Institutionen för datavetenskap

Department of Computer and Information Science

Master's Thesis

Model-Based Hazard Analysis of Undesirable Environmental and Components Interaction.

by

Hoda Mehrpouyan

LIU-IDA/LITH-EX-A—11/037—SE.

2011-08-23

Linköpings universitet

Master's Thesis

Model-Based Hazard Analysis of Undesirable Environmental and Components Interaction.

by

Hoda Mehrpouyan

LIU-IDA/LITH-EX-A—11/037—SE.

2011-08-23

Supervisor: Peter Bunus

Examiner: Peter Bunus

Presentation Date 2011-09-23 Publishing Date (Electronic version) _____	Department and Division Department of Computer and Information Science. Software and Systems.	Linköpings universitet
--	--	--

Language ☒ English ☐ Other (specify below) Number of Pages 72	Type of Publication ☐ Licentiate thesis ☒ Degree thesis ☐ Thesis C-level ☐ Thesis D-level ☐ Report ☐ Other (specify below) _____	ISBN -- ISRN: LIU-IDA/LITH-EX-A—11/037—SE. Title of series -- Series number/ISSN --
--	---	--

URL, Electronic Version http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-xxxx (Replace xxxx with the correct number)
--

Publication Title Model-Based Hazard Analysis of Undesirable Environmental and Components Interaction. Author(s) Hoda Mehrpouyan

Abstract <p>Identifying the detrimental effect of environmental factors and subsystem interactions are one of the most challenging aspects of early hazard assessment in the design of complex safety critical systems. Therefore, a complete understanding of potential failure effects before the catastrophe happens is a very difficult task. The thesis proposes a model-based hazard analysis procedure for early identification of potential safety issues caused by unexpected environmental factors and subsystem interactions within a complex safety critical system. The proposed methodology maps hazard and vulnerability modes to specific components in the system and analyzes the hazard propagation paths for risk control and protection strategies. The main advantage of the proposed method is the ability to provide the designers with means to use low-fidelity, high level models to identify hazardous interactions. Using this technique, designers can examine the collective impacts of environmental and subsystem risks on overall system during early stages of design and develop a hazard mitigation strategy.</p>
--

Keywords Hazard and vulnerability analysis; Conceptual modelling; environmental and subsystem interaction risks; Fail-free component design; Risk-based design.

Dedication

To my parents, for their unconditional love, affection and support.

Abstract

Identifying the detrimental effect of environmental factors and subsystem interactions are one of the most challenging aspects of early hazard assessment in the design of complex safety critical systems. Therefore, a complete understanding of potential failure effects before the catastrophe happens is a very difficult task. The thesis proposes a model-based hazard analysis procedure for early identification of potential safety issues caused by unexpected environmental factors and subsystem interactions within a complex safety critical system. The proposed methodology maps hazard and vulnerability modes to specific components in the system and analyzes the hazard propagation paths for risk control and protection strategies. The main advantage of the proposed method is the ability to provide the designers with means to use low-fidelity, high level models to identify hazardous interactions. Using this technique, designers can examine the collective impacts of environmental and subsystem risks on overall system during early stages of design and develop a hazard mitigation strategy.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisor and advisor Dr. Peter Bunus. Peter taught me how to question thoughts, express ideas and become more confident as a researcher. I am indebted to him for his valuable help, support and encouragement throughout my study. I hope one day, I also become as good an advisor as Peter to be able to steer my students to the higher realms of the imagination.

I would also like to thank Dr. Tolga Kurtoglu at Palo Alto Research Center for initiating this thesis and for his contribution and help during the process.

Contents

CHAPTER 1.....	- 3 -
INTRODUCTION.....	- 3 -
1.1 Background	- 3 -
1.2 Goals	- 4 -
1.3 Method	- 4 -
1.4 Thesis Outline.....	- 4 -
CHAPTER 2.....	- 6 -
BACKGROUND.....	- 6 -
2.1 Modeling and Simulation.....	- 6 -
2.2 Modeling Paradigms and Languages.....	- 6 -
2.2.1 Object-Oriented Modeling.....	- 6 -
2.3 Model-Based System Engineering (MBSE)	- 7 -
2.3.1 System Modeling Language (SysML).....	- 8 -
2.4 Metamodel.....	- 9 -
2.4.1 Heavyweight vs. lightweight design.....	- 10 -
2.4.2 Profile and Stereotype	- 11 -
2.4.3 SysML Profile	- 12 -
2.4.4 SysML Diagrams.....	- 12 -
2.4.5 SysML Block Definition Diagram.....	- 12 -
2.4.6 SysML BDD Metamodel and Elements supported by MagicDraw	- 12 -
2.4.7 SysML Requirement Diagram.....	- 13 -
2.4.8 SysML Cross Connecting Model Element	- 15 -
2.5 SysML MagicDraw modeling Environment.....	- 16 -
CHAPTER 3.....	- 17 -
RELATED WORK.....	- 17 -
3.1 Technical Background and Related work.....	- 17 -
3.2 Design of Complex Systems	- 17 -
3.2.1 Characteristics of a Complex System	- 17 -
3.2.2 Complex System Engineering Life Cycle	- 18 -
3.2.3 Safety analysis in Complex System.....	- 19 -
3.3 Imofis industrial European project	- 19 -
3.4 New Generation of End-to-End Tools for Safety-critical System Design.....	- 20 -
3.5 Functional Hazard Analysis (FHA)/ Functional Failure Analysis (FFA).....	- 21 -
3.6 Failure Mode and Effects Analysis (FMEA).....	- 22 -
3.7 Systems-Theoretic Accident Modeling and Processes (STAMP).....	- 23 -
3.8 Hazard and Operability Studies (HAZOP).....	- 23 -
CHAPTER 4.....	- 25 -
TOWARDS A MODEL-BASED HAZARD ANALYSIS FRAMEWORK.....	- 25 -
CHAPTER 5.....	- 27 -
HAZARD TYPES AND IDENTIFICATION.....	- 27 -

CHAPTER 6.....	- 29 -
MODEL-BASED AUTOMATED SAFETY ANALYSIS	- 29 -
6.1 <i>The NASA Adapt System</i>	- 30 -
6.2 <i>Satellite Electrical Power System</i>	- 31 -
6.3 <i>Identifying Hazard-Vulnerable Pairs</i>	- 32 -
6.4 <i>Hazard Path Analysis</i>	- 34 -
6.5 <i>Hazard Analysis Results in Design Mitigation</i>	- 36 -
CHAPTER 7.....	- 38 -
DESIGN SPACE EXPLORATION	- 38 -
CHAPTER 8.....	- 40 -
CONCLUSIONS	- 40 -
8.1 <i>Accomplishments</i>	- 40 -
8.2 <i>Future Work</i>	- 40 -
REFERENCES.....	- 41 -
APPENDIX A	- 44 -
XMI HAZARD PROPAGATION PATH	- 44 -
APPENDIX B	- 47 -
EPS XMI CODE FROM BLOCK DEFINITION DIAGRAM	- 47 -

List of Figures

Figure 1 Annotated diagram of the Mars Polar Lander spacecraft.....	4 -
Figure 2 Relationship between SysML and UML [31]	9 -
Figure 3 Language definition and use for a vehicle controller interface specification in model-based engineering form: metamodel definition (a), user-defined model (b), and model repository (c) [33].	10 -
Figure 4 Uses of metamodeling techniques related to concepts of the real-time domain: heavyweight approaches (a), lightweight approaches with reused elements (b), and optional connection to domain-specific modeling add-ins (c) [33].	11 -
Figure 5 Block Element and MagicDraw SysML Block Subtypes Metamodel [34].	13 -
Figure 6 Requirement Metamodel [34]	14 -
Figure 7 Cross Connecting Model Elements [31]	15 -
Figure 8 Double V Lifecycle Model for Complex System Engineering [39]	18 -
Figure 9 Tool integration via the Speculative and Exploratory Design in System Engineering (Speeds) [39]	20 -
Figure 10 The HAZOP Process [54]	24 -
Figure 11 Overview of the Model-Based Hazard Analysis Process.....	25 -
Figure 12 SysML diagram taxonomy [5]	29 -
Figure 13 ADAPT System at NASA Ames Research Center	31 -
Figure 14 Model of the existing electrical power system design architecture	31 -
Figure 15 An Example of Text-Based Requirement Transformation to SysML Requirement Diagram -	33 -
Figure 16 SysML Block Definition Diagram of Simple EPS System.....	34 -
Figure 17 EPS XMI File of Block Definition Diagram for Hazard-Vulnerable Pairs (Source of Power and Fan)	34 -
Figure 18 Input and Output of Hazard Path Analyzer.....	35 -
Figure 19 SysML Requirement Diagram of reconfigured EPS System	36 -
Figure 20 Model of the electrical power system redesigned architecture	37 -

