evolutions. For instance, when adding new resources, the models can be updated to consider this evolution and can be further used for reconfiguration purposes.

The initial aim of this research is to support system reconfiguration using a model-based approach. During this PhD thesis, we conducted an industrial audit to identify more precisely the encountered difficulties related to our research objective. We, then analyzed these difficulties with regard to the literature review that allowed further refinement of research questions. Section 3.1 details this refinement and summarizes the research questions.

1.3. Thesis Outline

This dissertation is organized into eight chapters. In chapter 1, we describe our motivation for conducting this research and detail the main research objective. In chapter 2, we present a review of the relevant literature. Chapter 3 gives an overview of the research steps and thesis contributions. In chapter 4, the research methodology adopted for this research work is presented. Chapters 5, 6, and 7 present the main research contributions. Chapter 5 identifies the difficulties and challenges related to system reconfiguration. In this chapter, we detail the qualitative study conducted in the industry. Chapter 6 presents the ontology developed to capture the knowledge related to the reconfiguration process. The method developed for system reconfiguration is presented in chapter 7.

Finally, chapter 8 discusses the answers that we have provided to the research questions as well as the research quality and the limitations of the research results. The generalizability and the applicability of the research results are also discussed in chapter 8. We conclude this thesis by presenting directions for future research.

2. Literature Review

This chapter presents definitions and literature from relevant domains. In order to understand reconfigurability and the general reconfiguration process, we explore the different research areas where the reconfiguration problem is discussed. System architecture and configuration domains are relevant as they both consider combining different elements into one system configuration. Platformization is relevant as product configuration during development can be achieved by introducing platform-based engineering. Modularity is briefly discussed as it is considered an enabler for system reconfiguration. Modularity deals with defining the modules that can be further used within the configuration/ reconfiguration process. In practice, modular systems can be easily configured and reconfigured. Understanding modularity is important to understand how the different modules within a system can be combined and arranged. Command and control and domains linked to embedded systems are interesting as they discuss reconfiguration strategies and causes based on the system's failures. Moreover, reconfigurable manufacturing systems domain considers requirement changes when discussing reconfiguration strategies. Both the autonomous systems and software and computing systems domains are significant to this thesis as they discuss the reconfiguration process addressing both system failures and requirement change. At last, we discuss a major domain that is significant for this research work, i.e., systems engineering, specifically model-based systems engineering, as the objective of this research work is to support system reconfiguration using a model-based approach.

2.1. Understanding Reconfiguration

This section gives a general understanding of reconfiguration and explains what reconfigurability essentially means. This step revealed that the reconfiguration problem has been treated differently in several domains. The key issue about reconfiguration is when it is prepared and actually happens during the system life cycle. Depending on when the system reconfiguration is done in the life cycle, the objectives and tasks incorporated in the process can be different. In this research work, we consider two life cycle phases:

engineering (design) and operations. Hence, we classify the reviewed domains into design-time and run-time related domains.

2.1.1. Design-time Related Domains

- **System Configuration and Reconfiguration**

The term reconfiguration means various things. One related term to reconfiguration is configuration, hence, we start by giving a definition of the term product configuration. Product or system configuration is related to several definitions or perceptions (Oddsson et al., 2014). Among these definitions are configuration as a product or artefact (Mckay et al., 1996; Mittal and Frayman, 1989), and configuration as a task (Aldanondo and Vareilles, 2008; Hofstedt and Schnee-weiss, 2013; Mittal and Frayman, 1989).

Table 2-1 gives a summary of configuration definitions as a product or artefact. Analyzing these definitions, one can notice that in general authors define configuration as a set of components that are arranged together (Krause et al., 1993; Magro, 2010; Männistö, 2000; Mckay et al., 1996; Mittal and Frayman, 1989; Oddsson et al., 2014; Zhou et al., 2008). Mittal and Frayman (Mittal and Frayman, 1989) precise that a product configuration also includes the ports allowing connecting the different components. Other clearly refer to product configuration as the final product or artefact that is resulting from the configuration process (Krause et al., 1993; Oddsson et al., 2014; Schwarze, 1996). The final product can be physical, information, or service (Oddsson et al., 2014; Schwarze, 1996). Product configuration is usually designed to deliver a function or purpose (Krause et al., 1993; Mittal and Frayman, 1989; Zhou et al., 2008), and it is intended to be sold (Mckay et al., 1996; Oddsson et al., 2014).

Table 2-1: Configuration as a product definitions

Source	Configuration definition as a product
(Mittal and Frayman, 1989)	<ul style="list-style-type: none"> • A product is a fixed, pre-defined set of components, where a component is described by a set of properties, ports for connecting it to other components, constraints at each port that describe the components that can be connected at that port, and other structural constraints and the components that can be used to design some artefact are fixed. • Artifacts are typically (but not always) designed with some purpose in mind.
(Krause et al., 1993)	<ul style="list-style-type: none"> • Products are materialized, artificially generated objects or groups of objects which form a functional unit. The materialization may contain mechanical parts, electrical components, electronic components, hydraulic components, and other components.
(Mckay et al., 1996)	<ul style="list-style-type: none"> • A product is a special kind of part that is, or is intended to be, sold. • A part can be described as either an assembly or a component. An assembly is a physical object that is built from two or more other physical objects (parts) whereas a component is a physical object that is treated as non-decomposable.
(Schwarze, 1996)	<ul style="list-style-type: none"> • A product is a thing, a substance or a service produced by a natural or artificial process.
(Männistö, 2000)	<ul style="list-style-type: none"> • A product is composed of components, which in turn are composed of further components, etc
(Zhou et al., 2008)	<ul style="list-style-type: none"> • a product can be seen as a functional unit with particular materials, a fixed form and other designated features.”
(Magro, 2010)	<ul style="list-style-type: none"> • Several products have a composite structure and an architecture which is based on a set of interacting modules.”
(Oddsson et al., 2014)	<ul style="list-style-type: none"> • A product is an artifact, a substance, information, or a service. • A configurable product is composed of several entities, produced by a natural or artificial process and is, or is intended to be, sold. • A product is the final product, the product instance, the product individual or the final product that is described after the configuration process.

As discussed previously, the configuration as a final product or artefact is the result of a configuration task or a configuration process. Table 2-2 summarizes the different definitions found for configuration task. These definitions highlight that the core problem of the configuration task is to select and arrange components in way to satisfy the specified requirements, while respecting the compatibility constraints.

Table 2-2: Configuration task definitions

Source	Configuration definition as a task
(Mittal and Frayman, 1989)	<ul style="list-style-type: none"> Given: (A) a fixed, pre-defined set of components, where a component is described by a set of properties, ports for connecting it to other components, constraints at each port that describe the components that can be connected at that port, and other structural constraints (B) some description of the desired configuration; and (C) possibly some criteria for making optimal selections. Build: One or more configurations that satisfy all the requirements, where a configuration is a set of components and a description of the connections between the components in the set, or, detect inconsistencies in the requirements.
(Brown, 1998)	<ul style="list-style-type: none"> Configuring = Selecting + Associating + Evaluating, where: <ul style="list-style-type: none"> Selecting = Choosing components; Associating = Establishing relationships between components; and Evaluating = Compatibility Testing + Goal Satisfaction Testing.
(Sabin and Weigel, 1998)	<ul style="list-style-type: none"> The configuration task is a special case of design activity where the artifact being configured is assembled from instances of a fixed set of well-defined component types" which can be composed conforming a set of constraints. Product configuration is to select and arrange combinations of parts or components that satisfy given specifications
(Dhungana et al., 2017)	<ul style="list-style-type: none"> Product configuration is a well-established methodology for generating and building individualized products.
(Aldanondo and Vareilles, 2008)	<ul style="list-style-type: none"> Product configuring is finding at least one set of components that satisfies all constraints and customer requirements
(Hofstedt and Schnee-weiss, 2013)	<ul style="list-style-type: none"> Product configuration describes the process of specifying a product according to user-specific needs based on the description of all possible (valid) products (the search space).
(Soininen et al., 1998)	<ul style="list-style-type: none"> Configuration as a task can be roughly defined as the problem of designing a product using a set of predefined components while taking into account a set of restrictions on how the components can be combined. The term product configuration is used to denote the routine engineering activity of this type in the sales-order-delivery process.

While product configuration deals with designing products to sell, reconfiguration is considered an important task to perform on systems during the after-sales life cycle (Manhart, 2005). Therefore, configuration activities and tasks undertaken to re-form an earlier configured product is referred to as reconfiguration (Zhang, 2014). In system configuration domain, reconfiguration is considered important as it allows system modifications to address new requirements from customers (Männistö et al., 1999).

Various product configuration methods have been reported in the literature (Falkner et al., 2011; Zhang, 2014). All these methods support the configuration task of the product to sell. On the other hand, after-sales reconfiguration is still a challenge to be addressed (Manhart, 2005).

- **System Architecture Design**

System architecture design and system configuration/ reconfiguration are research domains that are highly linked and often not easy to separate. In fact, both domains deal with combining building blocks to propose system configurations while respecting their mutual constraints (Ziv-Av and Reich, 2005).

Systems architecture is concerned with defining the structure and behavior of complex systems to meet their operational and functional needs (NAF v4, 2018). Several scholars have attempted to provide definitions for system architecture (Stone et al., 2000; Ulrich and Eppinger, 1995; Yassine and Wissmann, 2007) According to Crawley et al. (2004), these definitions share the idea that system architecture is an abstract description of the system entities and the relationships between them. Therefore, they define **product architecture as a model** of an engineering artifact in terms of components linked by relations.

Architecting activities consist of generating architecture alternatives, exploring their strengths and weaknesses, and selecting one based on defined requirements and specifications (Ulrich and Eppinger, 1995). In the literature, the architecture definition process is guided by many architecture frameworks (DODAF, 2009; MODAF, 2010a; NAF v4, 2018; TOGAF, 2009; Voirin and Bonnet, 2013), and norms and standards (ISO/IEC/IEEE:42010, 2011; ISO/IEC/IEEE:42020, 2019; ISO/IEC/IEEE:42030, 2019)

The literature is rich with system architecture design methods that vary according to the objective they address. Amongst these objectives is generating a variety of architectures while ensuring their viability (Bryant et al., 2005; Chen et al., 2019; Hamza et al., 2011; Kurtoglu and Campbell, 2009). Many system architecture design methods have been developed to support the conceptual phase when more innovative architectures are expected (Albarello et al., 2012; Helms and Shea, 2012; Moullec et al., 2013). Other researchers concentrate on the issue of interfaces when generating system architectures (Jankovic et al., 2012; Wyatt et al., 2008, 2012; Ziv-Av and Reich, 2005). In order to allow further assessments during the architecting process, researchers suggest to integrate different system and domain requirements (Condat et al., 2012; Judt and Lawson, 2012; Rosenstein and Reich, 2011). Haris and Dagli (2011) suggests to consider architecture evolutions (reconfigurations) to account for customer requirement evolutions during the detailed design phase.

- **Platformization**

This section presents a review of the product platformization literature as product configuration during development can be achieved by introducing platform-based engineering.

In today's competitive market, companies are concerned with expanding their product lines and varying their product offerings (Ho and Tang, 1998). Product variety can be achieved effectively by designing and developing product families (Meyer and Utterback, 1993; Sundgren, 1999). One fundamental approach of platform-based product family design is the configurational product family design (Du et al., 2001; Ulrich, 1995). The configurational approach aims at developing modular product architectures from which products can be designed by adding, substituting, and/or removing one or more functional modules (Du et al., 2001; Ulrich, 1995). In this context, product architectures that involve mappings of functional, structural, and physical elements allow changing functional elements of a product by only changing the corresponding physical component incorporated in the product (Ulrich, 1995). Therefore, one can see that product variation can be achieved by configuring existing building blocks.

In platform-based design, configuration refers to the arrangement of different components to satisfy customer requirements (Deciu et al., 2005). The change in configurations layout to meet the needs of multi-product manufacturers is referred to as reconfiguration (Benjaafar et al., 2002).

Two research currents exist for platform-based design. A significant research current concentrates on arranging a predefined set of components into a product structure as the main problem (Männistö, 2000). The second current considers designing the different components constituting the configurable product as an essential problem as they characterize the achievable degree of freedom in configuring products (Deciu et al., 2005).

- **Modularization**

Salvador (2007), in a review of the literature defining modularity, identifies different perspectives in the existing definitions. These dimensions include component separability and combinability, commonality, function binding, interface standardization, and loose coupling.

From the separability and combinability perspective, products are considered modular when different product configurations are obtained by mixing-and-matching components taken from a given set (Baldwin

and Clark, 1997; Langlois, 1992). To achieve this combinability, modularity is concerned with separating systems or products into modules, which are physical or conceptual groupings of components that share some characteristics (Newcomb et al., 1996). Modules can be defined based on functional, technical, and physical modularity (Jiao et al., 2007). From a practical point of view, increasing system modularity facilitates achieving a wide variety of system configurations without loss of the system's functionality or performance (Baldwin and Clark, 1997; Langlois, 1992).

When considering the life cycle perspective, modularity can have different objectives and approaches over the life cycle stages. For example, improving the system design and decreasing time-to-market imply standardization of components that can be kept similar while keeping the customer's perception of product variety and differentiation. Designing for assembly is another objective which implies diverse approaches to product modularization to facilitate its production by reducing fixing points and using plug-and-play interfaces. Modularity plays a crucial role in supporting system reconfigurability during operations. Product design modularity aims at improving usability by making the product usable independently from other modules. Making the subsequent versions of the same product compatible and upgradable is another approach of product design modularity to improve serviceability. Making the system modular is also a way to reduce recovery time in case of failures. Therefore, designing for reparability is also an objective of product design modularity to support its use and operations.

2.1.2. Run-time Related Domains

- **Command and Control**

In operations, it is important to supervise and control component or sub-system operations to maintain the system's desired behavior. Once a fault in one component has been identified within a feedback loop, the controller must react by reconfiguring the system to tolerate abnormal behaviors. The literature is addressing these concerns as fault detection, isolation, and reconfiguration or FTC. The primary purpose of FTC systems is to overcome the malfunctions while maintaining desirable stability and performance properties (Noura et al., 2009; Qi et al., 2013; Stoican and Olaru, 2013; Zhang and Jiang, 2008).

Passive and active FTC systems exist, depending on their management of detected faults. Passive FTC systems are robust control systems that handle faults within a predefined class. On the other hand, active FTC systems react to a detected fault and reconfigure the control actions so that the stability and the

performances can be maintained (Eterno et al., 1985). In active FTC systems, the fact that the controller is reconfigurable means that one can adaptively address non-predefined faults.

A typical active FTC system relies on two fundamental mechanisms: fault detection and isolation (FDI) sometimes referred to as “fault diagnosis” (Isermann, 1997), and reconfiguration control mechanisms (RC) (Zhang and Jiang, 2008). The reconfiguration control aims at masking the fault either by switching to a redundant system/component or by revising the controller structure. In some cases, the available resources do not allow masking faults. In such cases, the best solution is to allow system degradation when the performance is accepted to be out of the optimal area (Noura et al., 2009; Stoican and Olaru, 2013).

There are different techniques used in fault detection and isolation; these techniques are classified into model-based and data-based (Zhang and Jiang, 2008). Model-based methods use system models to estimate the system states and parameters. Data-driven techniques, on the other hand, rely on classifiers and signal processing (Zhang and Jiang, 2008). In this research, the interest lies in changes and deviations in the system state addressed by model-based techniques while data-driven techniques fall out of this research interest.

Reiter (Reiter, 1987), in his theory of diagnosis, proposes a method that requires a model describing a given system. Based on the observations, the diagnosis system compares it with the expected behavior to determine the malfunctioning components. Reiter’s theory has been extended to deal with the model-based diagnosis of different kinds of systems in different domains of applications (Kuntz et al., 2011; Ng, 1990).

Identifying faults in malfunctioning systems is important but repairing the system so that it can continue its mission is an essential problem to be addressed. Reiter’s theory of model-based diagnosis has been extended to a theory of reconfiguration (Crow and Rushby, 1991). Much research has been conducted to use the model-based analysis concepts in the reconfiguration controller design and analysis algorithms (Cui et al., 2018; Provan and Chen, 1999b; Russell and Broadwater, 2012; Shan and Hou, 2016).

- **Embedded Systems**

Embedded systems are microprocessors or microcontrollers, integrating both hardware and software elements that are designed to achieve a dedicated function within a given application (Krichen, 2010). The integrated embedded system is designated to carry out computation for real-time operations (T. Henzinger and Sifakis., 2006). Embedded systems monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa (Lee and Seshia, 2017). These systems connect to the outside world through peripherals, linking input, and output devices. Embedded systems rely on several components to achieve their control over the physical process (Lee and Seshia, 2017). These

components include a sensor, an analog to digital converter, the processor itself, a digital to analog converter, and finally an actuator. The evolution of user requirements and the change of execution environment constraints require the dynamic reconfiguration of systems providing the physical processes (Krichen, 2010). In embedded systems, the process of modifying a system's structure and behavior during its execution is called dynamic reconfiguration (Polakovic et al., 2007). In general, a configuration corresponds to a functional mode (Krichen and Zalila, 2011). This mode corresponds to how components, elements, and processes are integrated and how they interact. Reconfiguration is referred to as a transition between systems modes when triggering events occur.

Most of the researchers in this domain focus on designing and developing the safe reconfiguration for embedded systems as this process can be error-prone (Ali et al., 2011; Ju et al., 2015; Krichen et al., 2012; Krichen and Zalila, 2011; Li et al., 2015; OMG, 2010; Polakovic et al., 2007; Witt et al., 2013). Polkavic et al. (Polakovic et al., 2007) implemented safe dynamic reconfigurations by using the component-based framework for building reconfigurable operating systems and using a domain-specific language to specify reconfiguration. Both studies by Ju et al. (Ju et al., 2015) and Li et al. (Li et al., 2015) use the Architectural Analysis and Design Language (AADL) to model the dynamic reconfiguration of the Integrated Modular Avionics (IMA) regarding system operating mode. Li et al. (Li et al., 2015) extends the AADL model and transform it into a Petri net, which is used to verify the logical properties of the dynamic reconfiguration process.

Model-based approaches to handle safe reconfiguration for embedded systems have been the focus of many researchers (Ali et al., 2011; Krichen and Zalila, 2011; Witt et al., 2013). Ali et al. (Ali et al., 2011) proposed an agent-based approach that relies on a meta-model. The proposed approach suggests that reconfiguration operations or scenarios (i.e., ordered sequence of operations) occur in response to several types of requests, including problem diagnosis, improvement requests, prevention requests, and adaptation requests. The MARTE profile specified by the Open Management Group (OMG, 2010) extends UML capabilities to allow for model-driven development of Real-Time Embedded Systems (RTES). In the UML meta-model presented by Krichen et al. (2011), a state machine is proposed to describe the meta-modes and transitions. Moreover, this model describes the structured components and connections between them to ensure the structural integrity of systems when performing reconfiguration. The behavior of real-time embedded systems was investigated by Witt et al. (Witt et al., 2013), who presented a SysML profile based on the ideas of state analysis.

- **Reconfigurable Manufacturing Systems**

Reconfigurable Manufacturing System (RMS) represents a paradigm shift in manufacturing systems where responsiveness is the new objective along with cost and quality. The aim is to achieve rapid system changes as needed and when needed while still being cost-effective (El Maraghy, 2006). The RMS paradigm is linked to new trends related to industry 4.0.

Reconfigurable Manufacturing Systems refer to systems in which a machining system is created by incorporating basic process modules, both hardware, and software, that can be rearranged or replaced quickly and reliably. The reconfigurability allows for adding, removing, or modifying specific process capabilities, controls, software, or machine structure to adjust production capacity in response to changing market demands or technologies (Mehrabi et al., 2000). In this context, researchers generally define configuration as the composing modules (software and hardware) of the manufacturing system and their on-site set up (Regulin et al., 2016). The change in structure, hardware, and software components to quickly adjust production capacity and functionality within a part family is called reconfiguration (Alsafi and Vyatkin, 2010).

Several researchers have contributed work that deals with understanding, modeling, and designing RMS. For instance, Alsafi et al. (Alsafi and Vyatkin, 2010) proposed an ontology-based approach to achieve fast reconfiguration of modular manufacturing systems. In the presented approach, Alsafi et al. suggest an agent that uses knowledge of the manufacturing environment for reconfiguration purposes without human intervention. A model-based approach using SysML to describe the manufacturing system modules and the abstract processes as well as a code generation to reconfigure this system is proposed by Feldmann et al. (Feldmann and Loskyll, 2013). This approach aims at enhancing the efficiency and flexibility of manufacturing systems. Weyrich et al. (Weyrich et al., 2014) concentrated on the design of reconfigurable manufacturing systems by integrating the reusable modules in the engineering and simulation process. Regulin et al. (Regulin et al., 2016) used a model-based description to derive and to online process the knowledge bases so that the automatic reconfiguration of manufacturing systems is enabled.

- **Autonomous Systems**

Autonomy represents a rising demand in the future development of critical applications such as spacecraft systems (Peng et al., 2012; Tipaldi et al., 2017), self-driving cars (Axhausen et al., 2017), unmanned aerial vehicles (Zugaj, 2017), smart grids (Khedertzadeh and Beiranvand, 2018). Autonomous systems are built to be capable of adapting themselves in response to uncertain circumstances (Hernández et al., 2015). In the

context of autonomous systems, reconfiguration plays an essential role as it allows the adaptation of systems by directly acting on them (Sanz et al., 2014). Therefore, reconfiguration can be defined as an adaptation that can occur when the context (represented by parameter values) changes (Rodriguez et al., 2009). Such a transformation occurs if and only if the adaptation rules are satisfied.

Various research works contributed by modeling, designing, analyzing autonomous systems in a variety of applications (Khederzadeh and Beiranvand, 2018; Peng et al., 2012; Tipaldi et al., 2017; Zugaj, 2017). Some research works contributed by proposing model-based approaches for reconfiguration. Bermejo-Alonso et al. (Bermejo-Alonso et al., 2011) developed an ontology for system reconfiguration. The ontology encompasses both generic and domain-specific concepts for autonomous systems description and engineering. Hernández et al. (Hernández et al., 2015) presented the TOMASys meta-model, which lies at the core of meta-controllers' operation for autonomous systems as it specifies the system functional model used by the controller. To support autonomous systems engineering, Bermejo Alonso et al. (Bermejo-Alonso et al., 2016) proposed an ontology-based approach that relies on models capturing both the instantaneous state of the system at runtime and the knowledge about the design of the control system and properties of its components.

- **Software and Computing Systems**

Reconfiguration in software and computing systems is widely implemented as the cost of implementation is reduced compared to other domains. Configuration in such systems corresponds to assembled software components that deliver functions (Saxena et al., 2010). Swapping faulty components for working ones in the event of either component failure or an evolved mission goal is referred to as reconfiguration (Saxena et al., 2010).

Many research works treating reconfiguration in the software and computing domain exist. The existing studies address different issues related to reconfiguration, such as 1) performance system management at runtime based on monitoring and model-based performance evaluation (Caporuscio, Mauro and Marco, Antinisca Di and Inverardi, 2005; Caporuscio et al., 2007; Saxena et al., 2010; Vogel et al., 2010), 2) generating plans for self-adaptive software systems (Dalpiaz et al., 2009; da Silva and de Lemos, 2011; Yeom, 2015; Yeom and Park, 2012; Ziegert and Wehrheim, 2013), 3) online validation (Akue et al., 2012) and reliability verification (Leger et al., 2015) of dynamic reconfigurations ensuring that these processes do not nullify the system functionalities and lead to inconsistent states introduced by runtime modifications. Among the model-based approaches investigated in the literature, the approaches presented in (Gogniat et

al., 2013; Meyer et al., 2013; Walsh et al., 2005, 2006) are interesting as they propose models describing the reconfiguration process at a high level of abstraction.

2.2. Systems Engineering (SE) and Model-Based Systems Engineering (MBSE)

This section gives the state-of-the art on the model-based systems engineering as the objective is to support system reconfiguration using a model-based approach. Model-based systems engineering has originated from systems engineering. In this section, we first address advances in SE. Next, MBSE is addressed.

2.2.1. Systems Engineering (SE)

Systems Engineering (SE) is a discipline that considers the design and application of systems in a holistic manner. In SE, the perspective is to look at the problem in its entirety, considering and relating its different facets, variables and aspects (INCOSE Systems Engineering Handbook V4, 2015; NASA, 2007).

The core functions of SE are related to the identification of stakeholders and their needs and expectations, the identification of systems goals and interfaces, the definition and management of budgets and schedules, establishing configuration management practices and maintenance processes, developing systems that are effective regarding the defined needs and expectations, the creation of all documents concerning design description, use and risk, and finally, monitoring system implementation, integration, test and acceptance (INCOSE Systems Engineering Handbook V4, 2015; NASA, 2007).

These core functions are formalized and organized in different lifecycle processes, including technical, management, agreement and organizational processes (ISO/IEC/IEEE:15288, 2015). System configuration is an essential aspect addressed by the system management processes.

The objective of system configuration management is to ensure effective management of the evolving systems, both resources and processes, during its lifecycle (INCOSE Systems Engineering Handbook V4, 2015; ISO/IEC/IEEE:15288, 2015). System configuration is characterized by operational aspects, activities, systems, software, hardware and human profiles that are put together to demonstrate a capability. The system configuration management ensures its main objective by establishing a reference point (configuration) to which the system is always compared.

2.2.2. Model-Based Systems Engineering (MBSE)

The Model-Based Systems Engineering (MBSE) paradigm has emerged to harness systems complexity and increasing scales. MBSE can be described as the formalized application of modeling principles, methods, languages, and tools to the entire lifecycle of large, complex, interdisciplinary, sociotechnical systems (Ramos et al., 2012). Developing and managing systems using MBSE involves placing models at the center of these processes (Wymore, 1993).

A model is an abstraction of reality integrating diverse aspects from different perspectives and usually it is designated to a purpose (Aßmann et al., 2006; Madni and Sievers, 2017). The importance of models lies in capturing and stating domain knowledge to be shared among stakeholders, exploring system design and solutions, mastering complexity, giving insights into experimentation, deployment, and system operation in real time, gaining insights into the world in which systems are operating (Booch et al., 1999; Buede, 2009). Different types of models can be identified based on their use and purposes (e.g., structural, functional, behavioral, performance, etc.) (Dori, 2016). A model can be represented using different modeling formalisms, including formal and informal languages (Buede, 2009; Sanford, Friedenthal et al., 2014). An informal model is usually a text description or concept diagram written in a standardized notation. A formal model relies on formalisms and their visual representation. Complex systems are generally modeled using formalisms and visual representations enriched with text descriptions. A diversity of MBSE modeling languages and tools exist (Dori, 2016; Roques et al., 2016; Sanford, Friedenthal et al., 2014)(OPL/OPM, SysML, UML, UPDM, Capella, Arkietct, Rhapsody).

In MBSE, ontologies and meta-models play a key role in the entire systems engineering process (Madni and Sievers, 2017). Madni et al. (Madni and Sievers, 2017) define ontologies as formal, explicit specifications of shared conceptualizations for specific domains. Ontologies help formalizing domain knowledge by providing the specific vocabularies for a domain and relations among them (Guarino et al., 2009). Ontologies form the basis for a common understanding among stakeholders and they allow to analyze domain knowledge. In this context, ontologies represent conceptual frameworks that allow to consider problems and solutions related to a specific domain (Chandrasekaran et al., 1999). In MBSE, tracing the abstract model elements into more specific models is achieved through meta-modeling and meta-models (Sanford, Friedenthal et al., 2014). A meta-model is the formal definition of the properties of a model, i.e., a model that specifies the abstract syntax used by a modeling language. Metamodeling aims to identify what can be expressed using a modeling language (Aßmann et al., 2006). Ontologies and meta-models are closely related as they both provide concepts and define relations and validity rules between them. However,

ontologies are different from meta-models in that they do not describe systems, only domains (Aßmann et al., 2006).

2.3. Synthesis

In this section, we only highlight the main research gaps. A detailed discussion of the literature is provided in chapters 6 and 7.

From the previously discussed literature addressing reconfiguration, one can notice that configuration refers to the arrangement of the different components into the system in a way to satisfy the stakeholders' needs. On the other hand, reconfiguration refers to either the re-arrangement of these components or the transition between two consequent arrangements in the response of triggering events. In both cases, reconfiguration occurs to accommodate system failures, stakeholders' needs, or missions' evolution. However, when considering configuration from the systems engineering perspective, one can see that configuration refers to the components and interfaces (structural aspect), the processes that govern the system (dynamic aspect), as well as their arrangement (management aspect). In the previous sections, we discussed many definitions of system reconfiguration. When considering the existing definitions, one can see that these definitions address the three aspects of system reconfiguration separately. Therefore, we propose to define system reconfiguration, in systems engineering, as the subsequent changes of the system configurations with the objective of maintaining or adapting (increasing or decreasing) the capabilities provided by the system. Since our objective is to support system reconfiguration using a model-based approach, we analyzed the existing data models or ontologies for system reconfiguration. Several studies from the reviewed domains discussed reconfiguration data and proposed related data models or ontologies. However, to the best of our knowledge, there is no data model or ontology that considers and synthesizes the three aspects of system reconfiguration (i.e., structure, dynamics, and management).

It can also be noted that the design-time related domains are concerned with designing system configurations while satisfying new requirements. Moreover, the need to make systems reconfigurable has motivated a tremendous work on modularity and platformization. Existing methods and approaches rely on system design data to achieve reconfiguration. In the run-time related domains, on the other hand, researchers focus on analyzing system transitions and changes while considering the measures of effectiveness (e.g., cost, safety, reliability, etc.). This aspect of reconfiguration relies mainly on observing systems during operations. Therefore, one can see that there is a gap in the existing body of research concerning methods that integrate data from both observations (from operations) and system design model (from engineering).

Furthermore, the literature review revealed that existing studies focused on a particular application (e.g., robots, space systems, microgrids, etc.). There is no generic method or tool addressing reconfiguration that is not application or domain-specific.

The research gaps are also highlighted in section 3.1.2 and discussed with regard to the main industrial issues in section 3.1.3.

3. Research Overview and Thesis Contributions

This chapter aims at providing a global understanding of this dissertation. We summarize the steps that were followed to conduct this research in section 3.1. First of all, an industrial audit was conducted in the System Domain of the Thales Technical Directorate (KTD) to identify the needs and difficulties encountered when dealing with system reconfiguration. Then, we reviewed the literature in light of the industrial audit results. Both are developed in Section 3.1.1 and 3.1.2. Based on the mapping between industrial needs and research gaps (section 3.1.3), we raised the research questions detailed in section 3.1.4. Finally, we give a summary of the thesis contributions in section 3.2.

3.1. Research Overview

3.1.1. Synthesis of main issues from the industrial audit

The objective of this thesis is to support system reconfiguration using model-based approaches. To identify the existing approaches and the main issues encountered when developing system reconfiguration functionalities, we conducted an industrial audit within Thales. The industrial audit covered the various Thales business units. This industrial audit is detailed in the descriptive study (chapter 5). In this section, we highlight the main challenges that we address in this thesis and that is related to the development of reconfiguration functionalities.

For reconfiguration functionalities development, the descriptive study revealed that it is essential to capture reconfiguration data, leading to exhaustive and efficient configuration definition and selection. Efficient system reconfiguration relies on understanding the system design in terms of resources and the functions they provide. Reconfiguration also needs to integrate feedback from operations to enhance overall system performances. Therefore, it is necessary to identify the required data with different sources and link them to reconfiguration mechanisms. As, within Thales, reconfiguration functionalities are developed individually

for each project, the collected reconfiguration data are also specific and highly domain-dependent. Moreover, when defining reconfiguration data, system engineers and architects rely on hints from the end-users which may result in insufficient consideration of mission variation and changes in the system operating conditions. Therefore, standard definitions or models identifying system reconfiguration data are needed to ensure efficient development of reconfiguration functionalities, leading to improving systems operational effectiveness.

In addition to data identification, the descriptive study also showed that developing reconfiguration functionalities requires identifying the mechanisms and processes necessary to combine reconfiguration data. To do so, it is important to understand how data can be manipulated to identify relevant system configurations. Moreover, the required mechanisms should support dynamic reconfiguration in terms of resource allocation and re-allocation, allowing optimization of system capabilities regarding its missions. Current system reconfiguration functionalities allow only changes between pre-defined configurations and do not support dynamic reconfiguration of systems during operations. Given that the reconfiguration data is related to design and operations, the required reconfiguration mechanism should be able to combine and integrate data from these two phases and use it to support system reconfiguration by defining, assessing, and selecting relevant system configurations.

The main issues concerning the development of reconfiguration support can be summarized in:

- Lack of standard definition of reconfiguration data.
- Lack of mechanisms and processes supporting dynamic reconfiguration during operations.

3.1.2. Scientific State-of-the Art and Research Gaps

Given the thesis objective, our literature review (cf. chapter 2) focused on understating reconfigurability and the general reconfiguration process as well as the systems engineering domain and, more specifically, model-based systems engineering. System reconfiguration has been treated in various domains, including: system architecture, system configuration, platformization, modularity, command and control, embedded systems, reconfigurable manufacturing systems, autonomous systems, and software and computing systems. Based on the synthesis of the literature provided in section 2.3, we identified research gaps. In this section, we underline the research gaps that we address in this thesis.

As discussed in section 2.3, system reconfiguration relies on three main aspects: structure, dynamics, and management. The structural aspect is mainly linked to the system components and interfaces. The dynamic

aspect deals with system behavior, evolution, and transitions. The management aspect is concerned with optimizing the existing resources regarding the considered context and the mission. In order to master system reconfiguration, it is essential to consider this process in its entirety.

Model-based systems engineering has emerged to support complex system design and management (Madni and Sievers, 2017) while reducing cost and time via model use and reuse (Wymore, 1993). An important approach used as a basis in MBSE is ontology development. Ontologies help to formalize domain knowledge by providing the specific vocabularies for a domain and relations among them, which is a precondition for developing MBSE tools. Therefore, we argue that an ontology or a data model is necessary to support system reconfiguration in an overall manner.

From our literature review, we identified several ontologies that have been developed to support system reconfiguration (Ali et al., 2011; Bermejo-Alonso et al., 2011, 2016; Gogniat et al., 2013; Hernández et al., 2015; Krichen and Zalila, 2011; Meyer et al., 2013; OMG, 2010; Walsh et al., 2005, 2006; Witt et al., 2013). However, to the best of our knowledge, there is no system reconfiguration ontology that considers the overall system reconfiguration process and addresses all its related aspects. Furthermore, when considering system reconfiguration methods and mechanisms, one can see that the existing methods rely either on integrating system design data or observation data from operations concerning the system health and its context. Therefore, existing methods fail to integrate the necessary data coming from both design and operations.

The main research gaps can be summarized in:

- There is a lack of data models or ontologies that consider and synthesize the three aspects of system reconfiguration (i.e., structure, dynamics, and management). An in-depth literature review that allowed identifying this gap is provided in chapter 6.
- There is a lack of reconfiguration methods integrating system reconfiguration data and bridging both the engineering and the operational phases. A specific literature review concerning this gap is provided in chapter 7.

3.1.3. Gaps between industrial practice and research

The comparison between the state of the art and the industrial needs allows identifying the gaps that need to be filled to support system reconfiguration.

This thesis focuses on the development of reconfiguration supports. As discussed earlier, capturing reconfiguration data is crucial for system reconfiguration as it is linked to configuration definition and selection. System reconfiguration relies on data from both system design and operations. From our industrial audit, we identified a lack of standard definitions or models describing system reconfiguration data that can be further used in the development of reconfiguration supports. Based on our literature review, we identified three aspects of system reconfiguration that need to be considered jointly (i.e., structure, dynamics, and management). However, up to now, these aspects are considered separately. Therefore, we argue that a data model or an ontology synthesizing these three aspects of system reconfiguration is needed.

Furthermore, developing reconfiguration supports and functionalities requires methods allowing resource allocation and re-allocation to support dynamic reconfiguration during operations. Given that the reconfiguration data is related to design and operations, the required reconfiguration process or mechanism should address combining data from both phases. The required method should support system reconfiguration by defining, assessing, and selecting relevant system configurations. However, based on the previously identified research gaps, we argue that, to support system reconfiguration, there is a need for a method integrating data related to observations (from operations) and system design (from engineering).

In light of the industrial audit and the discussion of the literature, we argue that in order to support the development of system reconfiguration functionalities, we need to:

- Identify and synthesize the required data for system reconfiguration;
- Propose an adequate reconfiguration method integrating data related to observations (from operations) and system design (from engineering).

Based on these requirements and the thesis objective we formulated the research questions presented in the next section.

3.1.4. Research questions

This research work has been initiated to address the following overarching objective:

Research objective: to support system reconfiguration using a model-based approach.

In order to achieve this research objective, we needed to identify the existing approaches, methods and tools that are used to support system reconfiguration. It was also crucial to capture the needs and shortages in the industrial landscape as they allow us to identify characteristics and the use of the required support. Hence,

the overall research objective has been refined into the following research question that was guiding the research and that allowed for an initial understanding and identification of the encountered challenges:

RQ1: What are the existing approaches, methods and tools for system reconfiguration, and what are their related difficulties?

In order to address this question, we conducted a descriptive study. From the descriptive study, we identified existing approaches, methods, and tools for system reconfiguration. Moreover, the descriptive study allowed us to identify the main issues and the research gaps that we have discussed in the previous sections (3.1.1 and 3.1.2). This question is further addressed in chapter 5.

Based on the results of the descriptive study, this research focused on the difficulty concerning the development of a system reconfiguration support. In particular, we identified two specific challenges related to this difficulty including reconfiguration data identification and integration. The body of research lacks data models that describe the reconfiguration data in an overall manner. Moreover, the existing methods do not integrate data from both observations (from operations) and system design models (from engineering).

Given both the industrial challenges and the research gaps resulting from the descriptive study, we formulated two research questions.

RQ2: What is the required data for system reconfiguration, and how can it be formalized?

RQ3: How can we integrate the pre-identified data to support system reconfiguration?

Figure 3-1 shows the steps that contributed to the proposal of a model-based approach for system reconfiguration during operations and their link to the research questions previously defined.

Starting from the research objective, in order to support system reconfiguration using a model-based approach, we needed to identify the existing approaches, processes, methods and tools. This need has led to our first research question (RQ1). To address this question, as a first step, we conducted an empirical study and a literature review as recommended by the Design Research Methodology (DRM) (Blessing and Chakrabarti, 2009). This step has resulted in a descriptive study which underlined the industrial challenges and the research gaps. The contribution resulted from this step is the descriptive study detailed in chapter 5 entitled “A Descriptive Study of the System Reconfiguration Process Over System Life Cycle Stages”.

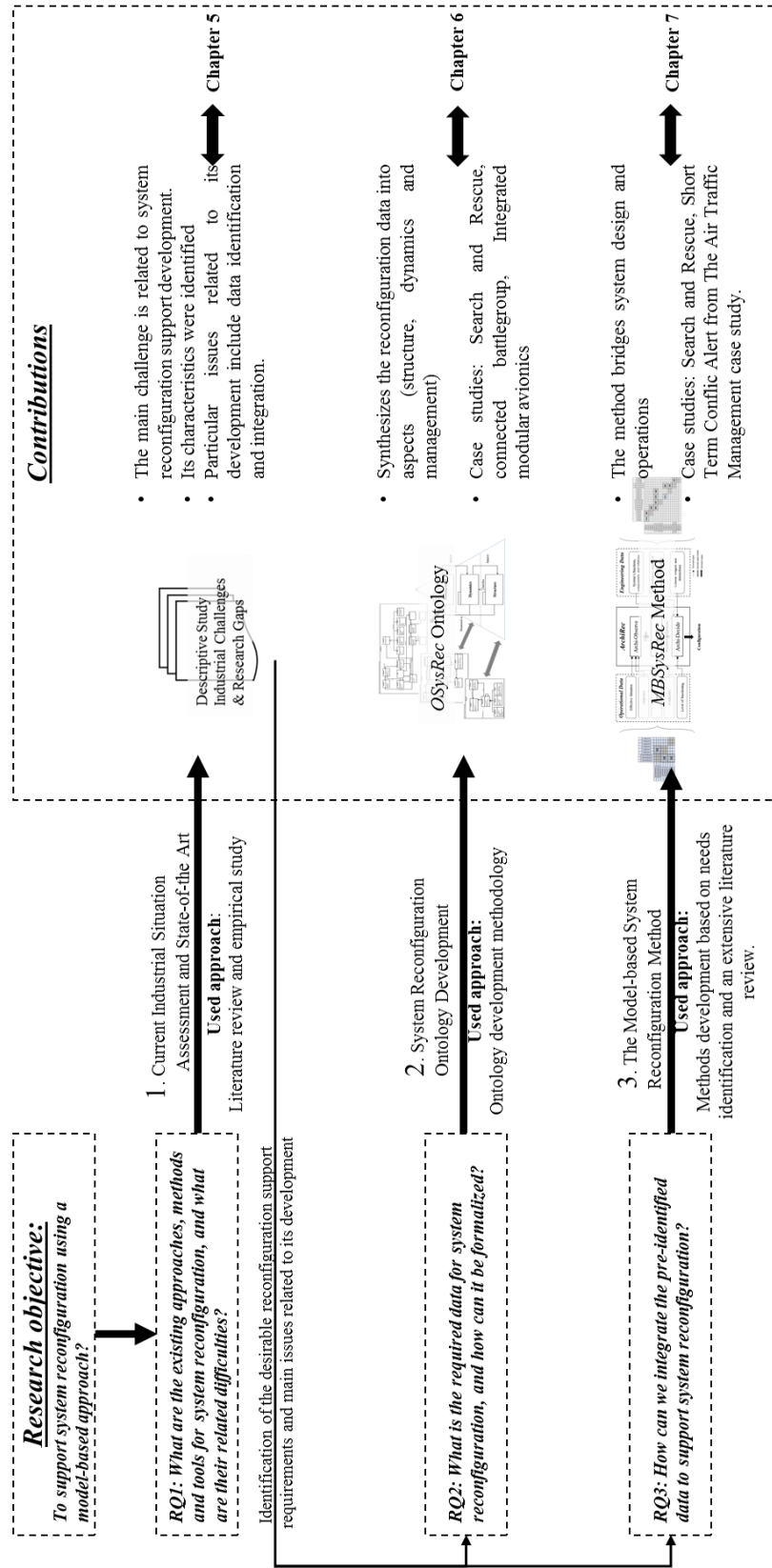


Figure 3-1: Summary of the thesis steps, used approaches, and their link to research questions and contributions

Based on the descriptive study, we identified a challenge regarding data identification and formalization. Quiet naturally, this leads to the second research question (RQ2). To address this question, the second step of this research focused on developing a system reconfiguration ontology. The methodology used to build this ontology was adapted from Pinto et al. (Pinto and Martins, 2004). The main result of this step is the OSysRec ontology. This contribution is detailed in chapter 6, entitled “An overall Ontology for System Reconfiguration using Model-Based System Engineering”.

The descriptive study also highlighted a challenge concerning data integration to support system reconfiguration. Therefore, the question related to data integration (RQ3) is necessary to address. As a third step, this part of the PhD thesis focused on developing reconfiguration support that allows for data utilization and integration. To develop the system reconfiguration support, we devised an approach based on identifying the requirements of the needed support. An extensive literature review was also performed to identify existing system configuration/reconfiguration methods and tools. This step resulted in the proposal of a model-based method to support system reconfiguration during operations. We call this method MBSysRec. This contribution is detailed in chapter 7, entitled “A Model-Based Method for System Reconfiguration”

In order to provide an overview of the PhD thesis, we propose to summarize the different contributions (section 3.2) in terms of objectives and main results necessary to give a global understanding of this research.

3.2. Thesis Contributions

3.2.1. A Descriptive Study of the System Reconfiguration Process Over System Life Cycle Stages

3.2.1.1. Objective

The objective of the research is to propose a model-based approach to support system reconfiguration during operations. In order to clarify the research topic and to better formulate the research problem, we conducted a research clarification step and completed it with a descriptive study. Even though our focus is on reconfiguration during operation, other life cycle stages are worth studying as they can provide other perspectives for future development. In the descriptive study, we consider reconfiguration during operations as well as other life cycle stages, the design phase, in particular, as system configurations result from design

activities. The main objective of the descriptive study is to capture the shortcomings in the current industrial landscape and to identify the industrial needs and challenges.

3.2.1.2. Results

When considering the operational phase, experts expressed the need for generic system management capabilities. As an enabler for system management, reconfiguration in terms of resource allocation and re-allocation is highly required. In particular, experts emphasized the need to manage the functions and resources of the system jointly, allowing the system to achieve mission goals while considering the impact of the context (e.g., environment) on both the system and its mission (see Figure 3-2). The challenges related to system reconfiguration during operations were identified. The main challenges include data availability, accessibility, storage, and verification, modeling issues, contracting and certification, system and context taxonomy, and challenges related to the reconfiguration support development.

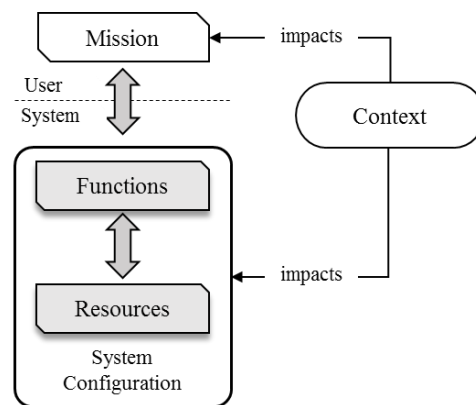


Figure 3-2: Management of functions and resources jointly

There is a need for reconfiguration support to define the resources needed to perform mission goals from a set of available resources (e.g., boats, operators, and cameras performing search and rescue missions). Specific challenges related to the development of reconfiguration support for the operational phase were also identified. First, identifying the data that are necessary for system reconfiguration. In particular, there is a lack of models that consider the reconfiguration process in an overall manner. Second, defining a reconfiguration method that uses and integrates data concerning system structure and functions as well as

observations. These two challenges are addressed in this research. These challenges have been discussed with regard to the existing literature and two research gaps were identified. The body of research lacks data models that synthesizes the data that are related to the different aspects of system reconfiguration. Moreover, the existing methods do not integrate data about both observations and system design model. Therefore, in the rest of this research we focus on addressing these research gaps.

3.2.2. An overall Ontology for System Reconfiguration using Model-Based System Engineering

3.2.2.1. Objective

Supporting system reconfiguration in industry remains challenging (Qasim, Jankovic, et al., 2019), especially in terms of data identification as it is directly linked to the development of industrial supports. The identification of reconfiguration data is difficult because reconfiguration is a multi-domain problem. Three key aspects can be identified when considering system reconfiguration from the systems engineering perspective: structure, dynamics and management. The structural aspect defines the resources constituting the system of interest, and the functions which they provide, as well as connections between the them. The dynamic aspect handles system behavior and evolution, based on events, conditions and transitions. Finally, the management aspect is concerned with optimizing the existing resources regarding the considered context and the mission. All these aspects, together, are important to cope with (internal or external) changes, impacting the system through its lifecycle towards effective system management (INCOSE Systems Engineering Handbook V4, 2015; ISO/IEC/IEEE:15288, 2015). Thus, there is a need to provide a unified representation of all aspects of the reconfiguration process.

Model-Based Systems Engineering (MBSE) is a model-based approach (Madni and Sievers, 2017) that supports complex system design by developing approaches and models (Chrisp and Richard, 2007), while reducing cost and time via using and reusing models (Wymore, 1993). In MBSE, ontology development represents an important approach used to support complex system design. The interest of developing ontologies gained prominence as they allow for the formalization of domain data by providing the specific vocabularies for a domain and relations among them. Facing the multi-aspect nature of reconfiguration requires addressing the overall process. Therefore, in chapter 6, we propose a system reconfiguration ontology, which we call *OSysRec*, synthesizing the key aspects necessary to support the system reconfiguration and that have been considered independently up to now.

3.2.2.2. Results

The System Reconfiguration Ontology (*OSysRec*) is a foundational upper ontology that addresses the reconfiguration process across domains. The aim is to synthesize data into an overall ontology. The *OSysRec* ontology provides its intended users with a comprehensive conceptual framework that allows a systematic analysis of various problems and solutions concerning systems' evolutions.

The *OSysRec* ontology macroscopic view was formalized within the Cameo System Modeler tool using an interconnected package diagram (see *Figure 3-3*). These packages corresponding to the system reconfiguration aspects (i.e., structure, dynamics, and management) are visualized in a way to show their hierarchal nature. In this chapter, we only describe the data flow between the different aspects. We also provide the necessary overview of the ontology to allow the comprehension of the rest of this chapter.

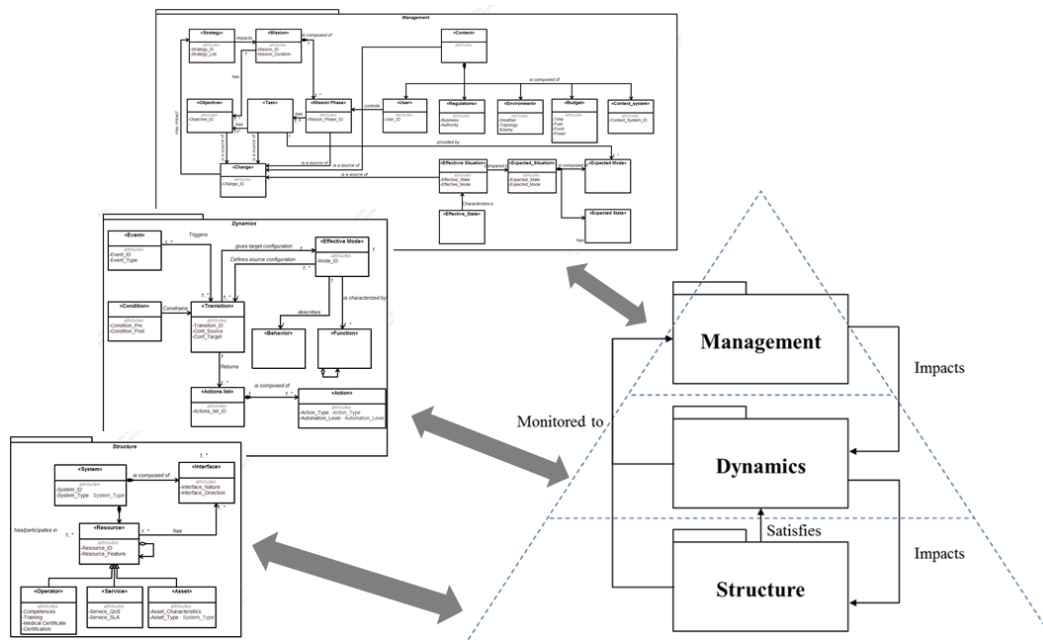


Figure 3-3: OSysRec macroscopic view and main packages

The structural part satisfies the dynamics required via the Configuration. Information about the structural elements and their related dynamics should be reported to the management level. These data flows are generated by health and usage monitoring systems and supervision mechanisms (ISO, 2012, 2015, 2018). The *Effective_Mode* and the *Effective_State* representing the dynamic and structural parts of this ontology

are used in the management part. Once a decision has been taken at the management level, this can impact the dynamic level and by consequence it impacts the structural level. This impact is translated as an *Event* at the dynamic level and an *Action* at the structural level.

Systems generally undergo reconfigurations to ensure operational effectiveness via the management of the available resources in the system in conjunction with missions and contexts evolutions. The most important concepts within the management aspect are *Objective*, *Context*, *Effective* and *Expected Situation*, as well as *Change*.

The mission, which is the general purpose to which all resources should be directed, has at least one *Objective*. For instance, in a maritime Search and Rescue (SAR) mission, one of the objectives is to be able to observe an object of 1 m² at 1 Km distance. Context is what characterizes the situation in which the system of interest is operated. Context is a composition of different categories, including Regulations, Environment, Time, Context System, and User. In the SAR example, a context can be stormy weather. When both the context and the objective are fixed, system reconfiguration can still happen in response to system failures. The failures can happen either because the resources are not delivering the required effects, or because they are not available. Therefore, the Effective Situation is characterized by Effective Mode and the Effective State, which reflects the correctness of the expected behavior and the health state of the system. The Effective Situation is compared to the Expected Situation, which is similarly composed of both the Expected Mode and an Expected State. For instance, being in a mode where only observation is required with a defective camera. Objectives and contexts evolutions, as well as a situation gap, are all considered sources of changes that may lead to reconfigurations. In a SAR example, the situation of observation with a defective camera can be corrected by engaging other available equipment.

Tasks of a mission are provided by the *Expected Modes*. One or more *Functions* characterize a *Mode* engaged at the dynamic level. These functions are provided by the system *Resources*, which are parts of the system at the structural level. At this level, the system is characterized by *Configurations*. For instance, in SAR, the detection task is provided by the observe mode, having observe and operate as functions. These two functions are provided by the camera and operator resources subsequently. In this specific case, the system is characterized by the observe configuration engaging the camera and the operator to satisfy the observe mode.

In chapter 6, the *OSysRec* ontology is illustrated using the SAR example. The ontology has been tested with many scenarios from different industrial case studies. From aerospace: the Integrated Modular Avionics (IMA) and Air traffic management have been used to instantiate the ontology. From the communication and military domains: the next-generation tactical radios incorporating innovative software-defined radio

technology, delegated fleet management for land forces systems (vehicles and payload), the future soldier system, and the connected battle group. In chapter 6, we detail the connected battle group case study and use it to illustrate the *OSysRec* ontology. IMA case study related scenarios are also illustrated.

3.2.3. A Model-based Method for System Reconfiguration

3.2.3.1. Objective

System Reconfiguration relies on different data that is related to system structure, dynamics, and management and synthesized in the *OSysRec* ontology. The most important concepts in the management aspect of the *OSysRec* ontology include *Objective*, *Context*, *Effective* and *Expected Situation*, and *Change*. For instance, in SAR missions, we may need to search for a victim during the night using a camera that is only functioning at 70%. In this context, reconfiguration is essential as life-saving relies mainly on how to manage the available resources and direct them towards the required purposes. Some of the identified challenges in proposing system reconfiguration supports within the current industrial practice in the company are 1) deployment context consideration, 2) missions and objectives evolutions, 3) sensor data integration. These concepts are linked to data regarding system architecture coming from the engineering phase (i.e., components, functions, and their relations), and data from observations (i.e., sensors data) reflecting the real level of functioning of the deployed system. Hence, chapter 7 aims to support system reconfiguration by proposing an approach bridging both phases, engineering, and operations.

3.2.3.2. Results

In chapter 7, we propose a model-based system reconfiguration method, which we call *MBSysRec*. This support integrates the information concerning system operational context, objective, and level of functioning that are essential for system reconfiguration.

Figure 3-4 synthesizes the proposed method by listing its main inputs, internal and external outputs, as well as its different phases. *MBSysRec* aims to link both the engineering and operational phases by integrating data coming from these two phases. Inputs from the engineering phase rely on the system data model in terms of functions, components, and their relations, as well as all information regarding the component

performance, criteria weights, and interactions. Inputs from the operational phase, on the other hand, are mainly related to sensor data reflecting information on the component's availability and level of functioning. In addition to sensor data, the system's operational context and the objective it intends to realize are also considered operational data inputs. All of this information is stored in matrices within a matrix system linking them. MBSysRec in its three phases (i.e., MBSysRec-Observe, MBSysRec-Orient, and MBSysRec-Decide) is implemented through three main functions within a Matlab program.

MBSysRec-Observe verifies continuously the context, objective, and the effective situation. The context and objective are entered by the user, while the effective situation is read directly from the matrices devoted to storing health and usage monitoring system data. When a change is detected, the instantiation information is updated, and the reconfiguration process is triggered by calling the MBSysRec-Orient block.

The **MBSysRec-Orient** function aims to identify a set of possible configurations, if found, by using information stored in the matrix system with regards to the context and the objective. MBSysRec-Orient searches the solution configurations in an updated set of available resources. The set of possible configurations is returned and entered in the MBSysRec-Decide function. For this function, in addition to matrix operations, we adopted the method proposed by Bryant et al. (Bryant et al., 2005) to generate system configurations.

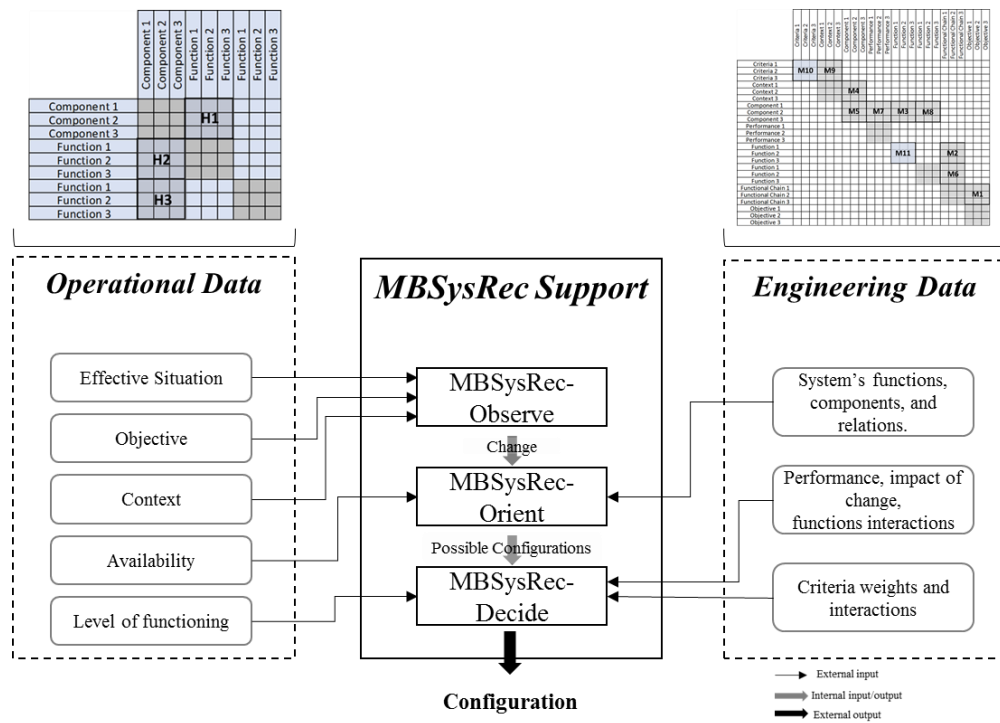


Figure 3-4: MBSysRec synthesis, inputs, outputs and steps

The *MBSysRec-Decide* function is a two-stage phase consisting of configurations evaluation and selection. In the decision support phase, we use both user-defined criteria, which can be found in the system data model, and two fixed criteria. The two fixed criteria that are, in our opinion, essential for system reconfiguration include 1) the overall configuration level of functioning, and 2) the impact of change. The configuration level of functioning gives the overall capacity that can be realized by a configuration. The impact of change describes the effort needed to deploy a configuration while considering the current configuration as a departure point. The configurations resulting from the orientation phase are evaluated based on these criteria, and finally, an overall satisfaction level is given using a multi-criteria decision-making technique. For the multi-criteria decision making, we adopted the Choquet integral method (Labreuche, 2011a) and its two indices, the Shapely index and interaction index, to account for the weight of the different criteria and their interactions. The selection step consists of 1) filtering the resulting configurations regards to the user-defined satisfaction threshold, and 2) choosing the configuration with the maximum satisfaction level among the threshold satisfying configurations.

4. Research Methodology

This chapter details the approach used in this thesis to conduct the research and validate its results. The context of this research is a Participation Action Research (PAR). This chapter gives an overview of the design research methodology and presents the strategies employed for validation. The research stages are also detailed based on the Design Research Methodology (DRM).

Research approaches constitute the plans and the procedures performed throughout the research process. These approaches involve general assumptions, data collection methods, analysis, and interpretation (Creswell, 2014). There are several research approaches to pursue a research project, namely, quantitative, qualitative, and mixed methods (Grover, 2015). Qualitative research deals with non-numerical data analysis and interpretation in order to understand concepts and develop new hypothesis. Various strategies, including case studies, narrative research, and action research, are employed within the qualitative research (Creswell, 2014).

Action research seeks a transformative change in the research environment by taking action and doing research simultaneously (Ottosson, 2003). This process implies a critical analysis to link observations and actions. In action research, the researcher is involved in the environment so that together as a team, they will work on solving problems and contribute to science. Participation action research (PAR) is a type of action research where the researcher is in direct contact with practitioners to improve strategies, practices, and knowledge of the environments within which they practice. Ottosson (Ottosson, 2003) describes the set-up of the PAR research in which the researcher is involved in both a scientific environment and the field under study.

The opportunity of industrial collaboration prompted us to adopt the PAR approach. The researcher is simultaneously an engineer in large international aerospace, space, ground transportation, defense, and security company (Thales) and a team member in the industrial engineering department (LGI).

This thesis concerns developing a model-based approach to support system reconfiguration during operations. The overall aim is to ensure the operational effectiveness of systems in conjunction with evolutions in their context and objectives. As the present thesis lies in the design research area, we studied the research methodologies used within this area. In the domain of engineering design, different research methodologies exist (Blessing and Chakrabarti, 2009; Cross et al., 1991; Eckert et al., 2003). In this thesis, we adopted the Design Research Methodology proposed by Blessing and Chakrabarti (Blessing and Chakrabarti, 2009) as it provides a set of supporting methods and guidelines that can be used as a framework for doing design research. Moreover, the different steps that we have conducted are linked and contribute altogether to achieve an overall research objective. In the following, we give an overview of the design research methodology, and we also explain how we used this methodology to address the previously defined research questions.

4.1. The Design Research methodology

Blessing and Chakrabarti (Blessing and Chakrabarti, 2009) developed the Design Research Methodology (DRM) to support research in design. The DRM methodological framework defines four stages to conduct a research project: Research Clarification (RC), Descriptive Study I (DS I), Prescriptive Study (PS), and Descriptive Study II (DS II). Iterations between the DRM stages are allowed. The different stages, their basic means, and main outcomes are shown in Figure 4-1.

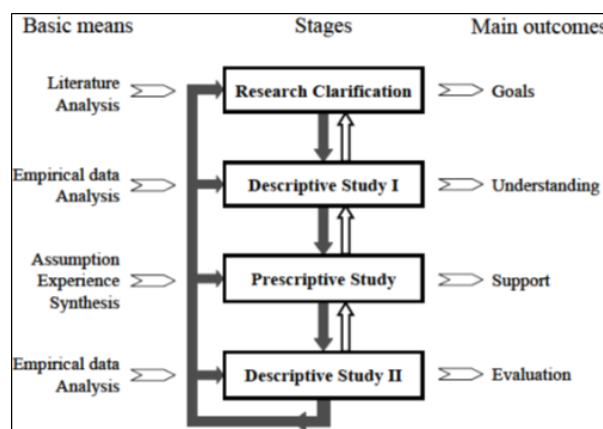


Figure 4-1: DRM methodological framework, from (Blessing and Chakrabarti, 2009)

The Research Clarification stage aims at formulating the research objectives and defining clearly the current and the target situations. During this stage, researchers also define different evaluation criteria to help measure the closeness to the target situation. Researchers come with these outcomes by mainly examining the literature. In the Descriptive Study I stage, researchers review the literature for more influencing factors. They also conduct empirical studies to describe the existing situation. The aim is to have a clear vision of both the current situation and state of the art to determine which factors should be addressed in priority. During the Prescriptive Study, researchers use their understanding of the current situation to build their theory on how addressing the improvement factors already identified in the previous stage. This stage involves the development of support methods and tools and also gives insights on how to perform their assessment. The closeness to the target situation after the support development is assessed and investigated within the Descriptive Study II stage. During this stage, researchers conduct empirical studies to evaluate the applicability, usability, and usefulness of developed support.

Research Clarification	Descriptive Study I	Prescriptive Study	Descriptive Study II
1. Review-based	→ Comprehensive		
2. Review-based	→ Comprehensive	→ Initial	
3. Review-based	→ Review-based	→ Comprehensive	→ Initial
4. Review-based	→ Review-based	→ Review-based Initial/ Comprehensive	→ Comprehensive
5. Review-based	→ Comprehensive	→ Comprehensive	→ Initial
6. Review-based	→ Review-based	→ Comprehensive	→ Comprehensive
7. Review-based	→ Comprehensive	→ Comprehensive	→ Comprehensive

Figure 4-2: Types of design research in DRM, from (Blessing and Chakrabarti, 2009)

Several types of design research exist within the DRM framework (see *Figure 4-2*). Depending on the research project, the researchers proceed differently with the DRM framework stages. Three dimensions are identified to describe the degree of depth of the conducted study: 1) review-based study, 2) comprehensive study, and 3) initial study. The review-based study relies mainly on the existing literature examination. The comprehensive study involves both the literature review and results production, such as empirical study or supports development. The study wherein the researcher conducts only the first few steps of any of the

stages is called an initial study. The initial step aims to show the consequence of the results. This thesis falls into design research type 5

4.2. Research Quality

Research quality is crucial to establish trust in engineering research results (Gerriker et al., 2017). In engineering research, results quality involves verifying and validating achieved results regarding the research objective or purpose (Seepersad et al., 2006). This thesis represents action-based research approach where the objective is to improve the current engineering practices in the industry while contributing to the state of the art. Thus, results verification and validation should address both the scientific quality and the suitability of the research results for industrial deployment (Björk and Ottosson, 2007; Sargeant, 2012).

For this thesis, we adopted the approach proposed by Björk and Ottosson (Björk and Ottosson, 2007) to ensure scientific quality. For academia, scientific quality involves research validity, reliability, and credibility (Björk and Ottosson, 2007). In few words, scientific quality should: 1) ensure the existence of the research results (credibility), 2) ensure their quality and novelty (reliability), 3) their accuracy in assessing the research questions. To establish a value of the research results from an industrial point of view, in this thesis, we adopt the evaluation recommended by Blessing and Chakrabarti (Blessing and Chakrabarti, 2009) within the DRM framework. The recommended criteria are 1) usability, 2) applicability, and 3) usefulness. Usability refers to the ease with which the results can be used for the intended task. Applicability ensures that the results have the intended effect on the current situation. Usefulness ensures that the direct effects of integrating the research results lead to an improvement in a high-level success factor.

Details on strategies used to ensure research quality in both science and industry are discussed individually for each contribution in section 8.1.

4.3. The Research Methodology used in the Present Thesis

In order to address the research questions, we propose an adaptation of the DRM framework (*Figure 4-3*). The adapted methodology involved four stages: Research Clarification (RC), Descriptive Study I (DS I), Prescriptive Study (PS), and Descriptive Study II (DS II). As shown in *Figure 4-3*, the research methodology

is not sequential; multiple iterations between the different stages are established. This thesis is considered Type 5 research as explained in section 4.1 (see *Figure 4-2*).

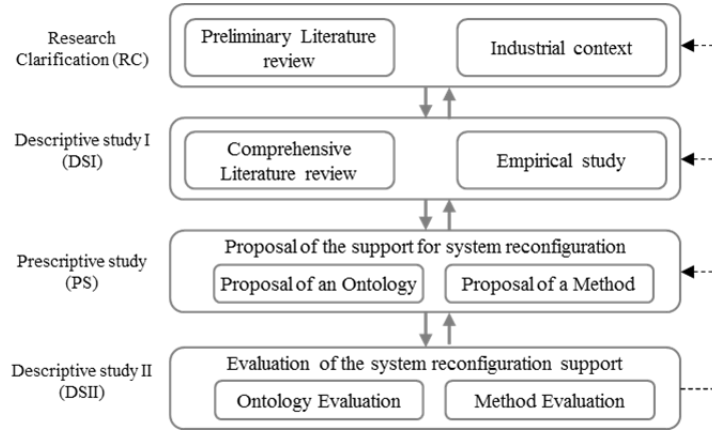


Figure 4-3: The adapted research methodology inspired by DRM (whose main stages are shown on the left side of the figure)

Figure 4-4 shows the link between the DRM stages and the corresponding thesis chapters, and the respective research questions.

This research started by conducting a research clarification with a preliminary literature review and the analysis of the industrial context. In the beginning of the thesis research, the research problem was formalised as a support for system reconfiguration using a model-based approach. Therefore, the purpose was to have a better understanding of the research context.

Next, we conducted a descriptive study (DS I) to understand how to support system reconfiguration during operation using a model-based approach. This study resulted in capturing the industrial landscape and highlighting the research gaps. Chapter 5 also contributed by defining reconfiguration in systems engineering. Therefore, research clarification (RC), descriptive study (DS I), and initial steps from the prescriptive study (PS) were addressed in chapter 5. Chapter 5 provides an answer to RQ1.

The development of an overall systems reconfiguration ontology is the second step of this research. The aim was to capture the data that is necessary for system reconfiguration and formalize it. Chapter 6 covers mainly the prescriptive stage within the DRM framework. In chapter 6, we also went further in the evaluation process and proposed a validation with a case study covering mainly the descriptive study II. Chapter 6 provides an answer to RQ2.

		RQ 1: What are the existing reconfiguration approaches and their related difficulties?	RQ 2: what data? How to formalize it?	RQ 3: How to integrate the data?
		Chapter 5: Descriptive study	Chapter 6: Ontology	Chapter 7: Method
DRM Stages	RC	x	x	x
	DS I	x		
	PS	x	x	x
	DS II		x	x

Figure 4-4: Research progress in terms of RQs and thesis chapters presented across DRM stages. Crosses indicate that the works included the chapters (column) address the specified DRM stage (row).

To support system reconfiguration using a model-based approach, we developed a reconfiguration method that aims at integrating reconfiguration data. In particular, the objective is to integrate the system's level of functioning as well as its operational context and objectives to ensure an effective system reconfiguration. By this proposal, we answered RQ3. Chapter 7, thus, covers the prescriptive study and the descriptive study II stages within the DRM framework.

The research work was performed at the Systems Engineering department within the technical directorate at Thales. The research progress in terms of hypothesis and results was discussed with transversal experts working on different systems engineering and system architecture issues. Being in an industrial set up, researchers could collect data of different nature and from different sources. Systems' technical specifications, as well as operational concept documents, were used to extract knowledge on the systems' functional and structural architectures. Interaction with experts has participated in increasing technical knowledge about the systems developed within Thales and the current engineering practices. For each stage of our research, the data used is detailed in corresponding chapters.

5. A Descriptive Study of the System Reconfiguration Process Over System Life Cycle Stages

This chapter is constituted from two papers published under the following references:

Qasim, L., Jankovic, M., Olaru, S. and Garnier, J.-L. (2019), “Model-Based System Reconfiguration: A Descriptive Study of Current Industrial Challenges”, Complex Systems Design & Management, Springer International Publishing, Cham, pp. 97–108.

Qasim, L., Hein, A.M., Jankovic, M., Olaru, S. and Garnier, J.-L. (2019), “Towards a reconfiguration framework for systems engineering integrating use phase data”, Proceedings of ICED 2019, the 22nd International Conference on Engineering Design: Responsible Design for Our Future, Delft, Netherlands, 05.-08.08. 2019.

Abstract. *System Reconfiguration is essential in complex systems management, as it is an enabler of system flexibility and adaptability with regard to system evolutions. System evolutions have to be managed to ensure system effectiveness and efficiency through its whole lifecycle, particularly when it comes to complex systems that take years of development and dozens of years of usage. In this context, system reconfiguration ensures system operation and increases reliability, availability, maintainability, testability, safety, and reuse of system entities and technologies.*

To understand how we can support system reconfiguration using a model-based approach, we conducted a field study in large international aerospace, space, ground transportation, defense, and security company to capture the reconfiguration process over the different system life cycle stages. A qualitative data analysis was performed to capture the strengths and shortages in the current industrial landscape. These challenges have been discussed regarding the existing literature on system reconfiguration.

Although our focus is on system reconfiguration during operations, in the descriptive study we consider the reconfiguration process over the different life cycle stages. Several industrial challenges have been identified for two main life cycle stages, i.e., design (engineering) and operations. The study revealed that the development of reconfiguration support is a major challenge as this process relies on heterogeneous data related to both system design and operation. In particular, two issues related to this challenge were identified. First, identifying and synthesizing data that is necessary for system reconfiguration. Second, integrating these data in order to develop the reconfiguration support. These findings are interesting as they give insights into the future development of reconfiguration supports.

5.1. Introduction

Large industrial companies developing products, systems, or solutions in critical domains (e.g., aerospace) are often concerned with system reconfiguration as it allows for the management of evolving systems. Systems are likely to evolve over their lifecycles to account for changing operational contexts, technology evolution, and environments (e.g., aircraft, ships, trains, etc.). The primary objective of systems engineering is to design systems and manage their evolutions over their lifecycle (INCOSE Systems Engineering Handbook V4, 2015; ISO/IEC/IEEE:15288, 2015). For an effective systems design and management, systems engineering formalizes the required processes including, technical, management, agreement, and organizational processes (ISO/IEC/IEEE:15288, 2015). System configuration management, addressed by the management aspect of systems engineering, is a critical activity ensuring effective management of an evolving system during its lifecycle. System configuration is defined in systems engineering as a set of elements that compose the system in terms of hardware devices, software, interfaces, human profiles, and processes. System configuration can be characterized according to economic, environmental, legal, operational, behavioral, structural, and social aspects necessary to demonstrate a capability. Any change in these aspects can lead to "System Reconfiguration" in order to maintain operational effectiveness. Consequently, "System Reconfiguration" is defined in this research by subsequent changes of the system configurations with the objective of maintaining or adapting (increasing or decreasing) the capabilities provided by the system.

This research has been conducted within Thales, which is a large international aerospace, space, ground transportation, defense, and security company. Thales is developing a variety of system types, including equipment such as radars, platforms such as ships, distributed systems such as Air Traffic Control and Management, and systems of systems such as the connected battlegroup. Thales systems can be used in different contexts (e.g., a radar can be used in military fighters or ships). Therefore, there is a specific need to develop a generic approach to suit the diversity of system types and business domains within the company. The generic approach should be as abstract as possible so that its instances can solve reconfiguration problems of any system type and in any domain.

A current trend in Systems Engineering is the Model-Based Systems Engineering (MBSE). Engineering systems with increasing scale and complexity has caused engineers to rethink system development. In MBSE, the system model is placed at the center of the system development process (Wymore, 1993). Similarly, one can consider MBSE as a possible venue to support system reconfiguration as it can tackle the abstraction issue.

To understand how we can support system reconfiguration using a model-based approach, we conducted a descriptive study based on a field study (empirical study) within Thales and an extensive literature review. The aim is to identify the challenges that the model-based approach should address. Although our focus is on reconfiguration during operations, in this study, we consider the different life cycle stages, particularly design, to get an overall understanding of the process and get insights on reconfiguration techniques that can be used for possible future developments. A synthesis of the relevant literature review is presented in section 5.2. Section 5.3 describes the industrial context and preliminary observations. The followed methodology is presented in Section 5.4, and the results are reported in Section 5.5. Results from the qualitative study have been confronted with the reviewed literature and discussed in Section 5.6. This section also discusses possible future work. Section 5.7 concludes this chapter by highlighting the most important difficulties and the research gaps that will be addressed in this research work.

5.2. Literature Review

The overall literature review was provided in chapter 2. Based on the synthesis of the literature provided in section 2.3, we identified the gaps in the body of knowledge. In this section we underline the research gaps that have been identified:

- The literature lacks a definition of system reconfiguration that considers the three aspects necessary to master the reconfiguration process from the systems engineering perspective. These aspects include the structural aspect related to components and interfaces, the dynamic aspect concerning the processes that govern the system, and the management aspect related to optimizing the existing resources with regard to the considered context and the mission.
- Up to now, the three aspects of system reconfiguration (i.e., structure, dynamics, and management), were considered independently. The analysis dedicated to the identification of this gap is provided in chapter 6.
- Existing reconfiguration methods do not integrate data regarding systems design and operations. Specific literature review concerning this gap is provided in chapter 7.
- There is a lack of a generic model. Existing models and methods are issued from studies that focused on a particular application (e.g., robots, space systems, microgrids, etc.).

5.3. Industrial Context

The field study (empirical study) was performed in the context of large international company developing systems (aerospace, space, ground transportation, defense and security) for military and civil applications. We were involved within the system department at Thales technical directorate. The technical directorate is a transversal team who aims to support the different business entities within the company with expertise in the system's engineering domain. Moreover, this team plays a crucial role in enhancing engineering practices and the reference engineering processes followed in the company.

A major objective of the team is to define the future technologies that are needed to enhance the operational effectiveness of the systems developed across the company business units. The operational effectiveness is a term used to describe the system performance, availability (reliability, maintainability, and supportability), process efficiency (system operations, maintenance, and logistics support), and system life cycle cost. In this context, the technical directorate is concerned with introducing system management capabilities in Thales systems. System management is defined within Thales as the processes, the methods, and tools aiming to satisfy and maximize system capabilities through efficient management of resources and their organization. System management aims to support Command and Control (C2) activities with systemic approach. It comprises four features including configuration, monitoring, supervision and reconfiguration. These features are essential to improve operational availability as well as a reduction in ownership cost through reducing support expenses. Moreover, beside system monitoring and supervision, the (Re)Configuration process is a key part as jointly they allow an effective and efficient system management. At the operational level, system configuration management supports operations through reporting the current system inventory to the Command and Control (C2) and allowing the C2 to launch reconfigurations. For this purpose, the technical directorate has initiated the Health and Usage Monitoring System (HUMS) working group in order to define the processes, methods and tools that are needed to guide and support product or system designers and customer service managers with the specification of system management features.

Preliminary observations revealed that there is a specific need to propose a model-based approach for system reconfiguration, as Thales develops a variety of system types (i.e., equipment, platforms, distributed, and systems of systems) in different domains (e.g., space or navy). A generic approach is paramount to ensure system flexibility and adaptability. The generic approach should be as abstract as possible so that its instances can solve reconfiguration problems of any system type and in any domain.

In the following, we present the methodology that we followed to conduct our field study (empirical study). Then, we discuss the results as well as the issues and the stakes that we identified based on our qualitative data analysis.

5.4. Methodology

The research methodology is based on the exploration of the current literature via the examination of papers and existing norms and standards, supported by data collection. Blessing and Chakrabarti (Blessing and Chakrabarti, 2009) highlight the importance of observation and data gathering (descriptive study) in order to identify the current challenges within an industrial context. In this descriptive study, we considered system reconfiguration over the different life cycle stages, though, our focus is to support system reconfiguration during operations. Studying reconfiguration in an overall manner allows to identify problems and solutions that need to be considered in the future development of the desirable model-based approach. To better understand the (re)configuration process in a multi-domain industrial context, we conducted an empirical study consisting of: 1) observing meetings where (re)configuration issues have been discussed, 2) interviewing experts in (re)configuration (directors, architects, experts, etc.) and 3) analyzing internal reference documents (Figure 5-1).

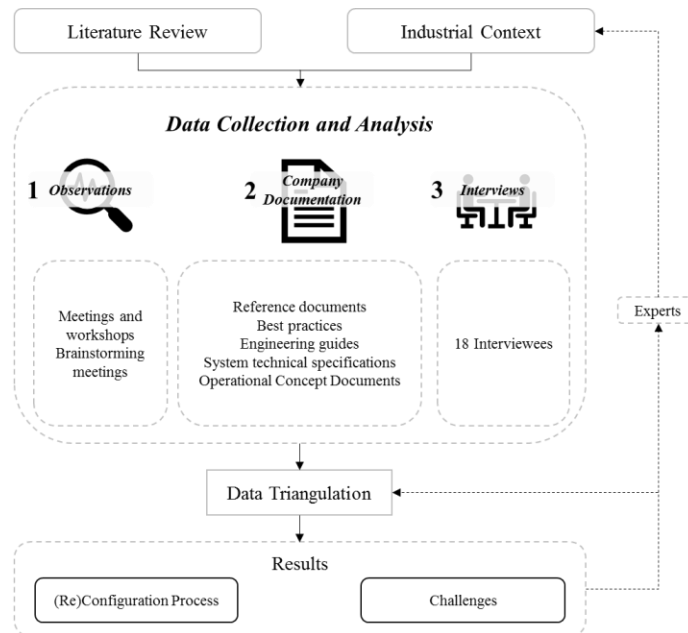


Figure 5-1: Empirical study methodology

This data was used to understand and model the existing (re)configuration process “as-is” and to identify its related difficulties. The modeled process as well as the identified shortages have been discussed with the stakeholders to validate our understanding of the process. Gathered data was analyzed using qualitative data analysis techniques such as coding and writing analysis memos. The research steering committee involving 7 experts from industry validated and confirmed the observed process and its related difficulties.

5.4.1. Observations

Over a period of six months, we observed different meetings, workshops and brainstorming sessions involving the Health and Usage Monitoring Systems (HUMS) working group at Thales. The HUMS working group aims at developing new engineering guide, methods and tools to support product/system designers and customer service managers with the specification of Health and Usage Monitoring System (HUMS).

The observed meetings treated different issues related to the HUMS design including main HUMS systems functions and architecture definitions; in particular, the (re)configuration function. Observed data concerning re(configuration) was collected and analyzed according to the project scope, necessary reconfiguration data to be collected, type of (re)configuration (i.e. on-board and on-ground reconfiguration), sensors technologies and possible reconfiguration alternatives.

In addition to the notes that taken during the HUMS meetings, we analyzed the working documents, meeting minutes and mail conversations in order to understand the difficulties and the challenges. Data analysis was mainly based on coding and categorizing the encountered issues and problems. Regular progress meetings with experts were conducted in order to discuss and review observations.

5.4.2. Documentation Exploration

In order to understand how we can support system reconfiguration during operations using a model-based approach, we studied documentation of different nature. In order to model the “as-is” (re)configuration engineering process in use, we analyzed and investigated different company documentations including: 1) Internal reference process documents, 2) engineering guides and best practices related to the (re)configuration process. To identify the various technical implementations of reconfiguration supports

within Thales systems, we studied the systems and products technical specification and operational concept documents. Data analysis consisted of highlighting the strengths and shortages in the current engineering practices. These remarks were coded and categorized as previously described to identify the related issues and challenges.

5.4.3. Experts interviewing

Interviewing is key in engineering-design research as it allows researchers to gain understanding of different design phenomena and to identify and explicate the needs for new tools and methods (Summers and Eckert, 2013). Although the effort needed to conduct and analyze the interviews is not trivial, this method allows interviewees to express naturally their concern without influencing them. The selected interviewees have different levels of involvement in system management and the configuration/reconfiguration process. Since the considered company deals with different types of systems in various operational contexts, the identified persons were classified into two categories: 1) people working in transversal activities and 2) subject matter experts. 18 experts with different roles have been interviewed (Table 5-1).

Table 5-1: Experts interviewed

Role	Number
System architects	10
Expert on engineering processes focusing on Norms & Standards	1
Expert on engineering processes focusing on Integration, verification, validation,	1
Expert on engineering processes focusing on reliability, availability, maintainability	1
Manager of future services	1
Modelling and Artificial intelligence expert	1
Research & Technology Engineers	2
Software development Engineer	1

To ensure objectivity, the interview has been designed according to a structured list of questions. A set of 16 questions has been prepared (Table 5-2). The objective of the interviews is to find out the definitions related to system management (including System Configuration and System Reconfiguration) in terms of artefacts and processes that govern system or product life cycle activities, such as Systems Engineering,

manufacturing and in-service operations. Moreover, questions about the different methods and tools used in the configuration management and System Reconfiguration processes have been included in our survey. The interviews had the following sequence: we started the interview with an introduction to our research work. This was followed by some warm up questions and the main body of the interview. The interview has been concluded with a cool off and wrap up. The interviews were different in terms of duration, ranging from 46 minutes up to 126 minutes with an average of 76 minutes.

Table 5-2: Questions list used for interviewing domain experts

Number	Question
0	Introduction by the interviewee: Name, Business unit, department, Role?
1	How do you define the configuration/Reconfiguration process (depending on your context) in the company?
2	What are the benefits expected by improving the reconfiguration process in the company?
3	What is the reference process of configuration/Reconfiguration in the company?
4	How is the configuration/reconfiguration process currently implemented in the company?
5	What are the different activities of the configuration/reconfiguration process in the company?
6	What are the methods used to implement configuration/reconfiguration in the company?
7	What are the tools used in the configuration/reconfiguration process in the company?
8	Is there any document describing the configuration/reconfiguration process followed in the company?
9	Who is involved in the configuration/reconfiguration process in terms of teams, services, roles?
10	What are the difficulties/problems/challenges which are faced when considering configuration/reconfiguration?
11	What are the data used in the reconfiguration process?
12	At which moment, the data are collected?
13	At which moment, the data are used for reconfiguration purposes?
14	According to the company, in which stage of the system life cycle, the reconfiguration process takes place? Is critical?
15	According to today's practices, how can you test/validate/evaluate/ select a configuration?
16	In your working context, what is the nature of Reconfiguration (cold/hot/warm)?

To avoid neglecting important facts and information audio-recording was preferred to note taking. The audio files then have been used to fully transcribe the interviews. Following the transcription, the coding process comes where we went through all textual data in a systematic way. Declarations of the interviewees were clustered into 16 categories, covering configuration and reconfiguration during different life cycle phases, and also a short biography and additional remarks. In order to avoid biases, each interviewee has been asked to verify and validate what has been discussed. The coding has been also done and validated by 3 additional

persons that have not participated in the interviewing process in order to avoid biases in our understanding. This has been done also for category identification. Category identification has been afterwards presented to a larger committee (10 people including Director of research in Systems Engineering and several experts in systems engineering) who have confirmed the observations coming from the field.

5.5. Results

During the field study (empirical study), we were interested to understand the (re)configuration process through the different lifecycle stages. This has allowed us to define the different activities and the actors involved in this process through the different life cycles. Moreover, studying the overall reconfiguration process allowed us to identify the different types of data and their use for reconfiguration purposes through the lifecycle stages. With a systematic approach, system (re)configuration is considered a means to improve the performance and quality of service provided by the system all over its life cycles. As for the perimeter considered in the company, two life cycle phases are judged critical: design and operations. During the engineering phase (Design), designers provide the standard configurations that are necessary to achieve customers' defined capabilities. While in the operational phase, command and control teams rely on the awareness of the system state concerning the health of the available resources to allow its management.

In this section, we, first, describe the engineering configuration process and discuss its related issues. Next, we discuss reconfiguration during operations and its related issue. Related difficulties and challenges have been identified and confirmed with stakeholders of this process.

5.5.1. The engineering (re)configuration Process

In the current practices and system engineering process in the company, the system technical specification document should define the different configurations of the product or the solution that is delivered to the customer. The configuration process described in this section details the steps for how to construct this document. The configuration process (represented by the black part in Figure 5-2) starts by understanding the customers' needs. For this, system architects translate the Operational Concept Document (OCD), which contains use case scenarios, into requirements in order to define standard configurations. The objective is that the system has one of these configurations at any time during its life cycle. On the other hand, domain

models (business, logistics, manufacturing, regulations, etc. provided by stakeholders) are used to define internal or external triggers, events or changes, yielding in a new system configuration. The defined configurations (system modes) and the corresponding triggering events are then described using States & Modes diagrams. The behavior of the system is then described for the different configurations. After this, a verification for the completeness and correctness of configurations is done. Once the obtained configurations are in agreement with the requirements, the configurations will be validated. Finally, the System Specification Document will be issued. If the validated configurations are different from the pre-defined ones, an engineering change request (ECR) will be issued in order to change the requirements for facilitating new product development.

5.5.2. Difficulties within the in-use engineering (re)configuration process

As discussed in the previous section, system architects and designers use operational scenarios to capture the essential needs of the end user when using the system. They define the standard configurations contractually with the end user company based on workshops and brainstorming sessions. However, this activity is found to be insufficient because of the lack of imagination, information as well as uncertainty that is related to the design process. This leads to limited creativity in defining configurations to make available for the operations phase. In this context, the major difficulty that has been raised by actors in this process is "how to pre-define the standard configurations and the triggers without relying completely on hints from end users who will be exploiting the system?"

To allow generating relevant configurations, a reliable knowledge base containing technical and operations data is needed. The elements composing the system need to be managed in terms of resources to satisfy system functions. Moreover, incompleteness and incorrectness of configurations can be a real issue, as there is no generic and automatic tool to support configuration generation. This activity is not trivial and demands a considerable engineering effort, especially when considering systems with increasing levels of complexity. To consider large number of configurations implies incompleteness of models.

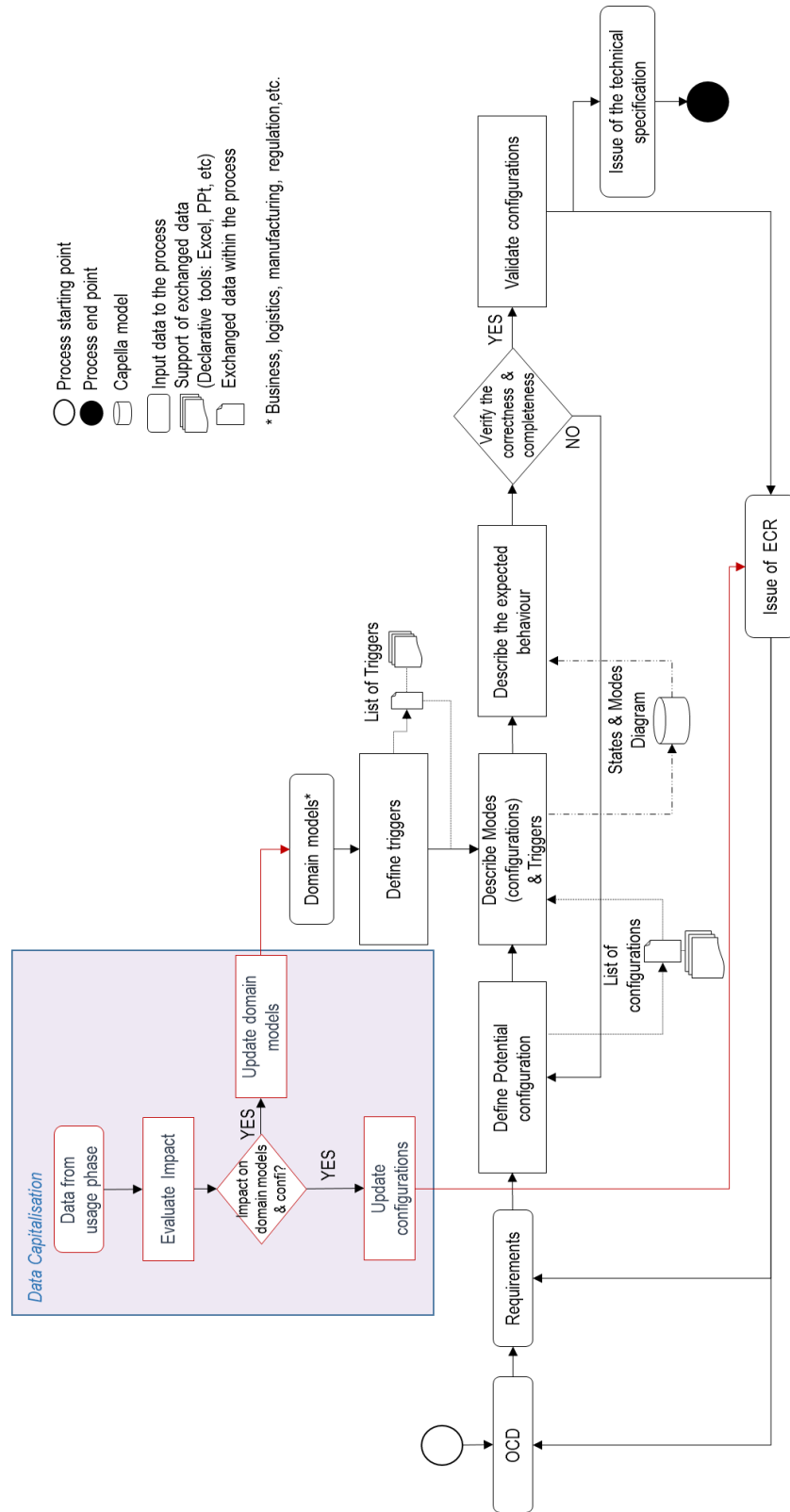


Figure 5-2: (re)configuration process with data capitalization shortage during design

The field (empirical) study revealed that the configuration process fails to integrate the use phase data (data from operations) that is essential for reconfiguration. The integration of such data is important to allow their use by system analysts to define system failures, peaks, needs, or simply abnormal behaviors that the system might need to handle. The impact of these behaviors is then evaluated, and a change of configurations and domain models is requested. The new configurations are then made available for operations. We highlighted this shortage and named it “data capitalization” in the (re)configuration process (Red part in *Figure 5-2*). A major difficulty lies in updating existing system configurations (re-organizing) when a change is detected. More specifically, it is crucial to understand how a change of one component in existing system configurations can propagate to other components in the system model

5.5.3. Reconfiguration as an enabler for system management during operations

System management relies on different functions and services including mission support, security management, maintenance support, simulations and tests and among all configuration management. Several tasks are needed in the system configuration management, e.g. configuration definition, configuration survey through health and usage monitoring, and reconfiguration. In an operational context, allocation and re-allocation of resources are highly desirable for: 1) maximizing the system capabilities, 2) optimizing the system resources regarding the defined capabilities, 3) maximizing the survivability and the availability of the system, 4) maximizing the number of possible missions that can be achieved by the system, and 5) allowing for mission evolution. Thus, experts highlighted the importance of managing jointly the systems functions and resources allowing the system to achieve mission goals (see *Figure 5-3*). It is worth noting that both the system mission and its constituting elements are impacted by the evolution of the operational context (e.g., Environment of the deployed system).

In general, system reconfiguration occurs to cope with changes related to: 1) mission and its phases progress, 2) system faults, 3) warning alarm regarding future possible failures, 4) collaboration between systems, especially in the context of Systems of systems management, and 5) user demanding a configuration change.

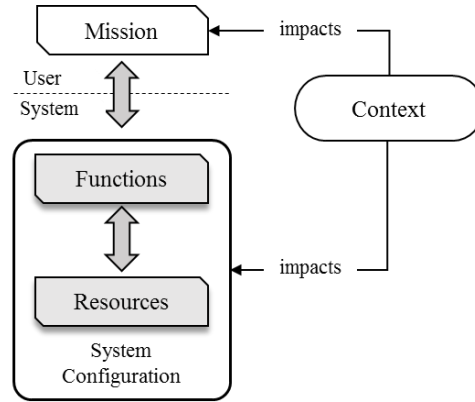


Figure 5-3: Management of functions and resources jointly to achieve missions during operations

In this specific context, there is a need to support command and control of the system operations at various levels of the system organization. Supervision and reconfiguration are the main tools and means that are used by C2 operators to allow system management. Across the company business units, various forms of health monitoring, supervision and reconfigurations tools are already implemented. However, their implementation is based on ad-hoc definitions. Moreover, these implementations are not supported by means to account for life cycle evolutions. Since the field study was conducted in a company covering multiple industrial sectors, there is a specific need for an adapted reconfiguration support that is based on a generic definition so as to suit the diverse systems developed across the company business units (see Figure 5-4).

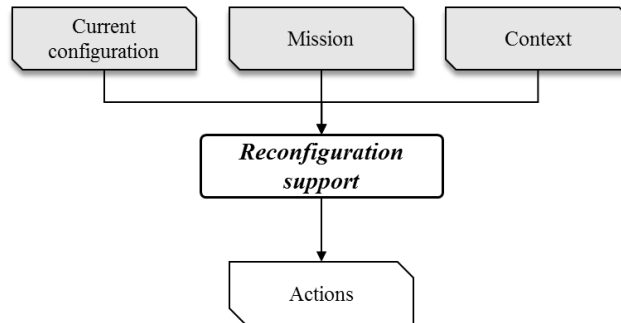


Figure 5-4: Reconfiguration support desired for systems operations

The desirable reconfiguration support should be able to evaluate the situation based on the observed context and mission goals, find relevant configurations, and finally apply them by defining the corresponding actions. Reconfiguration actions could act directly on the system to instantiate the selected configuration.

5.5.4. Challenges related to system reconfiguration during operations

Several challenges pertaining to system reconfiguration were identified: challenges related to data, modeling issues, contracting and certification, system and context taxonomy, as well as difficulties related to development of system reconfiguration support. In the following paragraphs we discuss these issues.

- **Data related challenges**

The (re)configuration process relies fundamentally on data. Having correct and verified data is crucial. Data collection is considered as one of the primary challenges within the (re)configuration process.

- *Data availability and accessibility*

Data availability and accessibility can be a real issue at the technical level. For instance, in some applications, data collection cannot be possible due to harsh working environments; for example, fuel rod temperature measurement within a nuclear reactor in operation. In other cases, measured data cannot be transmitted directly; therefore, communication technologies (i.e. communication by RFID radio wave or satellite communication) are needed to give access to these data. However, when communicating data, one should consider all the measures to secure these data. For example, the information systems on military airplanes which contain all flight data is critical. Hacking this system during a mission would lead to a critical situation. When dealing with operational data, security and confidentiality become challenging. Consequently, secured data processing for critical and tactical applications (e.g. military systems) is essential.

- *Data shared across stakeholders*

In complex systems, different stakeholders and system actors are involved, such as: system designers, developers, customers and end-users. Usually at each stage of the system lifecycle, a specific team is responsible for the system and therefore, owns its related data. For instance, the developing company is the owner of the design models that are restrictively maintained within the company or even the business unit. The end-user is the one using the product or the system, and thus operational data, in particular, data that are related to usage and missions is considered confidential. Therefore, it could be difficult to collect data shared across the different stakeholders because of non-clearly defined interfaces and intellectual property.

- *Data storage*

Depending on the data processing strategy (i.e., in flight or on ground data processing), the quantity of data to be stored can be different. In flight data processing strategy, data processing based on a threshold

detection is needed in order to avoid high space, energy consumption or weight related to data storage and processing units. On ground data processing allows a more advanced data processing, however there is a need to store the flight data which may need a high storage capacity leading to increase weight. To avoid storing data continuously, front-end pre-processing can be implemented as a way to lessen data storage. As one can see, both strategies relies on thresholds detection and hence the problem of threshold definition can arise.

- *Uncertainty and data verification*

The reconfiguration process relies on data that can have different sources: 1) system technical data collected from sensors, and 2): external observation of the system from operators or maintenance teams. In both cases, the data collected have some degree of uncertainty that is related either to the measurement process itself or the approximations done when exact measurements are not possible. Uncertainty handling is challenging because it requires a knowledge of the system components and interactions.

- **Modeling issues**

System reconfiguration should be considered at different system levels ranging from the product level up to the solution level. At each hierarchical level, different levels of the system are involved. Thus, multi-level modeling techniques is needed to combine and jointly manage these levels.

- **Contracting and certification**

In the industry, developers and solution providers are usually concerned with contracting and certification. Contracts include information about usage profiles, configuration alternatives, operational contexts, quality of service, reliability, availability, safety, security, etc. The contracted configurations are tested, validated and certified in advance. However, this activity may last for a long time leading to penalties due to a schedule overrun. Consequently, challenges related to certifying, assessing and selecting the newly emerging configurations need to be considered.

- **System and context taxonomy**

Thales Group develops a very large set of system types. These systems do not stand alone and are often integrated into different operational contexts (e.g., platform or infrastructure). Consequently, the concerns related to these systems are extremely different. For example, reconfiguring a closed system, assembly or a platform; is needed to propose new configurations while integrating new technologies. When considering distributed system (re)configuration, there is a need to address the problem of connectivity between the system elements. Moreover, in a system of systems (SoS), the constituent systems are independent leading to emerging effects. In addition to that, the interfaces between these constituent systems of a SoS may

evolve. Consequently, the SoS (re)configuration must be considered as an agile capability. The methods and mechanisms for reconfiguration might be slightly different according to the system types. Indeed, this variation leads to complexity when trying to build the reconfiguration support progressively with the aim of overall generalization across systems and industry. A holistic method for reconfiguration is an essential challenge because it must be as abstract as possible to adapt to system and context taxonomies.

- **Challenges related to the reconfiguration support development**

The development of an adapted reconfiguration support depends on the data that is manipulated. The system data related to its structure and their characteristics should be defined in a standard way. Reconfiguration depends on feedback loops from operation, so there is a need to identify the data to be collected. Moreover, the reconfiguration process relies on a priori and a posteriori data that needs to be combined. Design models coming from the engineering phase can be seen as the system reference models. An effective view of the system during its operation can also be captured via health and usage monitoring systems. These two models (engineering model and operations model) should be compared. If a gap is detected a reconfiguration should be triggered. However, data combination is challenging and need to consider uncertainties related to data sources and the phase from which it originates. The reconfiguration support development relies on mechanisms to allow action modeling, relevant configuration and reconfiguration methods and tools need to be clearly identified to address any system context.

5.6. Discussion and Future Work

The validity of our field study (empirical study) was investigated using different strategies including the triangulation of the used methods, investigators, theory and data sources. The results of the descriptive study were reviewed and discussed within workshops involving experts and the research steering committee. The review workshops and meetings verified the internal validity of the descriptive study results regarding the considered perimeter within the company. As for external validity, the key issues identified were also discussed for the multiple industrial sectors (e.g., aerospace, navy, etc.). Moreover, validation interviews and workshop reviews involving international stakeholders (from both academia and industry) were organized to discuss the relevance of research directions and the identified research questions.