Figure 21 Three different alternative architectures that are generated by the graph grammar technique.
..... - 39 -

List of Tables

Table 1 Benefits of Model-Based System Engineering - 7 -

Table 2 SysML Extension to UML - 8 -

Table 3 Steps in FHA [52] - 21 -

Table 4 Example of FMEA Output [52]..... - 22 -

Table 5 HIERARCHICAL HAZARD TYPE [12]..... - 27 -

Table 6 EPS HAZARD ONTOLOGY-INTERNAL INTERACTION: ELECTRICAL..... - 32 -

Terms and Definitions

EPS: Electrical power system

OMG: Object Management Group

SysML: System Modeling Language

STAMP: Systems-Theoretic Accident Modeling and Processes

ESA: European Space Agency

NASA: National Aeronautics and Space Administration

SOHO: Solar and Heliocentric Observatory

HAZOP: Hazard and Operability Studies

XMI: XML Metadata Interchange

INCOSE: International Council of Systems Engineering

MBSE: Model-Based System Engineering

UML: Unified Modeling Language

DSL: Domain Specific Language

MOF: Meta Object Facility

BDD: Block Definition Diagram

SOS: System Of Systems

MARTE: Modeling and Analysis of Real Time and Embedded Systems

fUML: foundational UML

SPEEDS: Speculative and Exploratory Design in Systems Engineering

HRC: Heterogeneous Rich-Component (HRC)

FHA: Functional Hazard Analysis

FFA: Functional Failure Analysis

FMEA: Failure Mode and Effects Analysis

IMOFIS: Ingénierie des MOdèles de FonctIons Sécuritaires

Chapter 1

Introduction

1.1 Background

For some years the rapid development of technology has led to the creation of systems and projects of increasing size and complexity, particularly in avionics and automotive industry. The creation of ever-complex systems is made possible by integrating increasingly complex subsystems, i.e. "systems of systems". Therefore, the system integration process has become more demanding due to the potential undesired interactions between subsystems and risks of components failure due to environmental factors. These types of influences are difficult to pinpoint during the design process, yet are often cause major malfunctions or failures throughout the system.

One example of such an accident includes failure in landing of the Mars Polar Lander [1], also known as Mars Surveyor '98 Lander, which was attributed to the noise created during the deployment of the landing gears. These signal noises were generated correctly and it was expected by hardware system. However, the Lander's controller software shuts the engines down precipitately as soon as the spurious signals are created, assuming landing is occurred. The result of this unwanted interaction caused the spacecraft to crash, even though the software and the landing legs were functioning as expected. The crash occurred as a result of designers not accounting for all interactions between the control software and the leg deployment component. Therefore, as Keating et al. [2] describes, conventional system engineering methodologies are not capable of addressing the ambiguity and uncertainty that naturally exists in complex systems. In addition, in [2] the authors propose that a more effective design procedure for complex systems should incorporate both the unpredictable interaction between subs-systems and the effect of environmental factors. In the same way, the Department of Defense Acquisition Guidebook [3] recognizes that the designers of individual components and subs-systems must be able to identify the impact of other systems on their design architecture. Such information can be applied to ensure the designed sub-system can deliver its functionality while at the same time mitigating or eliminating a variety of integration issues. Fig. 1 illustrates the major hardware components of the Mars Polar Lander Spacecraft.

Figure 1 Annotated diagram of the Mars Polar Lander spacecraft

1.2 Goals

This thesis proposes a model-based hazard analysis methodology for early identification of undesirable interactions between subsystems and components failure due to environmental factors within a highly integrated safety critical system. In order to help with this goal, the fundamentals of hazard-vulnerability pairs and propagation path identification were presented by Malin and Fleming in [4]. The proposed scheme in this thesis is applied to a satellite electrical power system (EPS) using the System Modeling Language (SysML) and XMISearch tool for scripting hazardous scenarios. The choice of SysML is motivated by its support for reason for specification, analysis, design, and verification of complex systems [5].

1.3 Method

Since the application of safety engineering methods for undesirable interactions between components in complex systems is a new area, small numbers of published research exists. Most of the literature studies for this thesis have been focused on the characteristics of the complex systems, existing models of the complex system development process, existing research such as Hazard-vulnerability pair, FHA/FFA, FMEA, HAZOP and STAMP techniques for hazard analysis.

SysML has been studied using the OMG user guide. MagicDraw is used as a system modeling tool. MagicDraw user guide is studied specially for requirement diagram and block definition diagram.

1.4 Thesis Outline

This thesis is structured as follows. Section II presents the background and related research on safety analysis of undesirable interactions and vulnerabilities in complex systems. Section

III provides an overview of the proposed model-based hazard process. Section IV specifies the hazard types to be considered for construction of hazard ontology. Section V, outlines the application of the proposed methodology in analyzing the safety issues for Satellite Electrical Power System Design. Section VI illustrates the application of proposed methodology for hazard analysis and elimination of unsafe design from the space of feasible conceptual designs of a complex system. Conclusion and future work are presented in Section VII.

Chapter 2

Background

2.1 Modeling and Simulation

Modeling and simulation has become one of the most powerful methods available for designing large complex systems, since it allows designers to examine whether design specifications are met by using prototypes instead of physical experiments. The use of modeling and simulation shortens the design cycle, provides designers with more inclusive exploration of design alternatives, and reduces the cost of design.

Design of a complex system involves modeling of individual subsystems often from different disciplines such as electrical, mechanical, etc. Therefore, one of the basic requirements of the modeling and simulation technique is to be able to sufficiently model multi-disciplinary and non-linear phenomenon occurs in the design.

The second requirement for an effective modeling and simulation technique is the ability to be easy to create and reuse. Design and development of high-fidelity simulation models of a complex system can become a convoluted and tedious task for the designer. Therefore, it is necessary to for the elected modeling and simulation methodology to fulfill the above requirements.

2.2 Modeling Paradigms and Languages

There are numerous general-purpose modeling and simulation paradigms and languages that have been developed. For the purpose of this thesis I am using object-oriented modeling paradigm as it fulfills the essential requirements mentioned in the previous section.

2.2.1 Object-Oriented Modeling

The same methodology that is applied to object-oriented programming is applied for system modeling, in order to create simplified and maintainable models.

Information hiding and encapsulation is one of the most important principles in object-oriented programming. Encapsulation enforces an object to be accessible only by its public interface, which decouples the object from underlying implementation. In object-oriented modeling same principle is applied to modeling the system components. There is a clear distinction between physical interactions of a component model with its operating environment which is called *interface* and its internal behaviour that is called *implementation* [26, 27].

Inheritance is the second important principle of object-oriented programming. In this concept inheritance means that a new class can be created from an existing class (parent class). The newly defined class contains the attributes and methods of the base class. In addition, new attributes and new methods can be defined for the subclass which then can be applied to the inherited attributes and methods. By using the inheritance mechanism, highly specialized class hierarchies can be created. In the same way, in the system modeling, a component model derives from a parent model and inherits parent model's interface and equations. In addition, the child component model is extendable as needed with additional physical interaction or ports in the interface, or even additional equations in the implantation of the model [28].

Object-oriented modeling allows hierarchical organization of system component models, which in turn results in easier reuse and maintenance of the system model. Several object-oriented modeling and simulation languages have been developed [28, 29]. One of these object-oriented languages is SysML which is based on UML and supports and encourages an object-oriented view of the system.

2.3 Model-Based System Engineering (MBSE)

System engineering is defined as a multi-disciplinary process to provide a system solution to ensure that the customer's needs are satisfied. System engineering plays an important role in addressing and managing the complexity of the engineering projects. For the more efficient practice of system engineering a transformation from document-based approach to model-based approach is necessary. Model-based approach improves the specification, design accuracy, and design integration of a complex system. As a result, the design quality and reuse significantly improves while development risk reduces. The main goal of MBSE is to provide a quality system model and use this model to verify and design the system.

Table 1 Benefits of Model-Based System Engineering

Benefit	Description
Communication	MBSE improve communications between the developers of a complex system and stakeholders
Manage Complexity	MBSE enables designers to manage system complexity by allowing a system model to be viewed from several perspectives, and to easier to analyze the impact of changes
Design Quality	MBSE results in higher design quality by providing quality and adequate model of the system that can be analyzed for consistency, correctness, and completeness.

Reuse and maintainability	MBSE provide the ability to better capture and reuse of the system information by collecting information in a more organized manner and decoupling subsystem information which is based on the abstraction mechanisms derived from model-based approach. Short development time cycle and decrease in maintenance cost is a result of this benefit.
---------------------------	---

An essential issue in modeling and simulation of a complex system is to verify that the model adequately reflects the system it is representing [32]. Since most of the modeling and simulation languages are formalized, there is no corresponding formal description of the system under consideration. As a result of this weakness, most of the verification process needs to be designed and developed in an ad hoc manner. However, the use of SysML model enables designers to complete the constraints and parametric analysis of the system model by analyzing and simulating component models.

2.3.1 System Modeling Language (SysML)

SysML is an extension of Unified Modeling Language (UML) 2.0 developed by the Object Management Group (OMG) in cooperation with the International Council of Systems Engineering (INCOSE) [31]. As depicted in the Venn diagram in Fig. 1, SysML borrows a subset of the UML language which is specifically designed for software system and adds extension to meet the requirements of a general purposed language for system engineering. SysML is a general-purpose modeling language for constructing models of complex, multi-disciplinary, and large-scale systems. SysML enable the designers of the complex system to model the system requirement, structure, behaviour, and parametric values for a more vigorous description of a system under consideration.

Each model element in one SysML diagram can be related to model elements on other diagrams allowing the designers of the system to construct an integrated model. The aim of diagrams in SysML language is to capture and organize information in a model repository to assist with specification, design, analysis, and verification of the system under construction. In addition, SysML has the capability of allowing the model repository information to be imported and exported via XMI standard.

Table 2 SysML Extension to UML

Element	Description
Requirements	Text-based requirements are transformed into the graphical view, therefore their relationships to each other and to other models are easier to understand

Blocks	It is extended to capture the structural information of system components and subsystems and their relationship with use of ports
Activities	Extensions to UML activities to support continuous behaviour of the system engineering requirements
Constraints	Allowing the creation of parametric models
Ports and Flows	extensions to the UML structural model to support flow of information, matter, and energy between system elements

Figure 2 Relationship between SysML and UML [31]

2.4 Metamodel

Library of notations and rules of a specific modeling language is called metamodel. Therefore, model confirms to the metamodel if the model follows the notions and rules used in metamodel. As illustrated in Fig. 3, in section (a) the graphical description of the metamodel of a chosen modeling language is represented. There are **interface** elements which are represented by a rectangle and correspond to an object. **Arrows** are a representation of relationships between the objects.

The designers of the complex system are not required to be familiar with the metamodel definition; however they use the metamodel definition via the use of graphical tools to create their specification of an interface and number of associated operations. As it is illustrated in Fig. 3 (b) and Fig. 3 (c) the VehicleControllerInterface is defined and associates with two operations 1- getSpeed() and 2- ControlEngine(). The getSpeed() operations returns and

integer type and the ControlEngine() operation accepts a “Torque” as typed parameter. The VehicleControllerInterface model together with the two associated operations conforms to the metamodel depicted in Fig. 3 (b) which prescribes that any NamedElement from this particular domain has an interface that can contain operations. Each operation associated with the interface can have typed parameters and is capable of returning a typed value.

Figure 3 Language definition and use for a vehicle controller interface specification in model-based engineering form: metamodel definition (a), user-defined model (b), and model repository (c) [33].

2.4.1 Heavyweight vs. lightweight design

In Fig. 4, an outline of the heavyweight and lightweight design for different languages and their effect on underlying tools is depicted. Heavyweight design demonstrated on the left-hand side of the Fig. 4, enables designers of the system to construct a new domain specific language (DSL) for modeling and simulation of the domain of interest, and specific metamodel is created for the system under consideration. As a result, the created DSL is

customized for the problem on hand and results in optimal solution. Since in the heavyweight design every discipline is required to have its own specific modeling language and metamodel, integration and verification of multi-disciplinary sub-systems of a complex system is very challenging.

On the other hand, the lightweight design depicted in Fig. 4b, is based on extension of an existing metamodel independent of system's disciplinary character. In this concept, the metamodel is consisting of modeling notions and rules that are suited for more general-purpose modeling languages. One example of general-purpose modeling language is UML.

Figure 4 Uses of metamodeling techniques related to concepts of the real-time domain: heavyweight approaches (a), lightweight approaches with reused elements (b), and optional connection to domain-specific modeling add-ins (c) [33].

2.4.2 Profile and Stereotype

In the UML language, the technique of lightweight extension is called profile. Each extension of an element from the UML metamodel is called stereotype. Each stereotype can have properties or constraints that are specific to the domain of the system under consideration. Stereotypes then further customized in the modeling language level as annotations so different modeling tools are able to access the information that is saved in the profile. The use of profile and stereotype allows the metamodel to be reused and customized for multi-disciplinary systems.

One of the disadvantages of using profile and stereotype is the requirement of selecting the most appropriate elements in the metamodel that must be extended. This is required for the designers of the system to have extensive knowledge of the metamodel classes. For example, knowledge about the differences between metaclasses and their interfaces, operations, and applicability is required in order to select the most appropriate metaclass and further extend it in the design process.

2.4.3 SysML Profile

SysML provides a wide range of support for system engineering architectures and supports modeling the system interaction with its environment. There are two main aspects in SysML modeling levels which are integrated together: 1- creating relationship between requirement documentation and model of the system. 2- The ability to describe the system architecture in detail with precise modeling of discrete and continue interactions. The profiling mechanism is consistent with the OMG Meta Object Facility (MOF)

2.4.4 SysML Diagrams

- SysML Block Definition Diagrams (BDD)
- SysML Internal Block Diagrams (IBD)
- SysML Package Diagrams
- SysML Parametric Diagrams
- SysML Requirement Diagrams
- SysML Activity Diagrams
- SysML Use Case Diagrams

2.4.5 SysML Block Definition Diagram

Block Definition Diagram used in SysML modeling to define the properties of a block and any relationships between blocks such as associations, generalizations, and dependencies, in terms of properties, operations, and relationships. Block Definition Diagrams are an expansion of UML class diagrams plus restrictions and extensions as defined by SysML model.

2.4.6 SysML BDD Metamodel and Elements supported by MagicDraw

As presented in Fig. 5 the Blocks in the metamodel of BDD diagram provide a general-purpose capability to define the structural design of a system, and correspond to the system hierarchy in terms of systems and subsystems. Blocks define the connectivity relationships

within / between a system and its subsystems. For more detail description of each element in the metaclass please refer to MagicDraw SysML plugin user guide.

Figure 5 Block Element and MagicDraw SysML Block Subtypes Metamodel [34].

SysML blocks are the best modeling elements to model multi-disciplinary systems and it is especially effective during system specification and design. It is effective because blocks are not only able to model logical or physical decomposition of a system; they also enable designers to define specification of software, hardware, or human elements.

Another advantage of using blocks during the design of the system under consideration is the ability to include both structural and behavioural features, such as properties and operations that represent the state of the system and behaviour that the system may display. In addition block allows for construction of the structure of the connectors between different components and subsystems. Blocks are reusable form of description that can be applied throughout the construction of system modeling if necessary.

2.4.7 SysML Requirement Diagram

SysML requirement diagram enables the transformation of test-based requirements into the graphical modeling of the requirement which can be related to other modeling elements. As a

result, the requirements are traceable throughout different phases of the design. Requirement diagrams display requirements, packages, other classifiers, test cases, rationales, and relationships.

There are different types of relationships available for Requirement diagrams are containments, derive requirement and requirement dependencies ('Copy', 'Refine', 'Satisfy', 'Trace', and 'Verify').

A Requirement defines ability or a condition that must be satisfied. Requirements are used to establish a contract between the user and designers and developers of the system. A requirement can also appear on other SysML diagrams to show its relationship to other modeling elements.

Fig. 6 demonstrates the SysML requirement diagram metamodel elements that are supported by MagicDraw. As illustrated in Fig. 6 the extended requirement is an expansion of requirement class and contains additional properties that are essential for requirement management. The extended requirement can be of extended, functional, interface, performance, physical, business, usability requirements or design constraints.

Figure 6 Requirement Metamodel [34]

2.4.8 SysML Cross Connecting Model Element

There are four cross connecting principles in SysML, which allows different diagrams to be connected to one another. **Allocate** is the first principle that enables the designers of the system to connect a behaviour to a block in a structure diagram. Since the behaviour diagram in SysML is similar as activity diagram and an activity or state can be allocated to the component in structure diagram. The requirement diagram can be connected to the structure diagram by second cross connecting element **satisfy**. A requirement can be satisfied by a component or subsystem. **Value binding** is third cross connecting element which binds a value in the structural diagram with a formula that is calculated in the behaviour diagram. And lastly, a formula in a parametric diagram can **verify** a requirement in the requirement diagram. With these four cross connecting elements requirements can be satisfied and verified and can be traced through the structural diagrams.