This field study aimed to investigate the reconfiguration process through the different life cycle stages, even though the focus is on supporting system reconfiguration during operations. Experts have been emphasizing the importance of reconfiguration over the different life cycle phases. However, our field study revealed

that two lifecycle stages are critical when considering reconfiguration. First, the engineering phase that aims at providing and delivering relevant configurations to customers and end-users. Second, the operational phase, where reconfiguration actually occurs in conjunction with evolutions in the system's missions and contexts, as well as its internal faults.

The challenges related to data collection, storage, and uncertainty are important when considering reconfiguration in both phases. However, a major challenge concerning development of reconfiguration support during operation was identified. More specifically, identifying reconfiguration data as well as their integration to support system reconfiguration are considered main issues challenging the development of the reconfiguration support. The data should address the reconfiguration process in its entirety. The methods used to integrate reconfiguration data should consider two sources of information that are different. While observation models are related to components and their behaviors, engineering models are governed by functions. Models representation should be done in a way that allows their comparison and combination. For instance, a car can provide a transport function; however, the effective ability of the car to provide this function is measured through other performance variables such as motor torque. Moreover, the measured ability deduced from observation models has uncertainties regarding the measurement process and its interpretation. In this context, it's crucial to identify reconfiguration data that are related to both system and its functions as well as its observation and their interpretations (analysis).

Several studies have discussed reconfiguration data (Ali et al., 2011; Bermejo-Alonso et al., 2011, 2016; Gogniat et al., 2013; Hernández et al., 2015; Krichen and Zalila, 2011; Meyer et al., 2013; OMG, 2010; Walsh et al., 2005, 2006; Witt et al., 2013). These data models were developed to support system reconfiguration in various domains: 1) software systems and computing domains (Gogniat et al., 2013; Meyer et al., 2013; Walsh et al., 2005, 2006), 2) embedded systems (Ali et al., 2011; Krichen and Zalila, 2011; OMG, 2010; Witt et al., 2013), 3) autonomous systems (Bermejo-Alonso et al., 2011, 2016; Hernández et al., 2015). However, at the system level, reconfiguration considers not only components or software elements but also the human profiles and a combination of these elements. Autonomy is an emerging capability of new generation systems; nevertheless, fully autonomous systems cannot be implemented in every domain (e.g., in military systems, a missile-firing cannot be an automatic action as it is related to responsibility and strategic issues). Therefore, reconfiguration data models should address domains involving manual, semi-automated, and fully automated actions. Furthermore, these models consider different facets of the reconfiguration problem. For instance, some data models consider reconfiguration data to be mainly related to managing systems via either capturing knowledge about the system structure (Bermejo-Alonso et al., 2011, 2016; Hernández et al., 2015; Meyer et al., 2013; Walsh et al., 2005, 2006), or the system dynamics in terms of behavior and functions. Other authors only concentrate

on describing of the system's structure and dynamics (Krichen and Zalila, 2011; OMG, 2010) to describe reconfiguration. However, all these aspects are important when considering reconfiguration at the system level; hence, a data model synthesizing these aspects is necessary.

Various configuration and reconfiguration methods exist in the literature. These methods vary depending on the reconfiguration aspect to be considered. Models of computations describing system behavior include dataflow, process networks and rendezvous, synchronous-reactive models, finite state machine, discrete-event models, modal models, continuous-time models, timed systems models, event-oriented model (Petra) (Lee and Seshia, 2017; Ptolemaeus, 2014). Another aspect considering functions allocation and re-allocation to the system's resources has also been studied in the literature (Albarello et al., 2012; Bryant et al., 2005; Lee and Lee, 2005; Liu and Liu, 2010). These methods rely on system architecture and configuration design to find the functions and their corresponding satisfying resources that are needed to perform a process. Our observations revealed that the reconfiguration process depends on data related to both system architecture and design and the real system behavior captured via health and usage monitoring systems. Therefore, an adapted method for configuration and reconfiguration integrating data from both observations and system design model is needed. The desirable method could allow the integration of the necessary data towards a reconfiguration support. Such a support can assist decisions within C2 activities via allocation and re-allocation of functions depending on the available resources and their accessible configurations.

Based on this discussion and considering reconfiguration during operations, this research focused on synthesizing data that are necessary for reconfiguration in an overall manner. Moreover, we extended the existing methods to integrate both data related to observations and system design. The reconfiguration support assists C2 activities via considering the system's level of functioning to allow system reconfiguration in conjunction with its evolving mission goals and contexts.

5.7. Conclusion

This chapter provided the results of the descriptive study that we have conducted to understand how we can support system reconfiguration during operations using a model-based approach. This descriptive study involved an empirical study (field study) and a comprehensive literature review. We conducted the field study in a large international company developing systems (aerospace, space, ground transportation, defense, and security) for military and civil applications.

Although our focus is on system reconfiguration during operations, in the descriptive study, we consider the reconfiguration process over the different life cycle stages. The overall consideration of the reconfiguration process, in particular engineering and operations, allowed us to identify challenges related to these phases, and that may impact reconfiguration during operations. Exploring the literature in an overall manner gave us insights on the possible use of methods originating from engineering to reconfiguration purposes during operations.

The field study revealed that there is a need for model-based support for system reconfiguration during operations. This desirable support should be able to evaluate the operational situation and generate relevant configurations. Two main challenges related to the development of such support were identified. These challenges include identification of reconfiguration data as well as mechanisms underlying system reconfiguration. These challenges were discussed with the existing literature on system reconfiguration. Therefore, two main research gaps were identified. First, the body of research lacks data models that describe the reconfiguration process in its entirety. Second, the existing methods do not integrate data from both observation (operations) and system design (engineering). Based on the major challenges and the corresponding research gaps, this research focuses on two main tasks. First, identifying and synthesizing data that is necessary for system reconfiguration. Second, investigating the possibility of extending the existing methods to integrate both data related to observations (from operations) and system design. In chapters 6 and 7, we present solutions for these two main issues.

6. An overall Ontology for System Reconfiguration using Model-Based System Engineering

This paper was submitted to IEEE Transactions on Systems, Man, and Cybernetics: Systems and is currently under minor revision.¹

Lara Qasim, Andreas-Makoto Hein, Soirn Olaru, Marija Jankovic and Jean-Luc Garnier

Abstract. *System Reconfiguration is essential in complex systems management, as it is an enabler of system flexibility and adaptability. It ensures system operation and increases reliability, availability, maintainability, testability, safety, and reuse of system entities and technologies. For the reconfiguration of a system in use, it is necessary to assess, in continuity, the system's state with regard to its context. Identifying data supporting system reconfiguration represents a major industrial challenge and is linked directly to the development of industrial reconfiguration tools. Reconfiguration tools are based on a data model, also called ontology, which represents key concepts of system reconfiguration and their relationships. A particular difficulty of developing the data model is the multi-domain nature of reconfiguration. Furthermore, it needs to address a considerable diversity of system types. Few publications propose an ontology supporting data identification and tool development for the entire process. Hence, in this chapter we propose to formalize the system reconfiguration process and propose an overarching ontology, which we call OSysRec. This ontology considers data at the management, dynamics, and structure level. The proposed ontology has been developed based upon expert knowledge and several industrial uses cases. The OSysRec ontology allowed a better understanding of the reconfiguration process, and hence it can be deployed for developing efficient and effective reconfiguration tools at the industrial scale. The ontology has been tested on an industrial case study to validate the proposed approach.*

¹ Case study in section 6.6 was extracted from the following paper "An Ontology for System Reconfiguration: Integrated Modular Avionics IMA Case Study", accepted in CSER 2020.

6.1. Introduction

Complex system engineering and development aims at designing systems that have considerable life-times and diversity of usages; sometimes even across several generations. Hence, it is necessary to consider the overall system life-cycle as well as its possible evolution with regard to changing operational contexts, technology evolution, and environments. System configuration management is a critical activity ensuring effective management of an evolving system during its lifecycle (INCOSE Systems Engineering Handbook V4, 2015; ISO/IEC/IEEE:15288, 2015).

In order to ensure the system's adequate operation with regard to its operational context and environment, systems engineering is discussing the notion of operational effectiveness. Operational effectiveness indicates the degree to which a system delivers defined capacities even though the system's environment and context might be evolving and changing. Hence, the system configuration depends on a diversity of aspects such as economic, environmental, legal, operational, behavioral, structural, and social that are critical to demonstrate a capability. Any change of these aspects can lead to "System Reconfiguration" to maintain its operational effectiveness. In accordance, "System Reconfiguration" is defined by the subsequent changes of system configurations with the objective of maintaining or adapting (increasing or decreasing) the capabilities provided by the system (Qasim, Jankovic, et al., 2019), (Qasim, Hein, et al., 2019). Optimizing configurations in terms of capabilities and available resources is needed to cope with environment or mission evolutions. In this context, reconfiguration allows for system performance, effectiveness and affordability improvement to ensure an increased reliability, availability, maintainability, testability, safety, reusability, and reuse of system entities and technologies (ISO/IEC/IEEE:15288, 2015).

Supporting system reconfiguration in industry still remains a challenge (Qasim, Jankovic, et al., 2019). Some of the identified challenges in proposing system reconfiguration tools within the current industrial practice of a large international aerospace, space, ground transportation, defense, and security company are: 1) to identify the data necessary for reconfiguration, 2) to provide a unified representation of all aspects of the reconfiguration process. To represent the data and arriving at a unified representation requires a model-based approach.

A model-based approach which has risen to prominence is Model-Based Systems Engineering (MBSE) (Madni and Sievers, 2017). MBSE aims at developing approaches and models supporting complex system design throughout different life cycle phases (Chrisp and Richard, 2007). The overall objective is to reduce the cost and time via using and reusing models (Wymore, 1993). An important approach used as a basis in MBSE is ontology development. Ontologies help formalizing domain knowledge by providing the specific

vocabularies for a domain and relations among them. They are also important to represent and manipulate complex models explicitly.

Several ontologies aiming at supporting system configuration and reconfiguration exist (Ali et al., 2011; Bermejo-Alonso et al., 2011, 2016; Gogniat et al., 2013; Hernández et al., 2015; Krichen and Zalila, 2011; Meyer et al., 2013; OMG, 2010; Walsh et al., 2005, 2006; Witt et al., 2013). However, to the best of our knowledge, no system reconfiguration ontology that takes into account the overall system reconfiguration process exists. Considering reconfiguration from the systems engineering perspective allows for categorizing the reconfiguration data into three classes according to the triggering aspects: structure, dynamics and management. The structural aspect defines the system of interest in terms of resources and the functions which they provide, as well as connections between the resources and functions. The Dynamic aspect deals with behavior and evolution, based on events, conditions and transitions, relying on causes and providing effects. The category of data which allows for optimizing the existing resources with regard to the considered context and the mission is referred to as the management aspect. The entire system development process relies not only on the development of the system-of-interest in terms of resources and interfaces (structural aspect), but also on the processes related to production, testing, operation, support (maintenance and logistics), and retirement (dynamic and management aspects). Thus, these aspects are important to ensure an effective system management by coping with (internal or external) uncertainties, impacting the system through its lifecycle (INCOSE Systems Engineering Handbook V4, 2015; ISO/IEC/IEEE:15288, 2015). Mastering these aspects supports system reconfiguration within the development process of reconfiguration tools. In order to address the overall system reconfiguration process, we propose a *SySRec* ontology, synthesizing structure, dynamics, and management aspects necessary to support the system reconfiguration process and that have been considered independently. For this purpose, a rigorous study of both the existing references and the industrial cases has been conducted to capture the knowledge about system reconfiguration and conceptualize it within a formal ontology. The remainder of this chapter is organized as the following: section 6.2 resumes the bibliographical review on the reconfiguration process in the different domains and the related ontologies or models used within these domains. Section 6.3 presents the methodology for ontology construction. Section 6.4 details the proposed ontology in terms of concepts used and formalization. Section 6.5 displays an industrial case study to illustrate and test the proposed ontology. Section 6.6 illustrates the application of the ontology to the Integrated Modular Avionics case study. The last section discusses the results and draws conclusions.

6.2. Literature review

In the aim of proposing an ontology for System Reconfiguration, we first investigate System Reconfiguration in different domains addressing this concept. Then we discuss System Reconfiguration from the system's engineering point of view (section 6.2.1). Section 6.2.2 introduces model-based systems engineering. Next, in section 6.2.3, we explore the relevant literature proposing System Reconfiguration ontologies and meta-models. Finally, section 6.2.4 provides a synthesis of the examined literature.

6.2.1. Reconfiguration in system's engineering

Runtime reconfiguration (or reconfiguration during system operation) has been addressed and researched by several scientific domains proposing related definitions of this concept. Mainly, reconfiguration has been discussed in the *control engineering* and *software systems and computing* domains. The *reconfigurable manufacturing systems* and the *embedded systems* domains have treated reconfiguration with less attention. Recently, runtime reconfiguration has been increasingly treated in the *autonomous systems* domain. In general, these domains refer to reconfiguration as a transition between system modes; functional or failure modes. In *embedded systems* and *control engineering*, configuration corresponds to a functional mode (Krichen and Zalila, 2011; Provan and Chen, 1999a). This mode corresponds to how components, elements, and processes are integrated and how they interact. Reconfiguration is referred to as a transition between systems modes when triggering events occur. In *reconfigurable manufacturing systems*, researchers define a configuration as the modules (software and hardware) of the manufacturing system and their on-site set up (Regulin et al., 2016). The change in structure, hardware, and software components to quickly adjust production capacity and functionality within a part family is referred to as reconfiguration (Alsafi and Vyatkin, 2010). Configuration of computing and information systems is defined as assembled components that deliver functions (Saxena et al., 2010). Saxena et al. (Saxena et al., 2010) define reconfiguration as swapping faulty components for working ones in the event of either component failure or an evolved mission goal. In *autonomous systems*, reconfiguration is an adaptation that can occur when the context (represented by parameters values) changes (Rodriguez et al., 2009). Such a transformation occurs if and only if the adaptation rules are satisfied.

In systems engineering, the principles of a system and system structure encapsulate system elements and relationships between them (INCOSE Systems Engineering Handbook V4, 2015; ISO/IEC/IEEE:15288,

2015). The aggregation of threads to the behavior (stimuli) of the system causing it to take specified actions and produce outputs represents a dynamic statement of what the system is required to do (INCOSE Systems Engineering Handbook V4, 2015; ISO/IEC/IEEE:15288, 2015). Thus, the description of transitions and actions that a system or its parts perform in response to events is referred to as the dynamic aspect. Moreover, during the operational phase, the system performance should be monitored such that the persistent stakeholders do not experience a breakdown in a service (INCOSE Systems Engineering Handbook V4, 2015; ISO/IEC/IEEE:15288, 2015). In systems engineering, changes are inevitable and their impact should be managed through the system lifecycle (INCOSE Systems Engineering Handbook V4, 2015; ISO/IEC/IEEE:15288, 2015). The impact of changes can cause system requirements to be added or deleted, increasing or decreasing system capabilities. From the previous discussion we conclude that the management aspect concerns controlling and optimizing the existing resources, in conjunction with the evolution in the operational context. More precisely, system configuration can be characterized with regards to economic, environmental, legal, operational, behavioral, structural and social aspects. The management of system configurations is essential to ensure an effective management of systems (INCOSE Systems Engineering Handbook V4, 2015; ISO/IEC/IEEE:15288, 2015). The definition of system configuration in systems engineering integrates different aspects of the processes discussed in previously mentioned domains. The provided definitions discuss system reconfiguration either in terms of transitions between modes (dynamic aspect), or changing software or hardware elements (structural aspect) in response to faults or changes in mission goals (management aspect). Therefore, these definitions do not address the overall concerns of system configuration as defined in systems engineering. Thus, Qasim et al. (Qasim, Hein, et al., 2019) have attempted to define system reconfiguration in Systems Engineering as the subsequent changes of the system configurations with the objective of maintaining or adapting (increasing or decreasing) the capabilities provided by the system.

Based on this analysis, we identified three main aspects that are highlighted when addressing system reconfiguration: structure, dynamics, and management. These three aspects are important for system management via reconfiguration in order to cope with internal and external uncertainties. Thus, capturing these data for the development of reconfiguration tools within the entire development process is crucial. The structural aspect defines the resources constituting the system, their functions, and connections between them. The mechanisms of transitions, and their causes and effects, are referred to as the Dynamic aspect. On the other hand, the management aspect concerns optimizing the existing resources with regard to the considered context and the mission.

6.2.2. Model-Based Systems Engineering (MBSE) and ontologies

The Model-Based Systems Engineering (MBSE) paradigm has emerged to harness systems complexity and increasing scales. MBSE can be described as the formalized application of modeling principles, methods, languages, and tools to the entire lifecycle of large, complex, interdisciplinary, sociotechnical systems (Ramos et al., 2012). Developing and managing systems using MBSE involves placing models at the center of these processes (Wymore, 1993).

A model is an abstraction of reality integrating diverse aspects from different disciplines and usually it is designated to a purpose (Aßmann et al., 2006; Madni and Sievers, 2017). The importance of models lies in capturing and stating domain knowledge to be shared among stakeholders, exploring system design and solutions, mastering complexity, giving insights into experimentation, deployment, and system operation in real time, gaining insights into the world in which systems are operating (Booch et al., 1999; Buede, 2009). Different types of models can be identified based on their use and purposes (e.g., structural, functional, behavioral, performance, etc.) (Dori, 2016). A model can be represented using different modeling formalisms, including formal and informal languages (Buede, 2009; Sanford, Friedenthal et al., 2014). An informal model is usually a text description or concept diagrams written in a standardized notation. A formal model relies on formalisms and their visual representation. Complex systems are generally modeled using formalisms and visual representations enriched with text descriptions. A diversity of MBSE modeling languages and tools exist (Dori, 2016; Roques et al., 2016; Sanford, Friedenthal et al., 2014) (OPL/OPM, SysML, UML, UPDM, Capella, Arkietct, Rhapsody).

In MBSE, ontologies and meta-models play a key role in the entire systems engineering process (Madni and Sievers, 2017). Madni et al. (Madni and Sievers, 2017) define ontologies as formal, explicit specifications of shared conceptualizations for specific domains. Ontologies help formalizing domain knowledge by providing the specific vocabularies for a domain and relations among them (Guarino et al., 2009). Ontologies form the basis for a common understanding among stakeholders and they allow to analyze domain knowledge. In this context, ontologies represent conceptual frameworks that allow to consider problems and solutions related to a specific domain (Chandrasekaran et al., 1999). In MBSE, tracing the abstract model elements into more specific models is achieved through meta-modeling and meta-models (Sanford, Friedenthal et al., 2014). A meta-model is the formal definition of the properties of a model, i.e., a model that specifies the abstract syntax used by a modeling language. Metamodeling aims to identify what can be expressed using a modeling language (Aßmann et al., 2006). Ontologies and meta-models are closely related as they both provide concepts and define relations and validity rules between them. However, ontologies is different from meta-models in that they do not describe systems, only domains (Aßmann et

al., 2006). This means that, ontologies describe the domain in a real world (problem space) and meta-models describe the systems to be designed (solution space)

6.2.3. System reconfiguration ontologies and meta-models

Ontologies and meta-models play an important role in the conceptualization and the formalization of the reconfiguration process. In the different domains mentioned previously in this section, several research studies discuss ontologies for the reconfiguration process. In *software systems and computing domain*, Meyer et al. (2013) proposed an approach for an automated knowledge-based IT management system which general architecture is based on MAPE-K components. Meyer's model takes into account systems components and connection between them. It also defines the events generated when the system performs state transitions. In another study, Gogniat et al. (2013) model the dynamic and partial reconfiguration system at a high level to allow technology-independent modeling. This model claim that reconfiguration is state-based (depending on context change) or strategy-based (defined by the user). Walsh et al. (2005, 2006) proposed a domain model for dynamic reconfiguration of component-based software systems. According to this model, reconfiguration occurs because of external interactions, internal behavior, dependencies and context of change.

To handle safe reconfiguration for *embedded systems*, Ali et al. (2011) proposed an agent-based approach. This approach relies on a meta-model which suggests that reconfiguration operations or scenarios (i.e. ordered sequence of operations) occur in response to a request. Problem diagnosis, improvement requests, prevention requests, and adaptation requests are types of reconfiguration requests. The Open Management Group (OMG, 2010) has specified a UML profile (MARTE profile) that adds capabilities to UML for model-driven development of Real-Time and Embedded Systems (RTES). In the MARTE profile, a mode identifies an operational segment within the system execution that is characterized by a given configuration. The transition between modes as a result of events is characterized as reconfiguration. In the same domain, Krichen and Zalila (2011) have proposed a model-based approach to specify the dynamic real-time embedded systems at a high level of abstraction. In their UML meta-model, they proposed a state machine of meta-modes and transitions. Moreover, this model describes the structured components and connections between them. For the description of RTES systems behavior in depth, Witt et al. (2013) developed a SysML profile based on the ideas of state analysis. A state in this SysML profile is more widely used than standard control theory; it can refer to various concepts (the health state, the operational state, the responsiveness, or other things). The model claims that everything we care about to meet mission objectives

can be “completely characterized” as knowledge of state and its behavior as well as transitions between the states.

To support the domain of *autonomous system* by a conceptual framework, Bermejo-Alonso et al. (2011) have developed an ontology, which contains both generic and domain-specific concepts for autonomous systems description and engineering. Hernández et al. (2015) presented the TOMASys meta-model which lies at the core of meta-controllers operation for autonomous systems. TOMASys has been developed to specify the functional model of the system used by the meta-controller at runtime. In further studies, the authors use an ontological approach for autonomous systems engineering (Bermejo-Alonso et al., 2016). Models proposed by these researchers include two kinds of elements: elements that capture the instantaneous state of the system at runtime, and elements that capture the knowledge about the design of the control system and properties of its components.

6.2.4. Synthesis and research gap

In our research we aim to propose an ontology for system reconfiguration that synthesizes the three aspects which we have previously identified, and upon which relies the reconfiguration process from the system’s engineering perspective: structure, dynamics, and management. Hence, we propose to analyze and classify the existing meta-models and ontologies with respect to these aspects. In the analysis task, we used the definitions of the concepts included in the existing meta-models and ontologies, their contexts of use and their objectives to classify them into the three aspects previously discussed. The classification is done via a comparison with regards to our definitions of these three aspects. In the following, we present the analysis of meta-models and ontologies in the previously discussed domains.

In the *software systems and computing* domain, Meyer et al. (2013) present a management system which relies on the system model, the system event model, and the service agreement model. The system model describes the *structure* of the managed system in terms of components, attributes and relationships. The system event model describes the simple events that are generated when the system performs state transition. Along with the service agreement model, the concepts in the system event model refer to the *management* aspect with regards to our definition. Gogniat et al. (2013) use a model to describe at a high level the dynamic and partial reconfigurations of systems. The authors use state and state change to describe the *dynamic* reconfiguration. The authors also describe the user and the context of use as *management* elements upon which the reconfiguration process depends. In another study, Walsh et al. (2005, 2006) present an ontology describing the types of changes. The types of changes include user driven change, structural change, domain

imposed change, internal behavior. As these changes refer to reasons why the reconfiguration occurs, we, therefore, classified them in the *management* aspect. In their model, Walsh et al. (2005, 2006) link the internal changes to the *structural* part represented by components, communication paths and connections.

Ontologies and meta-models in the *embedded systems* domain address mainly the *dynamic* aspect and the *structural* and *management* aspect depending on the application. Ali et al. (2011) propose an agent-based approach to handle safe reconfigurations of control systems. In their model, the authors include the causes of the fault (hardware misbehavior, software misbehavior, actor misbehavior, environment cause), all these elements correspond to concepts that we need to consider for system management, therefore, we classified them into the *management* aspect. The reconfiguration model contains concepts such as reconfiguration operation, initial and final modes, strategy of actions application (automatic, manual and hybrid) and a typology of reconfiguration actions. Since these concepts describe the dynamic operation of reconfiguration, we then classified them into the *dynamic* aspect. The MARTE profile (OMG, 2010) describes the behavior of the system in terms of modes, transition and triggers. These concepts describe the reconfiguration process in accordance with our definition of the *dynamic* aspect. The system configuration, which includes the set of active system elements or their functional parameters, refers to our definition of the *structural* aspect. The high-level model proposed by Krichen et al (2011) links *dynamic* reconfiguration process, represented by mode, transition, event, condition and action, to the *structural* configuration of the system in terms of components, connections and interaction ports. To describe the system behavior, Witt et al. (2013) use concepts such as states, state machines, constrains, transitions and effects. This information corresponds to the *dynamic* aspect as they describe transitions and their causes and effects. The only element from this model that can be linked to the *management* aspect is the current state, as we compare it to the constraints and goals to perform actions.

Bermejo-Alonso et al. (2011, 2016) and Hernandez et al. (2015) have presented models and an ontology for the *autonomous systems* domain. The resulting models and ontology have been used to develop a functional model that links the ontology to the system architecture. As the authors refer to components, their configurations and interconnections when describing the system model, we classified these concepts into the *structural* aspect. In addition to that, the authors included other concepts such as the instantaneous state, functional state, objectives and reconfiguration actions. These concepts correspond to our definition of the *management* aspect as they are used to compare the state of the system to deduce reconfiguration actions. The reconfiguration actions in this ontology refer to requests rather than actions, as they are provided to the lower control loop in a meta-control architecture. This classification is illustrated in *Table 6-1*. The structural aspect has been developed mainly in Meyer et al. (2013), and less developed in Bermejo-Alonso et al. (2011, 2016), Hernandez et al. (2015), Krichen et al. (2011), and Walsh et al. (2005, 2006). In Ali et al. (2011) ,

Gogniat et al. (2013), and Wlah et al. (2005, 2006), the management-related aspect has been treated by introducing exhaustive associated concepts. On the other hand, Meyer et al. (2013) refer to management in their model but, the work does not cover it in a consistent manner. While Ali et al. (2011) and Krichen and Zalila (2011) explain in details the dynamic aspect of the reconfiguration process within their respective models, others although mentioning this aspect in their papers (Bermejo-Alonso et al. (2011, 2016), Hernández et al. (2015) , Walsh et al. (2005, 2006), remain fuzzy, with no clear associated concepts.

Analysis of the system reconfiguration literature highlights that no model or ontology takes into account the main three aspects altogether: structure, management and dynamic (see Table 6-1). However, in the industry and from the System's Engineering perspective, an overall model covering all three aspects is needed for the management of complex systems via reconfiguration. In order to address this research gap, we propose to extend system reconfiguration ontologies integrating all three aspects of system reconfiguration in order to support effective and efficient management of systems while in use.

Table 6-1: Meta-models/ontologies for reconfiguration classification

Paper	Structural aspect	Management aspect	Dynamic aspect	Modeling language
Meyer et al. (2013)	x	x		OWL/API
Gogniat et al. (2013)		x	x	UML
Walsh et al. (2005, 2006)	x	x		UML
Ali et al. (2011)		x	x	UML
MARTE OMG (2010)	x		x	UML
Krichen et al. (2011)	x		x	UML
Witt et al. (2013)		x	x	SysML
Bermejo-Alonso et al. (2011)	x	x		UML
Bermejo-Alonso et al. (2016)	x	x		UML
Hérmendez et al. (2015)	x	x		UML

6.3. Methodology

Ontologies are a commonly used approach for domain knowledge construction (Ramos et al., 2012) and formalization. Ontology is agreed to be a shared conceptualization (Henderson-Sellers, 2011) and focus on the description and conceptualization of things (Aßmann et al., 2006). The categorization of ontologies depend on the knowledge they include and their application (Gómez-Pérez, 1999),(Šercar, 2002). The generic or common sense ontologies, also called the upper or foundational ontologies, aim at capturing general knowledge about the world, providing basic notions and concepts for things (Noy and Hafner, 1997), (Pirlein, Th and Studer, 2000). An upper ontology is limited to concepts that are meta, generic, abstract and philosophical, and therefore are general enough to address, at a high level, a broad range of domain areas (Aßmann et al., 2006). The object-oriented model is a way of representing ontologies through classes, properties of the classes and instances. Classes represent the domain concepts, and the association relations represent their interactions (Gruber and Özsu, 2009). Ontologies have been already used to describe different domains as presented in (Ebrahimipour and Yacout, 2015; Liang et al., 2011; Obitko and Mərík, 2002; Razmerita, 2011; Wan et al., 2018). The methodology used to define and evaluate the system reconfiguration ontology is that proposed by Pinto et al. (2004). We have adopted this methodology because it describes ontologies building from scratch and proposes techniques, guidelines and methods to help in the construction task. This approach proposed several steps in building ontologies: specification, conceptualization, formalization, implementation and maintenance. This chapter is mainly covering steps from specification to implementation, and is completing these steps with evaluation which includes both verification and validation tasks.

6.3.1. Ontology specification

Ontology specification consists of identifying both the purpose and the scope of the ontology (Pinto and Martins, 2004). The *OSysRec* ontology is a foundational upper ontology which aims to describe the reconfiguration process across domains through the use of concepts and relations for explicit model representation. The intended use of the *OSysRec* ontology is to provide a comprehensive conceptual framework allowing engineers to study various problems and solutions related to dynamic evolution in a systematic manner.

6.3.2. Ontology conceptualization, formalization, and implementation

Pinto et al. (2004) recommend to acquire knowledge by either using elicitation techniques on domain experts or by referring to relevant bibliography. A mixture of both techniques has been used to build a variety of ontologies in different domains (Bonjour and Micaëlli, 2010; Hildebrandt et al., 2018; Maleki et al., 2017). To construct our ontology, we used a mixture of top-down and bottom-up methods. This methodology is based upon the exploration of the current literature on reconfiguration and model-based systems engineering via the examination of scientific papers and existing norms and standards (top-down). The literature provides definitions for reconfiguration concepts from different domains. At the second phase, domain expert interviews and analysis of case studies from the industry have been conducted conjointly as a bottom-up method. The domain experts recommend which concepts need to be included based upon their knowledge of the industrial practices and the operational context. The total number of interviews was 18 in the specific case of our study. The selected interviewees have different levels of involvement in system management and the configuration/reconfiguration process. Since the considered company deals with different types of systems in various operational contexts, the identified persons were classified into two categories: 1) people working in transversal activities and 2) subject matter experts. The involved experts include: 10 system architects, an expert on engineering processes focusing on norms and standards, an expert on engineering processes focusing on integration, verification, validation, qualification and testing (IVVQT), an expert on engineering processes focusing on reliability, availability, maintainability and testability (RAMT), a manager of future services, a modeling and artificial intelligence expert, 2 research & technology engineers, and a software development engineer.

To ensure objectivity, we have designed the interview according to a structured list including 16 questions. The interviews aim to capture the concepts and definitions related to system management (including System Configuration and System Reconfiguration). More precisely, we asked questions about the different data, methods and tools used in the System Reconfiguration process. These questions have been used to extract the ontological elements and understand their nature, source and use within the reconfiguration process. The sequence of the interviews included: an introduction to our research work, warm up questions and main body of the interview, then a cool off and wrap up conclusion. The interviews durations ranged from 46 minutes up to 126 minutes with an average of 76 minutes. During the interviews, audio-recording was preferred to note taking to avoid neglecting important facts and information. Data analysis started by transcribing the audio files. Then, the coding process comes where we went through the declarations of interviewees in a systematic way. As a result, the declarations were clustered into 16 categories, covering configuration and reconfiguration during different life cycle phases, with a concentration on the in-service

operational phase, and also a short biography and additional remarks. The criteria that were used to accept the interview results for further analyses are reliability and trustworthiness. To ensure reliability and trustworthiness, we asked participants to verify and validate what has been discussed. In addition, the data analysis including coding has been validated by 3 additional persons that have not participated in the interviewing process. Results have been afterwards presented to a larger committee (10 people including Director of research in Systems Engineering and several experts in systems engineering) who have confirmed the observations coming from the field.

Several industrial case studies have been used within the knowledge elicitation process. For this purpose, five different case study workshops have been organized within a multi-national industry group which is the sponsor of this ontology development. This step is essential for capturing particularities of different system types, i.e., embedded system, platform, distributed systems, and systems of systems.

The different sources were analyzed to extract the ontological elements (key concepts), checking for commonalities, mismatches, and level of granularity. Subsequently, the relations between the concepts were developed from small examples within the case studies.

In the ontology development, two criteria were particularly important: clarity and exhaustiveness. These criteria were used to discuss the validity and the acceptance of the developed ontology. To address the clarity criteria, the authors discussed the extracted concepts to ensure that the adopted definitions are relevant to our domain and research context. In order to satisfy the exhaustiveness criteria, we categorized and classified the ontological elements into aspects. This strategy ensures that we cover three of the aspects of our ontology. A software engineering general- and specific-purpose language, SysML (Sanford, Friedenthal et al., 2014), was chosen to implement the ontology. Our decision was based on our review of ontologies for reconfiguration where UML and SysML have been widely used (Krichen and Zalila, 2011; OMG, 2010; Witt et al., 2013). SysML is relevant to our specific context because it supports system models, structural and functional, and traceability matrices that are essential for reconfiguration purposes. SysML allows the instantiation of our generic ontology to different systems in different domains. From the instantiated SysML model, traceability matrices can be extracted and further used within future reconfiguration tools. In addition to that, it is also important to use a language that is familiar to system architects and engineers to facilitate the industrial deployment of the ontology.

6.3.3. Ontology evaluation

Ontology evaluation aims, on the one hand, at testing the correctness of the theories and the assumptions underlying the ontology. On the other hand, it determines that the model's representation of the problem entity and the model's structure, logic, and mathematical and causal relationships are "reasonable" for the intended purpose (Sargent, 2013). Ontology evaluation involves several tasks: verification, validation, and user assessment (Pinto and Martins, 2004). The verification task is necessary to guarantee the correctness of the ontology according to the accepted understanding of the domain by specialized knowledge sources. Ontology evaluation is essential as it guarantees that what is built corresponds to the application requirements. Direct users should also assess the usefulness and usability of the ontology.

As discussed earlier in this section, the domain experts are the source of implicit knowledge. Interviewing experts is, thus, essential to verify the ontology in terms of correctness, completeness, and coherence. Five internal experts from the company were interviewed individually, and a workshop with ten internal and external experts was conducted. The verification sessions had the following sequence: presenting the ontology in terms of concepts and relations, discussing the definitions of the concepts with participants, testing the applicability of the ontology on operational scenarios specific to the contexts of the participants. The collected comments and remarks were used to update the ontology. The ontology has been presented to the steering committee, including the director of research in systems engineering and several experts in systems engineering within the company, who have confirmed the correctness and the coherence of the proposed ontology. During interviews and discussion sessions with engineers and system architects involved in the reconfiguration process, users have highlighted the importance of such a formalization of the reconfiguration process. According to them, this ontology allows engineers and system architects to evaluate the relevance of their designs and their reconfiguration tools.

In the literature, various techniques have been used for models and ontologies validation (Kösters et al., 2001; Sargent, 2013). The main tool used to validate the ontology was the Scenario-Based Test. A Scenario is used to describe the functionality and behavior of a system from a user-centered perspective (Ryser and Glinz, 1999). The validation task included testing the applicability of the ontology to different case studies. The applicability relies mainly on the ability of the ontology to describe the different scenarios completely. In other words, one should find all the necessary concepts to describe the tested scenario. To this end, different scenarios related to five case studies from the industry have been used. These case studies cover different domains from the industry; aerospace, communication, and military domains. From aerospace: the Integrated Modular Avionics (IMA) and Air traffic management have been used to instantiate the ontology. From the communication and military domains: the next-generation tactical radios incorporating innovative

software-defined radio technology, Delegated fleet management for land forces systems (vehicles and payload), the future soldier system, and the connected battle group. Moreover, different scenarios from the search and rescue system have also been used to test the validity of the ontology. The proposed ontology is detailed in the following section. In addition, the connected battlegroup case study is presented in order to illustrate and validate the proposed ontology.

6.4. OSYSREC: Ontology for systems reconfiguration

Although a considerable effort has been made to develop new ontologies and meta-models for systems reconfiguration, as highlighted by the literature review there is a lack of the ontology that considers all three aspects of system reconfiguration process conjointly. With regard to the identified gap and based on the methodology explained in section 6.3, we propose an extended ontology for systems reconfiguration (*OSysRec* ontology) integrating all three aspects. To illustrate this ontology, a maritime SAR (Search and Rescue) example has been used. The Search and Rescue system seeks to detect persons or vessels in distress; and recover them to a safe place in a maritime context. Section 6.4.1 is dedicated to the ontology main concepts description and definitions. Section 6.4.2 describes the main concepts and relations between them using a SysML formalism.

6.4.1. OSysRec conceptualization

The proposed ontology is developed with an operational perspective based on a literature survey and the analysis of use case scenarios. This extensive analysis has allowed us to extract the key concepts and their characteristics. The identified concepts have been analyzed and abstracted following their common meanings. **Table 6-2** gives an overview of the identified key concepts and their definitions. This table also gives the sources of these definitions which are adopted from the systems engineering domain. The column “concept source” indicates the reference of existing meta-models or ontologies if the concepts have been reused from them. The concepts which were identified from the bottom-up approach, consisting of expert interviews or case studies analysis, are indicated with “EI/CS” as a source. We have classified the identified concepts into three main categories representing the main three aspects of the reconfiguration process: aspects related to structure, management and dynamics.

Table 6-2: Key concepts in the OSysRec Ontology

Concept	Definition	Definition Source	Concept source
Context	Any information that can be used to characterize the situation of an entity. Elements for the description of this context information fall into five categories: individuality, activity, location, time and relations. Based on (ISO/IEC/IEEE:24765, 2017), the context of use includes users, tasks, equipment (hardware, software and materials), and the physical and social environments in which a system, product, or service is used.	(ISO/IEC/IEEE:24765, 2017),(Zimmermann et al., 2007)	(Gogniat et al., 2013)
Context_System	Systems that can be integrated in the SoS (System of systems) context and that can give new functionalities from their interaction.	(Silva and Batista, 2015)	EI/CS
Budget	Based on (ISO/IEC/IEEE:42010, 2011), budget helps understanding the setting of the system and evaluating its ability to ensure the continuity of the mission until the end. Budget is a vital aspect for context understanding and classification because most situations are constrained by the temporal dimension.	(Gross et al., 2001; ISO/IEC/IEEE:42010, 2011)	EI/CS
Environment	All elements in interaction with the system and its constituents (hardware, software or services). It includes the weather conditions, topology, etc.	(ISO/IEC/IEEE:42010, 2011; Zimmermann et al., 2007)	(Ali et al., 2011)
Regulations	Based on (ISO/IEC/IEEE:42010, 2011), regulatory constraints can dramatically orient developmental, technological, business, operational, organizational, political, economic, legal, regulatory, ecological and social influences.	(ISO/IEC/IEEE:42010, 2011)	(Meyer et al., 2013; Walsh et al., 2005, 2006)
Users	Individual or group that interacts with a system or takes benefit from a system during its utilization.	(ISO/IEC/IEEE:15288, 2015)	(Gogniat et al., 2013; Walsh et al., 2005, 2006)
Effective_State	A snapshot of the current state of the system and its components. Data allowing to derive the effective state are collected by means of health and usage monitoring systems.	(Buede, 2009),(ISO, 2018)	(Ali et al., 2011; Bermejo-Alonso et al., 2011, 2016; Gogniat et al., 2013; Hernández et al., 2015; Walsh et al., 2005, 2006; Witt et al., 2013)
Effective_Mode	The effective functional status of an entity (system) at a point of time.	(NAF v4, 2019), (Francois et al., 2014), (ISO, 2018)	(Bermejo-Alonso et al., 2011, 2016; Hernández et al., 2015; Walsh et al., 2005, 2006)
Effective_Situation	Logical combination of modes and states.	(Bonnet et al., 2017)	EI/CS
Expected_Situation	Situations resulting from confronting modes (defining the expected behavior of the system) to different situations that can harm it and are likely to occur (health states and physical absence for instance)	(Bonnet et al., 2017)	EI/CS
Mission	A mission is defined as a purpose to which resources may be directed.	(MODAF, 2010b)	EI/CS
Task	An assigned activity directly contributing to deliver a specified type of effect or to achieve a goal.	(MODAF, 2010b; NAF v4, 2018)	EI/CS
Objective	An increase for an organization used to demonstrate progress towards a goal.	(NAF v4, 2018)	(Bermejo-Alonso et al., 2011, 2016; Hernández et al., 2015)
Mission_Phase	A period of time in the life cycle during which activities are performed that enable achievement of objectives for that phase	(NAF v4, 2018)	EI/CS
Strategy	An organization's overall plan of development, describing the effective use of resources in support of the organization in its future activities.	(ISO/IEC/IEEE:24765, 2017)	EI/CS
Change	Anything that happens or takes place in the setting of the system, especially one of importance.	(NAF v4, 2019)	(Bermejo-Alonso et al., 2011, 2016; Gogniat et al., 2013; Hernández et al., 2015; Meyer et al., 2013)

System reconfiguration: A Model based approach
From an ontology to the methodology bridging engineering and operations

	Concept	Definition	Definition Source	Concept source
Dynamic	Mode	A functional status of an entity (system) at a point of time. The mode is subject to change during the mission. Thus, it may pass by different transitions.	(NAF v4, 2019), (Francois et al., 2014)	(Ali et al., 2011; Gogniat et al., 2013; Krichen and Zalila, 2011; OMG, 2010; Witt et al., 2013)
	Behavior	The way in which an entity acts or conducts oneself, especially towards others	(NAF v4, 2018, 2019)	EI/CS
	Function	An activity which is specified in context of the resource (human or machine) that performs it.	(NAF v4, 2018, 2019)	EI/CS
	Event	A trigger that is the cause of a particular action, process, or situation.	(Francois et al., 2014; NAF v4, 2019)	(Krichen and Zalila, 2011; OMG, 2010)
	Transition	A process or a period of changing from one mode to another.	(Francois et al., 2014; NAF v4, 2019)	(Ali et al., 2011; Krichen and Zalila, 2011; OMG, 2010; Witt et al., 2013)
	Condition	When transitions are triggered, a pre and post condition are verified before the transition starts and at the end of the transition in order to be sure that the transitions are allowed.	(Francois et al., 2014)	(Krichen and Zalila, 2011; Witt et al., 2013)
	Action_list	Group of Actions that are performed on (acted on) the resources to change the deployed configuration. Based on (Francois et al., 2014), the transitions may contain actions.	(Francois et al., 2014)	EI/CS
	Action	An information item explaining what has to be performed.	(Francois et al., 2014)	(Ali et al., 2011; Krichen and Zalila, 2011; Witt et al., 2013)
structure	Configuration	An arrangement of entities in a particular form, figure, or combination.	(NAF v4, 2018, 2019)	(Krichen and Zalila, 2011; OMG, 2010)
	Asset	A useful or valuable resource owned by an organization	(NAF v4, 2018, 2019)	EI/CS
	Service	A function, capability or behavior that is provided by a producer to a consumer	(NAF v4, 2018, 2019)	EI/CS
	Operator	Human resource is a human considered in a business or organization, regarded as a significant asset in terms of skills and abilities	(NAF v4, 2018, 2019)	EI/CS
	System	A combination of interacting elements organized to achieve one or more stated purposes	(ISO/IEC/IEEE:15288, 2015)	(Bermejo-Alonso et al., 2011, 2016; Hernández et al., 2015; Meyer et al., 2013)
	Resource	Anything that can be drawn on by a person or organization in order to function effectively	(NAF v4, 2018, 2019)	(Bermejo-Alonso et al., 2011, 2016; Hernández et al., 2015; Krichen and Zalila, 2011; Meyer et al., 2013; Walsh et al., 2005, 2006)
	Interface	A point or boundary where two entities can meet and interact	(NAF v4, 2018, 2019)	(Bermejo-Alonso et al., 2011, 2016; Hernández et al., 2015; Krichen and Zalila, 2011; Meyer et al., 2013; Walsh et al., 2005, 2006)

6.4.2. OSysRec description with SysML formalism

With regard to management, structure and dynamics aspect of system reconfiguration, the whole ontology is modeled as interconnected packages. This approach gives a macroscopic picture of the three aspects and the relations between them. Moreover, this representation allows visualizing the hierarchal nature between these aspects. In this section, we present the macroscopic view of our ontology. In the following sections, we will detail the different packages.

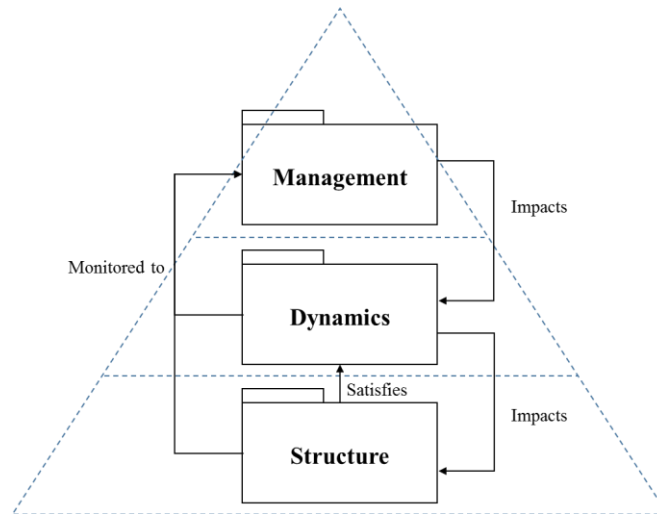


Figure 6-1: OSysRec Macroscopic view

The *OSysRec* macroscopic view (Figure 6-1) is presented by means of a package diagram. This global view shows the data flows between the different aspects of reconfiguration. The structural part satisfies the dynamics required via the *Configuration*. In order for the management aspect to work correctly, information about the structural elements and their related dynamics should be reported to the management elements. These data flows are generated by health and usage monitoring systems and supervision mechanisms (ISO, 2012, 2015, 2018). The *Effective_Mode* and the *Effective_State* representing the dynamic and structural parts of this ontology respectively are used within the management part. Once a decision has been taken at the management level, this can be translated into impacts at the dynamic level and by consequence impacts at the structural level. This impact is translated as an *Event* at the dynamic level and an *Action* at the structural level.

6.4.2.1. OSysRec: Management package

Systems generally exist to satisfy a general purpose to which all resources should be directed (see the Management package of the *OSysRec* ontology Figure 6-2). This purpose is referred to as a *Mission*, and the applied *Strategy* impacts its definition. A *Mission* is composed of one or more *Mission Phase*, each of which has its *Objectives* that can be achieved by performing the mission *Tasks*. For example, a Search and Rescue system (SAR) mission is to find a boat in the ocean and rescue its passengers. This mission is composed of 2 mission phases: search and rescue. The Search phase is composed of 3 tasks: detect, identify, and localize. In this scenario, the detection has the following objective: being able to observe an object of $1 m^2$ at 1 Km.

Systems are not isolated from their contexts. One of the main objectives of systems reconfiguration is the adaptation to context changes. Thus, it is vital to understand what elements are construing the context of the system of interest (SOI). In the *OSysRec* management package, the Context is composed of *Regulations*, *Environment*, *Time*, *Context System*, and *User*. Regulations from different domains can constrain the use of the system or parts of its resources. For example, within the detection phase, the helicopter (dedicated system for this objective) might not be allowed to cover some regions due to political or technological reasons. The environment can be of significant influence on the system and the way it is used to achieve the mission. For example, the presence of fog can impact the visualization capability provided by a camera. Thus, there might be a need to switch to another type of camera.

Budget is a critical element as it allows for evaluating the ability of the system to continue its mission within the current conditions. The *Budget* is an abstraction of all external resources that change over time (e.g., duration, power, money, food or fuel). For example, if the SAR mission should last 4 hours and the helicopter was provided with a quantity of gas enough for 4 hours, then if we still have 30 minutes left and the objectives have not yet been realized, actions should be undertaken in order to optimize the remaining resources to the maximum. The *Context System* can be part of the system as it interacts with the system of interest via interfaces. For instance, the SAR operators can use a helicopter or a boat. In the case they change their context system, there should be actions to be taken in order for them to continue their mission. The system *User* is the one who interacts with the system. The users may demand the mission to evolve. For instance, they might demand to pass the SAR mission from the detection phase to the identification phase.

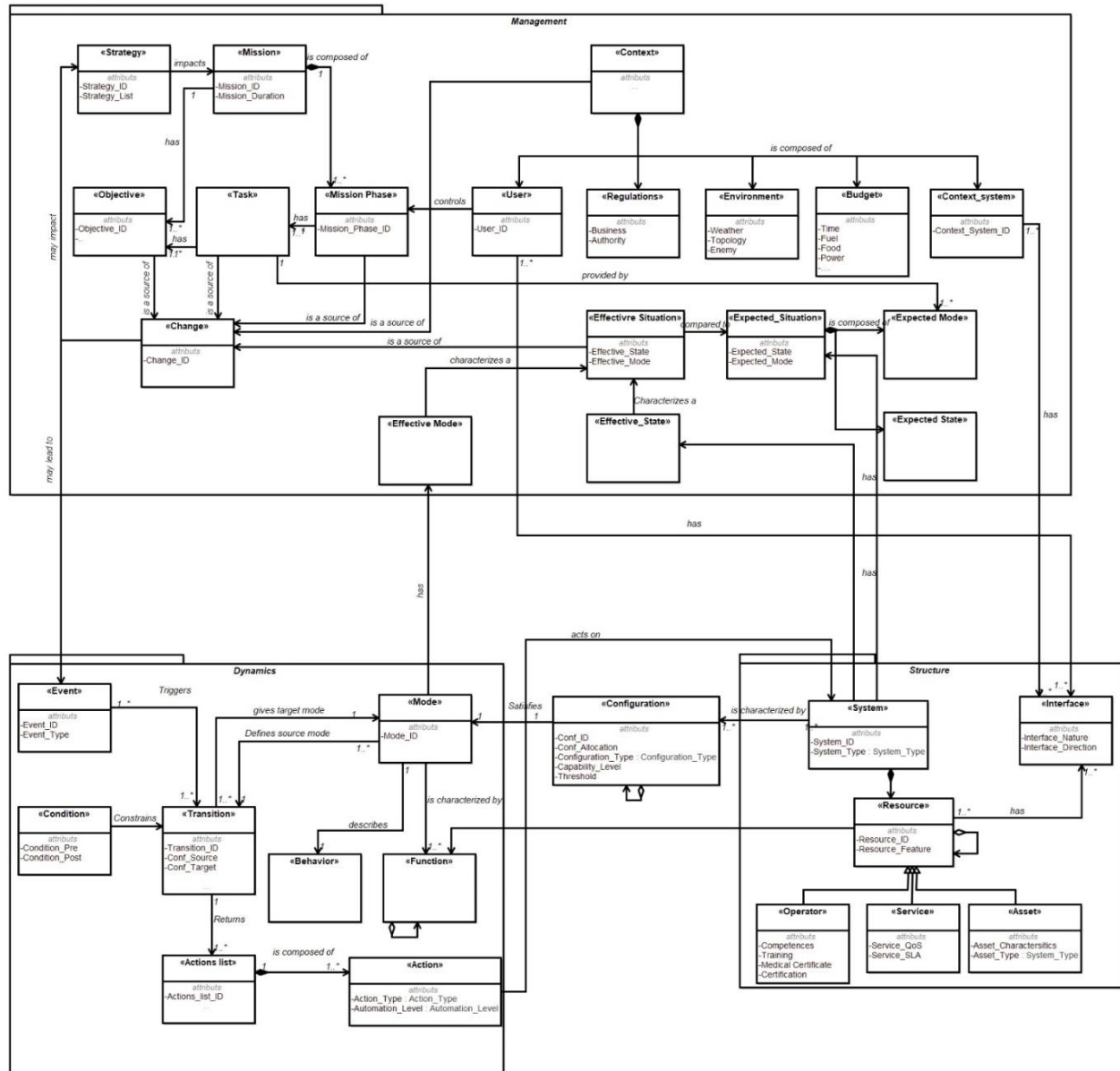


Figure 6-2: OSysRec ontology²

² A « Zoom-in » is provided in the appendix (see Figure A- 1 and Figure A- 2).