Figure 7 Cross Connecting Model Elements [31]

2.5 SysML MagicDraw modeling Environment

MagicDraw is a commercial modeling tool, developed by No Magic Inc.[34]. MagicDraw UML 17.0 is the latest version of the product by the time of this work. This tool supports UML 2 and provides a plugin for SysML.

Chapter 3

Related Work

3.1 Technical Background and Related work

It is widely recognized that risk-free design of highly integrated systems is a challenging task. As observed in [6], the sub-systems in complex systems are required to interact directly or indirectly with many other systems which results in countless interactions. For example, a typical weapon system is connected to other systems, vehicles, soldiers, and satellites with infinite interactions [6]. Leveson, in [7] argues that traditional hazard analysis techniques are based on assumptions that are only valid for specific domains such as simple mechanical and automotive systems. The hazard analysis in such designs is based on the main assumption that an accident in the system is the result of component failures. However, in the complex systems more often than not an accident is a result of erroneous interaction between components despite the fact that components are in nominal mode. Referring to the Mars Polar Lander Spacecraft in section 1.1, both the controller software and landing gears are reported as being in nominal mode and the erroneous interaction is recognized as a signal noises created by the hardware component which causes the spacecraft accident.

3.2 Design of Complex Systems

3.2.1 Characteristics of a Complex System

Complex system definition is a controversial issue precisely because different interpretations result in substantial variation in the conclusions reached regarding what does and does not constitute a complex system. For example, the US Department of Defense Acquisition Guidebook [3] defines a combat aircraft as a complex system, while the authors in [35] clearly reject this statement.

Author in [36] defines a complex system to be a large-scale distributed systems consisting of sub-systems which themselves are complex in design.

Another definition is provided by the University of Newcastle [37] which defines a complex system as a system of systems that contains hardware and software components with independent management and ownership. This definition rejects the Ktove's [36] definition of a complex system as a large-scale distributed system, since each sub-system has its own management and ownership and they can operate even though disassembled from the system under consideration.

A more complete definition that is used in this thesis is given by Maier in [38] which also reject's Ktove's description of a complex system as a large monolithic system. In [38]

complex system is defined as collection of components and subsystems that are independent with evolutionary nature and emergent behaviours.

These definitions of a complex system with the emergent behaviour of the components and their evolutionary nature is recognized as cause of accidents in most complex systems, the purpose of this thesis is to be able to identify these emergent hazards and prevent their occurrences during conceptual design of the system.

3.2.2 Complex System Engineering Life Cycle

Figure 8 Double V Lifecycle Model for Complex System Engineering [39]

Fig. 8 illustrates the double V lifecycle model used for design and development of complex systems or system of systems (SOS). Close collaboration between system developers and engineering controlling the SOS is required. The system developers are required to provide the required information for SOS engineers so they can evaluate and engineer how the whole system operates. On the other hand, the SOS engineers are required to provide the needed information to system developers so they are aware how to evaluate and develop their system as part of the complex system under consideration.

Keating et al. [2] recognize that it is not practical to presume that the above system engineering is appropriate for design and development of all complex systems and result in success of the project. As a result, the author in [2] suggest to concentrate on engineering methodology, as described in [40] not only one method but a set of principles that is appropriate and practical for any particular design and development of a complex system.

Therefore the metamodeling approach is recommended so the design and development of a complex system is based on the requirement of each specific system under consideration.

3.2.3 Safety analysis in Complex System

Safety requirements are very important part of the design and development of a complex system. Most of the time, safety requirements indicate the type of system architecture that a complex design should be based on; therefore it is very important to integrate the safety requirement in system development as early as possible. However, traditionally the safety engineers and system development team act independently, therefore the integration process becomes complicated and strenuous. As a result two different methodologies are available:

- 1- Safety team and development team to develop their own models and tools and then integrate the results. However, there is an issue with integrating the tools themselves. There are advantages and disadvantages using this approach. The advantages are that existing tools are used and customized for specific application. The disadvantage is that the safety team provide their result after development team has completed the specification phase, the late integration of safety causes architectural changes and redesign of those parts of the system with safety issues. This late integration result in long redesign process which reduces productivity and in some cases correctness.
- 2- The second approach is based on lightweight design described in section 2.4.1 and 2.4.2.

3.3 Imofis industrial European project

The Ingénierie des MOdèles de FonctIons Sécuritaires (Imofis) industrial European project started in mid-2008 is one of the best cases for defining a conceptual data model for safety-critical applications [41]. The conceptual data model is based on the ontology that is designed and created by safety engineering teams and contains hazard, accident, and a safety barrier. The second step in the process is for the development team to create a profile as described in section 2.4.2. The newly created safety profile then is integrated with other existing profiles of the complex system under consideration. Now that the profiles are created, at the modeling level, the designers are able to specify the model elements and define their properties using different modeling tools. In addition, new conception of a profile [42] is developed which contains a group of related languages. In order to customize a DSL and reduce the potential semantic and syntactical conflict between different profiles only a subset of profiles is used in the Imofis project. This approach is used in several European projects [43-45].

Since the Imofis approach requires the integration of different profiles, there is a need for formal execution description to be developed to be able to implement different form of profiles. For example, the designers are required to use SysML profile for requirement

specification together with Modeling and Analysis of Real Time Embedded (Marte)'s profile for its annotations and concepts for timing analysis. Even though, each profile has its own syntax and rules, there is a possibility of overlap and causing consistency issues when profiles are integrated together in a model of the system under consideration.

The issue with integrating different profiles not only lies on structural point, but on executing and implementing behavioural concept into profile description. One specific formalism standard is required to permit and ease the integration process of multiple profiles. One such formalism for specifying the encapsulated semantics of multiple profiles is the concept introduced by OMG which is called foundational UML (fUML) [46].

As the complexity of modern systems is increasing, the requirement for integration of various profiles is rising [41, 43-45, 47-50].

3.4 New Generation of End-to-End Tools for Safety-critical System Design

Based on the new Speculative and Exploratory Design in System (Speeds) project [51], the new definition of a semantic-based modeling approach is introduced that enables the designers of a complex system to integrate new and existing tools to create a complex system by composing heterogeneous subsystems. In the Speeds project a new definition for heterogeneous rich-component (HRC) model which characterizes the functional and architectural abstractions such as safety, timing, and other non-functional performances.

Figure 9 Tool integration via the Speculative and Exploratory Design in System Engineering (Speeds) [39]

3.5 Functional Hazard Analysis (FHA)/ Functional Failure Analysis (FFA)

Functional Hazard Analysis (FHA) is a technique to identify hazards that exists in the functional description of a system. FHA analyses the system function description to identify any failure that might cause unsafe behaviour during the operation of the system under consideration. The FHA process starts with analyzing the description of external functions and all the potential failures that might be possible for each function is identified and documented. Next, effects of each failure on the rest of the system is studied and examined. The result of these analysis leads to the improvement of the design of the system under consideration. Table 3 describes the steps required for FHA analysis. FHA analysis sometime is knows as Functional Failure Analysis (FFA). In FFA analysis explicit type of failures are considered for analysis:

- 1- Function is not available when it is required by the system.
- 2- Function is provided when it is not required by the system.
- 3- Function provides incorrect result.

Step	Description
1	Identify functions.
2	For each function identified, suggest failure modes, using the three failure types as prompts.
3	For each failure mode, consider the effects (at different levels, e.g. function / subsystem / system, if necessary).
4	Identify and record any actions necessary to improve the design.

Table 3 Steps in FHA [52]

The author in [52] suggests that FFA should be applied early in the design of the complex system to be able to identify the hazard problems and improve the safety related requirements of the design. In the concept of complex systems FHA or FFA can be applied on functional requirements before the configuration of the complex system is decided, therefore the result FHA is very valuable.

However, the disadvantage of FFA is that only identifies one single failure at a time and it is unable to identify multiple failures that contribute to a single hazard. Another, problematic issue with FFA is that many subsystems may function correctly in isolation, however in the context of a whole system it might cause a hazardous interaction which is not recognizable by FFA. This is especially true for the dynamic reconfigurable characteristics of a complex system.

3.6 Failure Mode and Effects Analysis (FMEA)

FMEA is based on divide and conquer methodology which divides the complex system under consideration into subsystems and further subsystems into components and based on previous experience of known failure modes of those components the hazardous impact at system level is analyzed. FMEA is a collection of related techniques [52,53].