When the mission and the context of use are constant, the system should adapt itself in case of faults. Therefore, information about the *Effective Situation* should be reported to the management part. The *Effective Situation* is characterized by *Effective Mode* and the *Effective State*, which reflect the correctness of the expected behavior and the health state of the system. As discussed earlier in this section, this data can be reported to the management part utilizing health and usage monitoring systems. The *Effective Situation* is compared to the *Expected Situation*, which is similarly characterized by the superposition of an *Expected Mode* and an *Expected Situation*. The information about the *Expected Situation* comes from the engineering phase. In the SAR mission, if the *Effective Situation* indicates that one of the engaged equipment is defective (e.g., a camera) while in the detection phase, then actions at the dynamic and the structural levels should be taken. These actions may consider engaging other equipment (e.g., another camera or just a radar), or continue the mission with a degraded mode. These elements are considered sources of *Change*. When a *Change* is detected, an evaluation should be done, and actions need to be taken either at the management level by changing the *Strategy*, or at the dynamic and structural levels by generating an *Event*. In the SAR mission, this may represent an evaluation of the ability of the boat to rescue another detected boat in distress and hence, change the strategy to rescue the new target. In such a scenario, other equipment might be needed. Thus, an *Event* is generated at the dynamics level.

6.4.2.2. OSysRec: Dynamics package

As discussed in the previous section, a *Change* in the system context, the mission it tends to execute or the context of use may lead to an *Event* at the dynamics level (see Figure 6-2). The *Event* triggers a *Transition* from one *Mode* to another. The *Transition* represents a rule deciding the target *Mode* if the defined *Conditions* are met. In a SAR mission, a foggy weather might generate a reconfiguration request towards a new mode (detection with a radar instead of a camera). This triggers the rule which decides the new mode containing the radar taking into account the condition on the availability of the radar.

Knowing the source and target *Modes*, an *Action List* which is composed of a group of Actions ensures the transition is generated. The *Actions* act directly on the structure (the system and its composing elements). Depending on the objective of reconfiguration, the Action can be of different types: predictive, adaptive, corrective, perfective. The automation level of actions execution can be fully manual, fully automatic or hybrid. For instance, if the current configuration is detection using the camera (nominal mode) by default, and the target configuration is detection using a radar (degraded mode), then the corrective Actions would be deactivate the camera and activate the radar.

The *Mode* describes the system's behavior and it is characterized by the *Functions* of the engaged *Resources*. The allocation of *Functions* on *Resources* is referred to as *Configuration*. A *Configuration* satisfies a *Mode*. Hence, it is obvious that *Configuration* represents the core element that links the dynamic and the structural level.

6.4.2.3. OSysRec: Structure package

As seen in the dynamics part of this ontology, the system modes are satisfied by the system's *Configurations* ontology (see Figure 6-2). Thus, the *System* is characterized by *Configurations*. Acting on *Systems* and their *Configurations* is the way to get a new *Effective_Mode*. Taking the SAR example, the SAR system can have one configuration engaging two cameras and radar, and another configuration engaging a camera and a radio depending on what *Functions* the system needs to accomplish. Based on the NATO Architecture Framework NAF (NAF v4, 2019), the *System* is composed of *Resources* and *Interfaces*. These *Resources* can be physical *Assets*, *Services*, or human *Operators*. In a context of maritime SAR, the *System* is composed of assets (e.g., boat, cameras, radars, radios), *Services* (means of communications, private boats), *Operators* (a resource to allow identification). Depending on the *System* type (i.e., SoS, distributed, platform or equipment), the *Resource* might be aggregated of other *Resources*. Thus, their *Configuration* is aggregated from the *Configuration* of the engaged *Resources* and the *Interfaces* (i.e., Choice of the *Function* to be accomplished by the *Resources* and using which *Interfaces*).

6.5. Scenario-based validation: Connected Battle group

Ontology development integrates the task of validation. The validation of *OSysRec* ontology has been done by experts and tests on different case studies. In this section, the Connected Battlegroup case study is presented and used to validate the proposed ontology. The battlegroup is a tactical military organization whose extent ranges from reinforced battalion to enterprise size, which can perform its mission throughout the full spectrum of military operations. In the context of network-centric warfare, we define a connected battlegroup as a cohesive fighting force relying on the coherent integration of sensors, effectors and decision makers as well as on efficient information resource sharing and exploitation by both humans and machines. This connected battlegroup will be our system of interest, and we will use a simple scenario within this system for illustrating concepts previously described.

6.5.1. Battlegroup system organization

The battlegroup system is composed of heterogeneous platforms such as the dismounted soldier, fighting vehicle, reconnaissance vehicle, unmanned ground or air vehicle, and support arms in terms of artillery and engineers; each one providing various capabilities which participate in achieving the overarching system mission. Although a battlegroup can conduct the full spectrum of military operations, it is only allocated one primary role for which it is configured. In that role it can perform a number of battlegroup related tasks such as attack, defense, etc. Changing this role may require additional training of the battlegroup. Thus, this fighting force is organized/structured in logical nodes, also called Community of Interest (COI) which performs specific operational activities (Command, combat, reconnaissance, and engineering).

By combining capabilities of each of the highly heterogeneous force elements, such a system has the ability to conduct collaborative combat. Collaborative combat refers to the network-centric collaboration between multiple units, shifting to a force level thinking resulting in an information advantage and ultimately leading to a decisive war fighting advantage by providing extended capabilities. For instance, collaborative combat enables a battlegroup to:

- Improve system knowledge and shared situational awareness,
- Enhance precision in navigation, observation, targeting and fire, therefore improving survivability and mobility,
- Accelerate decision making processes and shrink decision loops, therefore increasing the speed of command,
- Possibly flatten the hierarchy and decentralize decision and command, therefore shrinking the OODA (Observe, Orient, Decide, and Act) loops.

One of the key challenges associated with collaborative combat are related to the command chain adaptability and the battlegroup node structure malleability. This malleability aims at optimizing or regenerating combat capabilities throughout the mission depending on the context and on the mission phase. For example:

- Formation of an ad-hoc combined arm unit (node) tailored for a specific purpose (protection, observation, or engagement),
- Temporarily reinforcement by a support node (e.g. Engineering) or to handle operational user mobility.

In order to enable this operational reconfiguration, it is important to pinpoint that a platform can be member of one or several COIs and that this membership can vary during the mission. For example, an engineering support vehicle might join a squad COI during a specific phase of the mission.

6.5.1.1. Node connectivity

The ability of the battlegroup to be dynamically reconfigurable ensures operational readiness and better management of capabilities provided by each platform; but it has an impact on node connectivity and therefore on the technical configuration of the wireless network. Indeed, depending on the context and the operational organization, the Information Exchange Requirements¹ (IERs) might change. As connected battlegroup platforms operating in a tactical environment are likely to be highly mobile, these IERs are supported by wireless communication system elements. Therefore, the underlying communication system needs to be adapted to fit the new operational needs and to support IERs update; especially in a denied, degraded, intermittent or limited bandwidth environment inherent to tactical battlefield.

6.5.1.2. Wireless communication system modes

The technical configuration of wireless communication systems typically includes a description of the provided Quality of Service, radio resource allocation, security parameters, etc.). These configuration items are defined during the mission planning phase but are subject to change due to planned or unplanned events. For example:

- Creation of an ad-hoc network supporting information exchanges within a node (e.g. association of an infantry fighting vehicle and an unmanned ground vehicle),
- Shifting from high-throughput (e.g. during mission briefing phase) to guaranteed-latency transmission service (e.g. during combat phase requiring a shortest response time).

6.5.1.3. The vehicle technical configuration

The vehicle is configured to support the observation mode using a radar, and a thermal imager. During the identification task, the soldier will observe the potential threat using different sensor resources available in the vehicle. While observing, if the resource providing this functionality has been detected to be faulty, then a reconfiguration is initiated aiming at engaging other available resources.

6.5.2. OSysRec: illustration scenarios

In this section, we illustrate the proposed Ontology with a simplified example of a battlegroup. This scenario does not reflect real life where battlegroups and their IERs are more complex. Initially, the platforms of the battlegroup system are organized in two COIs. This organization is shown in Figure 6-3. The COI 1 has 4 fighting ground vehicles (e.g. a armored squad), and the COI 2 has 4 dismounted soldiers (e.g; a infantry squad). In this scenario, COI 1 is responsible for the surveillance, and COI 2 is responsible for the reconnaissance, and the overall mission is to protect the borders of a village. To meet the requirement of this operational configuration, the technical configuration of the wireless communication networks of COI 1 & 2 is set to enable high throughput information exchange such as video stream or large data. Each vehicle is equipped with sensors (radar, thermal imager) allowing the surveillance task.

The battlegroup undergoes reconfigurations in response to changes in its mission and context. The sources of changes considered in the following scenarios are heterogeneous and include: detection of a potential threat, a confirmed enemy, a sensor failure, soldier non-availability, certification requirements, budget, and applied regulations in terms of engagement rules.

- Scenario 1: Detection of a potential threat

At time t_1 , a detection of a potential threat is considered a *Context* change (Enemy). Thus, it might be a source of Change. After evaluation with regard to initial mission objectives, the resulting Action would be to inform the Command post (CP) of the inability of the battle group to meet the mission objectives with the current organization of the COIs. Thus, the strategy which is defined by the command post might be impacted. Changing the *strategy* might imply an *Action* of restructuring the battlegroup to optimize the surveillance capabilities with improved observation or targeting capabilities. Thus, a creation of the ad-hoc COI 3 and reassignment of COIs tasks are done. These Actions lead to *Change* in the *Tasks* assigned to *Resources* (ie. Vehicle or dismounted soldiers). Technically, this means that there should be an

implementation of an ad-hoc wireless communication network to support the information exchange within COI 3. Moreover, an *action* of reconfiguring the radio network is necessary to be compatible with the high-throughput mode of COI3 (Resources reconfiguration). The configuration corresponding to step 2 is shown in Figure 6-3. This scenario has also been instantiated (Figure 6-4) using SysML internal block diagram. Instances corresponding to scenarios 2 to 7 are shown in the appendix section (Figure A- 3 to Figure A- 8)

- Scenario 2: Sensor failure

At time t_2 , the COI3 is observing the potential threat using a radar. If the radar fails, then the *Effective State* (failure in the radar) of the *System* of interest (the vehicle in this case) along with *Effective Mode* (observation) characterize an *Effective Situation* (observation mode with failure state). When compared to the *Expected Situation* which is characterized by the *Expected Mode* (observation mode) and the *Expected State* (operative radar), a *Change* is detected. This Change may lead to an Event at the dynamic level (internal failure) to reconfigure the system. This internal failure triggers a *Transition* to another *Mode* (observation with a thermal imager). This Mode corresponds to the new *Expected Mode*, and therefore, the *Expected Situation*. In order to allow this Transition, the availability of the thermal imager is verified (*Condition*). The Expected Mode becomes the Effective Mode by applying the *Configuration* which corresponds to it. For this purpose, an *Action list* containing *Actions* such as stopping the radar and running the thermal imager should be returned. These actions act directly on the System (Vehicle) and its Resources (radar and thermal imager) and their *Interfaces*. In this scenario, the impacted Resources are physical *Assets*. This scenario corresponds to step 3 in Figure 6-3.

- Scenario 3: Soldier non-availability

At time t_3 , while the COI3 is still in the observation Mode, if the *Operator* (Soldier 4) is not responding then the COI3 (*System*) has an *Effective State* (Soldier 4 is not available). The *Effective Situation* is again different from the *Expected Situation* and is considered a source of *Change*, leading to a new restructuring of the COI2 and COI3 where the soldier 3 leaves COI2 and joins COI3. At the technical level, the radio of soldier 3 should be reconfigured to join the COI3 radio network as explained in the scenario (t_1). This scenario corresponds to step 4 in Figure 6-3.

- Scenario 4: Detection of smoke

At time t_4 , smoke in the area of interest (*Environment*) will make it impossible to accomplish the observation (*Behavior*) while being in the observation *Mode*. The *Change* may lead to an *Event* triggering a *Transition*. The observation with the available *Assets* is not possible, thus, a *Transition* towards a *Configuration* where the observation *Mode*, and the needed *Functions* which characterize it, is ensured by a *Service* provided by external actors such as a drone is necessary. As the *Service* realizes a *Function* of the *System*, it is considered as a *Resource*. This scenario corresponds to step 5 in Figure 6-3.

- Scenario 5: a confirmed enemy

At time t_5 , if the threat is confirmed to be an enemy, a Change in the *Context* is detected. This Change is then evaluated, and an *Event* can be generated at the dynamic level to allow adaptation to the new *Context*. The *Event* (presence of an enemy) then triggers a *Transition* (the rule of being in an observation mode and an enemy is present), to a more suitable *Effective_Mode* (combat mode) with regard to the new *Context* (positive identification of the enemy). If the wireless communication network of COI3 can be reconfigured to satisfy the combat mode (*Condition* on this *Transition*), then the *Transition* is validated. The *Actions* allowing to achieve the targeted Configuration (which provides the combat mode) will be generated. In this scenario, the actions are: at first reconfigure the wireless communication network of COI 3 to satisfy the latency guaranteed service mode, and then reconfigure the radio to be compatible with this information exchange requirements. This scenario corresponds to step 6 in Figure 6-3.

- Scenario 6: Engagement rules

At time t_6 , the objective is to neutralize the enemy while in the combat mode. However, soldier 3 of COI3 does not have the skill to do the targeting and hence is not certified in a context of fire support. The applied rules of engagement (*Regulations*) are part of the *Context*, which is a source of a *Change*. This change is then evaluated, and an action at the management level is taken to adapt the battlegroup (*System*) to the new *Context*. The *strategy* can be impacted and the mission *Tasks* can be then allocated differently, i.e. combat and firing to COI3 and providing firing position to COI2. This scenario corresponds to step 7 in Figure 6-3.

- Scenario 7: lack of fuel (budget)

At time t_7 , if vehicle 4 is running out of fuel (*Budget*), then soldier 3 will not be able to continue doing the observation in the area of interest. Lack of fuel (*Budget*) is a new *Context* to which the connected battle group (*System*) should respond. This Change will be evaluated at the management level and restructuring of COIs should be done such that vehicle 3 leaves COI1 and joins COI3 (*Resources*). Soldier 3 should now use vehicle 3 (*Context_System*), and thus reconfiguration actions to connect radios to this platform are necessary. This will allow the adaptation to the new *Context*. This scenario corresponds to step 8 in Figure 6-3.

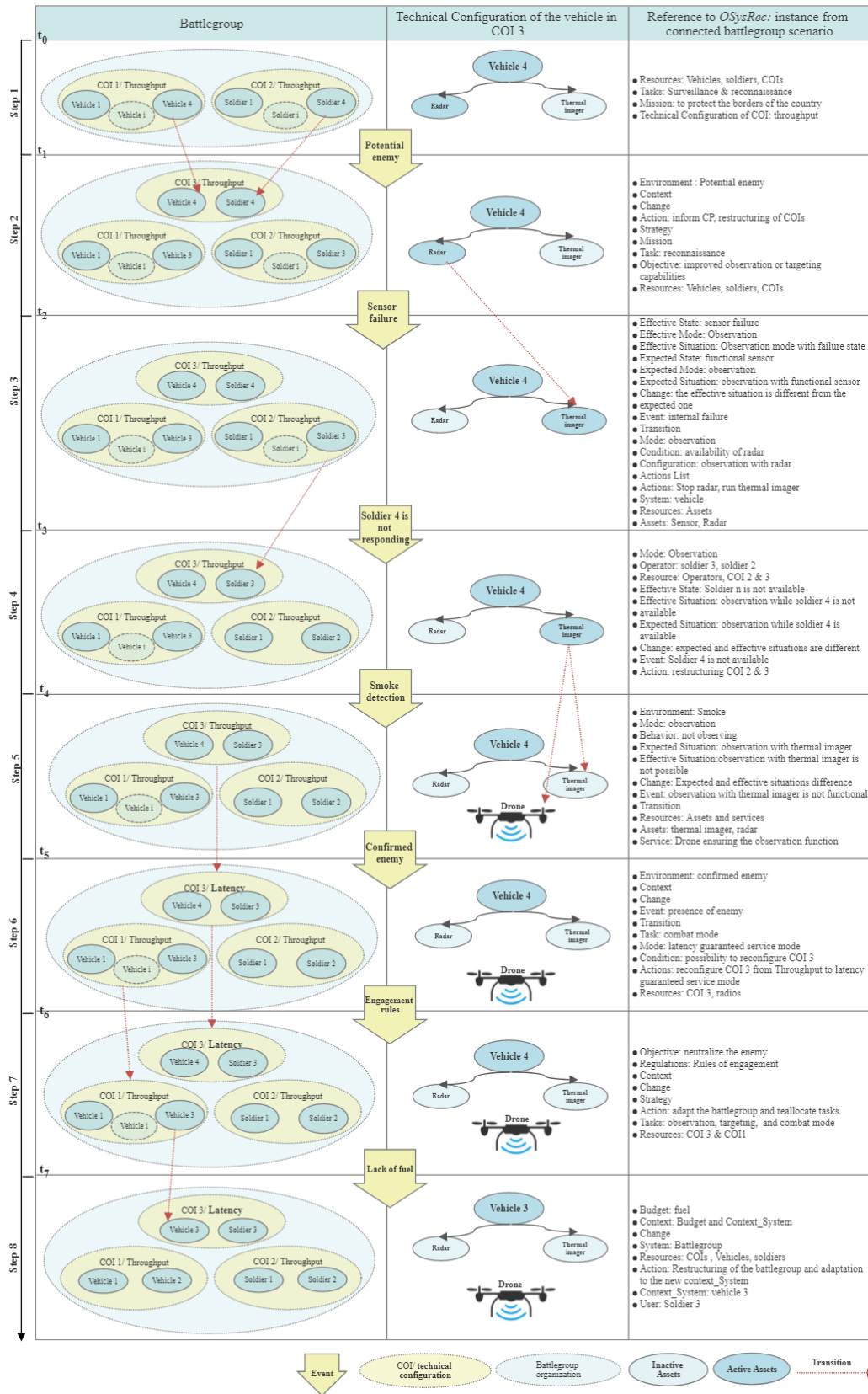


Figure 6-3: Battlegroup illustration scenarios

System reconfiguration: A Model based approach
From an ontology to the methodology bridging engineering and operations

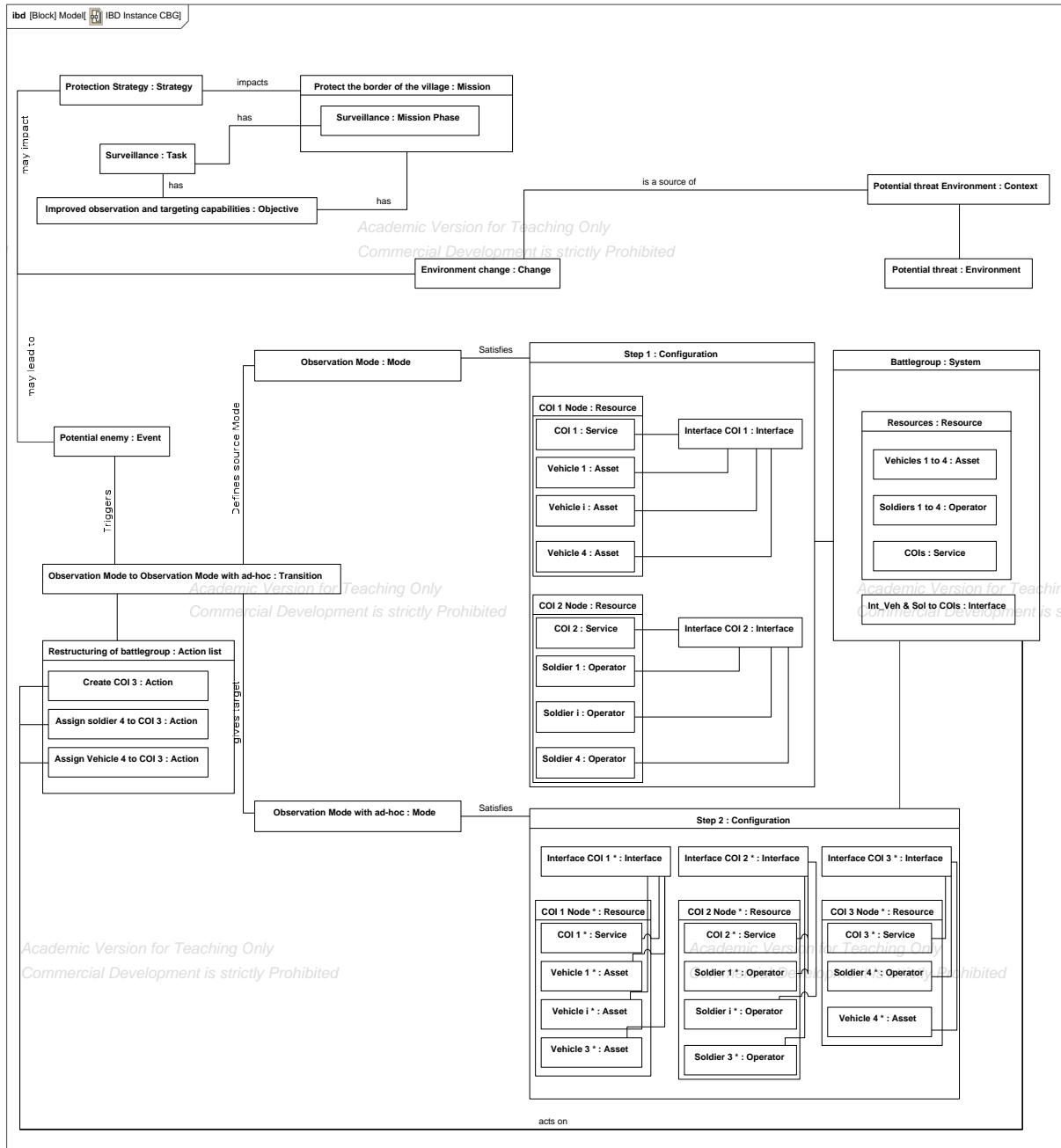


Figure 6-4: OSysRec instance for scenario 1

6.6. Scenario-Based Validation: Integrated Modular Avionics (IMA)

In this section, we demonstrate the application of the *OSysRec* ontology to Integrated Modular Avionics (IMA) system which is currently deployed in modern aircraft architectures for both civil and military applications such as the Airbus A380 and the Rafale (Personnic, 2002). IMA is a real-time computer network airborne system. IMA consists of a number of common function modules (CFM) populated in racks to allow for the replacement of the different modules. Six different CFM exist for IMA systems: 1) Data Processing Module (DPM) for data dependent processing activities, 2) Signal Processing Module (SPM) for data independent processing activities, 3) Graphics Processing Module (GPM) for image composition and formatting, 4) Mass Memory Module (MMM) for non-volatile storing, 5) Network Support Module (NSM) for network and protocol control, 6) Power Conversion Module (PCM) to allow for two stage power conversion to 48V. These CFM will be used to run different application processes depending on the type of the application (program). A functional configuration in this context refers to a mapping of application processes to CFM. The functional configuration can have a logical configuration and a physical configuration (ASAAC, 2004). The logical configuration defines the application requirements in terms of types and numbers of CFM, number of processing elements and communication channels. The physical configuration, on the other hand, is one implementation of a logical configuration which is translated (instantiated) for a specific physical configuration.

In order to operate properly, the system will require a number of tested, verified and certified configurations that should have been made available for operating the aircraft. During operation, the crew or the operator requests a system mode which can be satisfied by one or more logical and physical configurations. Mode selection is done via an application manager, and the reconfiguration is realized by the generic system management element. Within IMA systems, each mode can be satisfied by a nominal configuration, meaning that the required applications run on the main CFM. For each nominal configuration, one or more safety configurations exist. These configurations allow for running the same applications but on different CFM. In this case, the visible functionality does not change. Furthermore, for each nominal configuration, one or more degraded configurations exist. The degraded configurations provide a limited functionality of the IMA system. Changing between these associated logical or physical configurations will not impact the measured functionality level (external visible functionality). Changing from one mode to another may require a reconfiguration through the execution of action predefined in the run-time model. Civil and military avionics have different natures of reconfiguration. Civil IMA undergo static reconfiguration where only changing the hardware element is involved. On the contrary, IMA in the military applications can reconfigure SW elements dynamically.

The types of events that can be encountered within the IMA systems are: 1) equipment fault, 2) software (application) fault, 3) change mode, 4) global re-launch of IMA systems. Depending on the detected event and the current configuration, the system management can propose different actions. These actions may or may not require a reconfiguration. The possible alternatives include:

- No action: In this case, the processing on the considered modules continues.
- Program re-initialization: the program run on a given module is stopped and re-launched. In this case, no module reconfiguration.
- Program stop: in case of a software error, the program is stopped. When stopping a program, a change of the execution module is necessary.
- Module re-initialization: all the programs run on one module are stopped and re-launched.
- IMA re-initialization: all the programs on all modules are stopped and re-launched.
- Module stop: in this case, the IMA system will detect an absence of this module and hence a reconfiguration is necessary.

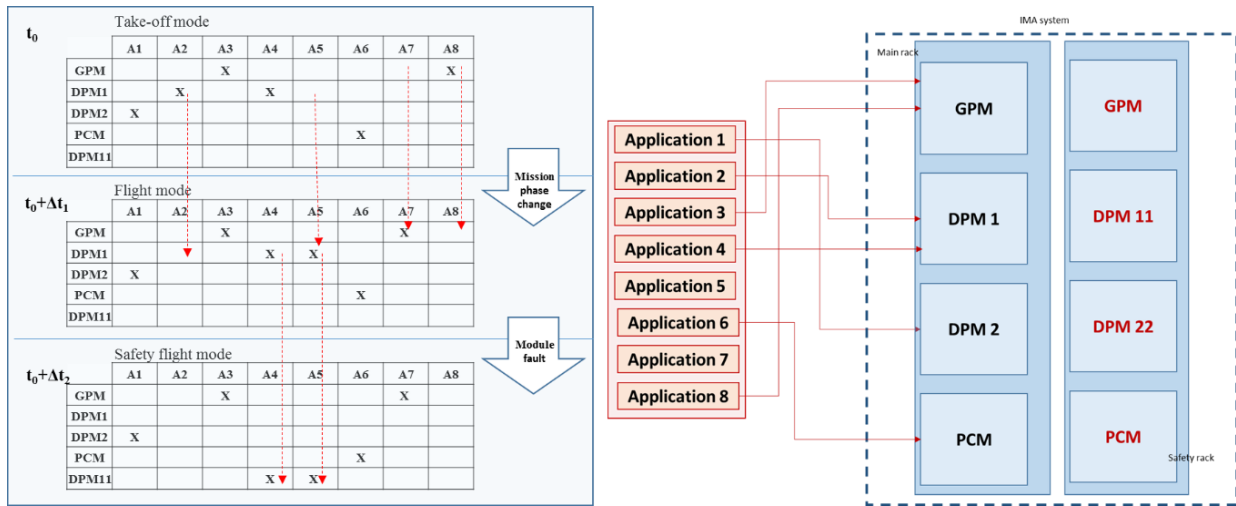


Figure 6-5: (a): Mapping of applications on available modules of IMA (b): Reconfiguration of IMA based on different events

To illustrate the ontology described in section 6.4, we use a simplified example of an IMA system with only 4 process modules and 8 applications. Time intervals refer to events sequencing. Initially, at time t_0 , the IMA system is organized to ensure the operation of the take-off (Figure 6-5,a). For this case, the needed applications for this logical configuration are 1, 2, 3, 4, 6 and 8 (see Figure 6-5,b). Applications 2 and 4 are run on DPM1. Application 1 is run on DPM2. Application 6 is run on the PCM module. Finally, the GPM module is used to run applications 3 and 8. In this context, the applications and the modules are considered

the *Resources*. The mapping of the applications on modules represents the *Configurations*. The Configuration satisfies a functional *Mode*. In this example the nominal take-off mode.

At time $t_0+\Delta t_1$, when the take-off is completed, the pilot requests the flight mode. The *User* controls the *Mission Phase*. For the flight mode (mission phase), the resources and their configuration are different. Hence, this mission change is considered as a *Change*. After evaluation with regard to the new mission objectives in terms of required resources, the result would be to generate an *Event* at the dynamic level in order to implement the Flight *Mode*. Technically, being in the take-off mode and receiving the *Event* to pass to the flight mode will trigger a *Transition* to implement the configuration corresponding to the Flight Mode (application 1 on DPM2, applications 4 and 5 run on DPM1, application 6 on the PCM module and applications 3 and 7 on the GPM module). A *Condition* on this transition would be the availability of all the required applications and modules (*Resources*). The resulting *Actions list* can include the following *Actions*: 1) stop application 2 on DPM 1 , 2) run application 5 on DPM 1, 3) stop application 8 on the GPM and finally 4) run application 7 on the GPM. This procedure is illustrated in Figure 6-5,b. This scenario has also been instantiated (*Figure 6-6*) using SysML internal block diagram.

At time $t_0+\Delta t_2$, a detection of a fault in the module DPM1 (*Resource*), while in the flight nominal mode (*Effective_mode*), will be reported to the management part via the *Effective_State*. Thus, comparing the IMA effective state for the current flight mode (*Effective_Situation*) to the expected one will lead to a change at the management level. This change when evaluated a signal of failure (SYSFAIL) is considered an event demanding a reconfiguration of the IMA at the dynamic level. This event triggers a transition towards the safety flight mode where DPM11 replaces DPM1. In this case, the condition on this transition is the availability of DPM11. In order to achieve the transition (reconfiguration) the actions needed are 1) stop application 5 on module DPM1 , 2) stop module DPM1, 3) re-launch DPM11, 4) run applications 4 and 5 on DPM11. This procedure is illustrated in Figure 6-5,b.

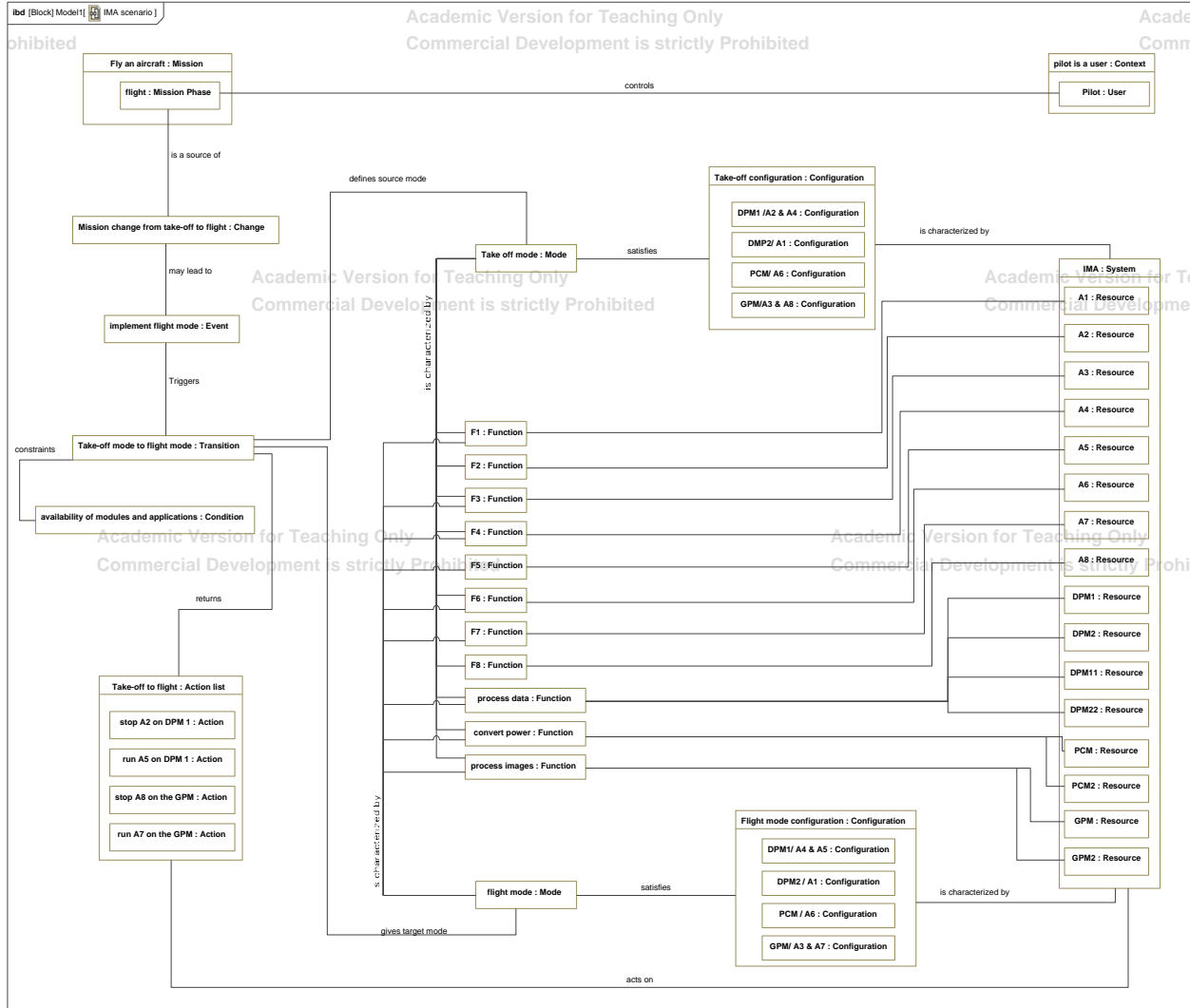


Figure 6-6: OSysRec instance for IMA scenario 1

6.7. Discussion

The novelty of the proposed *OSysRec* ontology lies in integrating and enhancing three main system reconfiguration aspects that have been considered, up to now, separately in the literature in order to address the overall system reconfiguration process. The proposed ontology covers the essential concepts of reconfiguration based on a validation by experts, and complements the concepts addressed by models and ontologies existing in the literature. Experts from industry assessed the *OSysRec* ontology regarding the clarity and exhaustiveness criteria. The assessment validated its intended use as a comprehensive conceptual framework aiming to support engineers in studying systematically various problems and solutions related

to dynamic evolution. The ontology intends to be a reference for key Systems Reconfiguration concepts on which parties from different domains may agree on. Furthermore, due to its development from case studies with different system types, we argue that it is sufficiently generic to be applicable to a wide range of systems in different domains. However, due to its generic nature, its application to specific cases is not trivial. It requires additional adaptation and tailoring steps.

As the ontology aims at presenting the essential concepts related to Systems Reconfiguration, not all of the concepts are discussed in detail (e.g., contracts, regulations, and business rules). This ontology defines what data are to be manipulated for Systems Reconfiguration. Different levels of data and information are presented without explaining the mechanism and the tools used to process them.

A potential threat to validity of the ontology is that the concepts were included based on the opinion of experts from one specific company. Furthermore, the ontology has been validated by experts that specifically work in the domain of defense, contrary to the experts that contributed to the elicitation of concepts who work in a variety of domains. This could introduce a bias into the generated ontology. In order to avoid the introduction of such a bias, we cross-validated the ontology via the existing literature and using a variety of case studies. The cross-validation via the existing literature, which covers several different domains intends to validate that those concepts or their equivalents selected by the experts were also considered as relevant in the literature. The variety of case studies covering different types of systems intends to avoid that bias. We also argue that this variety of system types makes the ontology sufficiently generic to be applicable to a wide range of systems in different domains.

The presented ontology is intended to be used as a basis for future tools to support system reconfiguration activities, which will be addressed in future work.

6.8. Conclusion and future work

Research presented in this chapter aims at supporting System Reconfiguration processes in industry setting. System Reconfiguration is a critical activity in complex system design and has been a subject of numerous research studies. The objective of this research is to provide an ontology as a comprehensive conceptual framework within which it is possible to study various problems and solutions related to dynamic evolution in a systematic manner. However, there is a lack of approaches that support the overall system reconfiguration process focusing only on structural, dynamic or management aspects. In order to address this gap an extensive literature review as well as industrial research has been conducted. In order to support

these activities, ontology development has been decided, and hence a Systems Reconfiguration (*OSysRec*) ontology was proposed. The ontology was developed based upon 18 expert interviews (both internal to the enterprise being sponsor of this research, and external) as well as several industrial use cases. Different use cases covering a wide range of system types and domains have been used for validation. A battlegroup scenario, currently under exploration and investigation in the military domain, has been used to illustrate and to test the *OSysRec* ontology.

This ontology is the initial step in a broader research which aims to develop a support for System Reconfiguration. Continuing work is investigating the possibility to develop such a feature using multi-agent modeling. This feature will evaluate systems ability to accomplish mission objectives based upon systems health and context. When necessary, this feature will inform system users with actions to be taken to ensure effectiveness.

7. A Model-based Method for System Reconfiguration

This paper was submitted to Journal of Mechanical Design.

Lara Qasim, Andreas-Makoto Hein, Soirn Olaru, Marija Jankovic and Jean-Luc Garnier

Abstract. *System Reconfiguration is essential in systems management as it enables system flexibility and adaptability. It plays a crucial role in ensuring system's operational effectiveness via increasing its availability and reliability, availability, maintainability, testability, safety, and reuse of system entities and technologies. Within current industrial practices, the development of reconfiguration supports is challenging. The development of these supports demands the integration of relevant reconfiguration data concerning the system objectives, operational context, and level of functioning. In this chapter, this fundamental data is integrated within a reconfiguration method, which we call MBSysRec. A maritime search and rescue case study is used to illustrate the presented method. The Short-Term Conflict Alert functionality of the Air Traffic Management system is also presented to challenge the presented approach with large scale industrial problems.*

7.1. Introduction

Companies are currently concerned with designing and developing systems, which have considerable lifetimes and diverse employments (i.e. ships, aircraft and military equipment). Throughout the extended lifecycles, systems can be upgraded to fulfill new demands, allow for evolution, and to improve survivability. System configuration management plays a crucial role in the effective management of an evolving system during its lifecycle (INCOSE Systems Engineering Handbook V4, 2015; ISO/IEC/IEEE:15288, 2015). The system's adequate operation in conjunction with the evolution of its operational context and environment is referred to as operational effectiveness. Operational effectiveness reflects the ability of the system to deliver defined capacities despite evolution and changes in system's environment and context. System reconfiguration, thus, depends on diverse aspects such as economic, environmental, legal, operational, behavioral, structural, and social that are critical to demonstrate a capability. Any change in these aspects can lead to "System Reconfiguration" to maintain its operational effectiveness. In accordance, "System Reconfiguration" is defined by the subsequent changes of system configurations with the objective of maintaining or adapting (increasing or decreasing) the capabilities provided by the system (Qasim, Hein, et al., 2019). System reconfiguration is of value for stakeholders. It allows improvements in system performance, effectiveness, and affordability while ensuring increased reliability, availability, maintainability, testability, safety, reusability, and reuse of system entities and technologies (ISO/IEC/IEEE:15288, 2015).

Within current industrial practices, the development of reconfiguration supports is challenging (Qasim, Hein, et al., 2019; Qasim, Jankovic, et al., 2019). In the context of a large international aerospace, space, ground transportation, defense, and security company, a major challenge related to the development of system reconfiguration supports consists of proposing adequate methods that address: 1) deployment context consideration, 2) missions and objectives evolutions, 3) sensor data integration.

Current research focuses on supporting system configuration during the design phase. However, to the best of our knowledge, there is no reconfiguration method takes into account the operational context and evolution of objectives as well as the level of functioning of the already deployed systems. Integrating this data is fundamental in proposing operationally effective configurations and supporting reconfiguration during operations.

In order to support system reconfiguration, we present a method, which we call MBSysRec, integrating information related to deployment contexts and objectives, along with the sensor data reflecting the functioning level of the already deployed system, which are lacking in the existing methods. For this

purpose, a rigorous study of both the existing references and the industrial cases has been conducted to present an adequate reconfiguration method.

The remainder of this chapter is organized as follows. Section 7.2 synthesizes the bibliographical review on methods addressing system configuration and reconfiguration and discusses their related issues. Section 7.3 presents an overview of the MBSysRec method. Section 7.4 illustrates the MBSysRec method through a Search and Rescue (SAR) case study. Section 7.5 displays the application of MBSysRec to an industrial case study concerning the reconfiguration of Short-Term Conflict Alert functionality within the Air Traffic Management system. Section 7.6 provides a discussion on the presented approach. The last section draws conclusions and outlines future work.

7.2. Literature Review

The objective of this research is to support the system reconfiguration process related to post-deployment modifications and updates. System reconfiguration relies on different information concerning the system architecture not only in terms of functions and components and their interfaces, but also system objectives and their operational contexts. The data concerning components or sub-systems availability is also needed to allow their reuse and to estimate the overall functioning level of the new set of possible configurations. Hence, a method jointly taking into account the information from system architecture design and configuration/reconfiguration principles is needed. Often system architecture design and system configuration/reconfiguration are research domains that are not easy to separate as they both rely on proposing system configurations by combining building blocks while respecting mutual constraints (Ziv-Av and Reich, 2005). This literature review addresses methods that specifically aim at supporting system architecture design (section 7.2.1). In section 7.2.2, we address methods pertaining, in particular, to system configuration/reconfiguration. In the last part we identify research gaps and underline future developments.

7.2.1. System Architecture Methods

Systems architecture aims at defining the structure and behavior of complex systems to meet their operational and functional needs (NAF v4, 2018). Several scholars presented definitions of system architecture (Stone et al., 2000; Ulrich and Eppinger, 1995; Yassine and Wissmann, 2007). Crawley et al.

(2004) underline that these definitions share the idea that system architecture is an abstract description of system entities and the relationships between them. In this research, we adopt this notion of system architecture.

To support the design of system architectures, system architecture frameworks have been presented to establish a common practice for creating, interpreting, analyzing and using architecture descriptions within a particular domain of application or stakeholder community (DODAF, 2009; MODAF, 2010b; NAF v4, 2018; TOGAF, 2009). The purpose of the architecture design process is to generate system architecture alternatives, to select one or more alternatives that frame stakeholder concerns and meet system requirements (NAF v4, 2018; Ulrich and Eppinger, 1995).

There are several objectives for system architecture design methods. Amongst these objectives is the generation of a variety of architectures while ensuring their viability. Both studies by Bryant et al. (2005) and Kurtoglu and Campbell (2009) suggest that designers should start by defining a functional model as an initial step in searching for configurations. The functional model can then be used along with the function-component relations and the component-component relations stored in a design repository to generate viable architectures. In the same vein, Chen et al. (2019) presented a method that maps functions to components while ensuring compatibility when combining components via two different types of compatibility. The first is the compatibility of physical effects, and the second is the compatibility of properties in terms of provided and accepted quantities (i.e. provided and accepted torque in machine-drives applications). The method supports system synthesis from its conceptual definition up to its detailed design. Hamza et al. (2011) proposed a feature-driven approach for generating pervasive systems architectures. This approach relies on establishing a link between features and components towards defining physical architecture based on the selected features.

Albarello et al. (2012) introduced a formal method for system architecture synthesis and optimization to support the conceptual phase when more innovative architectures are expected. The method relies on a predetermined functional model and uses rules concerning function to component consistency and component to component consistency to ensure the feasibility of the presented solutions. Innovative architectures have also been the concern of Moullec et al. (2013). Moullec et al. aimed to combine data from previous projects and data provided by expert estimations related to new technologies to generate innovative architectures and assess their performance under uncertainty. The method starts from a higher-level structural (physical) architecture and finds alternatives for design variables based on their performances and their overall confidence level. In another study aiming to present innovative solutions for different architecture alternatives, Helms and Shea (2012) use the functional decomposition in terms of overall

functions, high-level functions, and sub-functions. Functions are used to find the required physical effect (behavior) that can be further mapped to components to end up with a structural architecture.

Researchers also investigated the integration of interface and constraint information to support architecture generation during the conceptual phase. For instance, Wyatt et al. (2008) presented a method that captures the rules governing architectures that “make sense”, and uses these rules to compare a particular architecture concept against the space of possible architectures for that product. The data considered here are the possibility of mapping component types and interface types to find physical architectures using component compatibility and their immunity to propagate changes across their interfaces. In a more recent study, Wyatt et al. (2012) suggested capturing the knowledge about the components and their relations in an ontology that can be further used to identify the arrangements of these elements that are ‘realizable’. Jankovic et al. (2012) focused on integrating failures related to system interfaces and design parameters shared across multiple domains in architecture generation and assessment. The method relies on a given physical model and related satisfying components with their interfaces. Zi-Av and Reich (2005) presented a method to find the arrangement of design variables (building blocks) satisfying a physical architecture. The method estimates the interactions between the building blocks, the mutual constraints among them, and the contribution of building blocks to achieving the product requirements.

Other researchers presented comprehensive methods allowing to integrate requirements in the architecture generation process. Rosenstein and Reich (2011) extended the method presented by Ziv-Av and Reic (2005) to allow further assessments based on customer requirements, market conditions, and the use of current technology. The extension is based on the integration of a second level of building blocks to represent the product requirements. Condat et al. (2012) presented a method that should reduce the design duration through an earlier consideration of safety constraints. The data considered within this method are functions and their relations, modules (components), and their characterization, and module locations. The constraints regarding function to module, module to location, or function to location mappings represent the safety requirements defined by the user. Judt and Lawson (2012) presented a method linking the requirements to functions, and function to components, to allow for architecture assessment. During the detailed design phase, the customer’s key performance attributes can evolve, reflecting a change on the relative importance of assessment rules, that can further affect architecture evolution. Haris and Dagli (2011) studied this phenomenon by simulating changes in the relative importance of rules and observed the architecture evolutions due to these changes. Their method uses functions, components, and the importance of rules representing the measure of effectiveness.

7.2.2. Product Configuration and Reconfiguration Methods

Product configuration has been treated by researchers over many years (Aldanondo and Vareilles, 2008; Hofstedt and Schnee-weiss, 2013; Mittal and Frayman, 1989; Soininen et al., 1998). All these researchers agree that product configuration means selecting and arranging combinations of parts or components that satisfy given specifications. The product configuration task, in general, involves three main activities: 1) selecting components that play a role in satisfying requirements and fit in the current configuration, 2) establishing relationships between components (association), and 3) configured product evaluation which consists of compatibility and goal satisfaction testing (Brown, 1998).

Various product configuration methods have been documented in the literature (Falkner et al., 2011; Zhang, 2014). Song and Kusiak (2009) presented a method that relies on rules reflecting customers' preferences to recommend new feasible product configurations. Product configuration based on a dynamic set of components where variables in final solutions are activated from the original variable set depending on specific conditions. was addressed by Wang et al. (2011). The presented method takes into account the activation of components and further constraints such as cost and compatibility in an integrated approach. Liu and Liu (2010) integrated information concerning the reliability of the configured product within the configuration process. To reduce the complexity of product configuration, Lee and Lee (2005) presented a method that uses a standard model (reference model) as a starting point in the configuration process that can be optionally modified to meet a customer requirement. Xie et al. (2007) presented a methodology for a generic configurator that can solve an engineering product configuration problem with complex constraints. The methodology relies on modeling functional requirements, component types representing the basic building blocks in a product, design parameters reflecting performance and component features, and design constraints that are general relationships among functional requirements, design parameters, and components types.

Newer research underlines that the future of product reconfiguration is in addressing the rearrangement of already deployed products to meet the new requirements to keep the product up to date (Männistö, 2000). While product configuration has been widely researched, product reconfiguration has been treated with less attention. Existing methods focus on describing and characterizing the problem rather on reconfiguration itself. For instance, Mannisto et al. (1999) presented a framework summarizing the factors affecting reconfigurability. These factors include product type, product cost, customer requirements, and technological change rate. Kreuz and Roller (1999) describe the main parts needed in a reconfiguration system and concentrates on the knowledge part regarding the system components and dependencies. Friedrich et al. (2011) presented logic-based knowledge representation to describe reconfiguration

problems. As one can see, these approaches rather describe the problem of product configuration than provide concrete methods to support this activity.

7.2.3. Synthesis and Research Gap

System architecture design and product configuration/reconfiguration activities can be seen as similar as they are based on combining different building blocks into one system, such as to satisfy system requirements or missions. Section 7.2.1 underlined that system architecture design is concerned with identifying the building blocks by either starting from the defined system functions (Albarelo et al., 2012; Bryant et al., 2005; Chen et al., 2019; Condat et al., 2012; Helms and Shea, 2012; Judt and Lawson, 2012; Kurtoglu and Campbell, 2009), or through the identification of possible compositions between the building blocks (Jankovic et al., 2012; Moullec et al., 2013; Rosenstein and Reich, 2011; Wyatt et al., 2008, 2012; Ziv-Av and Reich, 2005). Depending on the objective, in general, the integrated data can be function-to-component relation, component-to-component relation, components types, a list of design variables that satisfy them, and compatibility relations. More specific information in these system architecture generation methods is integrated depending on the addressed issue such as the consistency of the physical effects (Chen et al., 2019), requirements (Condat et al., 2012; Judt and Lawson, 2012; Rosenstein and Reich, 2011) (Condat et al., 2012; Judt and Lawson, 2012; Rosenstein and Reich, 2011), component performance and uncertainties (Moullec et al., 2013) and interfaces characteristics (Jankovic et al., 2012; Wyatt et al., 2008, 2012; Ziv-Av and Reich, 2005).

On the other hand, product configuration, starting from the high-level physical architecture definition, aims at identifying all possible combinations of components that would correspond to this defined architecture. Data considered in product configuration methods, in general, include components types, instances of design variables, and compatibility relations. Additional data can be considered with regard to a specific reconfiguration problem, e.g. customer requirements or rules allowing configurations evaluation (Song and Kusiak, 2009; Xie et al., 2007), uncertainties reflecting configurations reliability (Liu and Liu, 2010), and components activation constraints (Wang et al., 2011).