FMEA follow a tabular implementation of listing all the components and their known failure modes followed by their failure effects on the system under consideration. This tabular format can be customized for different domains.

Component	Failure Mode	Subsystem Effects	Vehicle Effects	Comments
Variable Reluctance Sensor	No signal	Vehicle speed will always be calculated as zero	1) No speed indication 2) Mileometer not incremented 3) Electronic gearbox control may select too low a gear, possibly resulting in wheel lockup or transmission damage	Effect (3) requires simultaneous failure of engine load calculation and mechanical interlocks on gearbox
Variable Reluctance Sensor	Noisy (too many edges)	...	...	...

Table 4 Example of FMEA Output [52]

The difference between FFA and FMEA is that FMEA is applied later in the product lifecycle; the reason for this late implementation is the requirement for identifying all the system's component and their failure modes. Table 4 depicts an example of FMEA output.

However FMEA has the same disadvantages as FFA that it is very difficult to predict the impact of failure mode on a complex system under consideration. The FMEA analysis only considers the effect of one single failure at a time while in a complex system multiple configurations requires multiple FMEA analysis.

3.7 Systems-Theoretic Accident Modeling and Processes (STAMP)

As a result, [7] suggests that safety analysis for complex system needs to be based on a new model called Systems-Theoretic Accident Modeling and Processes (STAMP). In systematic models such as STAMP, accidents result from several causal factors that occur unexpectedly in a specific time and space [8]. Therefore, the system under consideration is not viewed as a static entity but as a dynamic process that is constantly adapting to achieve its goals and reacting to internal and environmental changes. Consequently, hazards are viewed as complex interactions between system components and their intended environments. The STAMP models are designed based on safety-related constraints and hazards are identified by violation of these safety constraints.

There are many benefits in using STAMP models as the basis for hazard analysis of a complex system. However, Johnson et al. [9] state that the STAMP approach has two fundamental weaknesses: the lack of methodological guideline in implementing the constraint flow taxonomy and the construction of control models in a complex system is complicated. In addition, [9] presents two independent studies of implementing STAMP hazard analysis techniques on the mission interruption of the joint European Space Agency (ESA) and National Aeronautics and Space Administration (NASA) Solar and Heliospheric Observatory (SOHO). The hazard analysis from each study resulted in significantly different conclusion regarding the cause of failure in the system under study.

3.8 Hazard and Operability Studies (HAZOP)

Another technique for safety analysis is Hazard and Operability Studies (HAZOP) [10] that is based on modeling the interaction flow between components and recognizing a hazard if components deviate from intended operation of design. A set of guidewords are provided to help with identification of such deviations. However, from the context of safety analysis based on interaction between components and their intended environments, HAZOP is unable to produce repeatable hazard analysis of the same accident. The reason for this weakness lies in the highly dynamic and unpredictable nature of interactions between different subsystems and their operational environment. Moreover, depending on the expertise and skills of the safety engineers the deviations can be identified differently. Fig. 10 illustrates required steps for HAZOP process.

Figure 10 The HAZOP Process [54]

As discussed in this section existing works on hazard analysis lack in accurate identification of potential safety issues caused by unexpected environmental factors and subsystem interactions. In addition, the algorithms in the literature do not attempt to identify the hazards within the system in the early stages of the conceptual design and analysis process.

The proposed model-based hazard methodology in this thesis improves the safety analysis process by emphasizing the importance of precision in hazard definitions and integration at the early stages of system design by using hazard ontology and requirement definition diagrams. This methodology also provides SysML notations for describing and reasoning about certain aspects of hazards.

Chapter 4

Towards a Model-Based Hazard Analysis Framework

An effective safety analysis methodology must be conducted as early as possible in the project, ideally at conceptual design level. The importance of this requirement stems from the fact that risks can be identified early and appropriate mitigation strategies can be designed early in the project to avoid potential costly remedial work later in the project. Therefore, the proposed methodology in the thesis is based on evaluation of the design architecture, identifying the hazards, and changing the design to mitigate the identified safety issues. The process is an iterative process and cycle is repeated until no hazard is exhibited. An overview of the process is shown in Fig. 11.

Figure 11 Overview of the Model-Based Hazard Analysis Process

Development of detail Hazard Ontology is required to assist the designers with risk and vulnerability information of all the potential components that is to be used during the design of the system under consideration. Second step in the proposed methodology is to transform the text-based requirement specification document into SysML requirement diagram. As

depicted in step 3 of Fig. 11, the block definition diagram is created by using the hazard information from the hazard ontology and the requirement diagram. The designer of the system includes the hazard and vulnerability information while designing the component model. Next step in the proposed process is to use SysML capability to transform the created block definition diagram into XMI format so it can be searched for hazard-vulnerability pairs by the hazard path finder. The hazard path finder which is a java-based application uses a sophisticated search algorithm to search for the source of hazard and components that are vulnerable to that hazard and examines the path to prove that the hazard propagates through the components and connections. If the hazard path finder application finds a hazardous path, the application notifies the system designers. In order to eliminate or mitigate the discovered hazard change in the requirement is required. Changing the requirement necessitate the hazard analysis process to be repeated once again to prove that the selected design is risk free. Therefore steps 3 and 4 of Fig. 11 are repeated.

Chapter 5

Hazard Types and Identification

In recent years, the design and development of design repositories and ontology-based framework has become popular [11] to capture design information. A knowledge-based ontology is used to specify a structured information model for organizing design knowledge; however these ontologies lack providing hazard information. Designers need hazard type information to take into consideration their threats and effects on the system. Therefore, a hazard ontology is required for each problem design depending on the type of environment being considered and the hazard-vulnerability effect of environment on components and between subsystems. In this context, hazard is the potential source that causes harm and constitutes a deviation from the intended design or operation. An example of such a common source of unsuspected hazard is sources and propagation paths of stored energy in electrical, chemical or mechanical form. For the purpose of this work, [12] is used as a reference to create a general hazard ontology illustrating these types of hazard sources. Each category of energy sources outlined in Table 1, is required to be methodically traced from the perspective of the conceptual design components and across subsystem interfaces to locate the possible hazardous deviation.

Table 5 HIERARCHICAL HAZARD TYPE [12]

Cause of Hazardous Source	Energy	e.g.
Internal Interaction	Electrical	ac/dc flows stored electric energy electromagnetic radiation static charges/flows
	Mass/Gravity/Height	falls and drops falling objects falling hazardous materials
	Rotational Kinetic	machinery fans
	Pressure/Volume	container ruptures and explosion vacuum creation liquids spill/flood vapor expansion
	Linear Kinetic	projectiles rams, moving parts shear press

		vehicular movements prints pre-stressed members
	Chemical Reactions	corrosion oxidation combustion polymerization decomposition toxic asphyxiant, anesthetic
	Thermal	heat, cold alternate heat/cold radiation/conduction convection, sublimation
	Etiologic	viral bacterial fungal
	Ionizing radiation	gamma alpha beta
	Noise and Vibration	
External Interaction	Explosion Projectiles Noise Vibration Fire	
	Atmospheric	wind rain snow lightning hail acid rain

Creating ontology for each design problem requires an identification of the source of hazard in almost any circumstance and that is only possible if detail knowledge of the system and its operation is known. The effectiveness of the ontology depends on the broad hazard coverage of software, hardware, environmental, and human systems which requires detail knowledge of the system and its intended environment.

Chapter 6

Model-Based Automated Safety Analysis

The proposed design and safety analysis process for early identification of the unexpected hazard sources and propagation paths is based on conceptual design information. Conceptual design is a preliminary stage of design that describes design requirements comprehensively and abstractly while identifying the optimized principle and solution to be used for the design. However, most existing functional models are difficult to translate to functional architecture for early hazard analysis during design of a complex system. Transformation is challenging because the construction of a complex system with integrated hazard information requires designer to map the design requirements, system components' hazard-vulnerability properties, and hazard ontology models using unified modeling environment. Therefore, a modeling language to provide simple but powerful constructs for modeling a wide range of engineering analysis is required. The System Modeling Language (SysML) [5] as a graphical modeling language for systems engineering applications is particularly effective in specifying system safety requirements, system structure, functional behavior during specification and design of system engineering [13]-[16].

Figure 12 SysML diagram taxonomy [5]

For the purpose of hazard analysis SysML is used to capture and integrate design and safety information during conceptual design, top-down from system level, with focus on system functions and structures. Therefore, as illustrated in Fig. 12 from the taxonomy of Sysml diagrams, two main categories of requirement and structure diagrams are used to provide ontologies and component-connection models for identifying and investigating system functions, treats, and safeguards. **Requirement Diagram** enables designers to construct a system and safety requirements model from a text-based requirement specification document and identify the relationship between requirements. In addition, requirement diagram allows composition of requirements and customized selection of types of components, functions, and treats from domain specific applications such as space systems, and automotive. **Block Definition Diagram** as a sub-category of structure diagram is used to connect components and define their properties, operations, relationships, hazards, vulnerabilities, and transmitted entities. The block definition diagram is derived from requirement diagram which in turn is derived from requirement specification document. In the block definition diagram, the default hazard, vulnerability, and transmitted entities are associated with each component by the use of hazard ontology. As a result, block definition diagram provides a structure for matching hazard and vulnerability type with the type of component used in the system.

6.1 The NASA Adapt System

Assessment and comparison of different diagnostics technologies can be difficult. To facilitate this task the researchers at NASA Ames Research Center have developed the Advanced Diagnostics and Prognostics testbed called ADAPT [19]. The testbed acts as a common platform where different diagnostics tools and technologies, called test articles, can compete against one another on equal conditions. To achieve this, ADAPT consists of a controlled and monitored environment where faults are injected into the system in a controlled manner and the performance of the test article is carefully monitored. The hardware in the testbed is an electrical power system (EPS) from a space exploration vehicle. The testbed is located in a laboratory at the NASA Ames Research Center. The ADAPT system consists of three major modules: a power generation unit, a power storage unit and a power distribution unit.

The testbed is controlled by a number of relays and monitored by a large set of sensors. Consequently, it is possible to detect an injected fault and recover from it if the correct action is taken. To facilitate the execution of the experiments performed with the testbed, three operating roles have been defined by [19]: user, antagonist and observer. The user simulates an actual crewmember or pilot who operates and maintains the EPS with the help of a vehicle health management application. The antagonist injects faults into the system, either manually by physically acting on the system, or remotely by spoofing sensor values through a computer connected to the system. The malicious actions of the antagonist are not known to the user who is responsible of choosing a suitable recovery action. The observer logs all data in the experiment and monitors how the user responds to the faults injected by the antagonist. The observer measures the effectiveness of the test article using the collected data and also acts as a safety officer for the experiment and can issue an emergency stop. Fig. 13 illustrates the ADAPT system at NASA Ames Research Center.

Figure 13 ADAPT System at NASA Ames Research Center

6.2 Satellite Electrical Power System

An electrical power system is designed to deliver power to select loads, which in an aerospace vehicle would include subsystems such as the avionics, propulsion, life support, and thermal management systems. The EPS is required to provide basic functionality common to many aerospace applications: power storage, power distribution, and operation of loads [17]. Fig. 14 displays the existing design of the EPS, containing a source of power connected through a series of relays to an inverter, and several loads consisting of a large fan, a DC resistor and an AC resistor. A series of four AC or DC voltage sensors and three current transmitters measure the voltage and current in different probing points of the circuit.

Figure 14 Model of the existing electrical power system design architecture

6.3 Identifying Hazard-Vulnerable Pairs

First step in identifying the hazard scenario is to construct a hazard ontology for the EPS design problem. Table 2 illustrates the developed hazard ontology and libraries of type of components, hazards, vulnerabilities, and transmitted entities for hazard analysis of EPS system.

Table 6 EPS HAZARD ONTOLOGY-INTERNAL INTERACTION: ELECTRICAL

Component	Source of Hazard	Vulnerability	Hazard Transmission
Source of Power	Over Current Spikes (OCS)		
AC Resistor		OCS	
DC Resistor		OCS	
FAN		OCS	
All Components			Transmit OCS
All Connections			Transmit OCS

Next Step is to create SysML requirement diagram for the specified satellite EPS system. Starting with EPS design requirements one example of the specified requirement is that ***AFGS-87219A: A battery relay shall be installed in each battery circuit to enable the flight crew to isolate the battery from the rest of the electric subsystem.*** Fig. 15 depicts the corresponding SysML requirement diagram derived from this requirement.

Figure 15 An Example of Text-Based Requirement Transformation to SysML Requirement Diagram

EPS requirement diagram plays an important role during the system modeling. It illustrates how the requirements are satisfied by the system elements in the block-definition diagram. EPS block definition diagram describes the internal system structure of EPS design using a block as its basic unit. Each block in the block definition diagram defines a collection of features such as properties, operations, relationships, hazards, vulnerabilities, and transmitted risks. With each operational mode a function actions and side effects action are defined. For example, in the EPS test-bed, the source of power component may have the operational modes ON and OFF. The ON mode has the functional action of generating power and a side effect action of generating over current spikes. As illustrated in Fig. 16, three hazard properties are defined for source of power block. On the other hand, the AC resistor, fan, and DC resistor are vulnerable to the generated over current spikes. This type of information is provided by the hazard ontology database constructed previously. As depicted in Fig. 16, Block definition diagram models the causal relationship between the source of hazard in this case the source of power, and impacted targets that are AC Resistor, DC Resistor, and the Fan.

Figure 16 SysML Block Definition Diagram of Simple EPS System

6.4 Hazard Path Analysis

Although, the analysis of the constructed block definition diagram identifies the source of hazards and susceptible components in design of the system, it does not verify safety violation. Since the threats introduced to the system by a hazard source may propagate from the source of hazard to the vulnerable components via components and connections that might mitigate the effect of this threat. In the block definition diagram all the components and connections are associated with the *hazard carrier type* to guide the path traversal procedure. Therefore, the path analyzer procedure is proposed to compare the hazard type with the specification of each component. If the component cannot mitigate the effect of the hazard, it is propagated to the next component or connection. While, if the component can eliminate the threat caused by the hazard, the proposed path analyzer deems that the specific hazard does not result in a safety violation. The proposed path analyzer is based on block definition diagram that is further transformed to a XML Metadata Interchange (XMI) file to enable quick and easy hazard path analysis through a java-based application called XMISearch.

```
<?xml version='1.0' encoding='UTF-8'>
<packageDefinition xmi:type='uml:PackageDefinition' xmi:id='_17_0_2_f4d037a_1310433890190_553579_11547' name='SourceofPower#10' visibility='public'>
  <packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310433890190_553579_11547' name='SourceofPower#10' visibility='public'>
    <ownedAttribute xmi:type='uml:Property' xmi:id='_16_9_39a017d_1313641212043_802391_13415' name='Vulnerability:Null' visibility='public'>
      <value xmi:type='uml:Literal' xmi:id='_16_9_39a017d_1313641212043_802391_13415' value='Null'>
      </value>
    </ownedAttribute>
    <ownedAttribute xmi:type='uml:Property' xmi:id='_17_0_2_f4d037a_1310434162129_729781_11663' name='Hazard: Over Current' visibility='public'>
      <value xmi:type='uml:Literal' xmi:id='_17_0_2_f4d037a_1310434162129_729781_11663' value='Over Current'>
      </value>
    </ownedAttribute>
    <ownedAttribute xmi:type='uml:Property' xmi:id='_16_9_39a017d_1313641248463_484328_13422' name='Carr:Electrical Current' visibility='public'>
      <value xmi:type='uml:Literal' xmi:id='_16_9_39a017d_1313641248463_484328_13422' value='Electrical Current'>
      </value>
    </ownedAttribute>
    <ownedOperation xmi:type='uml:Operation' xmi:id='_17_0_2_f4d037a_1310433936737_522046_11649' name='unnamed1' visibility='public'>
      <value xmi:type='uml:Literal' xmi:id='_17_0_2_f4d037a_1310433936737_522046_11649' value='unnamed1'>
      </value>
    </ownedOperation>
  </packagedElement>
  <packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310500827932_728300_14261' name='FANFAN416' visibility='public'>
    <ownedAttribute xmi:type='uml:Property' xmi:id='_17_0_2_f4d037a_1310500827932_728300_14261' name='Vulnerability Over Current' visibility='public'>
      <value xmi:type='uml:Literal' xmi:id='_17_0_2_f4d037a_1310500827932_728300_14261' value='Over Current'>
      </value>
    </ownedAttribute>
    <ownedAttribute xmi:type='uml:Property' xmi:id='_16_9_39a017d_1313686346319_567793_14115' name='Hazard:Null' visibility='public'>
      <value xmi:type='uml:Literal' xmi:id='_16_9_39a017d_1313686346319_567793_14115' value='Null'>
      </value>
    </ownedAttribute>
    <ownedAttribute xmi:type='uml:Property' xmi:id='_16_9_39a017d_1313686365645_869499_14122' name='Carr:Electrical Current' visibility='public'>
      <value xmi:type='uml:Literal' xmi:id='_16_9_39a017d_1313686365645_869499_14122' value='Electrical Current'>
      </value>
    </ownedAttribute>
  </packagedElement>
</packageDefinition>
```