In general, system architecture design (Jankovic and Eckert, 2016) and product configuration/reconfiguration (Zhang, 2014) methods were developed to support the conceptual phase. Up to now, studies treating reconfiguration of already deployed systems to assist the operational phase are limited (Männistö, 2000; Zhang, 2014). In particular, the difficulty of deployed system reconfiguration lies in identifying and managing functional variability that is essential to satisfy evolving system objectives and

missions (Qasim, Jankovic, et al., 2019). In addition to functional variability, system reconfiguration needs to capture the data concerning the operational context and its link to the available resources performing missions (Qasim et al., 2020). As reconfiguration deals with evolving systems, the set of available resources is also varying. Therefore, searching for configurations is based on an updated components list that reflects the information regarding their availability and levels of functioning (Qasim et al., 2020).

Few methods recently started to address issues related to system reconfiguration in the operational phase, including component activations during the configuration task (Wang et al., 2011), changes in the customers' requirements (Haris and Dagli, 2011; Lee and Lee, 2005), and legacy system evolutions (Friedrich et al., 2011). However, the difficulty lies in the fact that these presented methods do not integrate the information about the functioning of components stemming from different sensors. Sensors are nowadays used extensively in order to support system management and reconfiguration.

To summarize, system architecture design and configuration methods are usually developed to support the conceptual and detailed design phases. Very few methods exist that consider reconfiguration in the operational phase. As discussed in sections 7.2.1 and 7.2.2, existing methods do not address the combinatorial exploration of building blocks considering the information pertaining to the operational context and its objectives. Moreover, information from the operational phase reflecting the availability of resources (components), and their levels of functioning (sensor data) is also not integrated within these methods. To address this gap, we present the model-based system reconfiguration method (*MBSysRec*) to support the reconfiguration process and to ensure the system's operational effectiveness while considering system evolutions in terms of structure, missions, and operational contexts.

7.3. MBSYSREC Method

7.3.1. MBSysRec: Objective and Requirements

In order to integrate the information concerning the system's operational context, objective, and level of functioning that are essential for system reconfiguration, we present the Model-based system reconfiguration method (*MBSysRec*). *MBSysRec* is a method dedicated to reconfiguring systems equipped with health and usage monitoring systems (i.e., sensors and data analysis facilities) as operational data can only be captured via such enabling systems. Health and usage monitoring systems provide information on the constituting components health assessment (level of functioning). The main objective of this method is

to identify possible system configurations with regard to the observed changes. In an operational context, systems evolve in response to several changes. These changes are mainly related to the system’s mission, its tasks and objectives, context, and system health. *Figure 7-1* provides the information integrated within the MBSysRec method as well as the relationships between them. The information represented in the diagram integrates the concepts related to system architecture, more specifically functional and physical architecture (i.e., functional chain, functions, components, and their compatibility). These concepts are usually used in system architecture design in the early stages of the engineering phase. However, in our approach, we use these concepts to reconfigure systems during operations by generating system configurations. Therefore, in our approach, we extend these concepts to address reconfiguration issues, and we also integrate further required concepts. These concepts include objective, context, and other concepts representing data from operations (i.e., effective situation, effective state, level of functioning).

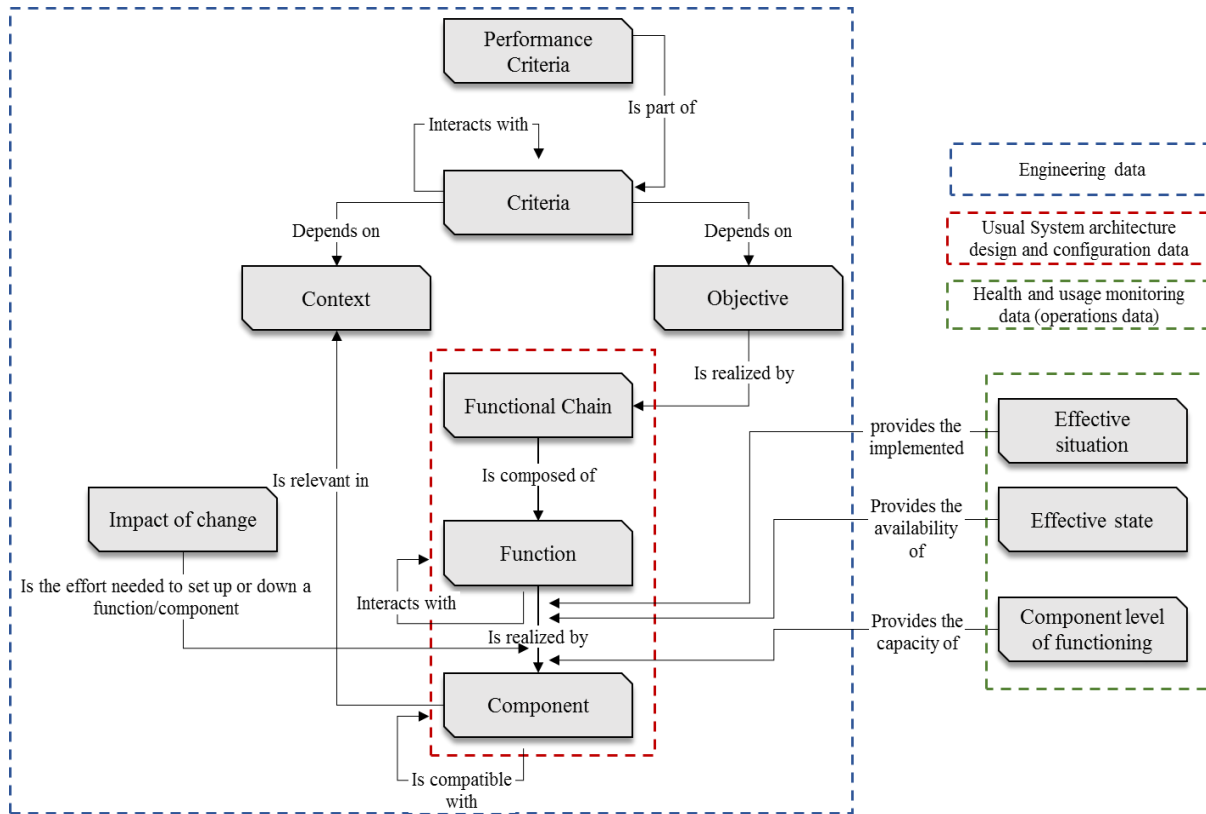


Figure 7-1: MBSysRec concept diagram

Figure 7-2 shows the main inputs and output of the method. *MBSysRec* aims to bridge both the engineering and operational phases by integrating data coming from system architecting activities during the engineering phase with the change data related to different sensors and component functions in the operational phase.

The engineering phase input data includes the system architecture data i.e., functions, components, and their relations, as well as all information regarding the component performance, criteria weights, and interactions allowing configurations trade-offs. The operational phase data are mainly related to sensor data reflecting information on the component's availability and level of functioning. In addition to sensor data, operational data encompasses also the system's operational context and objective.

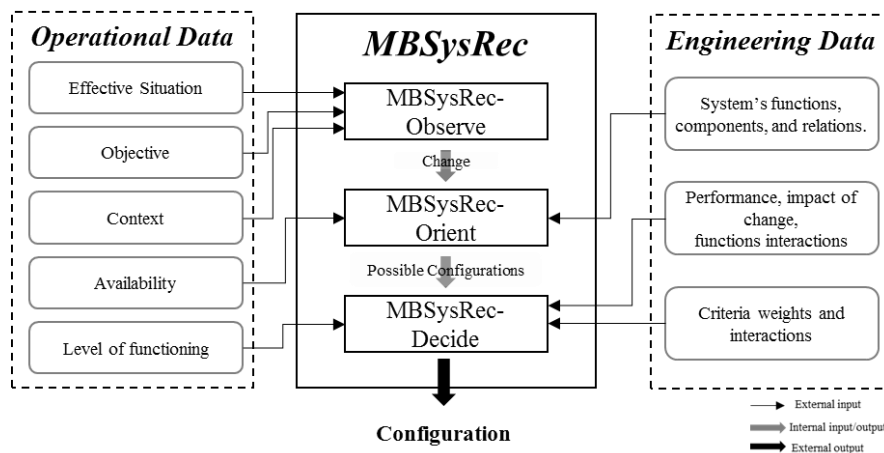


Figure 7-2: Overview of MBSysRec method

The *MBSysRec* method is defined to address the following requirements:

- Detecting changes in an operational context or objectives. Changes can also be related to system failures.
- Searching, with regularity, for a configuration with a better satisfaction level if no change is detected.
- Generating possible configurations with regards to the operational context and objective.
- Evaluating the generated configurations based on context and objective dependent criteria or policies (safety, business, cost, etc.).
- Selecting one configuration from the evaluated solution space.

7.3.2. MBSysRec: Process Steps

MBSysRec is an automated method, including three steps (see *Figure 7-2*): 1) observe changes (*MBSysRec*-Observe), 2) identify possible system configuration regarding the observed changes (*MBSysRec*-Orient), and 3) perform system configuration performance evaluation and selection (*MBSysRec*-Decide). In order to find an optimal configuration, *MBSysRec* relies on information stored in a matrix system (see *Figure 7-3*). We adopted this strategy as matrices are generic and a widely used way to represent the engineering model data. Matrices can also be translated from model instances that are used in the company. Furthermore, applying algorithms and mathematical operations can be eased using matrices. A Multiple Domain Matrix MDM represents the matrix system. It is composed of three DSMs (Design Structure Matrix) along its diagonals and eight DMMs (Domain Mapping Matrix) outside the diagonal.

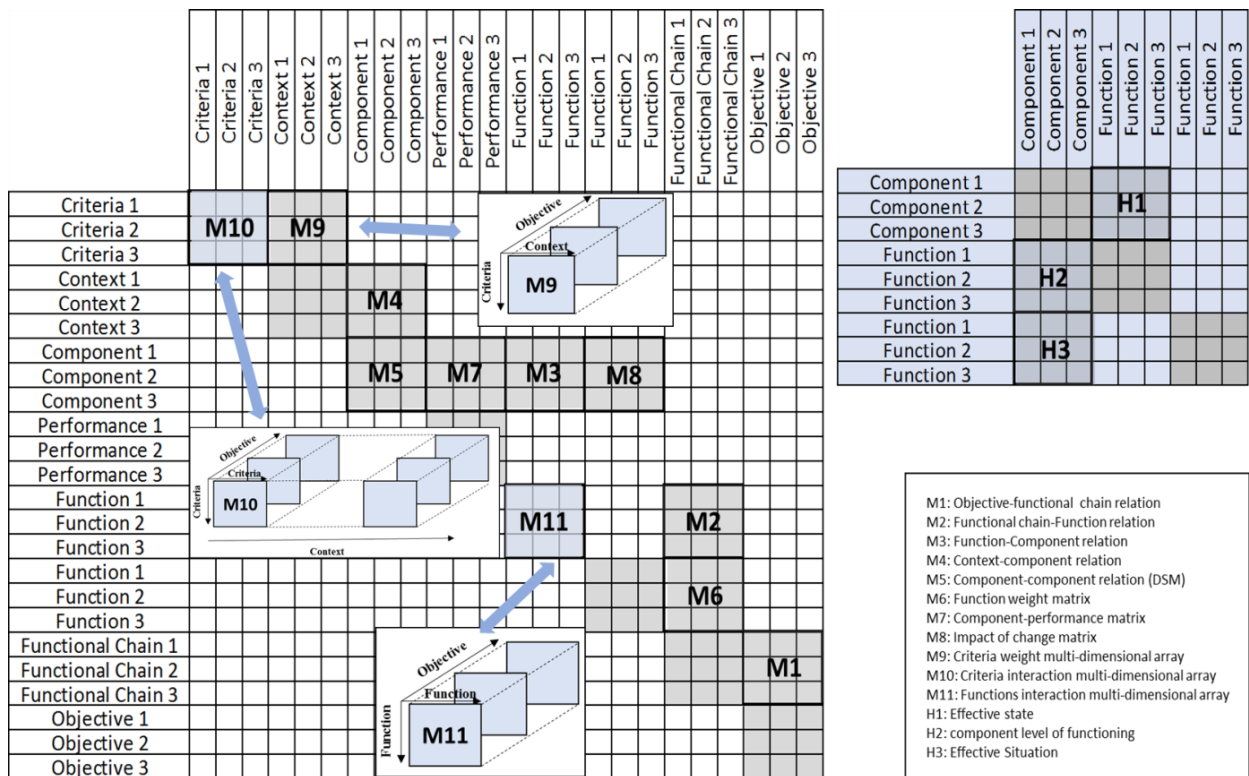


Figure 7-3: The matrix system used in MBSysRec

The information stored within the matrix system has different types and represent different elements regarding the engineering and the operations model (obtained from the health and usage monitoring system).

Table 7-1 summarizes the matrices in terms of the information type they include, description, size, and source. Figure 7-4 details the *MBSysRec* steps and the information used in each step (i.e., matrix system). The method is implemented in MATLAB.

Table 7-1: Matrices description

Matrix	Type	Description	Size	Source
M 1	Binary	Describes whether an Objective (O) is realized by a Functional chain (FC)	2D	Engineering
M 2	Binary	Describes whether a Function (F) belongs to a Functional chain (FC)	2D	Engineering
M 3	Binary	Describes whether a component (C) can realize a function (F).	2D	Engineering
M 4	Binary	Describes whether a Component (C) is relevant for a context.	2D	Engineering
M 5	Binary	Describes whether a Component (C) is compatible with another component (C)	2D	Engineering
M 6	Percentages	Represents the weight of each function (F) in the estimation of the functioning level of a configuration realized by the Functional chain (FC)	2D	Engineering
M 7	Percentages	Describe the score of each component (C) on the different performance criteria.	2D	Engineering
M 8	Percentages	Provides the impact of each change that is related the average proportion of work that needed to set up or down a Function (F) by a component (C).	2D	Engineering
M 9	Percentages	Gives the weighting coefficients of each criteria related to the context and objective specific technical priorities and benchmarks.	3D	Engineering
M 10	Percentages	Contains the context and objective dependent criteria interaction indices.	4D	Engineering
M11	Percentages	corresponds to the interaction between the different functions constituting the functional chain.	3D	Engineering
H1	Binary	Provides the effective state reflecting the availability of the components (whether a component can effectively realize a function or not),	2D	Operations
H2	Percentages	Gives the level of functioning of the individual components (“how much” the component is able to realize a function).	2D	Operations
H3	Binary	Provides the effective situation of the instantiated configuration (whether the instantiated components are realizing the required functions or not).	2D	Operations

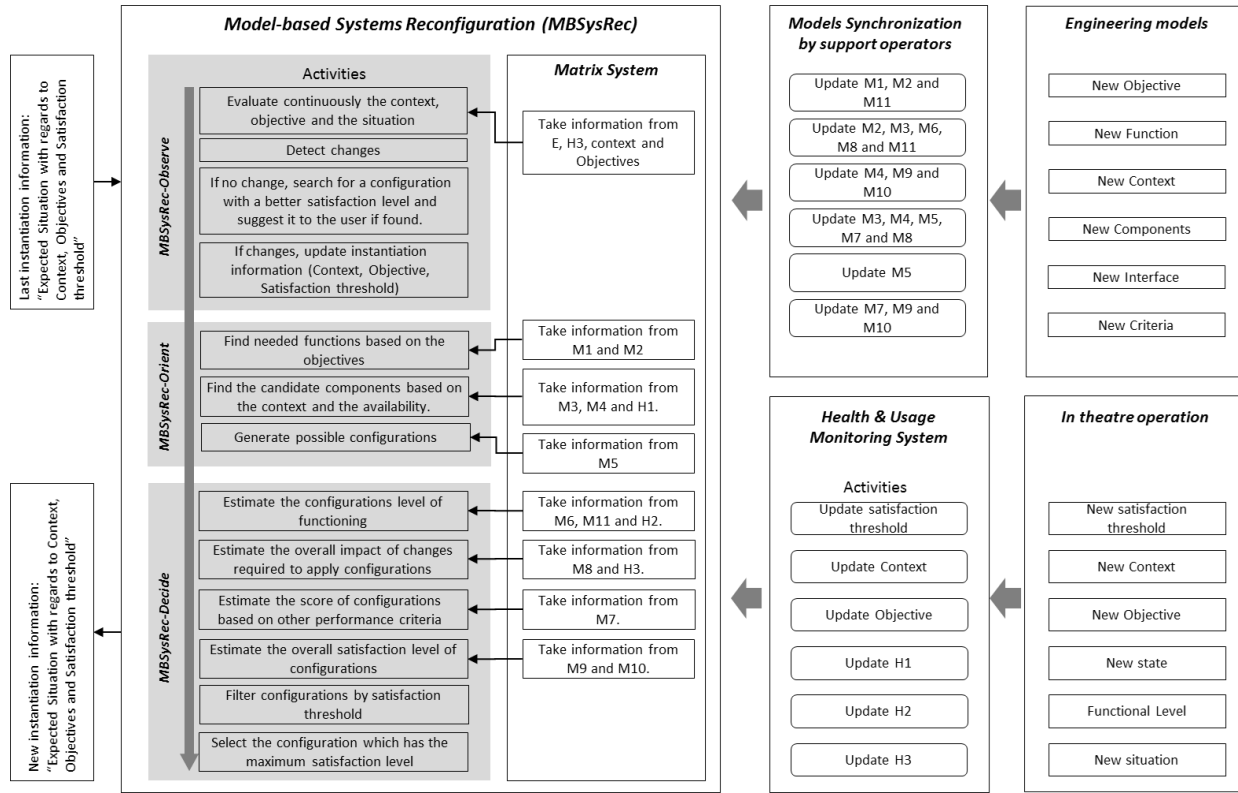


Figure 7-4: MBSysRec steps and link to the matrix system

MBSysRec-Observe (step1) satisfies the first two requirements that were defined previously in section 7.3.1. MBSysRec-Observe aims at detecting changes related to the system operational context, objective or system health. For this purpose, MBSysRec-Observe verifies regularly the context, objective and the effective situation, and compares this information with the last instantiation information (Expected Situation regarding the objective, context and satisfaction threshold) that are stored in memory. When a change is detected, the instantiation information is updated, and the reconfiguration process is triggered.

The information needed in this step includes *Objective, Context, Effective Situation* (stored in H3). This information is reported to MBSysRec-Observe thanks to a health and usage monitoring system (which is considered an enabling system). In our proposal, the user is asked to enter the *Objective* and *Context* via a dialog box, while the Effective Situation is read directly from matrix H3.

In the case of no-change detection during the observation phase, MBSysRec-Observe will search for another configuration satisfying both the context and the objective, and that is better in terms of satisfaction level. This can be done with collaboration with other functions of MBSysRec. If such an option is found, MBSysRec-Observe will suggest it to the user.

MBSysRec-Orient (step 2) satisfies the third requirement of the reconfiguration support concerning generating possible configurations regarding the operational context and objective. MBSysRec-Orient integrates sensor data so that the solution space is investigated among only available resources. First, MBSysRec-Orient searches the functions that the system needs to realize by browsing the matrix system for the functional chain that is linked to the specific objective, and then by finding the functions that are linked to this functional chain. Next, in order to allocate the captured functions, the set of candidate components (resources) should be found. The first step in defining the candidate components is to filter the overall set of components (M3), regarding their relevance in the considered context (M4). The second step is to integrate sensors data concerning components availability stored in H1 and use them to filter the overall set of candidate components.

These filtering steps are done via the matrix multiplication in (1). The operator (\cdot^*) is a notation for element-wise multiplication, and the resulting matrix is referred to as $M3'$.

$$M3' = M4_{Context}^T \cdot^* M3 \cdot^* H1 \quad (1)$$

Finally, MBSysRec-Orient generates configurations by relying on the updated set of available resources $M3'$. For each required function, MBSysRec-Orient will find the possible resource candidates in $M3'$ and systematically generate all possible configurations. For this purpose, we proposed an algorithm (see (2) to (4)), which we adapted from the work of Bryant et al. (Bryant et al., 2005). In (4), the operator (\cdot^*) is a notation for matrix multiplication, and the operator ($\overrightarrow{\cdot}_{M3'}$) is a vector from matrix $M3'$. For each configuration, the compatibility between different components is verified by using the information stored in M4. The result of the orientation phase is a set of matrices representing the candidate configurations.

$$Construct FC = \{f_1, f_2, \dots, f_n\} \quad (2)$$

$$\forall f_i \in FC \text{ get the corresponding line in } [M3'], \text{ where } i = \{1, \dots, n\} \quad (3)$$

$$\forall f_i \in FC \text{ do } \overrightarrow{M3'_{f_i}} \cdot^* \overrightarrow{M3'_{f_{i+1}}}, \text{ where } i = \{1, \dots, n-1\} \quad (4)$$

MBSysRec-Decide (step 3) addresses the last two requirements of MBSysRec and consists of two main sub-phases: evaluation and selection. The evaluation phase aims at calculating the overall satisfaction level of all possible configurations generated in the previous step. The configuration's overall satisfaction level is an aggregation of 1) the level of functioning of a configuration, 2) the impact of changes when applying this configuration, 3) other performance criteria that are specific to a domain or a problem. In this section, we will detail the overall evaluation process.

The estimation of a configuration level of functioning is based on the aggregation of the level of functioning of each component participating in the candidate configuration (z_i). Z_i is calculated by multiplying, element-wise, the matrix corresponding to the configuration under evaluation by the level of functioning of all the available resources (see (5)). The information considering the functioning level of each resource is provided by the health and usage monitoring system and is stored in H2.

$$\text{Level of functioning of components in configuration } X = \text{Configuration } X \text{ matrix} \cdot H2 \quad (5)$$

For the determination of the level of functioning aggregation model, we use the multi-criteria Choquet integral method (Labreuche, 2011b). We selected this method as it allows considering functions inter-dependency and interactions while other aggregation models, such as the weighted sum or the ordered weighted sum, fail to address this issue (Grabisch, 2013). Functions inter-dependency and interaction mean that the way the score of each function is handled depends on the value of the scores of the other functions. The Choquet integral is the most general approach as it collapses to a weighted sum when no interaction is present. The Choquet integral method is based on the disaggregation and aggregation phases. The disaggregation phase aims at finding the two additive Choquet integral capacities (i.e., the Shapely index (weight index) and the interaction index) using a game-based algorithm. Employing these indices allows handling the notion of functions' importance as well as functions redundancy and complementarity. These two indices can be used to estimate the configuration functioning level, as indicated in (6). The Shapely index (v_i) represents the importance of the function in realizing the objective (stored in M6), and the interaction index ($I_{i,j}$), stored in M11, gives insights on how these functions interact for the considered objective.

$$\text{Estimation of configuration } X \text{ level of functioning} = \sum_{i=1}^n v_i z_i - \sum_{\{i,j\}} \frac{I_{i,j}}{2} |z_i - z_j| \quad (6)$$

The overall impact of changes represents the total effort required to set up or down the components (resources), leading to the application of the configuration under evaluation (target configuration). The calculation of the impact of change is mainly based on M8, which incorporates the components' individual impact of change estimated by domain experts (IoC). In addition to M8, MBSysRec-Decide uses information about the effective situation (source configuration) and the potential expected situation (target configuration). The effective situation is provided by the health and usage monitoring system (stored in H3), while the expected situations correspond to all possible configurations generated during the MBSysRec-Orient step. The total number of changes NoC is calculated based on the distances between the source (H3) and target configuration (X) using (7) (where C is the total number of components included in the system). The impact of change for a configuration X is given by (8) (where F is the total number of functions realized

by the system). Equation 8 has two parts. The first one is related to non-impacted functions (2F-NoC), divided by 2 F for normalization. 2F corresponds to the total number of possible changes as a function can either be activated or/and deactivated. The second part is related to the sum of the Impact of changes related to component activation and deactivation (for each impacted function), divided by 2 F for normalization. A high score on the impact of change indicator means that either a few changes are needed to achieve the target configuration, or the required changes are relatively easy to implement.

$$\text{NoC} = \sum_{i=1}^F \sum_{j=1}^C |H3 - X| \quad (7)$$

$$\text{Impact of change of configuration } X = \frac{2F - \text{NoC} + \sum_{i=0}^C \text{IoC}}{2F} \quad (8)$$

Other specific performance criteria can be required as they play an important role in the evaluation process. The domain or application specific criteria are defined in the system data model. The performance of each component related to these specific criteria is given by experts and is stored in M7. The estimation of a configuration X performance for a criterion C is given by (9). In this equation, F^* corresponds to the number of functions that are concerned by the criterion C, and S is the satisfaction of the function F^*_i by the corresponding component in configuration X.

$$\text{Performance of a configuration } X \text{ on criteria } C = \frac{1}{F^*_i} \times \sum_{i=1}^F S(F^*_i \text{ by a Component in } X) \quad (9)$$

The final step of the evaluation phase is the overall satisfaction level calculation. To this end, we also adopt the multi-criteria Choquet integral method (Labreuche, 2011b) as it allows considering criteria interdependency and interactions. As explained previously, the Choquet integral method covers the most general case (criteria-dependency and interaction) and collapses to a weighted sum when no interaction is present. The weights v_i and the interaction indices $I_{i,j}$, are context and objective dependent, and their values are given by experts and stored in the multi-dimensional arrays M9 and M10. For a total number of criteria n, the overall satisfaction level of a configuration X is calculated using (10) (where z_i and z_j are the scores of a configuration X on any criteria "i" and "j" respectively).

$$\text{Overall satisfaction level estimation of a configuration } X = \sum_{i=1}^n v_i z_i - \sum_{(i,j)} \frac{I_{i,j}}{2} |z_i - z_j| \quad (10)$$

In the selection phase, first the configurations are filtered based on the user-defined satisfaction threshold. Next, the configuration having the maximum satisfaction threshold is chosen. This configuration is returned as the new expected situation for the system.

7.3.3. MBSysRec: Information Sources for the Matrix System

As explained earlier, the data used within MBSysRec are coming from both phases; engineering and operations. The health and usage monitoring system provides information about the operated system. This information is used to obtain the matrices H1, H2, and H3. In the engineering phase, on the other hand, the engineering teams and domain experts work together to define the models necessary for the reconfiguration process. These models define the structure of the deployed system, the functions provided by each constituting component (resource), and the applied management policies specific to a domain such as Land, Air, Naval, or joint domains, providing information on the relevance of the used components for the considered contexts. These models are maintained synchronized with the living system with the help of support operators. Models are updated to consider systems evolutions, including resources commissioning and decommissioning as well as the evolution of the applied policies. Information from domain experts is used to construct the matrices from M1 to M11. In the following we explain how expert knowledge is transformed into percentages and stored in the matrices from M6 to M11.

In the MBSysRec method, the satisfaction levels, stored in M7, are defined as “interval scales” or percentages to describe “how much” a component satisfies a criterion. Using interval scales allows to aggregate the satisfaction levels estimated by experts and the observed level of functioning. The values used for representing satisfaction range from 0.1 (for very inadequate solution) to 1 (for ideal solution). The component assessment is done during workshops involving domain experts who provide linguistic terms to estimate the satisfaction levels. In the MBSysRec method, we use an adapted quantification of the linguistic terms given by experts (see Table 7-2) based on (Fiod-Neto and Back, 1994). In case of non-relevance of a criterion for certain components the value “0” is given. Hence, the overall satisfaction level is not penalized since the calculation is done based on non-zero values.

Table 7-2: Linguistic terms for performance estimation

Linguistic terms	Satisfaction level
Very inadequate solution	0.1
Weak solution	0.2
Tolerable solution	0.3
Adequate solution	0.4

Satisfactory solution	0.5
Good solution with few drawbacks	0.6
Good solution	0.7
Very good solution	0.8
Solution better than requirements	0.9
Ideal solution	1

Furthermore, to quantify the impact of a change (M8), we rely on expert judgement to estimate the effort needed to implement it. This has been adapted from software development effort estimation methods (Jorgensen, 2005). The impacts of changes are also represented in percentages or interval scales and assigned values between 0.1 to 1. Ten values which are used to quantify the impact of a change range from 0.1 to 1, are listed in Table 7-3. While, the value “1” refers to a very low or negligible effort, the value “0.1” refers to a very high and considerable effort. The “0” value is reserved for non-authorized changes.

Table 7-3: Linguistic terms for the impact of change estimation

Linguistic terms	Impact of a change
Very high or considerable	0.1
Very high	0.2
high	0.3
significant	0.4
tolerable	0.5
accepted	0.6
medium	0.7
low	0.8
Very low	0.9
Very low or negligible	1

Expert judgment has also been used to obtain the two aggregation models included within the MBSysRec method. The Shapely index stored in M6 and M9 corresponds to the weight of functions and criteria, respectively. The weights are given as positive coefficients and sum to 1. M6 contains the weights of the different functions participating in the functional chain indicating “how important” these functions are in realizing the different objectives. M9 is a three-dimensional array that encloses the weights, allowing the different criteria to be prioritized depending on the context and the objective. A Shapely index above $1/n$, where n is the number of considered functions/criteria, corresponds to a function/criterion that is more important than average. A high “Shapely index” indicates that the corresponding function/criterion has a

veto among the others. The functions/criteria interaction indices are stored in M11 and M10, respectively. The M11 matrix is a multi-dimensional array comprising the interaction indices between functions that depend on the functional chain in which these functions participate (stored in layers of the multi-dimensional array called pages). M10 is a four-dimensional array that contains the context and objective-dependent interaction indices between criteria (similarly, these indices are stored in pages of the multi-dimensional array). A positive interaction index represents the complementarity between a pair of criteria/function. This complementarity means that both criteria/ functions need to be simultaneously satisfied to get a higher score. On the other hand, a redundant pair of criteria/functions has a negative interaction index indicating that a configuration can reach a high satisfaction level if any of these criteria/ functions is satisfied. An interaction index which equals zero indicates that there is no interaction between the corresponding criteria/ function.

7.4. Case Study: Search And Rescue

In order to demonstrate MBSysRec, we use the search and rescue case study. In section 7.4.1, we describe the SAR case study. The following sections illustrate the MBSysRec method via a reconfiguration scenario from the search and rescue case study.

7.4.1. Case Study Description

As Search and Rescue (SAR) aims at assisting persons and property in potential or actual distress in a maritime or land context. Maritime SAR provides assistance to save sailors and passengers in distress, or the survivors of downed aircraft using available resources. Maritime SAR organizations deploy lifeboats, aircraft, and surface vessels to rescue victims and return them to land when a distressed or missing vessel is located. In addition to that, the maritime SAR stations are equipped with other components, also called resources, (assets, services, and operators) that can be directed towards this end. Moreover, the operational contexts of SAR missions are variable; in particular, the target and the environment. For instance, SAR missions can be operated during day or night, rainy or sunny weather dealing with small to large objects. For the sake of simplicity, in this case study, we only consider small, medium, and large objects in both day and night environments. For the same reason, only two objectives are considered Search and Rescue. *Table 7-4* provides a global overview of the problem considered.

In general, the SAR coordinator in the command and control C2 center is confronted with situations in which he/she needs to quickly direct the available resources to realize objectives. When looking to the different resources and their corresponding level of functioning levels as well as the other criteria, diverse combinations or options are possible and should, then, be considered. Due to the increased number of resources and correspondingly their possible combinations, the coordination task is considered costly and time-consuming. This task also requires knowledge concerning the resources and their availabilities and abilities to provide functions. The MBSysRec method can assist the SAR coordinator in the C2 center by generating all possible configurations and recommending a configuration with a maximum satisfaction level.

Table 7-4: SAR problem data

6 Contexts	Small target night
	Small target day
	Medium target night
	Medium target day
	Large target night
	Large target day
2 Objectives	Search
	Rescue
2 Functional chains	FC 1 - FC 2
9 Functions	Reach
	Recover
	Apply first aid
	Transport
	Move
	Com with Victim
	Com with base
	Observe
	Operate
31 Components	Swimmer (3 instances)
	Life preserver (3 instances)
	Pilot (3 instances)
	Driver (3 instances)
	First aid agent (3 instances)
	Aircraft (3 instances)
	Boat (3 instances)
	Communication mean with victims (2 instances)
Communication mean with base (2 instances)	

	Camera (3 instances)
	Operator (3 instances)
	<hr/>
	Cost
4 criteria	Resolution
	Impact of change
	Level of functioning
	<hr/> <hr/>

7.4.1.1. Step 1: MBSysRec-Observe

During operation, the SAR coordinator sets an objective for the SAR mission and defines its operational context. The satisfaction level is also defined. In this step, the coordinator is asked to input the required instantiation information in a dialog box, as indicated in *Figure 7-5*. Moreover, information about the effective situation is needed.

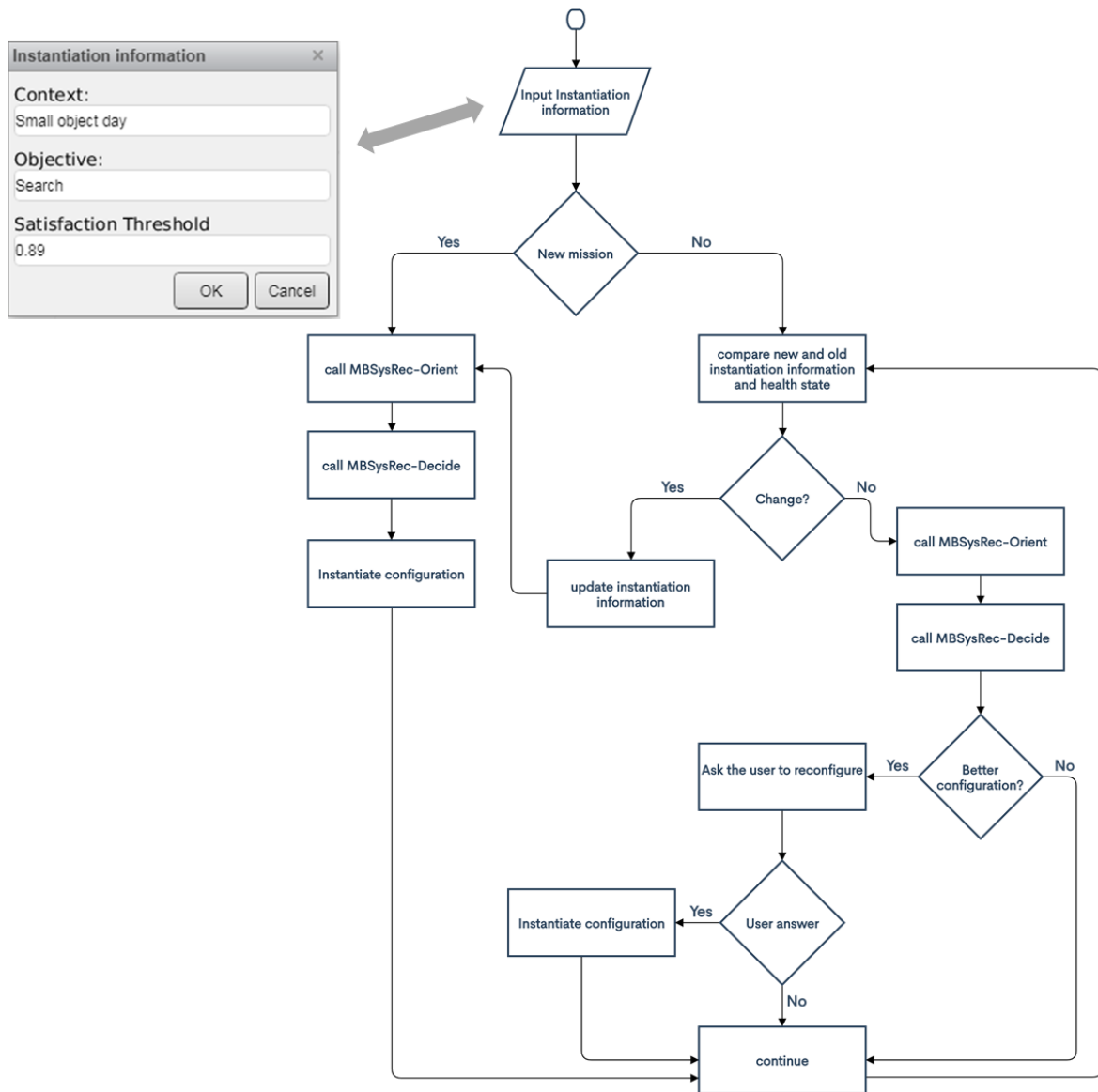


Figure 7-5: Instantiation information input to MBSysRec and MBSysRec-Observe flowchart

This information is provided by the health and usage monitoring system and collected to update the matrix H3. This phase aims to use MBSysRec to update the instantiation information in terms of operational context and objective, and the satisfaction threshold with regards to the effective situation. MBSysRec-Observe has three main activities: 1) setting up instantiation information for new missions, 2) update instantiation information if evolutions are detected, and 3) Proposing configuration with a better satisfaction level. The flowchart representing these activities is shown in *Figure 7-5*.

As illustrated in the above flowchart, MBSysRec performs a comparison between the old and the new instantiation information. If no change is detected, MBSysRec performs an additional comparison between the expected situation and the effective one. The expected situation corresponds to an instantiation matrix and is stored in the system from the last instantiation information, while the effective situation is a matrix provided by the health and usage monitoring system and stored in H3. The two matrices are then superimposed to check for changes. In this step, as well as the following ones, we will consider only new instantiation with the *Small object day* as context, *Search* as objective, and a satisfaction threshold of 0.89. Completing scenarios will be provided in the last section. Once the instantiation information is entered for a new mission, the MBSysRec-Observe will call MBSysRec-Orient function. In the next section, we detail the different activities of the MBSysRec-Orient function.

7.4.1.2. Step 2: MBSysRec-Orient

For new instantiation information, this step aims to find the solution space that is relevant for both the context and the objective. The solution space should, therefore, be searched, based on only the available resources (components).

Existing information related to objectives is stored in the MBSysRec matrix system. When necessary, the support operator can update M1 and M2 to add or remove objectives and functional chains. The objective specified in the instantiation information is, then, used to allocate the required functional chain by using the Objective - Functional chain relation stored in M1 (see *Figure 7-6*). In M1, “1” means that the objective can be realized by the functional chain listed in the corresponding row. In our scenario, the Objective *Search* is realized by the Functional chain *FC1*. MBSysRec-Orient, afterwards, propagates this information into M2 (*Figure 7-6*) to find the functions that are allocated to this functional chain. In our example, the functions: “Transport”, “Move”, “Com with Victim”, “Com with base”, “Observe”, and “Operate” participate in the functional chain *FC1*.

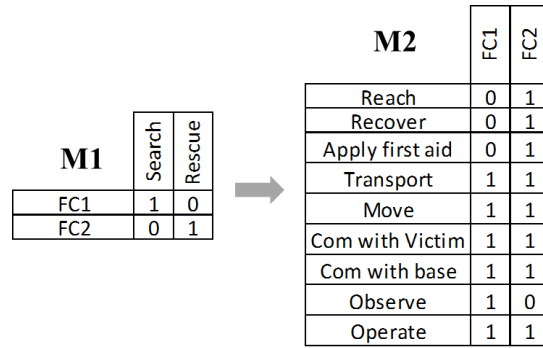


Figure 7-6: Objective-Functional chain relation and Functional chain - function relation matrices

The overall description of system components and the functions they can realize is provided by the Function-Component relation and is stored in M3 (see Figure 7-7). In this matrix, “1” represents the ability of a component to realize the corresponding function. The swimmer-A3, for instance, can provide both “Reach” and “Apply first aid” functions. However, not every component in this matrix is available and relevant for the considered context. Thus, MBSysRec-Orient performs a two-stage filtration process to find the candidate components based on the availability.

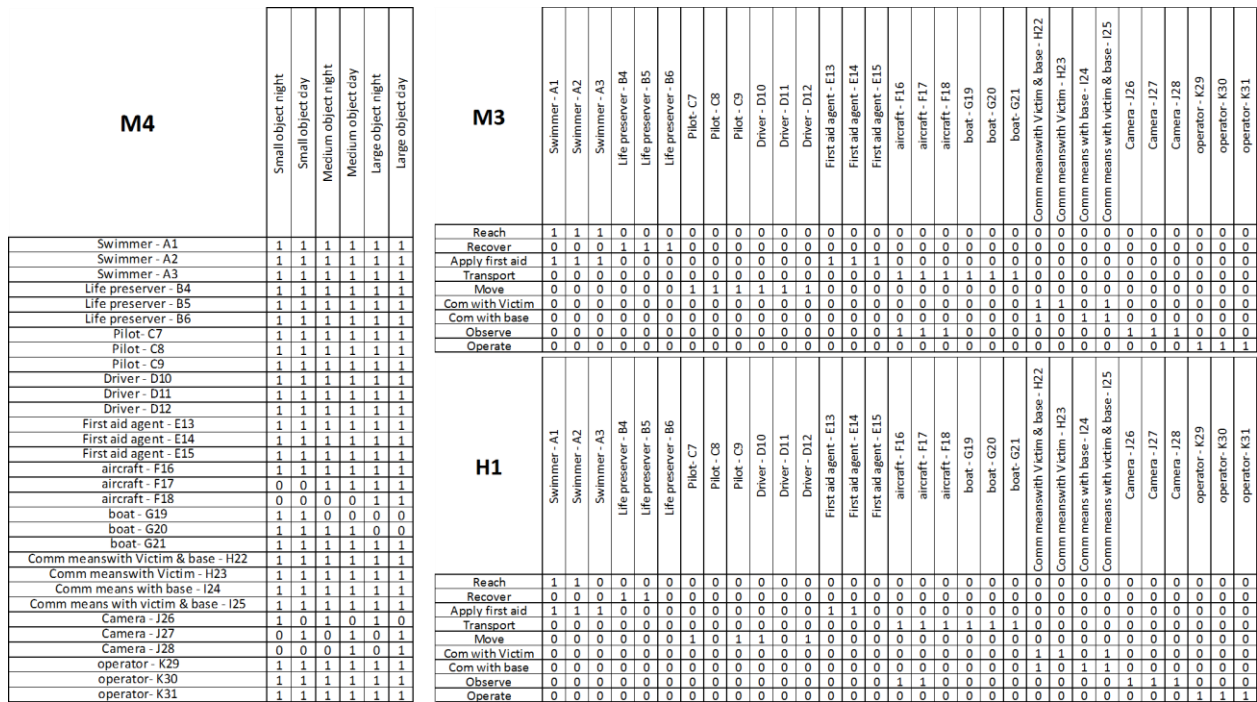


Figure 7-7: Context-Component relation, Function-Component relation, and availability information

In order to define the relevant set of components depending on the context, MBSysRec-Orient uses the context specified in the instantiation information to find the corresponding relevant set via the Context – Component relation, which is stored in M4 (see *Figure 7-7*). In the considered context “Small object day”, all resources (components) are relevant except for aircraft F17 and F18, camera J26 and J28. These resources have a “0” value in the column corresponding to the “Small object day” context.

After defining the relevant set, MBSysRec-Orient will now collect information about resource availability. This information is provided by the health and usage monitoring system and is stored in H1 (see *Figure 7-7*). In this matrix, “1” indicates that a resource is available to perform the corresponding function. At the moment MBSysRec-Orient collects this information, the resource “Swimmer A3”, for instance, is available only for the “Apply first aid” function and not the “Reach” function. In order to find the updated set of candidate components M3’ (see *Figure 7-8*) we can use equation (1).

M'3	Swimmer - A1	Swimmer - A2	Swimmer - A3	Life preserver - B4	Life preserver - B5	Life preserver - B6	Pilot - C7	Pilot - C8	Pilot - C9	Driver - D10	Driver - D11	Driver - D12	First aid agent - E13	First aid agent - E14	First aid agent - E15	aircraft - F16	aircraft - F17	aircraft - F18	boat - G19	boat - G20	boat - G21	Comm meanswith Victim & base - H22	Comm meanswith Victim - H23	Comm means with base - I24	Comm means with victim & base - I25	Camera - J26	Camera - J27	Camera - J28	operator - K29	operator - K30	operator - K31
	Reach	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Recover	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apply first aid	1	1	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Transport	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
Move	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Com with Victim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0
Com with base	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0
Observe	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Operate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Figure 7-8: Updated Function - component relation M'3

The last activity of the MBSysRec-Orient function is to generate the possible configurations based on the results from the previously explained activities M'3, the functional chain, as well as components compatibilities. Information about the components' compatibilities is given by the component - component relation and is stored in M5 (see *Figure 7-9*). The first row of M5 indicates "1" for compatibility and "0" for non-compatibility which means that "Swimmer -A1" is compatible with all other components except for "Swimmer – A2" and "Swimmer - A3.

M5	Swimmer - A1	Swimmer - A2	Swimmer - A3	Life preserver - B4	Life preserver - B5	Life preserver - B6	Pilot - C7	Pilot - C8	Pilot - C9	Driver - D10	Driver - D11	Driver - D12	First aid agent - E13	First aid agent - E14	First aid agent - E15	aircraft - F16	aircraft - F17	aircraft - F18	boat - G19	boat - G20	boat - G21	Comm meanswith Victim &	Comm meanswith Victim -	Comm means with base - I	Comm means with victim &	Camera - J26	Camera - J27	Camera - J28	operator - K29	operator - K30	operator - K31	
Swimmer - A1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Swimmer - A2	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Swimmer - A3	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Life preserver - B4	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Life preserver - B5	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Life preserver - B6	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Pilot - C7	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1
Pilot - C8	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1
Pilot - C9	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	0	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1
Driver - D10	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1
Driver - D11	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
Driver - D12	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	0	0	0	1	1	0	0	0	1	1	1	1	1	1	1	1	1
First aid agent - E13	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
First aid agent - E14	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
First aid agent - E15	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
aircraft - F16	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
aircraft - F17	1	1	1	1	1	1	1	0	1	0	0	0	1	1	1	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
aircraft - F18	1	1	1	1	1	1	0	1	0	0	0	0	1	1	1	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1
boat - G19	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
boat - G20	1	1	1	1	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
boat - G21	1	1	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
Comm meanswith Victim & base - H22	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
Comm meanswith Victim - H23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Comm means with base - I24	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Comm means with victim & base - I25	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
Camera - J26	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Camera - J27	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Camera - J28	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
operator - K29	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
operator - K30	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
operator - K31	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 7-9: Component - Component relation for the SAR system

To realize the functional chain FC1 (corresponding to the Search objective), the functions "Transport", "Move, "Communication with Victim", "Communication with base", "Observe', and "Operate" need to be allocated to components. Therefore, by taking one component for each function from M'3 and by respecting the components' compatibility stored in M5, all possible configurations are generated (see (2), (3) and (4)). For this SAR scenario, the total number of possible configurations is 147 which are combinations of "Pilot-C7", "Drivers -D10 and D12", "Communication means with victim H22, H23, and I25", "Communication means with base H22, I24, and I25", "Camera -J27", "Aircraft- F16" "Operators – K29, K30, and K31" and "Boats- G19, G20, and G21".

7.4.1.3. Step 3: MBSysRec-Decide

In this section, we will detail the calculation of configurations scores on the different criteria: configuration level of functioning, the impact of change, cost, resolution. The aggregation of these scores to find the overall satisfaction level is also explained.

- Configurations level of functioning estimation

The estimation of a configuration level of functioning is based on the aggregation of the level of functioning of each component participating in the candidate configuration (see (5)). *Figure 7-10* shows the functioning levels for all resources in the system (H2), and the matrix (X) represents the configuration under consideration. For the determination of the aggregation model, we used the two additive capacities of the Choquet integral (see (6)). The Shapely index (v_i), and the interaction index ($I_{i,j}$) are stored in M6 and M11 respectively (see *Figure 7-11*).

H2	Swimmer - A1	Swimmer - A2	Swimmer - A3	Life preserver - B4	Life preserver - B5	Life preserver - B6	Pilot - C7	Pilot - C8	Pilot - C9	Driver - D10	Driver - D11	Driver - D12	First aid agent - E13	First aid agent - E14	First aid agent - E15	aircraft - F16	aircraft - F17	aircraft - F18	boat - G19	boat - G20	boat - G21	Comm meanswith Victim & base - H22	Comm meanswith Victim - H23	Comm means with base - I24	Comm means with victim & base - I25	Camera - J26	Camera - J27	Camera - J28	operator - K29	operator - K30	operator - K31					
	Reach	1	0.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Recover	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Apply first aid	1	0.9	1	0	0	0	0	0	0	0	0	0	0	1	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Transport	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Move	0	0	0	0	0	0	1	0.8	1	1	0.75	0.88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Com with Victim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9	1	0	0.97	0	0	0	0	0	0	0	0	0	0
Com with base	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9	0	1	0.97	0	0	0	0	0	0	0	0	0	0
Observe	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Operate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Configuration X	Swimmer - A1	Swimmer - A2	Swimmer - A3	Life preserver - B4	Life preserver - B5	Life preserver - B6	Pilot - C7	Pilot - C8	Pilot - C9	Driver - D10	Driver - D11	Driver - D12	First aid agent - E13	First aid agent - E14	First aid agent - E15	aircraft - F16	aircraft - F17	aircraft - F18	boat - G19	boat - G20	boat - G21	Comm meanswith Victim & base - H22	Comm meanswith Victim - H23	Comm means with base - I24	Comm means with victim & base - I25	Camera - J26	Camera - J27	Camera - J28	operator - K29	operator - K30	operator - K31							
	Reach	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Recover	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Apply first aid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Transport	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Move	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Com with Victim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Com with base	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Observe	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Operate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 7-10: Components levels of functioning (H2) and configuration X matrices

For this scenario, if we consider the configuration 53, which contains "Driver - D10", Boats- G19", " Communication means with victim H23", "Communication means with base I24", "Camera -J27", and "Operator – K30" then the configuration matrix of X is shown in *Figure 7-10*. Applying (5), we can get the level of functioning of configuration X components (z_i in *Figure 7-11*). Then by applying (6) with v_i and $I_{i,j}$ values of the functional chain FC1 that is related to the "Search" objective (column 1 of M6 and page 1 in the multi-dimensional array M11), the estimation of configuration X overall level of functioning is found

and is equal to 0.965. MBSysRec-Decide repeats this calculation for all possible configurations. The results are shown in *Figure 7-12*, and the values vary between 0.774 and 0.996. The positive interaction indices $I_{i,j}$ in M11 means that the functions are complementary. Moreover, the most complementary functions are "Transport" and "Move". From M6, we can notice that both functions have equal weights; thus, there is no veto among these functions. However, some functions are more important than others in realizing the "Search" objective.