Figure 17 EPS XMI File of Block Definition Diagram for Hazard-Vulnerable Pairs (Source of Power and Fan)

Fig. 17 illustrates the XMI transformation of "Source of Power" and "Fan" elements in the EPS block definition diagram. First step in the java-based hazard path analyzer is to search for hazardous components, in this case the source of power with hazard type of "over current". This thread maybe propagated in all direction through conducting wires. Second step in the process is to search for the potential vulnerable components that are susceptible to the identified hazard in previous step. The susceptibility of component is recognized by comparing the type of vulnerability of the component with the type of identified hazard. In the case of EPS system, simple depth-first search is used, however more sophisticated search algorithms are necessary for a complex system. As illustrated in Fig. 18, there are three hazard paths to be examined by the hazard path analyzer: From source of power to AC resistor, fan, and DC resistor. In order to analyze the path from source of power to the AC resistor, the path analyzer inspects the Current Sensor (IT240), Relay (EY244), Inverter (INV2), Current Sensor (IT267), Relay (E272), and including all the connections between identified components are checked for matching "carrier type" to make sure that hazard is traversed from source to potentially vulnerable component. In the case of EPS system all the connections and components are carrier of over current spikes. Therefore, the examined path is recognized as hazardous. Fig. 18 illustrates the input and output of hazard path analyzer.

Figure 18 Input and Output of Hazard Path Analyzer

6.5 Hazard Analysis Results in Design Mitigation

A possible resolution to the identified hazards identified in the previous section is the introduction of new requirement to the existing design requirement. ***MIL-STD 7080: The battery relay shall be controlled by a crew station battery switch. Any circuits which must remain connected to the battery switch OFF shall be connected directly to the battery though suitable fuses or circuit breakers.*** Taking to the account the newly added design requirement results in a new SysML requirement diagram, as depicted in Fig. 19.

Figure 19 SysML Requirement Diagram of reconfigured EPS System

As demonstrated in Fig. 20 the circuit breakers are added to the design at various points in the distribution network to prevent over currents from causing unintended damage to the system components. The circuit breakers can be commanded externally to be closed or open. In order to verify and validate safety aspects of the new architecture, the proposed model-based framework is reapplied to the new EPS design. The outcome of the hazard analysis identified one sources of hazard (Source of Power) and three vulnerable components (AC/DC resistors, and Fan), which resulted in three hazard paths to be probed by path analyzer for the possibility of hazard propagation. Path analyzer identified that circuit breaker is not a carrier of over current spikes and constrains hazard propagation in all three paths.

Chapter 7

DESIGN SPACE EXPLORATION

The second application of the proposed model-based hazard methodology is to prevent unsafe design architecture to be selected from the space of feasible conceptual designs. A design space exploration process helps designers to explore the vast space of potential design architects that meet configurational requirements for the system in a systematic fashion. With recent progress in engineering design research, graph grammar has emerged as promising technique to provide structured approach to the creation of complex engineering systems [55],[56].

In the case of EPS system, this generative technique takes user specified EPS loads as input and satisfies system-level configuration requirements to generate feasible EPS candidate architectures. For instance, if there were configurational requirements to provide reliable power to certain electrical loads, this approach generates many feasible designs with alternative configurations for power generation (redundant and dissimilar sources), distribution (e.g., redundant buses), and switching. The generative graph grammar gives an EPS architect the ability to systematically explore a large number of alternative EPS architectures that meet a given set of design constraints and objectives. From this pool of alternative architectures, it is the responsibility of the designer to verify the risk-free design of EPS. In order to make the safety analysis more effective the proposed model-based hazard methodology is used. For instance, Fig. 21 provides three different alternative architectures that are generated by the graph grammar technique.

A. Design Architecture #1

B. Design Architecture #2

C. Design Architecture #3

Figure 21 Three different alternative architectures that are generated by the graph grammar technique.

With the use of proposed model-based hazard methodology the safety verification of different design alternative is performed. Design alternative in Fig. 21.B is recognized as an unsafe design, because the two source of power are too close together and may cause a short circuit. A short circuit of the conductors may also generate excess heat or cause a fire, resulting in damage to electrical devices.

Chapter 8

Conclusions

8.1 Accomplishments

Conventional safety-analysis approaches are not adequate enough to predict and prevent types of system accidents, where the cause of accident is not the result of an individual element failure or human error. A key to a risk-free design of such complex systems is to ensure that not only the individual components and technologies but also their integrations are reliable and effective, resulting in turn in safe and reliable systems. In our approach, to ensure safety and reliability, hazard and vulnerability information of component and subsystem is incorporated into the design process of the system as early as possible. At the early stages of design, where the decision about selection of specific components and connections have not been made yet, system designers need hazard identification and propagation information that is the result of proposed model-based hazard methodology to make an early design decisions. The proposed methodology provides the ability to transform the requirement and hazard information so it is possible to explores system interactions and identify the hazard scenarios.

8.2 Future Work

In the case of EPS system only internal hazards has been analysed. However, environmental hazard can be modeled and evaluated using the same methodology. More complex system or sub-system with combination of hazards and vulnerabilities associated with each components could be evaluated. In the case of multiple hazards and vulnerabilities, stereotypes can be used to extend the hazard and vulnerability elements in the model.

In addition, estimation of the degree of hazard impact on the system will be helpful system diagnosis and impact assessment.

References

- [1] A. Albee, S. Battel, R. Brace, G. Burdick, "Report on the Loss of the Mars Polar Lander and Deep Space 2 Missions" NASA STI/ Recon Technical Report N, March 2000.
- [2] C. Keating, R. Rogers, R. Unal, D. Dryer, A. Sousa-Poza, R. Safford, et., "System of Systems Engineering," Engineering Management Journal, vol. 15, no. 3, pp. 36–45, Sept. 2003.
- [3] "Defence Acquisition Guidebook," US Department of Defence, vol. 1st ed., Jul. 2006.
- [4] Jane T. Malin, L. Fleming, "Vulnerabilities, Influences and Interaction Paths: Failure Data for Integrated System Risk Analysis," Aerospace Conference, 2006 IEEE, 2006.
- [5] SEDSIG, "Systems Engineering Domain Special Interest Group," <http://syseng.omg.org/>, March 2006.
- [6] D. Raheja and B. Moriarty, "New Paradigms in System Safety," Journal of System Safety, vol. 42, no. 6, Nov.-Dec. 2006.
- [7] N. Leveson and N. Dulac, "Safety and Risk Driven Design in Complex Systems of Systems," in Proceedings of the 1st NASA/AIAA Space Exploration Conference, Orlando, Florida, Feb. 2005.
- [8] E. Hollnagel, "Barriers and Accident Prevention." Ashgate Pub Ltd, 2004.
- [9] C.W. Johnson and C. M. Holloway, "The ESA/NASA SOHO Mission Interruption: Using the Stamp Accident Analysis Technique for a Software Relationship," Software: Practice and Experience, vol. 33, p. 1177-1198, 2003.
- [10] CISHEC, "A Guide to Hazard and Operability Studies." The Chemical Industry Safety and Health Council of the Chemical Industries Association Ltd., 1977.
- [11] IAN R. Grosse, JOHN M. Milton Benoit, JACK C. Wileden, "Ontologies for Supporting Engineering Analysis Models," Artificial Intelligence for Engineering Design, Analysis, and Manufacturing, vol. 19, pp. 1–18, 2005.
- [12] "Facility System Safety Guidebook," NASA STD 8719.7, 1998.
- [13] R. C. R. Stone, K. Wood, "A Heuristic Method to Identify Modules from a Functional Description of a Product," Proceedings of DETC98, DETC98/DTM-5642, Atlanta, GA, 1998.
- [14] ———, "A Heuristic Method for Identifying Modules for Product Architectures," Design Studies, vol. 21, no. 1, pp. 5–31, 2000.
- [15] T. Weikiens, "Systems Engineering with SysML/UML: Modeling, Analysis, Design. Morgan Kaufmann, 2008.
- [16] S. Friedenthal, A. Moore, R. Steiner, "A Practical Guide to SysML: The Systems Modeling Language." Addison-Wesley Professional, 2004.
- [17] Y. Vanderperren, W. Dehaene, "Uml2 and Sysml: An Approach to Deal with Complexity in SOC/NOC Design," ACM Transactions on Programming Languages and Systems, vol. 2, pp. 716 – 717, 2005.
- [18] E. Huang, R. Ramamurthy, L.F. McGinnis, "System and Simulation Modeling Using Sysml," IEEE Press Piscataway, pp. 796–803, 2007.
- [19] S. Poll, A. Patterson-Hine, J. Camisa, D. Garcia, and D. Hall, "Advanced Diagnostics and Prognostics Testbed," 18th International Workshop on Principles of Diagnosis (DX-07) Nashville, TN, 2007.
- [20] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, "Introduction to Algorithms." MIT Press and McGraw-Hill, 2001.
- [21] Michael T. Goodrich, "Algorithm Design: Foundations, Analysis, and Internet Examples." Wiley, 2001.