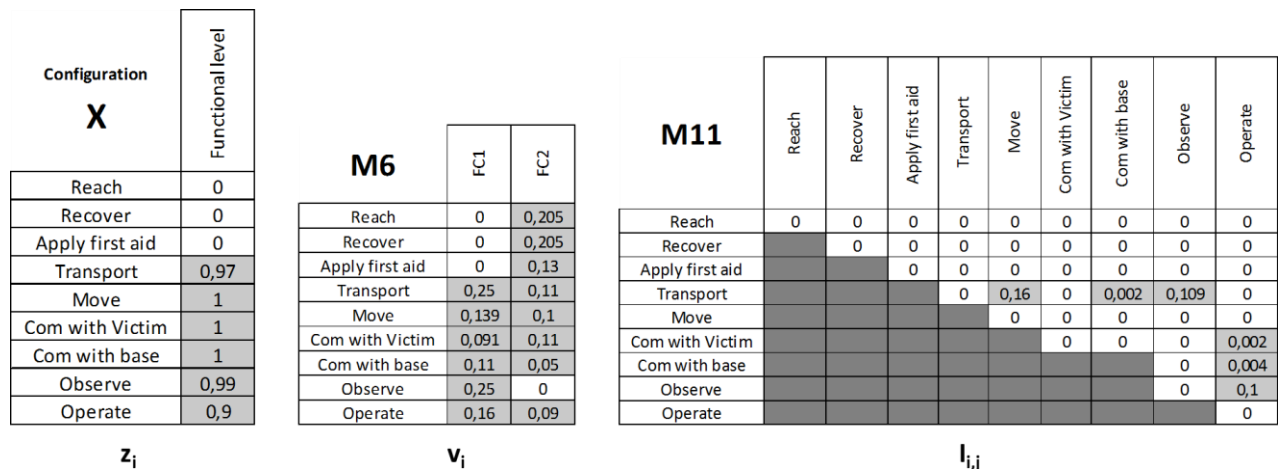


Figure 7-11: Configuration X functional level, Functions weights(M6) , and Function interaction indices (M11) matrices

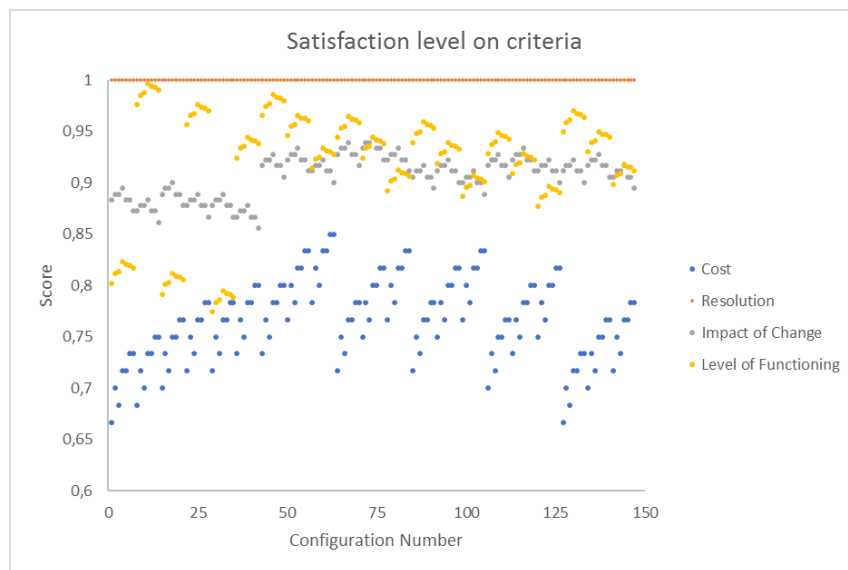


Figure 7-12: Score on each criterion of all possible configurations

Next, the MBSysRec-Decide will estimate the impact of change towards each of the configurations in the solution space. We detail this calculation in the next sub-section.

- Impact of change estimation

The impact of change is calculated based on the total number of functions that can be realized by the system (F), the number of changes (C), and the impact of individual changes (IoC) stored in M8 (see *Figure 7-13*). The total number of changes is calculated using equation (7). Next, the impact of change estimation can be found using equation (8).

M8	Swimmer - A1	Swimmer - A2	Swimmer - A3	Life preserver - B4	Life preserver - B5	Life preserver - B6	Pilot - C7	Pilot - C8	Pilot - C9	Driver - D10	Driver - D11	Driver - D12	First aid agent - E13	First aid agent - E14	First aid agent - E15	aircraft - F16	aircraft - F17	aircraft - F18	boat - G19	boat - G20	boat - G21	Comm meanswith Victim & base - H22	Comm meanswith Victim - H23	Comm means with base - I24	Comm means with victim & base - I25	Camera - J26	Camera - J27	Camera - J28	operator - K29	operator - K30	operator - K31	
	Reach	0,6	0,8	0,4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Recover	0	0	0	0,8	1	0,4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apply first aid	0,8	0,9	0,5	0	0	0	0	0	0	0	0	0	0,7	0,8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Transport	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,3	0,4	0,5	0,9	0,7	0,6	0	0	0	0	0	0	0	0	0	0	0
Move	0	0	0	0	0	0	0,4	0,5	0,7	0,6	0,7	0,8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Com with Victim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,8	0,9	0	0,7	0	0	0	0	0	0	0
Com with base	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,8	0	0,9	0,7	0	0	0	0	0	0	0
Observe	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,8	0,8	0,8	0	0	0	0	0	0	0	0,7	0,6	0,7	0	0	0	0
Operate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,8	0,9	0,7	

Figure 7-13: Impact of change matrix

In the same scenario, if we consider applying configuration 53, then the source configuration corresponds to a zero matrix, and the target configuration is the same as the one shown in *Figure 7-10*. In this case, F, C, and IoC equal 9.6, and 4.8, respectively, and the impact of change is equal to 0.933. The 147 possible configurations undergo the same evaluation; the results of this evaluation are shown in *Figure 7-12*. The impact of change estimation varies between 0.855 and 0.944.

- Specific performance estimation

For our application (SAR), we consider both resolution and cost as criteria. In this matrix, “0” means that the resolution is not relevant for the considered component and thus, will be eliminated from the estimation of the configuration resolution. The performance of the different SAR components with regards to these criteria is stored in M7 (see *Figure 7-14*).

M7	cost	Resolution
	Swimmer - A1	0,6
Swimmer - A2	0,8	0
Swimmer - A3	1	0
Life preserver - B4	0,8	0
Life preserver - B5	0,9	0
Life preserver - B6	0,9	0
Pilot- C7	0,7	0
Pilot - C8	0,9	0
Pilot - C9	1	0
Driver - D10	0,7	0
Driver - D11	0,8	0
Driver - D12	0,6	0
First aid agent - E13	0,9	0
First aid agent - E14	0,8	0
First aid agent - E15	1	0
aircraft - F16	0,6	1
aircraft - F17	0,8	0,9
aircraft - F18	1	0,7
boat - G19	0,9	0
boat - G20	0,8	0
boat- G21	0,6	0
Comm meanswith Victim & base - H22	0,7	0
Comm meanswith Victim - H23	0,9	0
Comm means with base - I24	0,8	0
Comm means with victim & base - I25	0,9	0
Camera - J26	0,5	0,9
Camera - J27	0,7	1
Camera - J28	0,9	0,7
operator - K29	0,7	0
operator- K30	0,9	0
operator- K31	1	0

Figure 7-14: Component performance matrix

In the considered SAR scenario, the resolution criterion is only considered for the “Observe” function. Consequently, only the components realizing this function have non-zero values in M6. The cost, on the other hand, has been provided for all components. For configuration 53, used in the previous evaluation steps, the configuration resolution is equal to the resolution of “Camera – J27”, which has been chosen to realize the “Observe” function. The cost is calculated using the cost of all components in the configuration 53 using equation (9), and is found to be 0.816. MBSysRec-Decide calculates the performance of all possible configurations. The results are shown in *Figure 7-14*. Since the generated configurations use only the “Camera – J27” or the integrated camera in “aircraft - F16” both having a resolution of 1, then all the configurations score 1 when considering this criterion.

- Overall satisfaction level aggregation

For the overall satisfaction level calculation, MBSysRec-Decide needs the information regarding the criteria weights and interactions, which is stored in M9 and M10, respectively (see *Figure 7-15*). The overall satisfaction level is aggregated using equation (10).

Configuration X	Score
cost	0,816
Resolution	1
Impact of change	0,933
Level of Functioning	0,965

z_i

M9	Small object night	Small object day	Medium object night	Medium object day	Large object night	Large object day
cost	0,1	0,2	0,2	0,3	0,2	0,3
Resolution	0,25	0,3	0,1	0,1	0,2	0,1
Impact of change	0,4	0,25	0,4	0,3	0,2	0,2
Level of Functioning	0,25	0,25	0,3	0,3	0,4	0,4

v_i

M10	cost	Resolution	Impact of change	Level of Functioning
cost	0	0,15	0	0
Resolution	0	0	0,3	0
Impact of change	0	0	0	0
Level of Functioning	0	0	0	0

$I_{i,j}$

Figure 7-15: Configuration X score, criteria weights(M9), and criteria interaction indices (M11)

To illustrate this calculation, we consider configuration 53 having the score matrix z_i (see *Figure 7-15*) obtained from the previously explained steps. Then, to apply equation (10), we need to find the v_i and $I_{i,j}$ values that are related to the "Search" objective and the "Small object day" context. The v_i values are extracted from the column "Small object day", which is stored in the page corresponding to the "Search" objective. However, the $I_{i,j}$ values are extracted from the page related to the considered objective and context. In *Figure 7-15*, we only show the pages that are relevant to the "Search" objective for M9, and "Search" objective and the "Small object day" context for M10. The positive interactions between both couples of criteria (Cost and Resolution) and (Resolution and Impact of Change) signify that the criteria forming each couple are complementary and that the criteria Resolution and Impact of Change are the ones having the highest complementarity. This complementarity means that a configuration should satisfy both criteria in order to have a good score. The v_i values shown in M9 for the "Small object day" context indicates that the Resolution is the most important criterion.

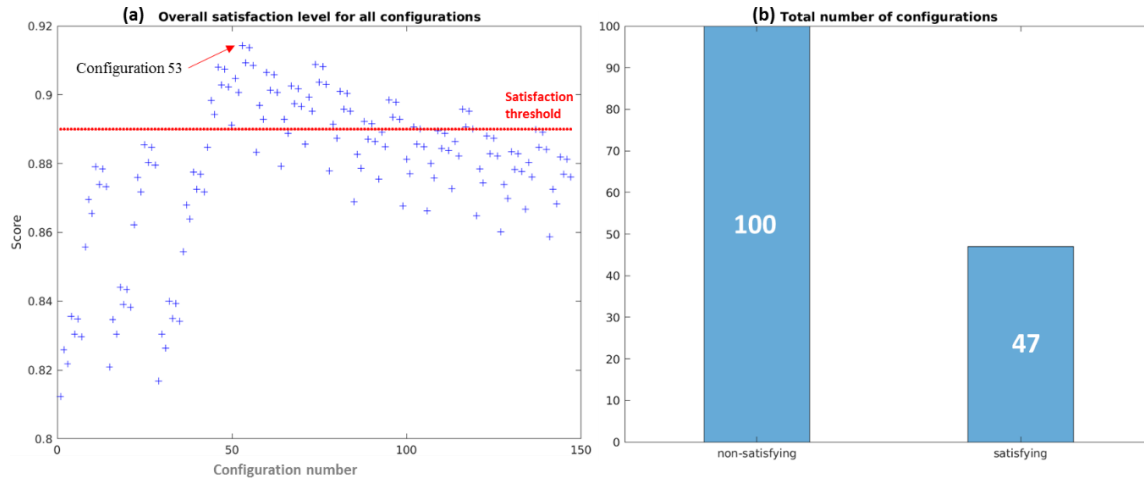


Figure 7-16: Overall satisfaction level and total number of configurations in a "Small object day" context and for a "Search" objective

By applying equation (10), the overall satisfaction level of configuration 53 is found and is equal to 0.914. Next, the MBSysRec-Decide replays this calculation considering another configuration from the 147-solution space. Figure 7-16-a shows the results of this replication and indicates a variation between 0.812 and 0.9143. After that, the MBSysRec-Decide uses the satisfaction threshold provided by the user to filter the solution space into satisfying and non-satisfying solutions (see Figure 7-16-b). Applying a threshold of 0.89 that has been provided by the user gives 47 satisfying configurations and 100 non-satisfying configurations. The final activity of the MBSysRec-Decide is to select a configuration from the satisfying solution space by maximizing over their satisfaction level. In the considered SAR scenario, configuration 53 is the one having the maximum overall satisfaction level and is recommended by the MBSysRec-Decide.

To understand the reason behind choosing configuration 53, we have analyzed the different patterns shown in Figure 7-16-a per function existing in the functional chain (see Figure 7-17). Considering the transport function, Figure 7-17-a shows that the configurations having the maximum overall satisfaction level generally have allocated the transport function to the "boat – G19" component (resource). In fact, in the problem data "boat -G19" has better score on both the impact of change and the cost criteria, and a level of functioning that is better than that of "boat -G20", and not so different from those of "aircraft – F16" and "boat - G21". Similarly, analyzing the rest of the functions in Figure 7-17 shows that components "Driver – D10", "H23", "I24", "Camera – J27", and "Operator – K30" are those maximizing the overall satisfaction level.

Overall configurations satisfaction level/ Analysis per function

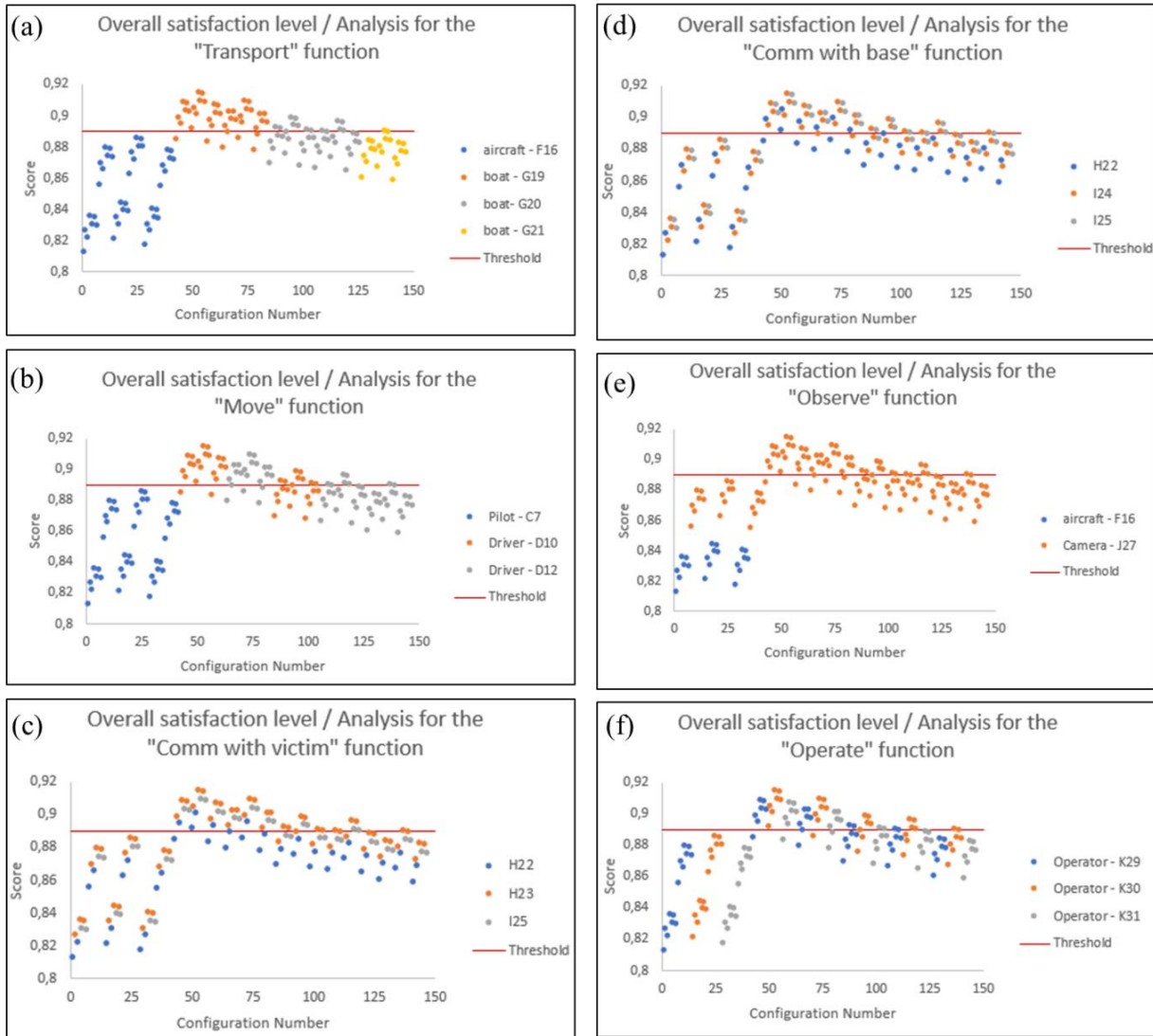


Figure 7-17: Configuration 53 analysis per function

Now, we consider only the two satisfying configurations that maximize the overall satisfaction level (see Table 7-5). Configuration 53 is better than configuration 55 on all criteria except the cost. The only difference between these configurations is the component realizing the "Comm with base" function. In configuration 53, this function is allocated to "I24", which is a digital communication means, while in configuration 55, this function is allocated to the radio "I25". Even though the radio is less expensive (in terms of acquisition and operation) than the digital equipment, the effort needed to set up this equipment is much more important. Thus, the impact of change for the configuration having "I25" is worse than the one having "I24". In the aggregation model that is used in the problem data, the resolution has the highest weight.

However, since the resolution of both configurations is the same, then this criterion becomes insignificant in the overall satisfaction level calculation. Now, considering the cost criterion and its interaction with other criteria, the score of configurations 55 is better than that of configuration 53. Thus, in the overall score calculation, the choice of configuration 55 will be advantageous. However, in the aggregation model, higher weights are given for both the impact of change and the level of functioning criteria. Considering the weights of these criteria as well as the interaction index between the impact of change and the resolution criteria, MBSysRec-Decide will favor configuration 53.

Table 7-5: Summary of configurations 53 and 55 scores and overall satisfaction levels

Configuration Number	Cost	Resolution	Impact of Change	Level of Functioning	Satisfaction Level
53	0.817	1	0.933	0.966	0.9143
55	0.833	1	0.922	0.962	0.9137

7.4.2. Further Scenarios from the SAR Case Study

This section provides the results of further scenarios from the SAR case study and aims to illustrate the variety of cases that MBSysRec can handle. *Figure 7-18* summarizes the considered scenarios. Having performed scenario 1, as previously explained, we have applied the configuration G19-D10-H23-D24-J27-K30. Next, for the second scenario, we consider a context change from small object day to small object night. The MBSysRec-observe detects this change and responds by updating the instantiation information and calling both MBSysRec-Orient and MBSysRec-Decide. Again, 147 different solutions engaging Camera – J26 and aircraft - F16 for the "Observe" function are generated, and among them, only 43 options meet the satisfaction threshold. As the impact of change has the highest weight among other criteria for this context, then MBSysRec-Decide chooses a configuration that maximizes the score of the impact of change criterion.

Scenario Number	Instantiation information	Treated event	Number of configurations		Overall satisfaction level	Configurations
			Satisfying	Non-satisfying		
1	Context: Small object day Objective: Search Satisfaction Threshold: 0,89	Context and Objective change	47	100	0,914	G19-D10-H23-I24-J27-K30
2	Context: Small object night Objective: Search Satisfaction Threshold: 0,89	Context change	43	104	0,926	G19-D10-H23-I24-J26-K30
3	Context: Small object night Objective: Search Satisfaction Threshold: 0,89	Situation change / "Boat - G19" KO	52	53	0,923	G20-D10-H23-I24-J26-K30
4	Context: Small object night Objective: Search Satisfaction Threshold: 0,89	"Boat - G20" degraded	-	-	0,902	G21-D12-H23-I24-J26-K30
5	Context: Small object night Objective: Rescue Satisfaction Threshold: 0,89	Objective change	207	801	0,939	A1-B5-E13-G21-D12-H23-I24-K30

Figure 7-18: Summary of SAR scenarios results

MBSysRec-Observe compares the effective situation that is provided by the health and usage monitoring system to the expected one. The expected situation corresponds to the configuration that is instantiated. In scenario 3, the instantiation information stays unchanged, hence MBSysRec-Observe expects the last instantiated configuration when analyzing the health and usage monitoring system data concerning the effective situation. However, MBSysRec-Observe detects a change related to the failure of the boat - G19 and calls MBSysRec-Orient and MBSysRec-Decide to overcome this situation. Due to this failure, MBSysRec-Orient will only consider "G20", "G21" and "F16" for the "Transport" function; therefore, only 105 configurations are generated, and out of them only 52 are satisfying the threshold. The choice was made among configurations engaging "G20" because they maximize the score on the impact of change criterion, which is the most significant in the considered context. Configurations using "G21" require a change of the component D10, which is not compatible with G20 resulting in a low score on the impact of change criterion. Similarly, configurations with "F16" require further changes. Moreover, "F16" needs a considerable effort compared to the effort necessary to set up "G20" or "G21".

If MBSysRec-Observe detects no change regarding the context, objective, or the situation, it will verify the availability of a better configuration. Better configurations can exist if we consider maintenance actions increasing the levels of functioning of the different resources, or simply a degradation in the current configuration that has not yet led to a situation change. In scenario 4, we consider a degradation in the component "G20" level of functioning from 0.9 to 0.2 leading to an overall satisfaction level decrease. In this case, MBSysRec will exclude all configuration engaging "G20" and will recommend a configuration having a score of 0.902 from the set using "G21" to the user via a dialog box (see *Figure 7-19*).

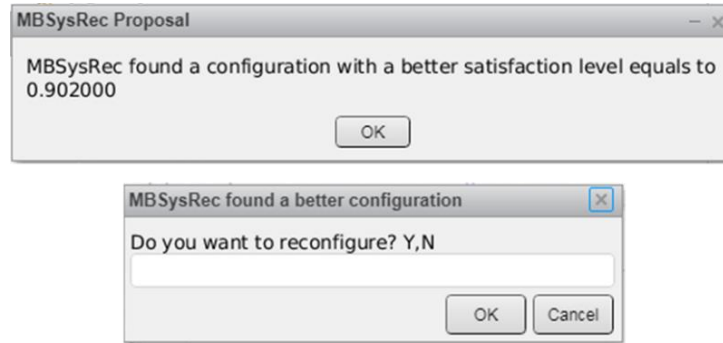


Figure 7-19: MBSysRec proposal of a better configuration

At last, we consider scenario 5 dealing with an objective change. In this case, the required functions as well as the interactions between them change. Moreover, the aggregation model used to calculate the overall satisfaction level is also different. Since the "Rescue" objective" does not need the "Observe" function anymore, the resolution criterion is no more significant; thus, its weight is set to zero, and higher weights are given for both the impact of change and the level of functioning criteria. The total number of the threshold satisfying configurations is 207, and among them, the configuration which maximizes the score of the impact of change criterion has been chosen.

7.5. Industrial Case Study: Short-Term Conflict Alert in Air Traffic Managnemt System

Addressing configuration and reconfiguration issues requires verification and testing of the presented approaches using large-scale industrial problems. Therefore, we have chosen the Short-Term Conflict Alert (STCA) functionality from the Air-Traffic Management (ATM) system. ATM corresponds to the entire system responsible for ensuring the safety of an aircraft departing from an aerodrome during its take-off, transit, and landing. The ATM has different missions, and among those is the operational control that addresses the en-route traffic control. STCA assists the controller in preventing collision between aircraft by detecting potential violations of the user-defined minimum separation distance (for instance, 9 km laterally and 300 vertically below flight level 2900 ft). When detecting violation or predicting a violation within a short time (usually 2 minutes), an alarm is generated by the STCA system and displayed via a visual alert display, or audible alert means. The STCA is a functionality of the ATM system, and its realization varies depending on the Air Traffic Control (ATC) center provider. STCA manages 1) the calculation and the display of STCA alerts, 2) manages the inhibition of alerts 3) manages the supervised areas. This

functional chain encompasses many functions; *Figure 7-20* shows a simplified functional chain that is required to implement the STCA system.

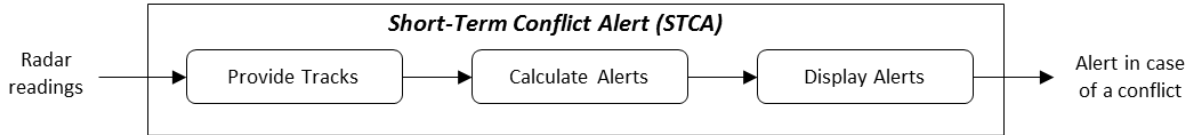


Figure 7-20: STCA simplified functional chain

In order to warn the ATC operator of any situation where a violation of the minimum separation is present or is predicted in a short time, the STCA system should be able to have the surveillance tracks of any pair of flying aircraft. Once the tracks are available, the STCA system should be able to calculate if there is a conflict. The conflict calculation is based on the flight plans and the current tracks of the flights under supervision. If a conflict is detected, the STCA system should be able to distribute these alerts and display them to the ATC operator, who gives directives to the conflicting flights preventing their collisions and ensuring air traffic safety.

The current implementation of STCA is rigid in terms of reconfiguration practices during operation. All configuration and reconfigurations sequences are previously defined, and the operators respect the regulations constraining this domain. However, emerging technologies, such as virtualization, can increase the flexibility of the system leading to a large number of configurations to be considered by the ATC design and support team. In this context, reconfiguration tasks can become burdensome. The difficulty lies in evaluating a large number of configurations quickly as conflicts can arise in a short time. Moreover, reconfiguration requires choosing adequate resources (components) to realize the different functions based on their performance and availability, while taking their compatibility into consideration.

7.5.1. Problem Definition

In order to verify and test the MBSysRec approach, we collected the data regarding the STCA case study from experts who have many years of experience in the ATM/ATC system. Moreover, we have also extracted data from the system's detailed specifications documents. Using the current industrial implementation of the STCA functional chain, we consider the system to be a series of modules

(components) connected and exchanging information to calculate the conflict alerts. As input, we know the number of existing modules (components), their compatibilities, and the functions they can provide. Because the STCA system should be operational in all contexts, only one set of relevant modules (components) is considered.

Table 7-6: STCA problem data

1 Context	All operational contexts
1 Objective	Detect and display a conflict
1 Functional chain	Raise STCA alerts
23 Functions	Including providing tracks, calculating and displaying alerts
	Conflict calculating module (6 instances)
	Alerts management and display module (2 instances)
	Area under supervision management module (2 instances)
	System tracks and flight plans coupling module (3 instances)
24 modules	Flight plans processing module (3 instances)
	Fallback tracks providing module (1 instance)
	Aeronautical information providing module (1 instance)
	System management module (2 instances)
	Tracks providing module (4 instances)
	Precision
3 criteria	Impact of change
	Level of functioning

The STCA has only one objective, which is to detect and warn the ATC operator in case of conflict. Thus, we use one aggregation model (single weight and interaction matrices) to calculate the level of functioning of all functions participating in this functional chain. Experts were asked to complete the M8 matrix with the efforts needed for setting up the different modules (components) in order to allow for the impact of change calculation. In this case study, we use the precision of modules as a performance criterion. For the sake of simplicity, further criteria such as cost, and user preferences were not considered in this case study. Finally, to estimate the overall satisfaction level, we rely on experts' opinions to complete the weights and interaction matrices. Table 7-6 summarizes the problem data.

7.5.2. Reconfiguration Scenario from the STCA Case Study

In this scenario, we consider a reconfiguration due to a failure of the “alerts management and display” module. This module is critical in the STCA system as the ability to communicate the alerts to the controller is essential to the success of the ATC operator mission. To account for this failure, MBSysRec-Observe detects a change and calls MBSysRec-Orient and MBSysRec-Decide functions. The MBSysRec-Orient uses the available resources to generate the solution space containing 864 configurations. MBSysRec-Decide, then, assesses the candidate configurations and gives an overall satisfaction level for each option. Next, MBSysRec-Decide reduces the solution space using the user-defined satisfaction threshold, which is set to 0.94 for this reconfiguration scenario (see Figure 7-21). The resulting solution space is, therefore, reduced to 54. Finally, to recommend one option among the reduced solution space configurations, MBSysRec-Decide chooses the configuration having the maximum score (Configuration 1 having 0.959).

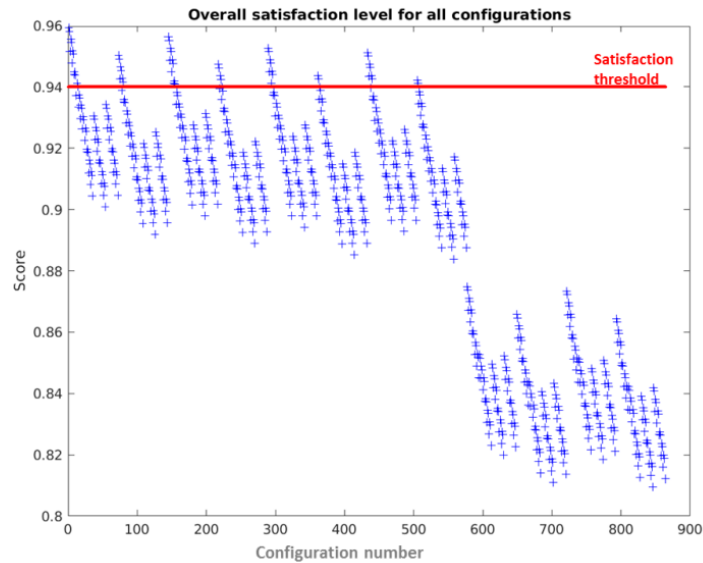


Figure 7-21: Overall satisfaction level for the STCA reconfiguration scenario

In order to understand the reason behind recommending configuration 1, we will analyze this configuration along with the second satisfying configuration on the list. Table 7-7 shows the score of both configurations on each evaluated criterion. Configuration 1 uses identical modules as Configuration 2 for all functions, except for the “Flight plans processing” module. Although both instances of this module have identical precision, the one used in configuration 1 has a lower level of functioning. The aggregation model used to

estimate a configuration level of functioning gives less importance to the function realized by the “Flight plans processing” module compared to other functions in the STCA functional chain. This explains the nearly negligible difference in the level of functioning estimation for both configurations. Finally, configuration 1 uses the same instance of the “Flight plans processing” module as the initially instantiated configuration giving a higher score for configuration 1 on the impact of change criterion. When calculating the overall satisfaction level for both configurations and based on the used aggregation model, MBSysRec-Decide will recommend the configuration having the higher rank.

Table 7-7: Summary of configurations 1 and 2 scores for the STCA reconfiguration scenario

Configuration Number	Precision	Impact of Change	Level of Functioning	Satisfaction Level
1	0.943	0.933	0.9826	0.959
2	0.943	0.928	0.9834	0.958

The STCA functionality and the ATM system, in general, are highly constrained by international aviation organizations due to their crucial role in ensuring air traffic safety and security. Therefore, those systems use redundancy to increase system safety. In this context, reconfiguration sequences are predefined and expected, as we have noticed when using MBSysRec to reconfigure the STCA system exploiting the current design and rules. The configuration recommended in the previously explained scenario was the one expected by the involved experts.

Experts have highlighted the importance of MBSysRec when shifting towards a virtual implementation of the STCA system. Virtualization implies using identical processors to run different applications yielding in a non-negligible number of the STCA system implementations. Consequently, reconfiguration, and MBSysRec in particular, would play an essential role in assisting the ATC design and support team in choosing the most suitable configuration with regards to the different criteria.

7.6. Evaluation and Discussion

The MBSysRec method has been tested using various reconfiguration scenarios from one case study (i.e., Search and Rescue) and one industrial project (i.e., Search and Rescue and the Short Term Conflict Alert from the Air Traffic Management). In order to evaluate the usability, applicability, and usefulness of the

presented method, we organized workshops involving experts, internal and external to the company. Experts included three system architects and one operation support engineer. Two of the involved experts are working on Air Traffic Management system architecture. The other experts were also involved in designing aspects within the search and rescue case. Experts were asked to evaluate the method using the previously mentioned criteria. The experts underlined that the method is easy to use once they became familiar with it. After a 90-minute presentation of the method with two more hours of simple example manipulations, engineers were able to apply the method to simple reconfiguration scenarios. Experts also emphasized that such a method would assist the engineering and operation team when reconfiguring already deployed systems. The results from the two case studies presented in sections 7.4 and 7.5 were discussed with the experts who validated the quality of the selected configurations in each reconfiguration scenario regarding the problem data.

The advantages of the presented method lie in using matrices within MBSysRec as well as using the Choquet integral method to estimate the configuration functioning level and the multi-criteria decision making. As explained in this chapter, the Choquet integral method is interesting as it considers the criteria interdependency and their interaction. Using allocation matrices is practical as they can be generated and extracted automatically from system models developed using engineering tools such as SysML. The fact that the company is familiar with using model-based engineering tools facilitates the integration of MBSysRec within the overall engineering process. Although there are prior works that also use matrices for configuration purposes within the design phase, MBSysRec is the first matrix-based method that aims at integrating the data related to the system during operations to support its reconfiguration. Moreover, the existing system architecture design and system configuration methods map requirements, function, and components in order generate system configurations. Our method is different from the existing methods in that it adds objective, context, and maps the following specific information Objective, Functional Chain, Function, Component, Context to generate system configuration. In addition to this specific information, our method integrates health and usage monitoring data (operational data) regarding the availability and the level of functioning of components as well as the effective situation. Integrating such data allows for considering system evolutions and support its reconfiguration.

However, there are several limitations to the presented method. Firstly, in this work, we make the hypothesis that health and usage monitoring data reflecting the availability and the level of functioning of the already deployed system is available in a matrix form that suits the MBSysRec method. Moreover, we also assume that it is possible to fully observe the deployed system and its components. In practice, full observability is not always possible, making the configuration level of functioning estimation complex and uncertain. Uncertainty might also be present in the observable data. These uncertainties should be considered to

increase our confidence in the configuration recommended by the MBSysRec method. Reconfiguration, sometimes, involves the integration of new technologies. The uncertainty regarding combining these components with the legacy system should also be considered to increase system reliability. Finally, some of the information used within MBSysRec, such as the impact of change and the performance information, are expert estimations. The subjectivity of expert estimations has not been explicitly modelled in the method. Therefore, the sensitivity of MBSysRec should be verified to ensure its robustness and usefulness for industrial projects.

Other industrial challenges linked to this proposal are also to be addressed. For instance, building complete and consistent models (structure, functions, rules, policies, etc.) to be used in the reconfiguration process is among these challenges. Finally, the present method presents a reconfiguration method of already deployed systems. Thus, its certification includes not only the certification of the reconfiguration algorithms but also the certification of the whole engineering process used to develop both reconfiguration algorithms and system models.

7.7. Conclusions

Systems that have considerable lifetimes (for instance, up to 80 or 100 years) are subject to upgrades due to evolutions in their operational context and objectives as well as to accommodate system faults, components malfunctioning, or the need for performance enhancements. System reconfiguration is, thus, essential to ensure the system's operational effectiveness in conjunction with the evolution of its operational context and objectives. The challenge of supporting and managing system reconfiguration lies in collecting and exploiting both the engineering design data and operational data (such as health monitoring data). The data related to the level of functioning of each component as well as interface management data are important to consider with regard to the actual system context, system objectives, and its environment.

In this chapter, we presented the model-based system Reconfiguration method (MBSysRec), which considers the context, the objectives, and the availability of the components in the generation of relevant configurations. To calculate the overall satisfaction level, MBSysRec uses the requirement satisfaction, the configuration level of functioning estimation, and the impact of change reflecting the degree of reuse of components from the legacy system. These criteria are aggregated using the two additive capacities of Choquet integral permitting to consider the criteria inter-dependency and their different levels of importance.

To the best of our knowledge, MBSysRec is the first reconfiguration method that integrates the operational context and objectives along with sensor data reflecting the level of functioning of the legacy configuration. MBSysRec has been tested for reconfiguration scenarios from a maritime search and rescue (SAR) case study. The Short-Term Conflict Alert (STCA) from the Air Traffic Management (ATM) industrial project has also been used to test the applicability of the method to large scale industrial systems. An evaluation of the method by experts supported its usability, applicability, and usefulness. Future work should focus on integrating uncertainties related to the level of functioning estimation and new arrangements reliabilities.

8. Conclusions

This chapter summarizes the key results of this thesis and discusses their quality as well as their limitations. This chapter also discusses the generalizability and the applicability of the research results, and finally gives directions for future research.

8.1. Research Key Results

The aim of this research is to support system reconfiguration using a model-based approach. This thesis focuses on the identification of reconfiguration data and its integration. The identified reconfiguration data is formalized thanks to the OSysRec ontology that synthesizes the three aspects of system reconfiguration. Then to support system reconfiguration, we presented the MBSysRec method which integrates data from the OSysRec at different levels.

Research quality is a key factor in evaluating research outcomes and results. In section 4.2, we identified the criteria that we used to ensure the research quality: validity, reliability, and credibility. Usability, applicability, and usefulness were also used to establish a value from an industrial point of view.

In the following, we review the research questions that framed this thesis and discuss the research contributions with regard to the previously identified quality criteria. Finally, we underline the challenges for each contribution.

RQ1: What are the existing approaches, methods and tools for system reconfiguration, and what are their related difficulties?

As explained in chapter 1, the research objective stems from the need for a generic system reconfiguration approach that can suit a variety of system types. Therefore, the objective of this research is to propose a model-based approach for system reconfiguration during operations. The descriptive study allowed us to

identify the difficulties and challenges that are related to the existing model-based approaches for system reconfigurations. The main identified issue is related to the development of reconfiguration support. Therefore, we proposed to concentrate on this issue, with a focus on the challenges regarding reconfiguration data identification and integration. Hence, system configuration can be identified by providing the necessary data, and by defining the underlying mechanisms allowing data integration. This would participate in increasing the operational effectiveness of systems.

The identified difficulties and challenges were discussed during workshops to verify the credibility and reliability of the descriptive study. For the empirical study, we used methods such as expert interviews, interviews recording and transcription, and triangulation of data from different sources. These methods are well-founded in the literature and widely used for action-based research. As for the validity, the results, including an overview of difficulties, challenges, and opportunities for future developments, represent an expected answer for the corresponding research question. The direct use of these results is to identify future developments and to improve the current industrial situation. The results of the research work proved the relevance of the identified challenges as they led to tangible improvements via the development of the OSysRec ontology and the MBSysRec method.

The empirical results reported within the descriptive study should be considered in the light of some limitations due to data collection and interpretation. We collected data from interviews with the experts that we have identified within the company. However, selection bias can be a threat to our results as we were limited in access to the appropriate participants within and across industry. Data analysis and interpretation are other sources of bias in this study. To avoid this bias, we cross verified the results with regard to the literature.

RQ2: What is the required data for system reconfiguration, and how can it be formalized?

Based on the results of the descriptive study and to address the issue related to the identification of reconfiguration data, we considered the development of an ontology for system reconfiguration that aims at capturing and formalizing the data necessary for this process.

In the *OSysRec* ontology, we synthesized the data that is crucial for system reconfiguration into three aspects based on the holistic systems engineering perspective. The three aspects that can be identified include structure, dynamics, and management. The management aspect is related to managing the system's available resources regarding its operational situation (i.e., context and mission). The system's resources and their connections are defined by the structural aspect. The dynamic aspect handles system behavior, evolution,

and transitions based on the triggering events and the applied conditions. The synthesized data, together, are essential to ensure effective system management.

The *OSysRec* ontology was the subject of different meetings involving the research team and the research steering committee. The reliability of the *OSysRec* ontology regarding its quality has been highlighted during verification and validation workshops. The ontology development method, as well as the validation and evaluation methods (i.e., experts' interviewing, scenario-based tests, and cross-checking with the literature) are also well established and widely used. *OSysRec* captured the necessary data for system reconfiguration and synthesized them into three packages ensuring the coverage of three aspects governing this process, structure, dynamics, and management.

The *OSysRec* ontology is easily applicable as definitions of the integrated concepts, as well as their interactions and relations, are well-documented. The *OSysRec* is usable and provides a conceptual framework for engineers and system architects to discuss problems and solutions related to system reconfiguration. To ensure the usefulness of *OSysRec*, we set the development of the reconfiguration support (in terms of integration methods) as a high-level success factor. This support has been developed in the form of the *MBSysRec* method.

As with the majority of research, the presented ontology is subject to limitations. Our proposal argues that the *OSysRec* ontology is overarching as it integrates key system reconfiguration concepts that have been shared and agreed on for different system types, and across several domains. However, the application of the *OSysRec* ontology to specific cases is not trivial as additional adaptation and tailoring steps are required. Moreover, the ontology validation workshops and interviews involved experts from one specific domain, while the concepts elicitation process involved experts from various domains. To avoid this bias, we cross-validated the ontology via the existing literature, and we also assessed the ontology using several case studies from the company.

RQ3: How can we integrate the pre-identified data to support system reconfiguration?

The required reconfiguration support is needed to bridge data, from both phases engineering and operations, which is related to the three aspects previously defined (i.e., structure, dynamics and management). Hence, in this thesis, we presented the **model-based system reconfiguration method (*MBSysRec*)** that integrates and combines the data synthesized within the *OSysRec* ontology.

The identified requirements of *MBSysRec* are change detection, configuration generation with regards to operational context, mission, and the available resources as well as configuration evaluation and selection

based on context and objective dependent criteria (e.g., safety, business, cost). *MBSysRec* maps these requirements to a three-step process, i.e., *MBSysRec-Observe*, *MBSysRec-Orient*, and *MBSysRec-Decide*. The orientation step relies on the combinatorial exploration of the solution space. The decision step is mainly based on the Choquet integral multi-criteria decision-making that considers criteria inter-dependency and interactions. *MBSysRec* provides decision support for configurations evaluation and selection within command and control activities to ensure effective system management via reconfiguration.

The credibility and the reliability of the *MBSysRec* method have been verified through workshops. *MBSysRec* is designed based on different methods and analytical equations. The configuration generation method that was used allows for the combinatorial exploration of the solution space and is well established in the literature and very widely used in publications of a similar nature. The Choquet integral multi-criteria decision-making method was highlighted as a powerful method in this domain. The analytical equations that were used, such as the configuration performance calculation or the distance between two configurations, have also been used in the literature in publications related to configuration and architecture generation.

Experts, internal and external to the company, were asked to evaluate the *MBSysRec* method with regards to the previously mentioned criteria. For this purpose, workshops were organized involving three system architects and one operation support engineer working on Air Traffic Management and Search and Rescue design and architecture. The usability of the method was confirmed by experts who manipulated different reconfiguration scenarios using the *MBSysRec* method. Experts ensured that *MBSysRec* is applicable and corresponds to the intended use, being a support in managing systems via reconfiguration. The results from the two case studies employed to test the method were discussed with the experts. The involved experts approved the quality of the configurations selected for each considered reconfiguration scenario. The quality of the selected configurations represents the improvement measure allowing to confirm the usefulness of the method.

Even though *MBSysRec* applicability has been proven, it is subject to several limitations. A potential threat to the quality of configurations recommended by *MBSysRec* is related to configurations uncertainties. The identified sources of uncertainties are related to the lack of observability, level of confidence in sensors measurements, integration of new technologies, and expert estimations included in the system data model. *MBSysRec* integrates the data related to the system in operations to support its reconfiguration. However, health and usage monitoring data are not always fully observable, or they are not available in the exact format we have chosen for *MBSysRec* (Matrix). Due to the lack of observability, we may rely on estimations of resources functioning level that may introduce uncertainties in our measurements. Observable data can also be uncertain as data provided by sensors is always subjected to some level of uncertainty and

inconsistency linked to different reasons such as the sensor physics, data treatment, or even the measurement procedure. Uncertainty makes the measured quantity inexact and hence should be addressed. Uncertainty related to the reliability of the selected configuration needs to be integrated. Reconfiguration relies on combining different components (resources) and put them all together to realize a given objective. The reliability of the selected configuration, therefore, depends on the reliability of each integrated component as well as their ability to work together when combined. The last source of uncertainties is linked to expert estimations regarding the information included in the system data model. For instance, experts provide information about the impact of change, performance, functions' interactions, and importance within a configuration, as well as evaluation criteria inter-dependency and interactions. The subjectivity of expert estimations has not been explicitly modeled in the method. Therefore, the sensitivity of *MBSysRec* should be verified to ensure its robustness and usefulness for industrial projects.

Other major challenges prevent the direct use of *MBSysRec* in the industry at larger scales. Building complete and consistent models (structure, functions, rules, policies, etc.) to be used in the reconfiguration process is among these challenges. The use of this method for reconfiguration purposes during operation needs to be certified by international organizations, in particular, if the domain of application is considered safety-critical, such as the aerospace domain. The certification concerns not only the reconfiguration algorithms but also the overall system engineering process contributing to the development of the required algorithms and the system models.

The proof of concept implementation of *MBSysRec* in the form of a MATLAB program suffers from challenges that should be considered for its industrial deployment. Our choice of the configuration generation algorithm relies on the combinatorial exploration of the solution space, which allows us to consider the solution space exhaustively. However, this process is very slow and needs to be optimized, especially if *MBSysRec* is to be deployed for online system reconfiguration, where response time is relatively short. *MBSysRec* allows us to analyze possible system configurations with regards to the functions considered within these configurations. Such analysis is not yet implemented in the proof of concept implementation. To summarize, the software is at early stage and requires further tests and improvements.

Finally, the data related to the different case studies, employed to create and assess the research results, was provided by Thales engineers and experts. However, we made some hypotheses when the data was lacking. These hypotheses need to be verified and confirmed to ensure the validity of the configuration satisfaction level estimations.

8.2. Generalizability and Applicability

The thesis focuses on supporting system reconfiguration with a generic approach. Therefore, to ensure the generalizability of the approach, we conducted an assessment through case studies based on real projects from the company. Moreover, to ensure the applicability, we assessed the results in terms of coherence and consistence regarding the enterprise instructions and technical reports and guidelines. In this section, we discuss the generalizability and applicability of the research results in the company.

8.2.1. Generalizability through case studies

The *OSysRec* ontology has been assessed using many case studies from the company. These case studies include

- the Integrated Modular Avionics (IMA).
- Air Traffic Management (ATM).
- the next-generation tactical radios incorporating innovative software-defined radio technology.
- Delegated fleet management for land forces systems (vehicles and payload).
- the future soldier system and the connected battle group.
- Further scenarios from the Search and Rescue (SAR) case study have also been used.

These case studies were chosen from different business domains of the company to ensure the generalizability of the ontology across domains. The chosen scenarios cover different systems types that are dealt with within Thales (i.e., equipment, platform, distributed, and systems of systems). Simple reconfiguration scenarios from each case study have been run to test and validate the ontology. The selected scenarios covered the different parts of the ontology.

The MBSysRec method has been tested and evaluated using the scenarios from the SAR case study. It has also been applied to reconfiguration scenarios of the Short-Term Conflict Alert (STCA) functionality from the Air Traffic Management case. The STCA was chosen as it is a representative case while being relatively simple compared to other functionalities within the ATM case. The STCA case has one context and one objective. In this case, we considered the raise STCA alerts functionality, which incorporates 23 functions and 24 realizing modules. For configurations assessments, in addition to the two generic criteria (i.e., the impact of change and level of functioning), specific criterion concerning the precision of the configuration

is also used. The configuration generated using this method were assessed with experts and recognized as relevant, given the problem data.

8.2.2. Applicability based on Thales References

Different levels of assessment exist within an industrial context. Thales reference processes (Chorus) incorporating the system engineering processes (Sys-EM) is aligned to the national norms and standards (e.g., NF X50). Thales instructions and technical reports and guidelines are tested for compliance, partially or fully, with regard to the national standards. These standards are developed by the French Association for Norms and Standards (AFNOR) and include standards on different industrial domains and the corresponding technical report and guidelines. Experts are developing the national standards to be aligned with the requirements of the international standards and technical reports and guidelines (e.g., ISO 15288) (see Figure 8-1).

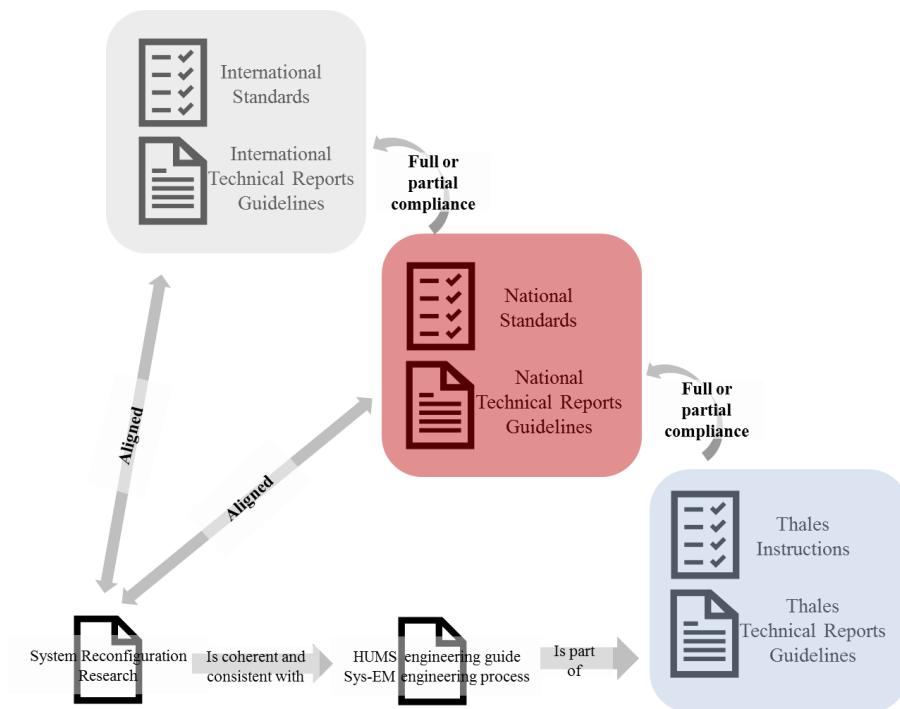


Figure 8-1: Thesis results applicability mapping

The applicability of this research is ensured by the consistency and the coherence of the results with regard to Thales reference system (Chorus).

This research work is aligned with the activities of the health and usage monitoring systems (HUMS) working group at Thales. The HUMS engineering guide as well as the global systems engineering methodology within Thales (Sys-EM) are parts of the company reference system (Chorus). These documents have been used as a starting point for this research work. The research results are consistent and coherent as they lie within the perimeter considered by the HUMS engineering guide and t Sys-EM. *Figure 8-2* shows the simplified HUMS architecture defined by Thales experts on systems engineering. System reconfiguration is linked to this process via the command and control activities.

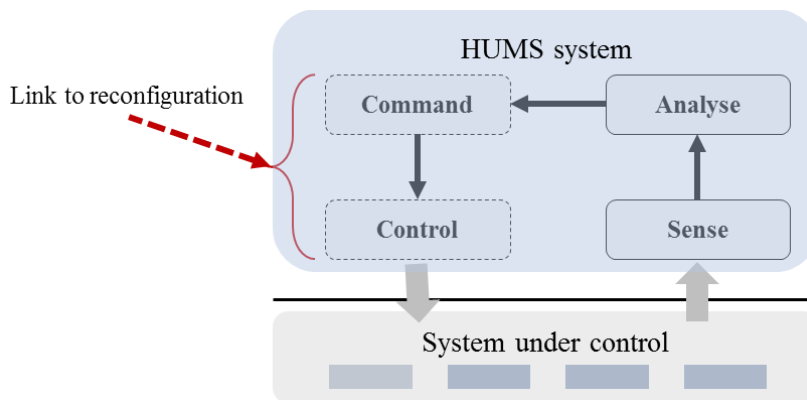


Figure 8-2: Simplified HUMS architecture from Thales

8.3. Future Work

Further improvements can be made to overcome the research challenges that we previously underlined and strengthen the proposals presented in this thesis. Such improvements will increase the technological readiness level of our proposals allowing their widespread use across business domains in Thales or even across companies.

To improve the documentation of the *OSysRec* ontology, future work should focus on discussing the concepts that were not comprehensively elaborated in this version of the ontology. Further improvements of the *OSysRec* ontology in terms of tailoring methods with regard to the considered cases can be made to ease its use. Finally, to ensure the generalizability of the *OSysRec* ontology, we suggest employing case

studies from industrial domains beyond the scope of our industrial partner (i.e., Thales) to assess its applicability (for instance, application to autonomous vehicles from the automotive industry).

The level of confidence of *MBSysRec* in terms of the recommended configuration quality should be increased. Therefore, *MBSysRec* can be improved by integrating the uncertainties related to the lack of observability, level of confidence in sensor measurements, integration of new technologies, and expert estimations included in the system data model.

Future work should also consider improving supporting domains. For instance, novel technologies and breakthroughs in the big data and artificial intelligence domains in terms of data collection, treatment methods, and uncertainty management can significantly contribute to enhancing our reconfiguration method (*MBSysRec*) and its application for real-life systems. Advances in modeling practices, methods, and tools can support building accurate system models and improve the efficiency of the reconfigurations implemented using *MBSysRec*.

To increase the technology readiness level of *MBSysRec*, more efficient configuration generation algorithms or strategies should be investigated. One can consider different generation algorithms, such as genetic algorithms, fuzzy logic, or Bayesian Networks. The genericity of these methods and their ability to deal with evolutions related to systems and their operational situations should be explored. Optimization strategies can also be investigated to avoid exploring the overall solution space of configurations. For instance, using reference configurations defined and stored in the matrix system can drastically reduce the calculation time. One also can consider defining the minimum acceptable functioning levels for each function needed in the configuration. This can serve for filtering the candidate components based on their accepted functioning level rather than the binary indicators reflecting their availability.

Finally, further tests and improvements are needed to increase confidence in the *MBSysRec* method allowing its industrial deployment within Thales systems and beyond.

References

- Akue, L., Lavinal, E. and Sibilla, M. (2012), “A model-based approach to validate configurations at runtime”, *4th International Conference on Advances in System Testing and Validation Lifecycle (VALID)*, pp. 133–138.
- Albarello, N., Welcomme, J.-B. and Claude, R. (2012), “A formal design synthesis and optimization method for systems architectures”, *Proceedings of MOSIM ...*, available at: http://hal.archives-ouvertes.fr/docs/00/72/85/77/PDF/paper_1.pdf.
- Aldanondo, M. and Vareilles, E. (2008), “Configuration for mass customization : how to extend product configuration towards requirements and process configuration”, *Journal of Intelligent Manufacturing*, pp. 521–535.
- Ali, A.B.H., Khalgui, M., Valentini, A. and Ben Ahmed, S. (2011), “Safe reconfigurations of agents-based embedded control systems”, *IECON Proceedings (Industrial Electronics Conference)*, IEEE, pp. 4344–4350.
- Alsafi, Y. and Vyatkin, V. (2010), “Ontology-based reconfiguration agent for intelligent mechatronic systems in flexible manufacturing”, *Robotics and Computer-Integrated Manufacturing*, Vol. 26 No. 4, pp. 381–391.
- ASAAC. (2004), *ASAAC Final Draft of Proposed Guidelines for System Issues Document Reference: ASAAC2-GUI-32450-001-CPG*.
- Aßmann, U., Steffen, Z. and Gerd, W. (2006), “Ontologies, meta-models, and the model-driven paradigm.”, *Ontologies for Software Engineering and Software Technology*, Springer, Berlin, Heidelberg, pp. 249–273.
- Axhausen, K.W., Meyer, J., Becker, H. and Patrick, M.B. (2017), “Autonomous vehicles : The next jump in accessibilities ?”, *Research in Transportation Economics*, Vol. 62, pp. 80–91.
- Baldwin, C.Y. and Clark, K.B. (1997), “Managing in an age of modularity”, *Harvard Business Review*, Vol. 75, pp. 84–93.
- Benjaafar, S., Heragu, S.S. and Irani, S.A. (2002), “Next generation factory layouts: Research challenges and recent progress”, *Interfaces*, Vol. 32 No. 6, pp. 58–76.
- Bermejo-Alonso, J., Hernandez, C. and Sanz, R. (2016), “Model-based engineering of autonomous systems using ontologies and metamodels”, *ISSE 2016 - 2016 International Symposium on Systems Engineering - Proceedings Papers*, available at: <https://doi.org/10.1109/SysEng.2016.7753185>.
- Bermejo-Alonso, J., Sanz, R., Rodríguez, M. and Hernández, C. (2011), “ENGINEERING AN ONTOLOGY FOR AUTONOMOUS SYSTEMS - The OASys Ontology”, pp. 47–58.
- Björk, E. and Ottosson, S. (2007), “Aspects of consideration in product development research”, *Journal of Engineering Design*, Vol. 18 No. 3, pp. 195–207.
- Blessing, L.T.M. and Chakrabarti, A. (2009), *DRM, a Design Research Methodology, Focus*, Vol. 1, available at: <https://doi.org/10.1007/978-1-84882-587-1>.
- Bonjour, É. and Micaëlli, J.P. (2010), “Design core competence diagnosis: A case from the automotive industry”, *IEEE Transactions on Engineering Management*, Vol. 57 No. 2, pp. 323–337.
- Bonnet, S., Voirin, J.-L., Exertier, D. and Normand, V. (2017), “Modeling system modes, states, configurations with Arcadia and Capella: method and tool perspectives”, *INCOSE International Symposium*, Vol. 27 No. 1, pp. 548–562.

- Booch, G., Rumbaugh, J. and Jacobson, I. (1999), *The Unified Modeling Language Reference Manual*, Addison-Wesley Reading.
- Brown, D.C. (1998), “Defining configuring”, *AI EDAM*, Vol. 12 No. 4, pp. 301--305.
- Bryant, C.R., Stone, R.B., Mcadams, D.A., Kurtoglu, T. and Campbell, M.I. (2005), “Concept generation from the functional basis of design”, *ICED 05: 15th International Conference on Engineering Design: Engineering Design and the Global Economy*, p. 1702.
- Buede, D.M. (2009), *The Engineering Design of Systems: Models and Methods*, John Wiley & Sons, Inc.
- Caporuscio, Mauro and Marco, Antiniscia Di and Inverardi, P. (2005), “Run-time Performance Management of the Siena Publish / Subscribe Middleware Mauro Caporuscio”, *Proceedings of the 5th International Workshop on Software and Performance*, pp. 65--74.
- Caporuscio, M., Di Marco, A. and Inverardi, P. (2007), “Model-based system reconfiguration for dynamic performance management”, *Journal of Systems and Software*, Vol. 80 No. 4, pp. 455–473.
- Chandrasekaran, B., Josephson, J.R. and Benjamins, V.R. (1999), “What are ontologies, and why do we need them?”, *IEEE Intelligent Systems and Their Applications*, Vol. 14 No. 1, pp. 20–26.
- Chen, R., Liu, Y., Fan, H., Zhao, J. and Ye, X. (2019), “An integrated approach for automated physical architecture generation and multi-criteria evaluation for complex product design”, *Journal of Engineering Design*, Vol. 30 No. 2–3, pp. 63–101.
- Chrisp, H. and Richard, W. (2007), “Technical Operations International Council on Systems Engineering (INCOSE)”, No. September 2007, available at: http://www.cose.org/media/upload/SEVision2020_20071003_v2_03.pdf.
- Condat, H., Strobel, C. and Hein, A. (2012), “Model-based automatic generation and selection of safe architectures”, *22nd Annual International Symposium of the International Council on Systems Engineering, INCOSE 2012 and the 8th Biennial European Systems Engineering Conference 2012, EuSEC 2012*, Vol. 1, pp. 645–665.
- Crawley, E., De Weck, O., Magee, C., Moses, J., Seering, W., Schindall, J., Wallace, D., et al. (2004), “The influence of architecture in engineering systems (monograph)”, *Engineering Systems Monograph, MIT Engineering Systems Division, March*.
- Creswell, J.W. (2014), *Research Design : Qualitative, Quantitative, and Mixed Methods Approaches*, 4th ed., SAGE Publications, Inc.
- Cross, N., Dorst, K. and Roozenburg, N. (1991), “Research in design thinking workshop”, Delft, Netherlands.
- Crow, J. and Rushby, J. (1991), “Model-Based Reconfiguration: Toward an Integration with Diagnosis”, *In Proceedings of AAAI-91*, pp. 836–841.
- Cui, Y., Shi, J. and Wang, Z. (2018), “Backward Reconfiguration Management for Modular Avionic Reconfigurable Systems”, *IEEE Systems Journal*, Vol. 12 No. 1, pp. 137–148.
- Dalpiazz, F., Giorgini, P. and Mylopoulos, J. (2009), “An architecture for requirements-driven self-reconfiguration”, *International Conference on Advanced Information Systems Engineering*, pp. 246--260.
- Deciu, E.R., Ostrosi, E., Ferney, M. and Gheorghe, M. (2005), “Configurable product design using multiple fuzzy models”, *Journal of Engineering Design*, Vol. 16 No. 2, pp. 209–233.
- Dhungana, D., Falkner, A., Haselböck, A. and Taupe, R. (2017), “Enabling integrated product and factory configuration in smart production ecosystems”, *2017 43rd Euromicro Conference on Software Engineering and*

- Advanced Applications (SEAA)*, IEEE, pp. 266--273.
- DODAF. (2009), *DoD Architecture Framework Version 2.0*, Working Group, Department of Defense, Washington, DC,.
- Dori, D. (2016), *Model-Based Systems Engineering with OPM and SysML*, *Model-Based Systems Engineering with OPM and SysML*, available at: <https://doi.org/10.1007/978-1-4939-3295-5>.
- Du, X., Jiao, J. and Tseng, M.M. (2001), "Architecture of product family: Fundamentals and methodology.", *Concurrent Engineering: Research and Application*, Vol. 9 No. 4, pp. 309–325.
- Dumond, R., Farcet, N., Slimani, M. and Garnier, J.-L. (2009), *A Model-Based Architecture to Manage Tactical Systems of Systems*, available at: [http://public.jlgarnier.fr/downloads/2009-NATO-IST-087 - A Model-based Architecture to Manage Tactical Systems of Systems.pdf](http://public.jlgarnier.fr/downloads/2009-NATO-IST-087-A-Model-based-Architecture-to-Manage-Tactical-Systems-of-Systems.pdf).
- Ebrahimipour, V. and Yacout, S. (2015), "Ontology-based schema to support maintenance knowledge representation with a case study of a pneumatic valve", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, IEEE, Vol. 45 No. 4, pp. 702–712.
- Eckert, C.M., Clarkson, P.J. and Stacey, M.K. (2003), "The spiral of applied research: A methodological view on integrated Design research", *Proceedings of the International Conference on Engineering Design, ICED*, Vol. DS 31.
- Eterno, J., Weiss, J., Looze, D. and Willsky, A. (1985), "Design issues for fault tolerant-restructurable aircraft control", *1985 24th IEEE Conference on Decision and Control*, pp. 900–905.
- Falkner, A., Haselböck, A., Schenner, G. and Schreiner, H. (2011), "Modeling and solving technical product configuration problems", *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, Vol. 25 No. 2, pp. 115–129.
- Feldmann, S. and Loskyll, M. (2013), "Increasing Agility in Engineering and Runtime of Automated Manufacturing Systems", *IEEE International Conference on Industrial Technology (ICIT)*, pp. 1303–1308.
- Fiod-Neto, M. and Back, N. (1994), "Assessment of product conception: A critical review", *The 1994 Lancaster International Workshop on Engineering Design*.
- Francois, F., Nain, G., Morin, B., Daubert, E., Barais, O., Plouzeau, N. and Jézéquel, J.-M. (2014), "Kevoree Modeling Framework (KMF): Efficient modeling techniques for runtime use", No. May 2014, pp. 1–24.
- Friedrich, G., Ryabokon, A., Falkner, Andreas A Haselböckk, A., and Schenner, G. and Schreiner, H. (2011), "(Re) configuration based on model generation", *LoCoCo*, pp. 26–35.
- Gausemeier, J., Gaukstern, T. and Tschimer, C. (2013), "Systems engineering management based on a discipline-spanning system model", *Procedia Computer Science*, Elsevier B.V., Vol. 16, pp. 303–312.
- Gerrike, K., Eckert, C. and Stacey, M. (2017), "WHAT DO WE NEED TO SAY ABOUT A DESIGN METHOD ?", *21th International Conference on Engineering Design (ICED 2015)*, 21-25 Aug 2017, Vancouver, Canada, Vol. 7, pp. 101–110.
- Gogniat, G., Diguët, J.-P., de Lamotte, F., Guillet, S. and Vidal, J. (2013), "Dynamic applications on reconfigurable systems: From UML model design to FPGAs implementation", *2011 Design, Automation & Test in Europe*, IEEE, pp. 1–4.
- Gómez-Pérez, A. (1999), "Ontological engineering: A state of the art", *Expert Update: Knowledge Based Systems and*

- Applied Artificial Intelligence*, Vol. 2 No. 3, pp. 33–43.
- Grabisch, M. (2013), “FUZZY MEASURES AND INTEGRALS IN MULTICRITERIA DECISION ANALYSIS”, *Wiley Encyclopedia of Operations Research and Management Science*, pp. 1–8.
- Gross, T., Specht, M., Hci, G. and Augustin, S. (2001), “Awareness in Context-Aware Information Systems”, pp. 173–182.
- Grover, V. (2015), “RESEARCH APPROACH: AN OVERVIEW”, *Golden Research Thoughts*, Vol. 4.
- Gruber, T. and Özsu, M. (2009), *Encyclopedia of Database Systems. Ontology*.
- Guarino, N., Oberle, D. and Staab, S. (2009), “What Is an Ontology?”, *Handbook on Ontologies*, pp. 1–17.
- Hamza, M., Aly, S.G. and Hosny, H. (2011), “An Approach for Generating Architectures for Pervasive Systems from Selected Features”, *Proceedings of the International Conference on Software Engineering Research and Practice (SERP)*.
- Haris, K. and Dagli, C.H. (2011), “Adaptive reconfiguration of complex system architecture”, *Procedia Computer Science*, Vol. 6, pp. 147–152.
- Helms, B. and Shea, K. (2012), “Computational synthesis of product architectures based on object-oriented graph grammars”, *Journal of Mechanical Design, Transactions of the ASME*, Vol. 134 No. 2, available at: <https://doi.org/10.1115/1.4005592>.
- Henderson-Sellers, B. (2011), “Bridging metamodels and ontologies in software engineering”, *Journal of Systems and Software*, Elsevier Inc., Vol. 84 No. 2, pp. 301–313.
- Hernández, C., Fernández, J.L., Sánchez-Escribano, G., Bermejo-Alonso, J. and Sanz, R. (2015), “Model-Based Metacontrol for Self-adaptation”, *International Conference on Intelligent Robotics and Applications*, Springer, pp. 643–654.
- Hildebrandt, C., Törsleff, S., Caesar, B. and Fay, A. (2018), “Ontology Building for Cyber-Physical Systems: A domain expert-centric approach”, *IEEE International Conference on Automation Science and Engineering*, Vol. 2018–August, pp. 1079–1086.
- Ho, T.H. and Tang, C.S. (1998), *Product Variety Management: Research Advances*, Boston: Kluwer Academic Publishers.
- Hofstedt, P. and Schnee-weiss, D. (2013), “FdConfig: A Constraint-Based Interactive Product Configurator”, *19th International Conference on Applications of Declarative Programming and Knowledge Management*.
- INCOSE Systems Engineering Handbook V4. (2015), *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, 4th Edition, available at: <http://eu.wiley.com/WileyCDA/WileyTitle/productCd-1118999401.html>.
- Isermann, R. (1997), “Supervision, fault-detection and fault-diagnosis methods - An introduction”, *Control Engineering Practice*, Vol. 5, pp. 639–652.
- ISO/IEC/IEEE:15288. (2015), *ISO/IEC/IEEE/15288: Systems and Software Engineering- System Life Cycle Processes*.
- ISO/IEC/IEEE:24765. (2017), “International Standard - Systems and software engineering -- Vocabulary”.
- ISO/IEC/IEEE:42010. (2011), *Systems and Software Engineering: Architecture Description*, IEEE New York, NY.
- ISO/IEC/IEEE:42020. (2019), “Enterprise , systems and software — Architecture processes”.

- ISO/IEC/IEEE:42030. (2019), *Software, Systems and Enterprise — Architecture Evaluation Framework*.
- ISO. (2012), “ISO 13379-1: Condition monitoring and diagnostics of machines -- Data interpretation and diagnostics techniques -- Part 1: General guidelines”.
- ISO. (2015), “ISO 13381-1: Condition monitoring and diagnostics of machines -- Prognostics -- Part 1: General guidelines”.
- ISO. (2018), “ISO 17359: Condition monitoring and diagnostics of machines — General guidelines”, available at: <https://doi.org/10.1126/science.ns-1.6.159>.
- Jankovic, M. and Eckert, C. (2016), “Architecture decisions in different product classes for complex products”, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, Vol. 30 No. 3, pp. 217–234.
- Jankovic, M., Holley, V. and Yannou, B. (2012), “Multiple-domain design scorecards: A method for architecture generation and evaluation through interface characterisation”, *Journal of Engineering Design*, Vol. 23 No. 10–11, pp. 746–766.
- Jiao, J.R., Simpson, T.W. and Siddique, Z. (2007), “Product family design and platform-based product development : a state-of-the-art review”, *Journal of Intelligent Manufacturing*, Vol. 18 No. 1, pp. 5–29.
- Jorgensen, M. (2005), “Practical guidelines for expert-judgment-based software effort estimation”, *IEEE Software*, Vol. 22 No. 3, pp. 57–63.
- Ju, H., Wang, S. and Zhao, T. (2015), “A Modeling Method of IMA Dynamic Reconfiguration Based on AADL”.
- Judt, D. and Lawson, C. (2012), “Methodology for Automated Aircraft Systems Architecture Enumeration and Analysis”, *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, American Institute of Aeronautics and Astronautics, Reston, Virginia, pp. 1–17.
- Khederzadeh, M. and Beiranvand, A. (2018), “Identification and Prevention of Cascading Failures in Autonomous Microgrid”, *IEEE Systems Journal*, Vol. 12 No. 1, pp. 308–315.
- Kösters, G., Six, H. and Winter, M. (2001), “Validation and Verification of Use Cases and Class Models”, *7th International Workshop on Requirements Engineering: Foundations for Software Quality (REFSQ'2001, Proc.)*.
- Krause, F.-L., Kimura, F., Kjellberg, T., Lu, S.-Y., Alting, Elmaraghy, HA Eversheim, W., Iwata, K., et al. (1993), “Product modelling”, *CIRP Annals*, Vol. 4 No. 2, pp. 695–706.
- Kreuz, I. and Roller, D. (1999), “Knowledge Growing Old in Reconfiguration Context Ingo Kreuz Exact configuration Onboard Starting the Reconfiguring Process”, *Configuration Papers from the AAI Workshop, AAI Technical Report WS99*.
- Krichen, F. (2010), “Position paper : Advances in Reconfigurable Distributed Real Time Embedded Systems”, *10th Annual International Conference on New Technologies of Distributed Systems (NOTERE) (Pp. 273-278)*. IEEE., IEEE, pp. 273–278.
- Krichen, F., Hamid, B., Zalila, B. and Jmaiel, M. (2012), “Design-Time Verification of Reconfigurable Real-Time Embedded Systems”, *IEEE 14th International Conference on High Performance Computing and Communications*, available at: <https://doi.org/10.1109/HPCC.2012.217>.
- Krichen, F. and Zalila, B. (2011), “Towards a Model-Based Approach for Reconfigurable DRE Systems Towards a

- Model-Based Approach for Reconfigurable DRE Systems”, *European Conference on Software Architecture*, Springer, pp. 295–302.
- Kuntz, F., Gaudan, S., Sannino, C., Laurent, É., Griffault, A. and Point, G. (2011), “Model-based diagnosis for avionics systems using minimal cuts.”, *Martin Sachenbacher and Oskar Dressler and Michael Hofbauer. DX 2011, Oct 2011, Murnau, Germany*, pp. 138–145.
- Kurtoglu, T. and Campbell, M.I. (2009), “Automated synthesis of electromechanical design configurations from empirical analysis of function to form mapping”, *Journal of Engineering Design*, Vol. 20 No. 1, pp. 83–104.
- Labreuche, C. (2011a), “A general framework for explaining the results of a multi-attribute preference model”, *Artificial Intelligence*, Vol. 175 No. 7–8, pp. 1410–1448.
- Labreuche, C. (2011b), “A general framework for explaining the results of a multi-attribute preference model”, *Artificial Intelligence*, Elsevier B.V., Vol. 175 No. 7–8, pp. 1410–1448.
- Langlois, R.N. (1992), “External economies and economic progress: the case of the microcomputer industry”, *Business History Review*, Vol. 66, pp. 1–50.
- Lee, E.A. and Seshia, S.A. (2017), *INTRODUCTION TO EMBEDDED SYSTEMS A CYBER-PHYSICAL SYSTEMS*.
- Lee, H.J. and Lee, J.K. (2005), “An effective customization procedure with configurable standard models”, *Decision Support Systems*, Vol. 41 No. 1, pp. 262–278.
- Leger, M., Ledoux, T. and Coupaye, T. (2015), “Reliable Dynamic Reconfigurations in a Reflective Component Model”, *International Symposium on Component-Based Software Engineering*, available at: <https://doi.org/10.1007/978-3-642-13238-4>.
- Li, Z., Wang, S. and Zhao, T. (2015), “A model based simulation verification method for IMA reconfiguration on system level”, *2015 First International Conference on Reliability Systems Engineering (ICRSE)*, pp. 3–7.
- Liang, Q., Wu, X., Park, E.K., Khoshgoftaar, T.M. and Chi, C.H. (2011), “Ontology-based business process customization for composite web services”, *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, IEEE, Vol. 41 No. 4, pp. 717–729.
- Liu, Y. and Liu, Z. (2010), “An integration method for reliability analyses and product configuration”, *The International Journal of Advanced Manufacturing Technology*, Vol. 50, pp. 831–841.
- Madni, A.M. and Sievers, M. (2017), “Model-based systems engineering: Motivation, current status, and needed advances”, *Disciplinary Convergence in Systems Engineering Research*, pp. 311–325.
- Magro, D. (2010), “COCONF: conceptual language-based configuration”, *AI Communications*, Vol. 23 No. 1, pp. 1–46.
- Maleki, E., Belkadi, F., Zwaag, B.J. van der and Bernard, A. (2017), “A sensor ontology enabling service implementation in Industrial Product-Service Systems”, *IFAC-PapersOnLine*, Elsevier B.V., Vol. 50 No. 1, pp. 13059–13064.
- Manhart, P. (2005), “Reconfiguration-A problem in search of solutions”, *IJCAI*.
- Männistö, T. (2000), *A Conceptual Modelling Approach to Product Families and Their Evolution*.
- Männistö, T., Soininen, T., Tiihonen, J. and Sulonen, R. (1999), “Framework and Conceptual Model for Reconfiguration”, *AAAI Technical Report WS-99-05*.
- El Maraghy, H.A. (2006), “Flexible and reconfigurable manufacturing systems paradigms”, *Flexible Services and*

- Manufacturing Journal*, Vol. 17 No. 4 SPECIAL ISSUE, pp. 261–276.
- Mckay, A., Erens, F. and Bloop, M.S. (1996), “Engineering Design Relating Product Definition and Product Variety”, pp. 63–80.
- Mehrabi, M.G., Ulsoy, A.G. and Koren, Y. (2000), “Reconfigurable manufacturing systems: key to future manufacturing”, *Journal of Intelligent Manufacturing*, Vol. 11 No. 4, pp. 403–419.
- Meyer, F., Kroeger, R., Heidger, R. and Milekovic, M. (2013), “An approach for knowledge-based IT management of air traffic control systems”, *2013 9th International Conference on Network and Service Management, CNSM 2013 and Its Three Collocated Workshops - ICQT 2013, SVM 2013 and SETM 2013*, pp. 345–349.
- Meyer, M. and Utterback, J. (1993), “The product family and the dynamics of core capability.”, *Sloan Management Review*, Spring, pp. 29–47.
- Mittal, S. and Frayman, F. (1989), “Towards a generic model of configuration tasks”, *IJCAI*, pp. 1395–1401.
- MODAF. (2010a), “UK Ministry of Defence Architecture Framework (MODAF)”, available at: <http://www.mod.uk/DefenceInternet/AboutDefence/WhatW...nformationManagement/MODAF/ModafDetailedGuidance.htm>.
- MODAF. (2010b), “MODAF, UK Ministry of Defence Architecture Framework”, available at: <http://www.modaf.com>.
- Moullec, M.-L., Bouissou, M., Jankovic, M., Bocquet, Jean-Claude Réquillard, F., Maas, O. and Forgeot, O. (2013), “Towards System Architecture Generation and Performances Assessment Under Uncertainty Using Bayesian Networks”, *Journal of Mechanical Design* 135, No. 4, available at: <https://doi.org/https://doi.org/10.1115/1.4023514>.
- NAF v4. (2018), “North Atlantic Treaty Organization: NATO architecture framework v4.0 documentation”.
- NAF v4. (2019), “NATO Architecture Framework, Version 4, Modeling Guidelines for use of the UAF DMM”.
- NASA. (2007), *NASA Systems Engineering Handbook, Systems Engineering*, Vol. 6105, available at: [https://doi.org/10.1016/0016-0032\(66\)90450-9](https://doi.org/10.1016/0016-0032(66)90450-9).
- Newcomb, P.J., Bras, B. and Rosen, D.W. (1996), “Implications of modularity on product design for the life cycle”, *ASME Design Engineering Technical Conferences, DETC96/DTM-1516, Irvine, CA*.
- Ng, H.T. (1990), “Model-Based, Multiple Fault Diagnosis of Time-Varying, Continuous Physical Devices”, *Proceedings 6th Conference on A. I. Applications*, pp. 9–15.
- Noura, H., Theilliol, D., Ponsart, J.-C. and Chamseddine, A. (2009), *Fault-Tolerant Control Systems: Design and Practical Applications*.
- Noy, N.F. and Hafner, C.D. (1997), “The state of the art in ontology design: A survey and comparative review”, *AI Magazine*, Vol. 18 No. 3, pp. 53–74.
- Obitko, M. and Mærik, V. (2002), “Ontologies for multi-agent systems in manufacturing domain”, *Proceedings - International Workshop on Database and Expert Systems Applications, DEXA, IEEE*, Vol. 2002–Janua, pp. 597–602.
- Oddsson, G., Ladeby, K.R. and Oddsson, G. (2014), “From a literature review of product configuration definitions to a reference framework”, pp. 413–428.
- OMG, (Object Management Group). (2010), “UML profile for MARTE Object Management Group”, V1.1.

- Ottosson, S. (2003), "Participation action research - A key to improved knowledge of management", *Technovation*, Elsevier Ltd, Vol. 23 No. 2, pp. 87–94.
- Peng, X., Yan, Z., Li, Y. and Wu, J. (2012), "Model-based Intelligent Configuration and Reconfiguration for Spacecraft Propulsion System", *4th International Conference on Intelligent Human-Machine Systems and Cybernetics Model-Based*, pp. 7–12.
- Personnic, G. (2002), "Asaac: the way to flying military open systems", *3rd European Systems Engineering Conference Systems Engineering: A Focus of European Expertise Pierre Baudis Congress Centre, Toulouse, 21st-24th May 2002*.
- Pinto, H.S. and Martins, J.P. (2004), "Ontologies: How can They be Built?", *Knowledge and Information Systems*, Vol. 6 No. 4, pp. 441–464.
- Pirlein, Th and Studer, R. (2000), "Integrating the Reuse of Commonsense Ontologies and Problem-Solving Methods", *Journal of Expert Systems*.
- Polakovic, J., Mazare, S., Stefani, J.B. and David., P.C. (2007), "Experience with safe dynamic reconfigurations in component-based embedded systems", *In 10th International Symposium in Component-Based Software Engineering*, pp. 242–257.
- Provan, G. and Chen, Y.-L. (1999a), "Model-based diagnosis and control reconfiguration for discrete event systems: an integrated approach", *Proceedings of the 38th IEEE Conference on Decision and Control*, Vol. 2, pp. 1762–1768.
- Provan, G. and Chen, Y. (1999b), "Model-based diagnosis and control reconfiguration for discrete event systems: An integrated approach", *IEEE Conference on Decision and Control*, available at:<https://doi.org/10.1109/CDC.1999.830888>.
- Ptolemaeus, C. (2014), *System Design, Modeling, and Simulation: Using Ptolemy II*, Ptolemy. org Berkeley.
- Qasim, L., Hein, A.M., Jankovic, M., Olaru, S. and Garnier, J.-L. (2019), "Towards a reconfiguration framework for systems engineering integrating use phase data", *Proceedings of ICED 2019, the 22nd International Conference on Engineering Design: Responsible Design for Our Future, Delft, Netherlands, 05.-08.08. 2019*.
- Qasim, L., Hein, A.M., Olaru, S., Jankovic, M. and Garnier, J.-L. (2020), "An overall Ontology for System Reconfiguration using Model-Based System Engineering", *Submitted to IEEE Transactions on Systems, Man, and Cybernetics: Systems*.
- Qasim, L., Jankovic, M., Olaru, S. and Garnier, J.-L. (2019), "Model-Based System Reconfiguration: A Descriptive Study of Current Industrial Challenges", *Complex Systems Design & Management*, Springer International Publishing, Cham, pp. 97–108.
- Qi, X., Theilliol, D., Qi, J., Zhang, Y., Han, J. and Song, D. (2013), "Fault Diagnosis and Fault Tolerant Control Methods for Manned and Unmanned Helicopters : A Literature Review", *Conference on Control and Fault-Tolerant Systems*, IEEE, pp. 132--139.
- Ramos, A.L., Ferreira, J.V. and Barceló, J. (2012), "Model-based systems engineering: An emerging approach for modern systems", *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, Vol. 42 No. 1, pp. 101–111.
- Razmerita, L. (2011), "An ontology-based framework for modeling user behavior-A case study in knowledge

- management”, *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, IEEE, Vol. 41 No. 4, pp. 772–783.
- Regulin, D., Schutz, D., Aicher, T. and Vogel-Heuser, B. (2016), “Model based design of knowledge bases in multi agent systems for enabling automatic reconfiguration capabilities of material flow modules”, *IEEE International Conference on Automation Science and Engineering*, Vol. 2016–Novem, pp. 133–140.
- Reiter, R. (1987), “A theory of diagnosis from first principles”, *Artificial Intelligence*, Vol. 32 No. 1, pp. 57–95.
- Rodriguez, I.B., Drira, K., Chassot, C. and Jmaiel, M. (2009), “A model-based multi-level architectural reconfiguration applied to adaptability management in context-aware cooperative communication support systems”, *2009 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture, WICSA/ECSA 2009*, pp. 353–356.
- Roques, P., Roques, P., European, T. and Roques, P. (2016), “MBSE with the ARCADIA Method and the Capella Tool To cite this version : MBSE with the ARCADIA Method and the Capella Tool Keywords : Abstract”:
- Rosenstein, D. and Reich, Y. (2011), “Hierarchical system concept generation”, *ICED 11 - 18th International Conference on Engineering Design - Impacting Society Through Engineering Design*, Vol. 10, pp. 24–34.
- Russell, K.J. and Broadwater, R.P. (2012), “Model-based automated reconfiguration for fault isolation and restoration”, *2012 IEEE PES Innovative Smart Grid Technologies (ISGT)*, IEEE, pp. 1–4.
- Ryser, J. and Glinz, M. (1999), “A scenario-based approach to validating and testing software systems using statecharts”, *Proc. 12th International Conference on Software and Systems Engineering and Their Applications*.
- Sabin, D. and Weigel, R. (1998), “Product configuration frameworks-a survey”, *IEEE Intelligent Systems and Their Applications*, Vol. 13 No. 4, pp. 42–49.
- Salvador, F. (2007), “Toward a product system modularity construct: Literature review and reconceptualization”, *IEEE Transactions on Engineering Management*, Vol. 54 No. 2, pp. 219–240.
- Sanford, Friedenthal, S., Alan, M. and Steiner, R. (2014), *A Practical Guide to SysML: The Systems Modeling Language.*, Morgan Kaufmann.
- Sanz, R., Hernandez, C. and Bermejo, J. (2014), “Improved Resilience Controllers Using Cognitive Patterns”, *IFAC Proceedings Volumes*, Vol. 47, Elsevier, pp. 683–688.
- Sargeant, J. (2012), “Qualitative Research Part II: Participants, Analysis, and Quality Assurance”, *Journal of Graduate Medical Education*, Vol. 4 No. 1, pp. 1–3.
- Sargent, R.G. (2013), “Verification and validation of simulation models”, *Journal of Simulation*, Vol. 7 No. 1, pp. 12–24.
- Saxena, T., Dubey, A., Balasubramanian, D. and Karsai, G. (2010), “Enabling Self-Management by Using Model-Based Design Space Exploration”, *Seventh IEEE International Conference and Workshops on Engineering of Autonomic and Autonomous Systems*, pp. 137–144.
- Schwarze, S. (1996), *Configuration of Multiple-Variant Products: Application Orientation and Vagueness in Customer Requirements*, vdf Hochschulverlag AG.
- Seepersad, C.C., Pedersen, K., Emblemsvåg, J., Bailey, R. and Allen, J.K. (2006), “The Validation Square: How Does One Verify and Validate a Design Method?”, *Decision Making in Engineering Design*, pp. 303–313.
- Šercar, T.M. (2002), “Ontologies: A silver bullet for knowledge management and electronic commerce”, *Organizacija*

- Znanja*, Vol. 7 No. 3–4, pp. 121–122.
- Shan, S. and Hou, Z. (2016), “Neural Network NARMAX Model Based Unmanned Aircraft Control Surface Reconfiguration”, *2016 9th International Symposium on Computational Intelligence and Design (ISCID)*, IEEE, pp. 154–157.
- da Silva, C.E. and de Lemos, R. (2011), “A Framework for Automatic Generation of Processes for Self-Adaptive Software Systems 2 Background 3 Framework for process generation”, *Informatica*, Vol. 35, pp. 3–13.
- Silva, E. and Batista, T. (2015), “A Mission-Oriented Approach for Designing System-of- Systems”, pp. 346–351.
- Soininen, T., Tiihonen, J., Männistö, T. and Sulonen, R. (1998), “Towards a general ontology of configuration”, *Ai Edam*, pp. 357–372.
- Song, Z. and Kusiak, A. (2009), “Optimising product configurations with a data- mining approach”, *International Journal of Production Research*, Vol. 47 No. 7, pp. 1733--1751.
- Stoican, F. and Olaru, S. (2013), *Set-Theoretic Fault Detection and Control Design for Multisensor Systems*.
- Stone, R., Wood, K. and Crawford, R.H. (2000), “A heuristic method for identifying modules for product architectures”, *Design Studies*, Vol. 21 No. 1, pp. 5–31.
- Summers, J.D. and Eckert, C.M. (2013), *Design Research Methods: Interviewing. Portland, Oregon, USA: Workshop in ASME Conference. 2013*.
- Sundgren, N. (1999), “Introducing interface management in product family development”, *Journal of Production Innovation Management*, Vol. 16 No. 1, pp. 40–51.
- T. Henzinger and Sifakis., J. (2006), “The embedded systems design challenge”, *In Proceedings of the 14th International Symposium on Formal Methods (FM), Lecture Notes in Computer Science*.
- Tipaldi, M., Glielmo, L. and Member, S. (2017), “A Survey on Model-Based Mission Planning and Execution for Autonomous Spacecraft”, *IEEE SYSTEMS JOURNAL 1 A*, pp. 1–13.
- TOGAF. (2009), *The Open Group, “Togaf ”Enterprise Edition” Version 9”*.
- Ulrich, K. (1995), “The role of product architecture in the manufacturing firm”, *Research Policy*, Vol. 24 No. 3, pp. 419–440.
- Ulrich, K.T. and Eppinger, S.D. (1995), *Product Design and Development*, McGraw-Hill Higher Education.
- Vogel, T., Seibel, A. and Giese, H. (2010), “The Role of Models and Megamodels at Runtime”, *International Conference on Model Driven Engineering Languages and Systems*, Vol. 6627.
- Voirin, J.-L. and Bonnet, S. (2013), “Arcadia: model-based collaboration for system, software and hardware engineering.”, *Complex Systems Design & Management, Poster Workshop (CSD&M 2013)*.
- Walsh, D., Bordeleau, F. and Selic, B. (2005), “A Domain Model for Dynamic System Reconfiguration”, pp. 553–567.
- Walsh, D., Bordeleau, F. and Selic, B. (2006), “Change types of dynamic system reconfiguration”, *Proceedings of the International Symposium and Workshop on Engineering of Computer Based Systems*, pp. 3–12.
- Wan, J., Yin, B., Li, D., Celesti, A., Tao, F. and Hua, Q. (2018), “An Ontology-Based Resource Reconfiguration Method for Manufacturing Cyber-Physical Systems”, *IEEE/ASME Transactions on Mechatronics*, Vol. 23 No. 6, pp. 2537–2546.
- Wang, L., Ng, W.K. and Song, B. (2011), “Extended DCSP Approach on Product Configuration with Cost Estimation

- Lin”, *CONCURRENT ENGINEERING*., Vol. 19 No. 2, pp. 123--138.
- Weyrich, M., Klein, P. and Steden, F. (2014), “Reuse of modules for mechatronic modeling and evaluation of manufacturing systems in the conceptual design and basic engineering phase”, *19th World Congress The International Federation of Automatic Control Cape Town, South Africa. August 24-29, 2014*, pp. 3450–3455.
- Witt, R., Kennedy, A., Baetz, B., Mohr, U., Bucher, N. and Eickhoff, J. (2013), “Implementation of Fault Management Capabilities for the Flying Laptop Small Satellite Project through a Failure-Aware System Model”, pp. 1–16.
- Wyatt, D., Wynn, D. and Clarkson, J. (2008), “Synthesis of product architectures using a DSM/DMM-based approach”, *Proceedings of the 10th International DSM Conference*, No. November, pp. 349–361.
- Wyatt, D.F., Wynn, D.C., Jarrett, J.P. and Clarkson, P.J. (2012), “Supporting product architecture design using computational design synthesis with network structure constraints”, *Research in Engineering Design*, Vol. 23 No. 1, pp. 17–52.
- Wymore, A.W. (1993), *Model-Based Systems Engineering*, CRC press.
- Xie, H., Henderson, P. and Kernahan, M. (2007), “Modelling and solving engineering product configuration problems by constraint satisfaction”, *International Journal of Production Research*, Vol. 43 No. 20, pp. 4455--4469.
- Yassine, A.A. and Wissmann, L.A. (2007), “The Implications of Product Architecture on the Firm”, *Systems Engineering*, Vol. 10 No. 2, pp. 118–137.
- Yeom, K. (2015), “Neurocomputing Morphological approach for autonomous and adaptive system : The construction of three-dimensional arti fi cial model based on self-recon fi gurable modular agents”, *Neurocomputing*, Vol. 148, pp. 100–111.
- Yeom, K. and Park, J. (2012), “Morphological approach for autonomous and adaptive systems based on self-reconfigurable modular agents”, *Future Generation Computer Systems*, Elsevier B.V., Vol. 28 No. 3, pp. 533–543.
- Zhang, L.L. (2014), “Product configuration: A review of the state-of-the-art and future research”, *International Journal of Production Research*, Taylor & Francis, Vol. 52 No. 21, pp. 6381–6398.
- Zhang, Y. and Jiang, J. (2008), “Bibliographical review on reconfigurable fault-tolerant control systems”, *Annual Reviews in Control*, Vol. 32 No. 2, pp. 229–252.
- Zhou, Z., Xie, S. and Yang, W. (2008), “A case study on STEP-enabled generic product modelling framework”, *International Journal of Computer Integrated Manufacturing*, Vol. 21, p. 2008.
- Ziegert, S. and Wehrheim, H. (2013), “Temporal Reconfiguration Plans for Self-Adaptive Systems *”, *Software Engineering*, pp. 271–284.
- Zimmermann, A., Lorenz, A. and Oppermann, R. (2007), “An Operational Definition of Context”, *Modeling and Using Context*, pp. 558–571.
- Ziv-Av, A. and Reich, Y. (2005), “SOS - Subjective objective system for generating optimal product concepts”, *Design Studies*, Vol. 26 No. 5, pp. 509–533.
- Zugaj, M. (2017), “Reconfiguration of Fixed-Wing UAV Control System in Autonomous Flight”, *22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, pp. 1063–1068.

Appendix

Appendix for chapter 6

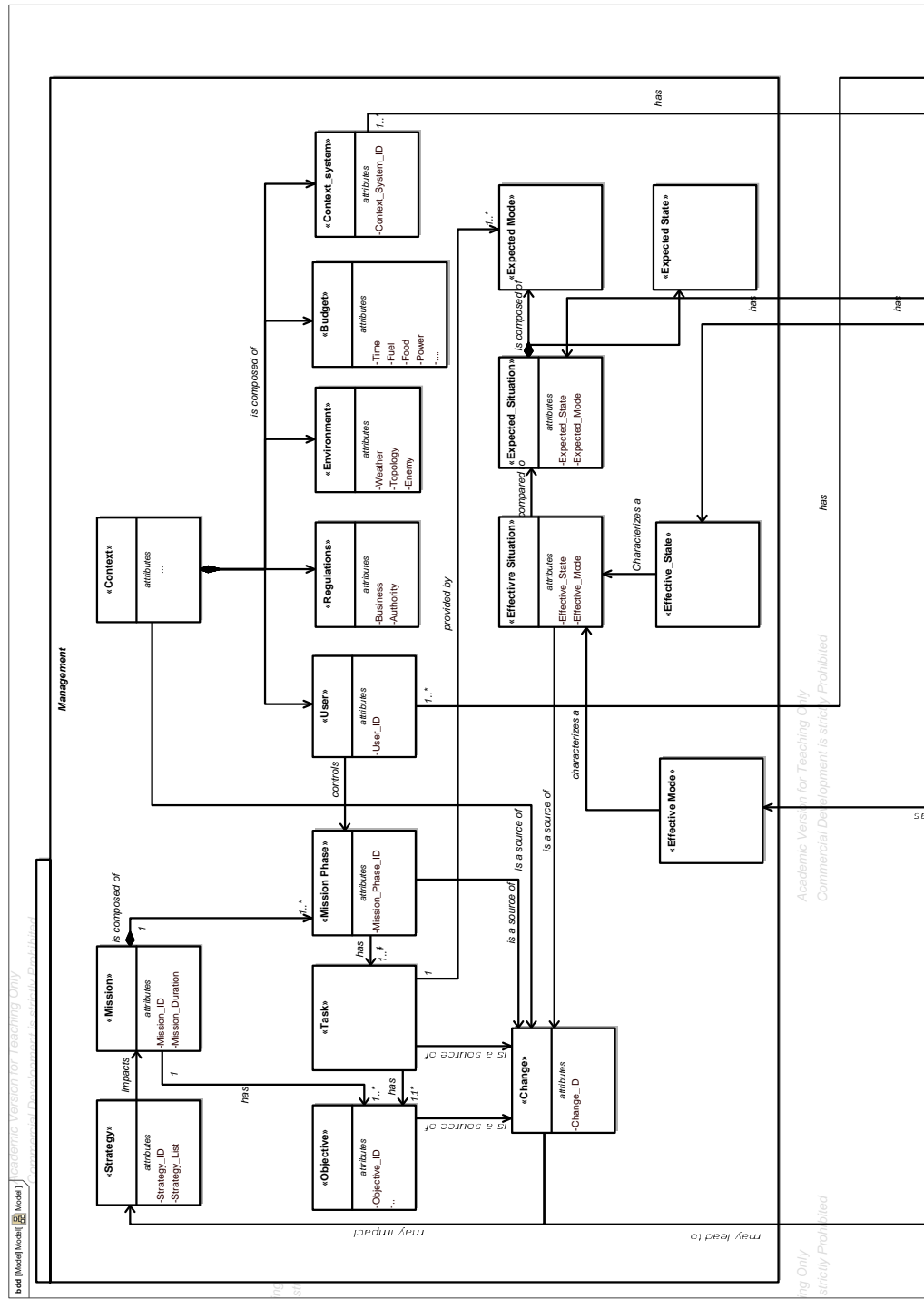


Figure A- 1: OSysRec “Zoom-in” part 1

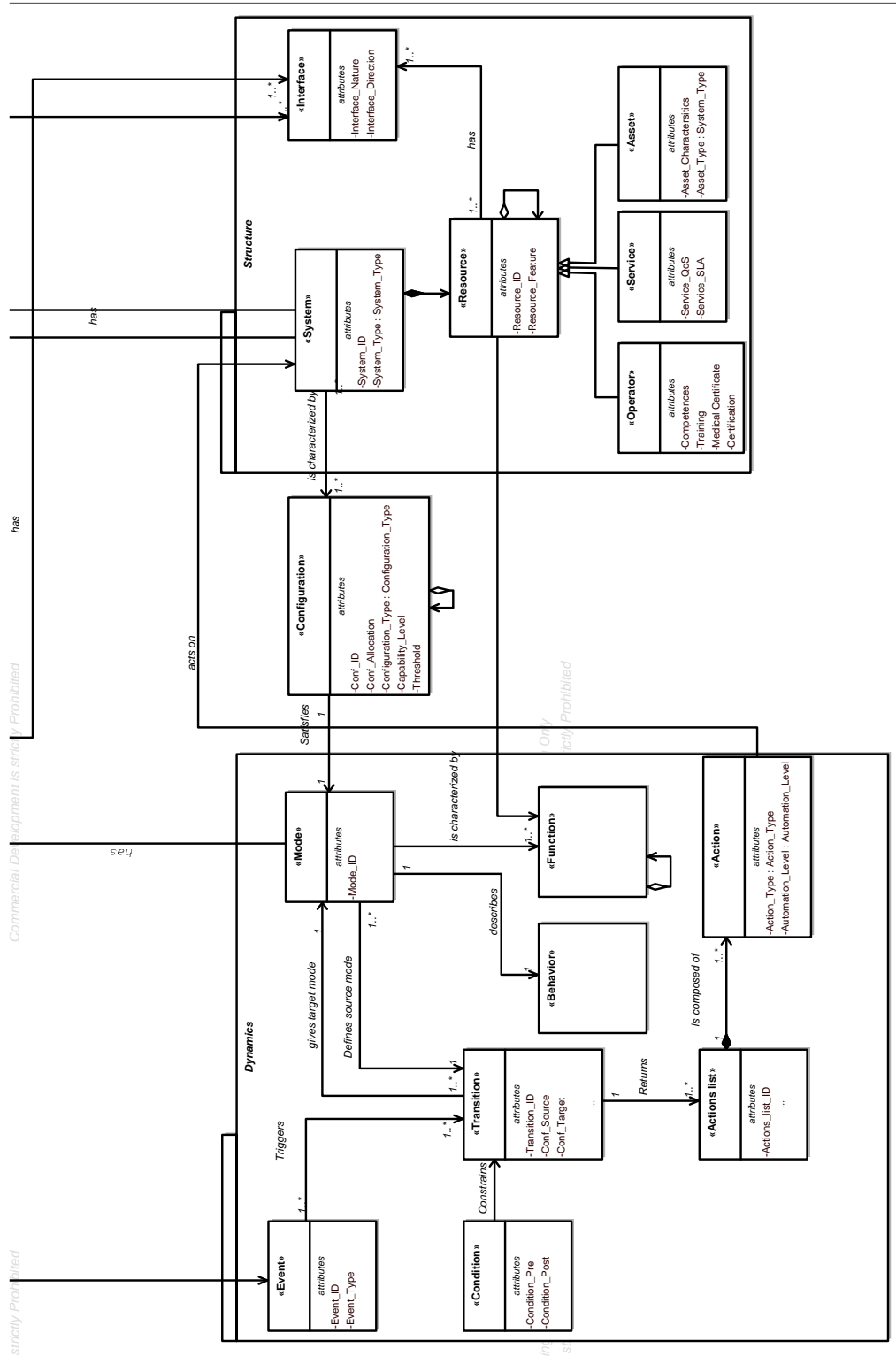


Figure A- 2: OSysRec "Zoom-in" part 2

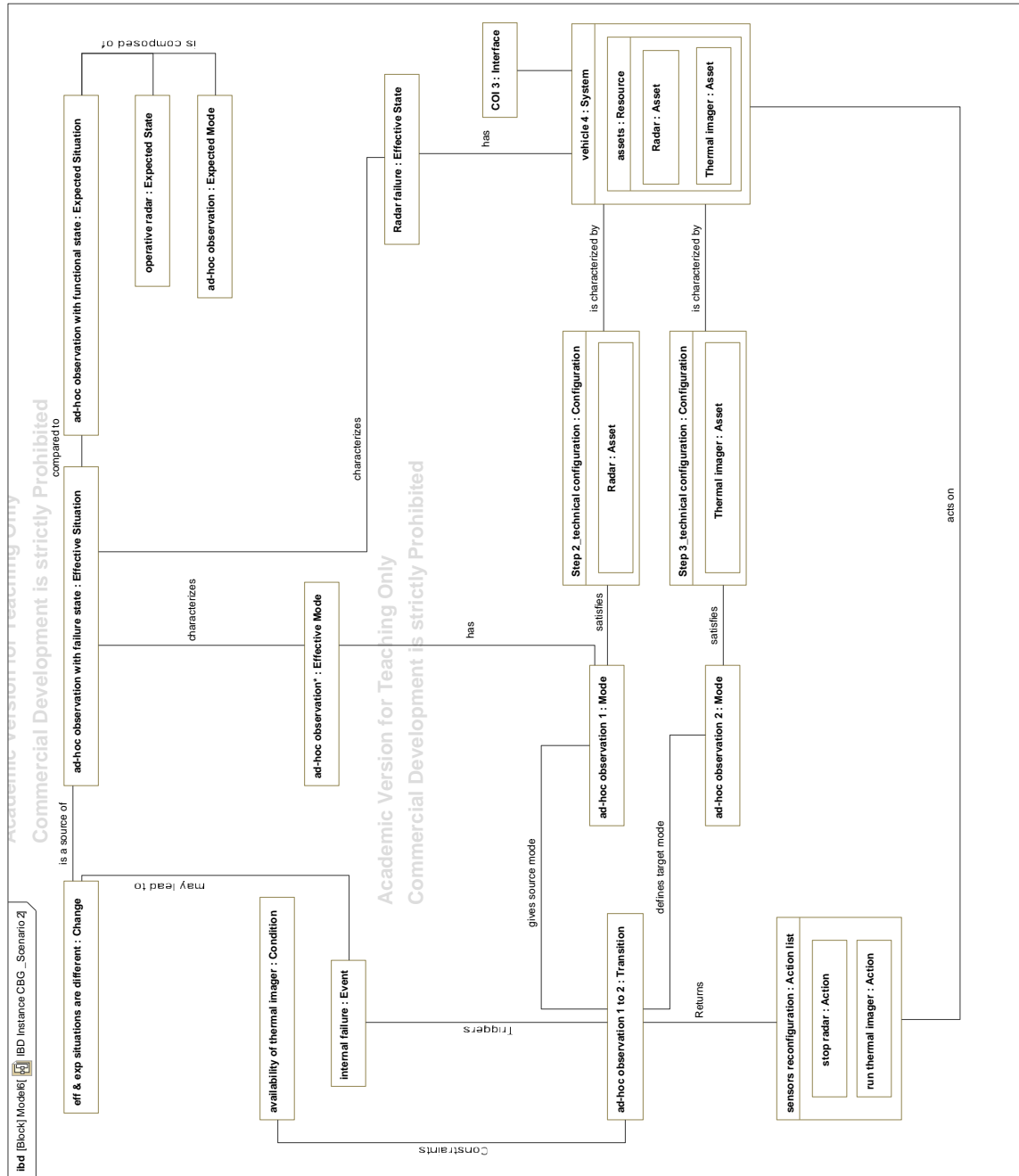


Figure A- 3:OSysRec instance for scenario 2

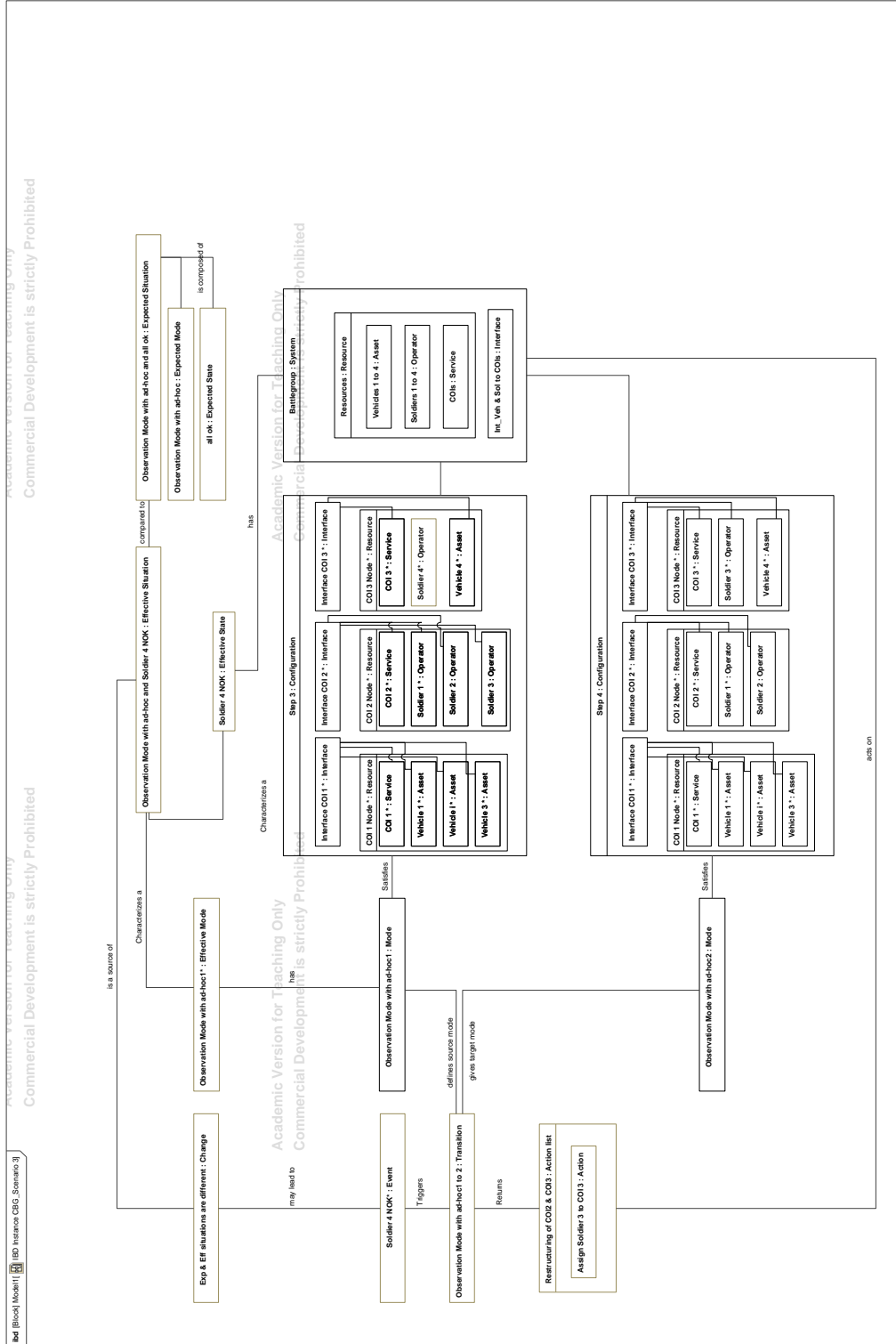


Figure A- 4:OSysRec instance for scenario 3

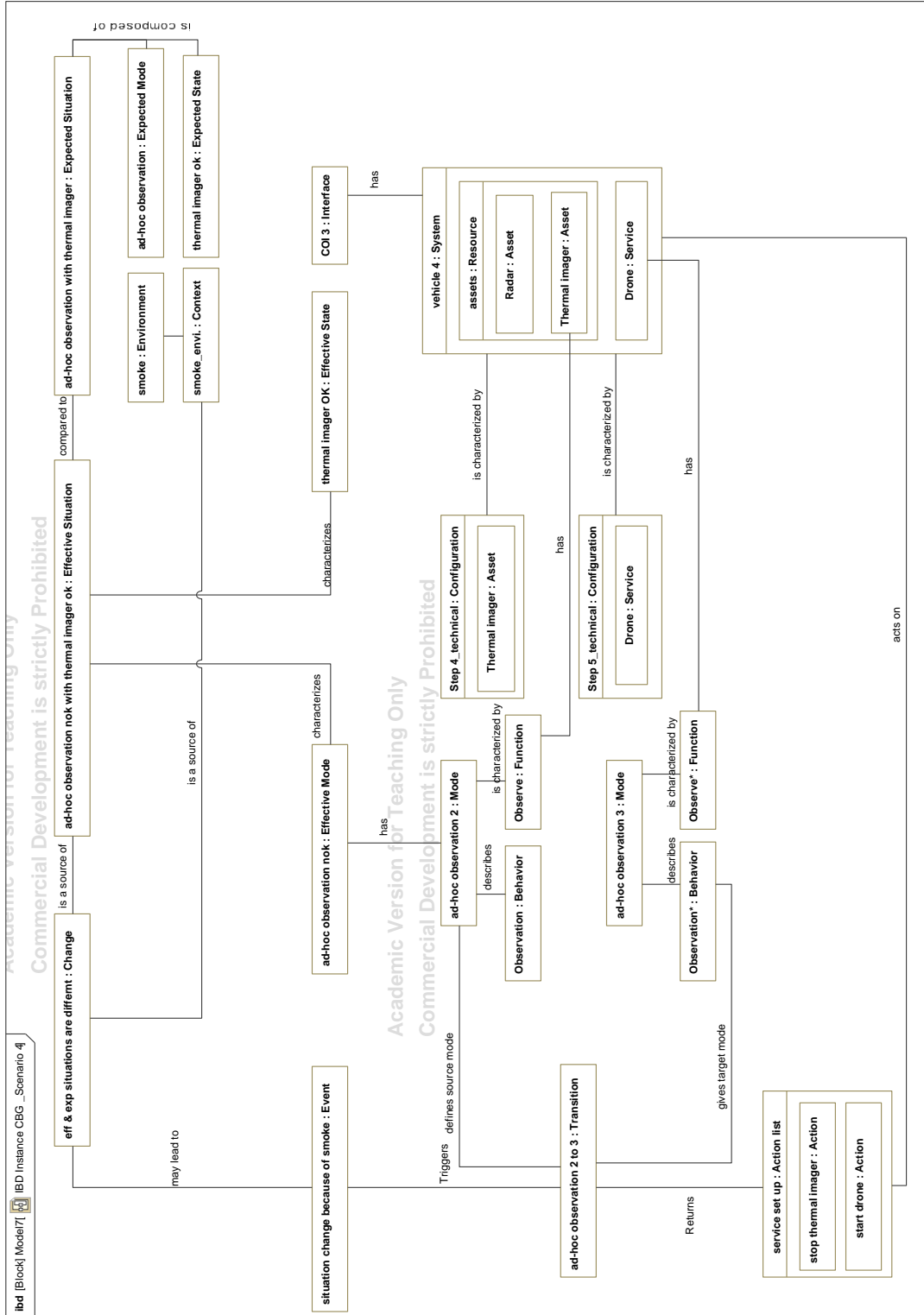


Figure A- 5: OSysRec instance for scenario 4

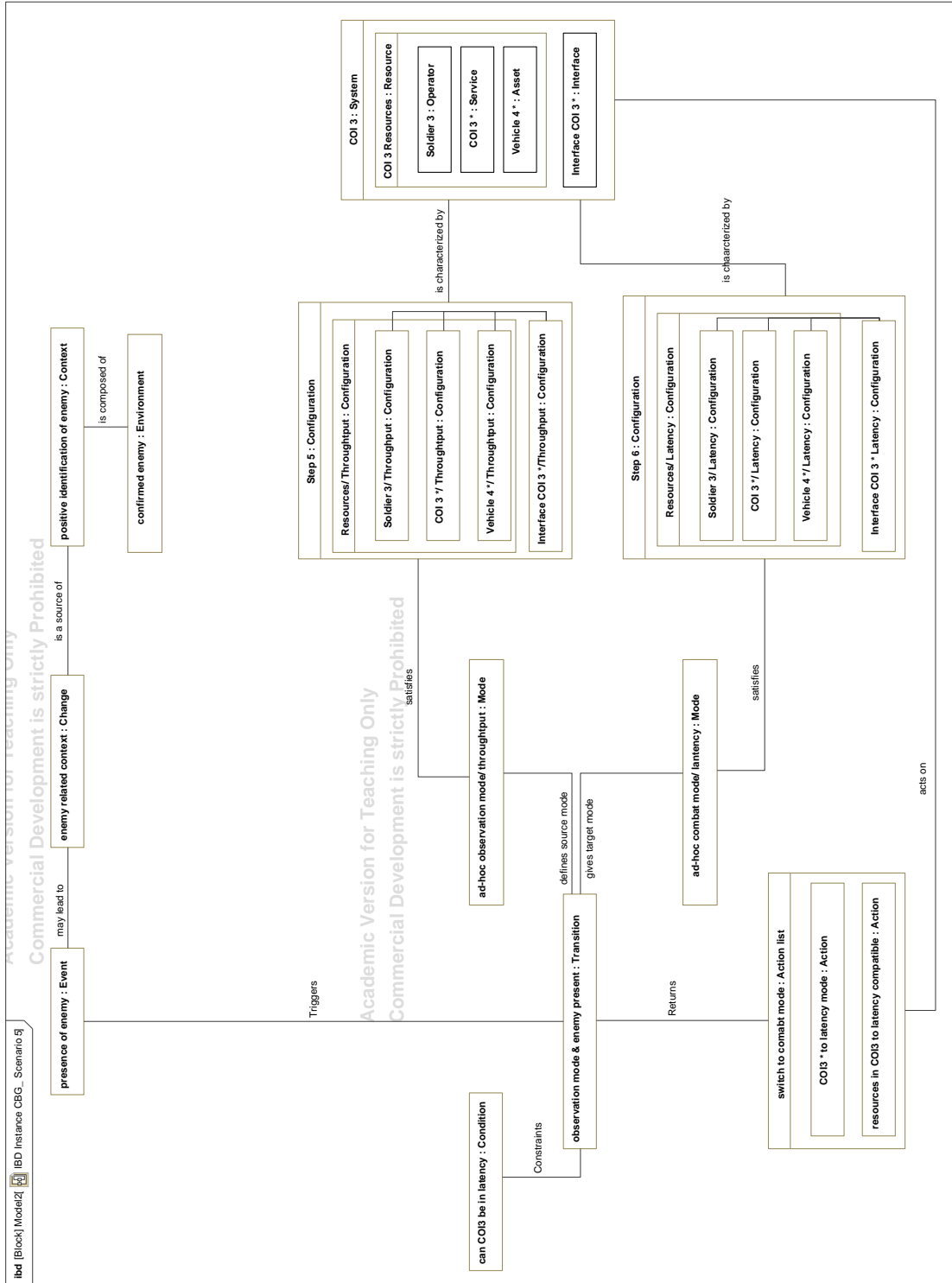


Figure A- 6: OSysRec instance for scenario 5

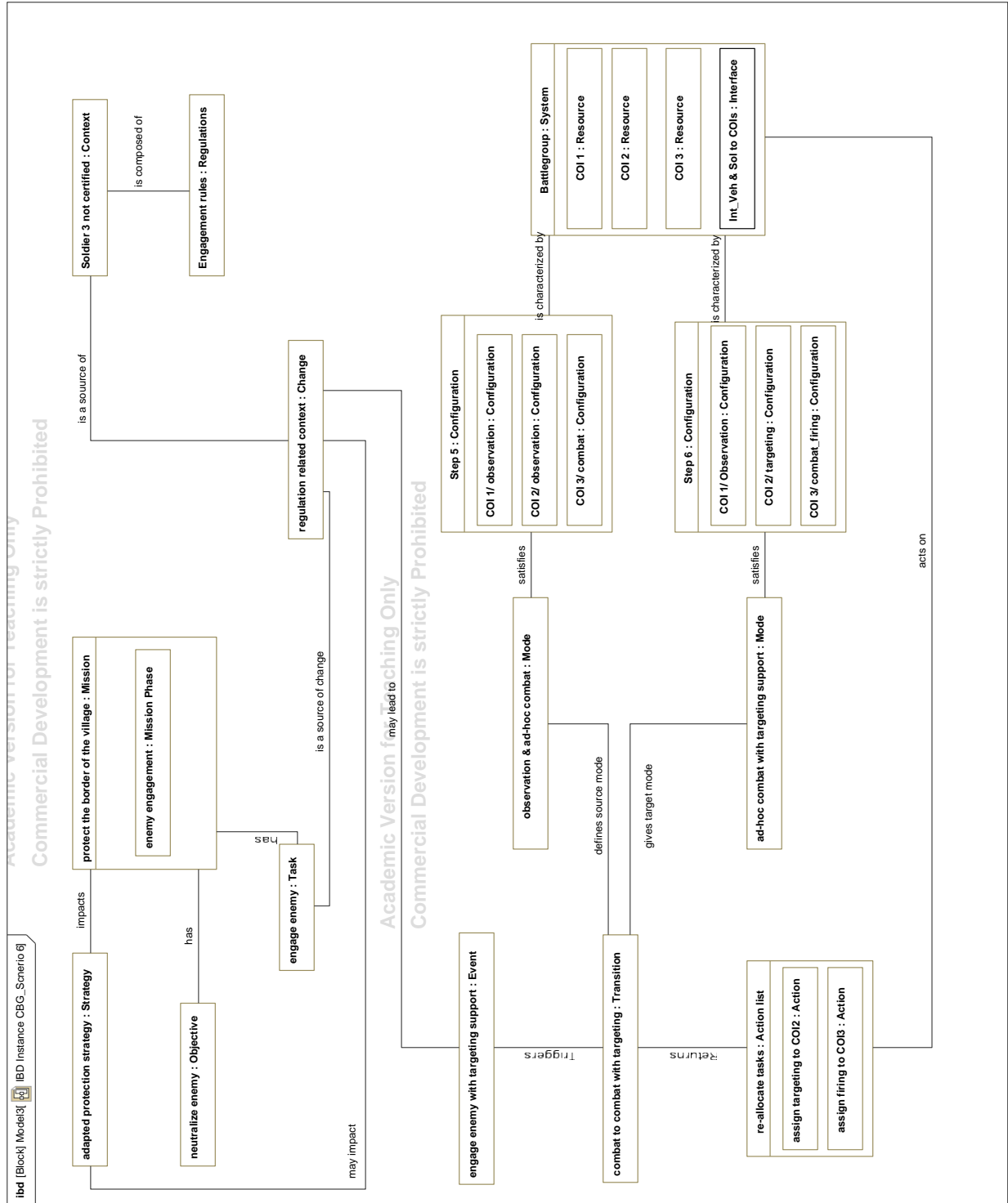


Figure A- 7: OSysRec instance for scenario 6

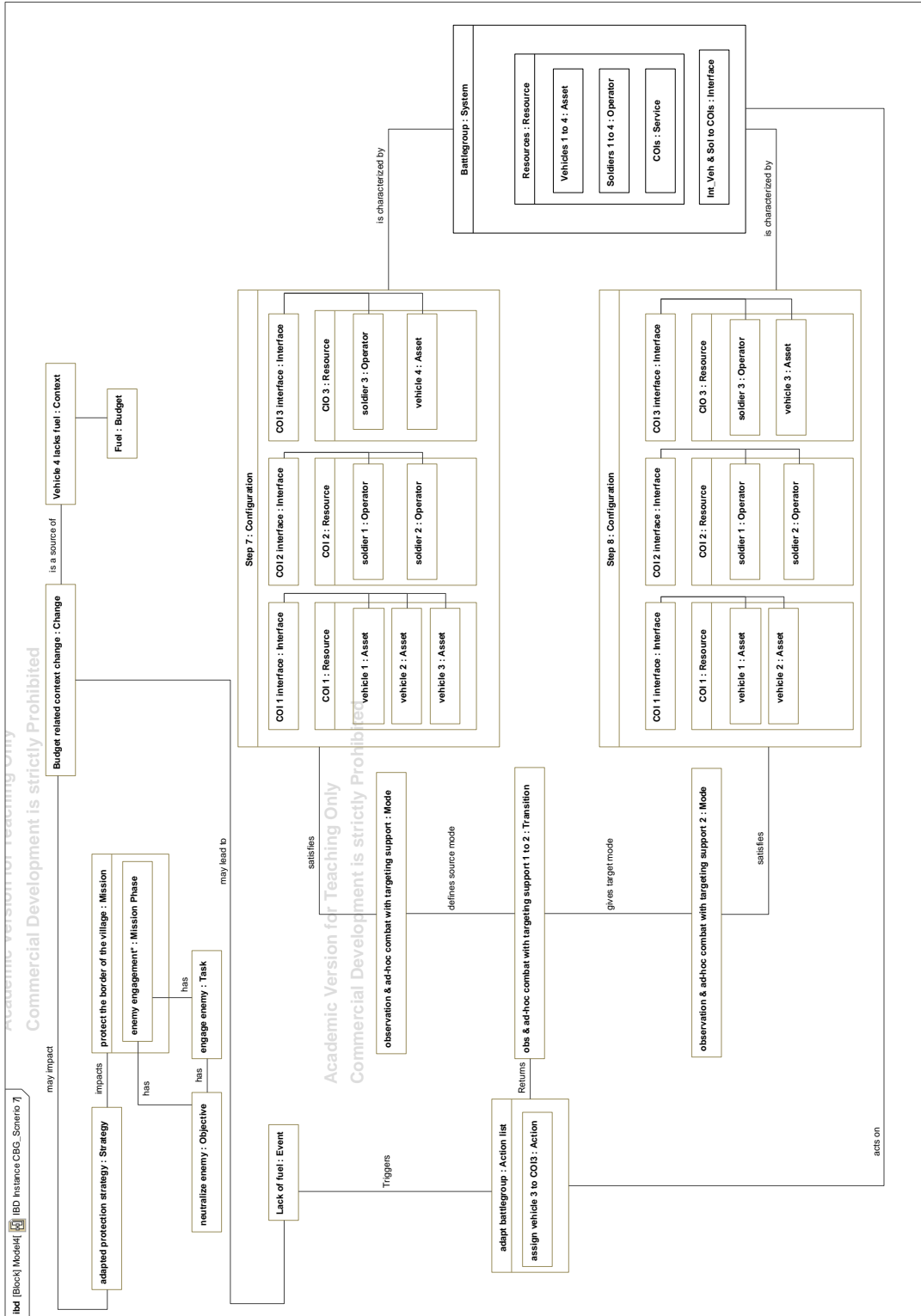


Figure A- 8: OSysRec instance for scenario 7

Résumé en Français

Synthèse et contributions de la thèse

Ce résumé a pour objectif de fournir une compréhension globale de cette thèse. Nous résumons les étapes qui ont été suivies pour mener cette recherche dans la partie 1. Tout d'abord, un audit industriel a été réalisé dans le domaine système de la Direction Technique de Thales (KTD) afin d'identifier les besoins et les difficultés rencontrés lors de la reconfiguration du système. Ensuite, nous avons effectué une revue de la littérature à la suite de l'analyse des résultats de l'audit industriel. Au regard des besoins industriels et les insuffisances dans la littérature, nous avons alors déterminé les questions de recherche. Enfin, nous présentons un résumé des contributions de la thèse dans la partie 2.

Partie 1: Synthèse

- **Synthèse des principaux points de l'audit industriel**

L'objectif de cette thèse est de supporter la reconfiguration de systèmes en utilisant des approches basées sur des modèles. Afin d'identifier les approches existantes et les principaux problèmes rencontrés lors du développement de fonctionnalités de reconfiguration de systèmes, nous avons réalisé un audit industriel au sein de Thales. L'audit industriel a couvert les différentes unités opérationnelles de Thales. Cet audit industriel est détaillé dans l'étude descriptive (chapitre 5). Dans ce résumé, nous soulignons les principaux défis que nous abordons dans cette thèse et qui sont liés au développement de fonctionnalités de reconfiguration.

Pour le développement des fonctionnalités de reconfiguration, l'étude descriptive a révélé qu'il est essentiel de capturer les données de reconfiguration, menant à une définition et une sélection exhaustive et efficace des configurations. Une reconfiguration efficace du système repose sur la compréhension de la conception du système en termes de ressources et de fonctions qu'elles fournissent. La reconfiguration doit également intégrer le retour d'information des opérations pour améliorer les performances globales du système. Il est donc nécessaire d'identifier les données requises venant de différentes sources et de les relier aux

mécanismes de reconfiguration. Etant donné que, chez Thales, les fonctionnalités de reconfiguration sont développées individuellement pour chaque projet, les données de reconfiguration collectées sont également spécifiques et fortement dépendantes du domaine. De plus, lors de la définition des données de reconfiguration, les ingénieurs et architectes système s'appuient sur les indications des utilisateurs finaux, ce qui peut entraîner une prise en compte insuffisante des variations de la mission et des changements dans les conditions d'exploitation du système. Par conséquent, des définitions ou des modèles standard identifiant les données de reconfiguration des systèmes sont nécessaires pour assurer le développement efficace des fonctionnalités de reconfiguration, ce qui permet d'améliorer l'efficacité opérationnelle des systèmes.

Outre l'identification des données, l'étude descriptive a également montré que le développement des fonctionnalités de reconfiguration nécessite l'identification des mécanismes et des processus nécessaires pour combiner les données de reconfiguration. Pour faire cela, il est important de comprendre comment les données peuvent être manipulées afin d'identifier les configurations pertinentes des systèmes. En outre, les mécanismes requis devraient soutenir une reconfiguration dynamique en termes d'allocation et de ré-allocation des ressources, permettant d'optimiser les capacités du système par rapport à ses missions. Les fonctionnalités actuelles de reconfiguration des systèmes ne permettent que des changements entre des configurations prédéfinies et ne prennent pas en compte la reconfiguration dynamique des systèmes pendant les opérations. Étant donné que les données de reconfiguration sont liées à la conception et aux opérations, le mécanisme de reconfiguration requis devrait pouvoir combiner et intégrer les données de ces deux phases et les utiliser pour soutenir la reconfiguration des systèmes en définissant, en évaluant et en sélectionnant les configurations de système pertinentes.

Les principales difficultés concernant le développement d'un support de reconfiguration peuvent être résumées en :

- L'absence de définition standard des données de reconfiguration.
- L'absence de mécanismes et de processus soutenant la reconfiguration dynamique pendant les opérations.

- **État de l'art scientifique et insuffisances de la recherche**

Compte tenu de l'objectif de la thèse, notre état de l'art (cf. chapitre 2) s'est concentré sur la compréhension de la notion de la reconfiguration ainsi que sur le domaine de l'ingénierie des systèmes et, plus spécifiquement, l'ingénierie des systèmes basée sur les modèles. La reconfiguration des systèmes a été traitée dans différents domaines, notamment : l'architecture des systèmes, la configuration des systèmes, la platformisation, la modularité, la commande et le contrôle, les systèmes embarqués, les systèmes de

fabrication reconfigurables, les systèmes autonomes, et les logiciels et systèmes informatiques. Sur la base de la synthèse de la littérature fournie dans la section 2.3, nous avons identifié des insuffisances en matière de recherche. Dans ce résumé, nous soulignons les insuffisances en matière de recherche que nous abordons dans cette thèse.

Comme nous l'avons vu dans la section 2.3, la reconfiguration des systèmes repose sur trois aspects principaux : la structure, la dynamique et la gestion. L'aspect structurel est principalement lié aux composants et aux interfaces du système. L'aspect dynamique concerne le comportement, l'évolution et les transitions du système. L'aspect gestion concerne l'optimisation des ressources existantes en fonction du contexte et de la mission considérés. Afin de maîtriser la reconfiguration du système, il est essentiel de considérer ce processus dans sa globalité.

L'ingénierie des systèmes basée sur les modèles est apparue pour soutenir la conception et la gestion de systèmes complexes (Madni et Sievers, 2017) tout en réduisant les coûts et le temps grâce à l'utilisation et à la réutilisation des modèles (Wymore, 1993). Une approche importante utilisée comme base dans le MBSE est le développement d'ontologies. Les ontologies aident à formaliser la connaissance des domaines en fournissant les vocabulaires spécifiques à un domaine et les relations entre eux, ce qui est une condition préalable au développement d'outils MBSE. Par conséquent, nous pensons qu'une ontologie ou un modèle de données est nécessaire pour soutenir la reconfiguration du système d'une manière globale.

À partir de notre analyse de l'état de l'art, nous avons identifié plusieurs ontologies qui ont été développées pour soutenir la reconfiguration des systèmes (Ali et al., 2011 ; Bermejo-Alonso et al., 2011, 2016 ; Gogniat et al., 2013 ; Hernández et al., 2015 ; Krichen et Zalila, 2011 ; Meyer et al., 2013 ; OMG, 2010 ; Walsh et al., 2005, 2006 ; Witt et al., 2013). Toutefois, à notre connaissance, il n'existe pas d'ontologie de reconfiguration du système qui prenne en compte le processus global de reconfiguration du système et qui traite tous ses aspects connexes. En outre, lorsque l'on examine les méthodes et les mécanismes de reconfiguration du système, on peut constater que les méthodes existantes reposent soit sur l'intégration des données de conception du système, soit sur les données d'observation des opérations concernant la santé du système et son contexte. Par conséquent, les méthodes existantes ne parviennent pas à intégrer les données nécessaires provenant à la fois de la conception et des opérations.

Les principales insuffisances dans la recherche peuvent être alors résumées comme :

- Un manque de modèles de données ou d'ontologies qui prennent en compte et synthétisent les trois aspects de la reconfiguration des systèmes (c'est-à-dire la structure, la dynamique et la gestion). Une analyse approfondie de l'état de l'art qui a permis d'identifier cette insuffisance est présentée au chapitre 6.

- Un manque de méthodes de reconfiguration intégrant les données de reconfiguration du système et faisant le lien entre les phases d'ingénierie et d'opérations. Une analyse bibliographique spécifique concernant cette insuffisance est fournie au chapitre 7.

- **Les écarts entre la pratique industrielle et la recherche**

La comparaison entre l'état de l'art et les besoins industriels permet d'identifier les insuffisances à combler pour soutenir la reconfiguration du système.

Cette thèse porte sur le développement des supports de reconfiguration. Comme nous l'avons vu précédemment, la saisie des données de reconfiguration est cruciale pour la reconfiguration du système car elle est liée à la définition et à la sélection de la configuration. La reconfiguration du système repose sur des données provenant à la fois de la conception et de l'exploitation du système. Notre audit industriel nous a permis d'identifier un manque de définitions ou de modèles standard décrivant les données de reconfiguration du système qui pourraient être utilisées dans le développement de supports de reconfiguration. Sur la base de notre analyse de l'état de l'art, nous avons identifié trois aspects de la reconfiguration des systèmes qui doivent être considérés conjointement (la structure, la dynamique et la gestion). Cependant, jusqu'à présent, ces aspects sont considérés séparément. Par conséquent, nous pensons qu'un modèle de données ou une ontologie synthétisant ces trois aspects de la reconfiguration du système est nécessaire.

En outre, le développement de supports et de fonctionnalités de reconfiguration nécessite des méthodes permettant l'allocation et la ré-allocation des ressources pour soutenir la reconfiguration dynamique pendant les opérations. Étant donné que les données de reconfiguration sont liées à la conception et aux opérations, le processus ou le mécanisme de reconfiguration requis doit permettre de combiner les données des deux phases. La méthode requise devrait soutenir la reconfiguration du système en définissant, en évaluant et en sélectionnant les configurations pertinentes du système. Cependant, sur la base des insuffisances de recherche précédemment identifiées, nous pensons que, pour soutenir la reconfiguration du système, il est nécessaire de disposer d'une méthode intégrant les données liées aux observations (provenant des opérations) et à la conception du système (provenant de l'ingénierie).

À la lumière de l'audit industriel et de l'analyse de l'état de l'art, nous pensons que pour soutenir le développement des fonctionnalités de reconfiguration des systèmes, nous devons :

- Identifier et synthétiser les données nécessaires à la reconfiguration des systèmes ;

- Proposer une méthode de reconfiguration adéquate intégrant les données relatives aux observations (provenant des opérations) et à la conception du système (provenant de l'ingénierie).

Sur la base de ces exigences et de l'objectif de la thèse, nous avons formulé les questions de recherche présentées dans la section suivante.

- **Questions de recherche**

Ce travail de recherche a été lancé pour répondre à l'objectif global suivant :

Objectif de recherche : soutenir la reconfiguration du système en utilisant une approche basée sur un modèle.

Pour atteindre cet objectif de recherche, nous avons dû identifier les approches, méthodes et outils existants qui sont utilisés pour soutenir la reconfiguration du système. Il était également crucial de saisir les besoins et les insuffisances dans le paysage industriel car ils nous permettent d'identifier les caractéristiques et l'utilisation du support souhaité. Par conséquent, l'objectif général de la recherche a été affiné à la question suivante qui a guidé la recherche et qui a permis une compréhension et une identification initiales des défis rencontrés :

QR1 : Quelles sont les approches, méthodes et outils existants pour la reconfiguration des systèmes, et quelles sont les difficultés qui y sont liées ?

Afin de répondre à cette question, nous avons mené une étude descriptive. A partir de cette étude, nous avons identifié les approches, méthodes et outils existants pour la reconfiguration des systèmes. En outre, l'étude descriptive nous a permis d'identifier les principaux problèmes et les insuffisances dans l'état de l'art que nous avons abordé au-dessus. Cette question est abordée plus en détail au chapitre 5.

Sur la base des résultats de l'étude descriptive, cette recherche s'est concentrée sur la difficulté concernant le développement d'un support de reconfiguration du système. En particulier, nous avons identifié deux défis spécifiques liés à cette difficulté, notamment l'identification et l'intégration des données de reconfiguration. L'état de l'art manque de modèles de données qui décrivent les données de reconfiguration de manière globale. De plus, les méthodes existantes n'intègrent pas les données provenant à la fois des observations (provenant des opérations) et des modèles de conception du système (provenant de l'ingénierie).

Étant donné les défis industriels et les manques de la recherche résultant de l'étude descriptive, nous avons formulé deux questions de recherche.

QR2 : Quelles sont les données requises pour la reconfiguration du système et comment peut-on les formaliser ?

QR3 : Comment pouvons-nous intégrer les données pré-identifiées pour soutenir la reconfiguration du système ?

En partant de l'objectif de recherche, afin de soutenir la reconfiguration du système à l'aide d'une approche basée sur un modèle, nous avons dû identifier les approches, processus, méthodes et outils existants. Ce besoin a conduit à notre première question de recherche (QR1). Pour répondre à cette question, dans un premier temps, nous avons mené une étude empirique et un état de l'art, comme le recommande la méthodologie de recherche dans la conception (DRM) (Blessing et Chakrabarti, 2009). Cette étape a abouti à une étude descriptive qui a souligné les défis industriels et les manques de la recherche. La contribution issue de cette étape est l'étude descriptive détaillée au chapitre 5 intitulé " A Descriptive Study of the System Reconfiguration Process Over System Life Cycle Stages".

Sur la base de l'étude descriptive, nous avons identifié un défi concernant l'identification et la formalisation des données. Naturellement, cela a conduit à la deuxième question de recherche (QR2). Pour répondre à cette question, la deuxième étape de cette recherche s'est concentrée sur le développement d'une ontologie de reconfiguration du système. La méthodologie utilisée pour construire cette ontologie a été adaptée de Pinto et al. (Pinto et Martins, 2004). Le principal résultat de cette étape est l'ontologie OSysRec. Cette contribution est détaillée dans le chapitre 6, intitulé "An overall Ontology for System Reconfiguration using Model-Based System Engineering".

L'étude descriptive a également mis en évidence un défi concernant l'intégration des données pour soutenir la reconfiguration du système. Il est donc nécessaire d'aborder la question liée à l'intégration des données (QR3). Dans un troisième temps, cette partie de la thèse de doctorat s'est concentrée sur le développement d'un support de reconfiguration qui permet l'utilisation et l'intégration des données. Pour développer le support de reconfiguration du système, nous avons conçu une approche basée sur l'identification des exigences du support nécessaire. Une analyse approfondie de l'état de l'art a également été réalisée afin d'identifier les méthodes et outils de configuration/reconfiguration du système existant. Cette étape a abouti à la proposition d'une méthode basée sur un modèle pour soutenir la reconfiguration du système pendant les opérations. Nous appelons cette méthode MBSysRec. Cette contribution est détaillée au chapitre 7, intitulé " A Model-based Method for System Reconfiguration".

Afin de donner un aperçu de la thèse de doctorat, nous proposons de résumer les différentes contributions (partie 2) en termes d'objectifs et de principaux résultats nécessaires pour donner une compréhension globale de cette recherche.

Partie 2 : Contributions de la thèse

- **Une étude descriptive du processus de reconfiguration du système au cours des étapes du cycle de vie du système**
 - **Objectif**

L'objectif de la recherche est de proposer une approche basée sur un modèle pour soutenir la reconfiguration des systèmes pendant les opérations. Afin de clarifier le sujet de recherche et de mieux formuler la problématique de recherche, nous avons mené une étape de clarification de la recherche et l'avons complétée par une étude descriptive. Même si nous nous concentrons sur la reconfiguration en cours d'opération, d'autres étapes du cycle de vie méritent d'être étudiées car elles peuvent offrir d'autres perspectives de développement futur. Dans l'étude descriptive, nous considérons la reconfiguration en cours d'exploitation ainsi que d'autres étapes du cycle de vie, la phase de conception en particulier, car les configurations du système résultent des activités de conception. L'objectif principal de l'étude descriptive est de saisir les insuffisances dans le contexte industriel étudié et d'identifier les besoins et les défis industriels liés.

- **Résultats**

Lors de l'examen de la phase opérationnelle, les experts ont exprimé le besoin de développer des capacités génériques de gestion du système. En tant qu'outil de gestion du système, une reconfiguration en termes d'affectation et de réaffectation des ressources est hautement nécessaire. En particulier, les experts ont souligné la nécessité de gérer conjointement les fonctions et les ressources du système, ce qui permettra au système d'atteindre les objectifs de sa mission tout en tenant compte de l'impact du contexte (par exemple, l'environnement) à la fois sur le système et sur sa mission. Les défis liés à la reconfiguration du système

pendant les opérations ont été identifiés. Les principaux défis comprennent la disponibilité, l'accessibilité, le stockage et la vérification des données, les questions de modélisation, la passation de marchés et la certification, la taxonomie du système et du contexte, ainsi que les défis liés au développement d'un support de reconfiguration.

Il est nécessaire de soutenir la reconfiguration afin de définir les ressources nécessaires à la réalisation des objectifs de la mission à partir d'un ensemble de ressources disponibles (par exemple, les bateaux, les opérateurs et les caméras effectuant des missions de recherche et de sauvetage). Les défis spécifiques liés au développement d'un support de reconfiguration pour la phase opérationnelle ont également été identifiés. Tout d'abord, l'identification des données nécessaires à la reconfiguration du système. En particulier, il y a un manque de modèles qui considèrent le processus de reconfiguration d'une manière globale. Deuxièmement, définir une méthode de reconfiguration qui utilise et intègre les données concernant la structure et les fonctions du système ainsi que les observations. Ces deux défis sont abordés dans cette recherche. Ces défis ont été examinés à la lumière de la littérature existante et deux insuffisances ont été identifiées dans la recherche. L'état de l'art manque de modèles de données qui synthétisent les données qui sont liées aux différents aspects de la reconfiguration du système. En outre, les méthodes existantes n'intègrent pas les données relatives aux observations et au modèle de conception du système. Par conséquent, dans le reste de cette recherche, nous nous concentrons sur la résolution de ces insuffisances.

- **Une ontologie globale pour la reconfiguration des systèmes à l'aide de l'ingénierie des systèmes basée sur les modèles**

- **Objectif**

La reconfiguration des systèmes de soutien dans l'industrie reste un défi (Qasim, Jankovic, et al., 2019), en particulier en termes d'identification des données, car elle est directement liée au développement des supports industriels. L'identification des données de reconfiguration est difficile car la reconfiguration est un problème multi-domaine. Trois aspects clés peuvent être identifiés lorsque l'on considère la reconfiguration d'un système du point de vue de l'ingénierie des systèmes : la structure, la dynamique et la gestion. L'aspect structurel définit les ressources constituant le système d'intérêt, et les fonctions qu'elles assurent, ainsi que les connexions entre elles. L'aspect dynamique traite du comportement et de l'évolution du système, en fonction des événements, des conditions et des transitions. Enfin, l'aspect gestion concerne

L'optimisation des ressources existantes en fonction du contexte considéré et de la mission. Tous ces aspects, ensemble, sont importants pour faire face aux changements (internes ou externes), ayant un impact sur le système tout au long de son cycle de vie vers une gestion efficace du système (INCOSE Systems Engineering Handbook V4, 2015 ; ISO/IEC/IEEE:15288, 2015). Il est donc nécessaire de fournir une représentation unifiée de tous les aspects du processus de reconfiguration.

L'ingénierie des systèmes basée sur les modèles (MBSE) est une approche (Madni et Sievers, 2017) qui soutient la conception de systèmes complexes en développant des approches et des modèles (Chrisp et Richard, 2007), tout en réduisant les coûts et le temps grâce à l'utilisation et la réutilisation des modèles (Wymore, 1993). Dans le MBSE, le développement d'ontologies représente une approche importante utilisée pour soutenir la conception de systèmes complexes. L'intérêt de développer des ontologies a gagné en importance car elles permettent de formaliser les données d'un domaine en fournissant les vocabulaires spécifiques à un domaine et les relations entre eux. Pour faire face à la nature multi-aspects de la reconfiguration, il faut s'attaquer au processus global. C'est pourquoi, dans le chapitre 6, nous proposons une ontologie de reconfiguration du système, que nous appelons OSysRec, qui synthétise les aspects clés nécessaires pour soutenir la reconfiguration du système et qui ont été considérés indépendamment jusqu'à présent.

- **Résultats**

L'ontologie de reconfiguration du système (OSysRec) est une ontologie fondamentale qui traite du processus de reconfiguration dans les différents domaines. L'objectif est de synthétiser les données dans une ontologie globale. L'ontologie OSysRec fournit à ses utilisateurs un cadre conceptuel complet qui permet une analyse systématique des différents problèmes et solutions concernant l'évolution des systèmes.

La vue macroscopique de l'ontologie OSysRec a été formalisée dans l'outil Cameo System Modeler en utilisant un diagramme de package interconnecté. Ces paquets, correspondant aux aspects de reconfiguration du système (c'est-à-dire la structure, la dynamique et la gestion), sont visualisés de manière à montrer leur nature hiérarchique. Dans ce résumé, nous ne décrivons que le flux de données entre les différents aspects. Nous fournissons également l'aperçu nécessaire de l'ontologie pour permettre la compréhension du reste de ce résumé.

La partie structurelle répond à la dynamique requise via la configuration. Les informations relatives aux éléments structurels et à la dynamique qui leur est associée doivent être communiquées au niveau de la gestion. Ces flux de données sont générés par les systèmes de suivi de la santé et des usages et les mécanismes de supervision (ISO, 2012, 2015, 2018). Le Mode_effectif et l'État_effectif représentant les

parties dynamiques et structurelles de cette ontologie sont utilisés dans la partie gestion. Une fois qu'une décision a été prise au niveau de la gestion, elle peut avoir un impact sur le niveau dynamique et, par conséquent, sur le niveau structurel. Cet impact se traduit par un événement au niveau dynamique et une action au niveau structurel.

Les systèmes subissent généralement des reconfigurations pour assurer l'efficacité opérationnelle via la gestion des ressources disponibles dans le système en liaison avec les évolutions des missions et des contextes. Les concepts les plus importants dans l'aspect gestion sont l'Objectif, le Contexte, la Situation effective et attendue, ainsi que le Changement.

La mission, qui est l'objectif général auquel toutes les ressources doivent être affectées, a au moins un objectif. Par exemple, dans une mission de recherche et de sauvetage maritime (SAR), un des objectifs est de pouvoir observer un objet de 1 m² à une distance de 1 km. Le contexte est ce qui caractérise la situation dans laquelle le système d'intérêt est exploité. Le contexte est une composition de différentes catégories, dont les règlements, l'environnement, le temps, le système de contexte et l'utilisateur. Dans l'exemple du SAR, un contexte peut être un temps orageux. Lorsque le contexte et l'objectif sont tous deux fixés, la reconfiguration du système peut toujours avoir lieu en réponse à des défaillances du système. Ces défaillances peuvent survenir soit parce que les ressources ne produisent pas les effets requis, soit parce qu'elles ne sont pas disponibles. Par conséquent, la situation effective est caractérisée par le mode effectif et l'état effectif, qui reflète la justesse du comportement attendu et l'état de santé du système. La situation effective est comparée à la situation attendue, qui se compose de la même manière du mode attendu et de l'état attendu. Par exemple, être dans un mode où seule l'observation est requise avec une caméra défectueuse. Les évolutions des objectifs et des contextes, ainsi qu'un écart de situation, sont tous considérés comme des sources de changements qui peuvent conduire à des reconfigurations. Dans un exemple de SAR, la situation d'observation avec une caméra défectueuse peut être corrigée en engageant d'autres équipements disponibles.

Les tâches d'une mission sont fournies par les modes attendus. Une ou plusieurs fonctions caractérisent un mode engagé au niveau dynamique. Ces fonctions sont fournies par les ressources du système, qui font partie du système au niveau structurel. À ce niveau, le système est caractérisé par des Configurations. Par exemple, dans le SAR, la tâche de détection est fournie par le mode d'observation, ayant pour fonctions d'observer et de fonctionner. Ces deux fonctions sont fournies par la caméra et les ressources de l'opérateur par la suite. Dans ce cas précis, le système est caractérisé par la configuration d'observation qui engage la caméra et l'opérateur pour satisfaire le mode d'observation.

Dans le chapitre 6, l'ontologie OSysRec est illustrée à l'aide de l'exemple du SAR. L'ontologie a été testée avec de nombreux scénarios issus de différentes études de cas industriels. Dans le domaine aérospatial, l'avionique modulaire intégrée (IMA) et la gestion du trafic aérien ont été utilisées pour instancier l'ontologie. Dans le domaine militaire et des communications : les radios tactiques de nouvelle génération intégrant une technologie radio logicielle innovante, la gestion déléguée de la flotte pour les systèmes des forces terrestres (véhicules et charge utile), le système du soldat du futur et le groupement tactique connecté. Dans le chapitre 6, nous détaillons l'étude de cas du groupement tactique connecté et l'utilisons pour illustrer l'ontologie OSysRec. Les scénarios liés à l'étude de cas IMA sont également illustrés.

- **Une méthode basée sur les modèles pour la reconfiguration du système**
 - **Objectif**

La reconfiguration du système repose sur différentes données liées à la structure, la dynamique et la gestion du système et synthétisées dans l'ontologie OSysRec. Les concepts les plus importants dans l'aspect gestion de l'ontologie OSysRec comprennent l'objectif, le contexte, la situation effective et attendue, et le changement. Par exemple, dans les missions de recherche et de sauvetage, nous pouvons avoir besoin de rechercher une victime pendant la nuit en utilisant une caméra qui ne fonctionne qu'à 70%. Dans ce contexte, la reconfiguration est essentielle car le sauvetage dépend principalement de la manière de gérer les ressources disponibles et de les orienter vers les objectifs requis. Certains des défis identifiés dans la proposition de support à la reconfiguration du système dans le cadre de la pratique industrielle actuelle de l'entreprise sont 1) la prise en compte du contexte de déploiement, 2) l'évolution des missions et des objectifs, 3) l'intégration des données des capteurs. Ces concepts sont liés aux données concernant l'architecture du système provenant de la phase d'ingénierie (c'est-à-dire les composants, les fonctions et leurs relations), et aux données provenant des observations (c'est-à-dire les données des capteurs) reflétant le niveau réel de fonctionnement du système déployé. Ainsi, le chapitre 7 vise à soutenir la reconfiguration du système en proposant une approche reliant les deux phases, l'ingénierie et les opérations.

- **Résultats**

Au chapitre 7, nous proposons une méthode de reconfiguration du système basée sur les modèles, que nous appelons MBSysRec. Ce support intègre les informations concernant le contexte opérationnel, l'objectif et le niveau de fonctionnement du système qui sont essentielles pour la reconfiguration du système.

MBSysRec vise à relier les phases d'ingénierie et d'exploitation en intégrant les données provenant de ces deux phases. Les entrées de la phase d'ingénierie reposent sur le modèle de données du système en termes de fonctions, de composants et de leurs relations, ainsi que sur toutes les informations concernant les performances des composants, les pondérations des critères et les interactions. Les apports de la phase opérationnelle, en revanche, sont principalement liés aux données des capteurs reflétant les informations sur la disponibilité et le niveau de fonctionnement des composants. Outre les données des capteurs, le contexte opérationnel du système et l'objectif qu'il entend atteindre sont également considérés comme des entrées de données opérationnelles. Toutes ces informations sont stockées dans des matrices au sein d'un système matriciel qui les relie. MBSysRec dans ses trois phases (c'est-à-dire MBSysRec-Observe, MBSysRec-Orient et MBSysRec-Decide) est mis en œuvre par le biais de trois fonctions principales au sein d'un programme Matlab.

MBSysRec-Observe vérifie en permanence le contexte, l'objectif et la situation effective. Le contexte et l'objectif sont saisis par l'utilisateur, tandis que la situation effective est lue directement à partir des matrices consacrées au stockage des données du système de surveillance de la santé et de l'utilisation. Lorsqu'un changement est détecté, les informations d'instanciation sont mises à jour, et le processus de reconfiguration est déclenché par l'appel du bloc MBSysRec-Orient.

La fonction MBSysRec-Orient vise à identifier un ensemble de configurations possibles, si elles sont trouvées, en utilisant les informations stockées dans le système de matrices en ce qui concerne le contexte et l'objectif. MBSysRec-Orient recherche les configurations des solutions dans un ensemble de ressources disponibles mises à jour. L'ensemble des configurations possibles est renvoyé et saisi dans la fonction MBSysRec-Decide. Pour cette fonction, en plus des opérations matricielles, nous avons adopté la méthode proposée par Bryant et al. (Bryant et al., 2005) pour générer les configurations du système.

La fonction MBSysRec-Decide est une phase en deux étapes consistant à évaluer et à sélectionner les configurations. Dans la phase d'aide à la décision, nous utilisons à la fois des critères définis par l'utilisateur, qui se trouvent dans le modèle de données du système, et deux critères fixes. Les deux critères fixes qui sont, à notre avis, essentiels pour la reconfiguration du système comprennent 1) le niveau global de configuration du fonctionnement, et 2) l'impact du changement. Le niveau de configuration du fonctionnement donne la capacité globale qui peut être réalisée par une configuration. L'impact du

changement décrit l'effort nécessaire pour déployer une configuration tout en considérant la configuration actuelle comme point de départ. Les configurations résultant de la phase d'orientation sont évaluées sur la base de ces critères, et enfin, un niveau de satisfaction global est donné en utilisant une technique de prise de décision multicritères. Pour la prise de décision multicritère, nous avons adopté la méthode intégrale de Choquet (Labreuche, 2011a) et ses deux indices, l'indice de forme et l'indice d'interaction, pour tenir compte du poids des différents critères et de leurs interactions. L'étape de sélection consiste à 1) filtrer les configurations résultantes par rapport au seuil de satisfaction défini par l'utilisateur, et 2) choisir la configuration ayant le niveau de satisfaction maximum parmi les configurations satisfaisant au seuil.