- [22] J. U. H. v.d. H. G.M.J-B. Chaslot, M.H.M. Winands and B. Bouzy., "Progressive Strategies for Monte-Carlo Tree Search," *New Mathematics and Natural Computation*, vol. 4, no. 3, pp. 343–357, 2008.
- [23] T. Cazenave and N. Jouandeau, "A Parallel Monte-Carlo Tree Search Algorithm." *Computers and Games*, vol. 5131, pp. 72–80, 2008
- [24] T. Kurtoglu, M. Campbell, "Automated Synthesis of Electromechanical Design Configurations from Empirical Analysis of Function to form Mapping," *Journal of Engineering Design*, vol. 20, no. 1, Feb. 2009.
- [25] T.Kurtoglu, P. Bunus, and J.de Kleer, "Simulation-based Design of Air-craft Electrical Power Systems," presented at the 8th International Modelica Conference 2011, Dresden, Germany, March 2011.
- [26] Cellier, F. E., "Object-Oriented Modeling: Means for Dealing with System Complexity," 15th Benelux Meeting on Systems and Control, Mierlo, Netherlands, 1996.
- [27] Diaz-Calderon, A., Paredis, C. J. J., and Khosla, P. K., "Reconfigurable Models: A Modeling Paradigm to Support Simulation-Based Design," 2000 Summer Computer Simulation Conference, Vancouver, Canada, 2000.
- [28] P. Fritzson. "Principles of Object-Oriented Modeling and Simulation with Modelica 2.1." WileyIEEE Press, 2004.
- [29] Fishwick, P. A., "Integrating Continuous and Discrete Models with Object Oriented Physical Modeling," 1997 Western Simulation Multiconference, Phoenix, Arizona, 1997.
- [30] Zeigler, B. P., Praehofer, H., and Kim, T. G., "Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems", 2nd ed. Academic Press, 2000.
- [31] OMG, inc., "OMG Systems Modeling Language (OMG SysML™)," Nov. 2008.
- [32] Law, A. M. and W. D. Kelton. "Simulation Modeling and Analysis." 3rd ed. Boston: McGraw-Hill., 2000
- [33] R. Passerone, I. B. Hafaiedh, S. Graf, A. Benveniste, D. Cancila, A. Cuccuru, S. Gérard, F. Terrier, W. Damm, A. Ferrari, L. Mangeruca, B. Josko, T. Peikenkamp, and A. Sangiovanni-Vincentelli, "Metamodels in Europe: Languages, Tools, and Applications," *IEEE Design & Test of Computers*, vol. 26, no. 3, pp. 38–53, May/Jun. 2009.
- [34] MagicDraw <http://www.magicdraw.com/>
- [35] A. Pyster and P. Gardner, "Critical Success Factors in System of Systems Engineering," SAIC Corporation, October 2006, [http://sunset.usc.edu/events/2006/CSSE Convocation/presentations/Pyster.ppt](http://sunset.usc.edu/events/2006/CSSE%20Convocation/presentations/Pyster.ppt), accessed 20 Jan 2007.
- [36] V. Kotov, "Systems of Systems as Communicating Structures," Hewlett-Packard Computer Systems Laboratory, Tech. Rep., 1997.
- [37] Dependable Systems of Systems Project, "Dependable Systems of Systems Project Summary," June 2003, <http://www.newcastle.research.ec.org/dsos/programme/summary.html>.
- [38] M. W. Maier, "Architecting Principles for Systems-of-Systems," in 6th Annual Symposium of INCOSE, 1996, pp. 567–574.
- [39] Alexander, R. "Using Simulation for Systems of Systems Hazard Analysis." PhD thesis, Department of Computer Science, University of York, 2007
- [40] P. Checkland, "Systems Thinking, Systems Practice". John Wiley & Sons, 1999.
- [41] IMOFIS Project, Ingénierie des Modèles Fonctions Sécuritaires "Model-Driven Engineering for Safety Functions"; <http://www.imofis.org>
- [42] B. Selic, "On the Semantic Foundations of Standard UML 2.0," *Formal Methods for the Design of Real-*

Time Systems, LNCS 3185, M. Bernardo and F. Corradini, eds., Springer-Verlag, 2004, pp. 181-199.

- [43] MeMVATEX French Project, *Me´ thodologie pour la Mode´ - lisation, la Validation et la Tracabilite´ des Exigences* “Methodology for Modeling, Validation, and Traceability of Requirements”; <http://www.memvatex.org> (in French).
- [44] ATESSST Project, “Advancing Traffic Efficiency and Safety through Software Technology,” ATESSST Specific Targeted Research or Innovation Project (STREP), 6th Framework Programme; <http://www.atesst.org>.
- [45] Lambda Project, “Lambda Libraries for Applying Model Based Development Approaches”; http://www.usine-logicielle.org/lambda/index_FR.html.
- [46] A. Cuccuru et al., “Enhancing UML Extensions with Operational Semantics: Behavioed Profiles with Templates,” Proc. 10th Int’l Conf. Model Driven Engineering Languages and Systems (MODELS 07), LNCS 4735, Springer, 2007, pp. 271-285.
- [47] “Interested FP7 IP: Interoperable Embedded Systems Tool-Chain for Enhanced Rapid Design, Prototyping and Code Generation”; <http://www.interested-ip.eu>.
- [48] “Genesys Project, Generic Embedded System Platform”, a 7th Framework program; <http://www.genesys-platform.eu>.
- [49] CESAR Project, “Cost-Efficient Methods and Processes for Safety Relevant Embedded Systems,” funded project from Artemis Joint Undertaking (JU); <http://www.cesarproject.eu>.
- [50] COMBEST, “Component-Based Embedded Systems Design Techniques project, Information Society Technologies”, STREP 215543, a 7th Framework program; <http://www.combest.eu>.
- [51] “Speculative and Exploratory Design in Systems Engineering, European Union 6th Framework Project in Embedded Systems Development”, IP contract 033471; <http://www.speeds.eu.com>.
- [52] D. Pumfrey, “The Principled Design of Computer System Safety Analysis,” DPhil Thesis, University of York, 2000.
- [53] A. Villemeur, “Reliability, Availability, Maintainability and Safety Assessment: Volume 1 Methods and Techniques”. Chicester, England: John Wiley & Sons, 1992.
- [54] “Introduction to Hazop+ 2008” <http://www.isograph-software.com/hazover.htm>
- [55] R. Rai, T. Kurtoglu, and M. Campbell, 2009, “Stochastic Interactive Graph Grammar Search for Conceptual Design” ASME Journal of Mechanical Design
- [56] M. I. Campbell, “A Graph Grammar Methodology for Creative Systems,” Journal of Computer and Information Science in Engineering, vol. accepted, 2009.

Appendix A

XMI Hazard Propagation Path

```
package XMI;
import java.io.*;
import java.util.Iterator;

import javax.xml.stream.*;
import javax.xml.stream.events.*;

public class XMISearch {

    private XMLInputFactory inputFactory = null;
    private XMLEventReader xmlEventReader = null;

    String name;
    public XMISearch() {
        inputFactory = XMLInputFactory.newInstance();
    }

    public void read() throws Exception{

        xmlEventReader = inputFactory.createXMLEventReader(
            new FileReader("EPSXMI4.xml"));

        while (xmlEventReader.hasNext()){

            XMLEvent xmlEvent = xmlEventReader.nextEvent();

            if (xmlEvent.isStartElement()){

                if(xmlEvent.asStartElement().getName().toString().equals("packagedElement"))
                {
                    Iterator<Attribute> attributes
                    =xmlEvent.asStartElement().getAttributes();
                    while (attributes.hasNext()) {
                        Attribute attribute =
                        attributes.next();
                        if
                        (attribute.getName().toString().equals("name")) {
                            name=attribute.getValue();
                        }
                    }
                }
                Iterator<Attribute> attributes
                =xmlEvent.asStartElement().getAttributes();
            }
        }
    }
}
```

```

        while (attributes.hasNext()) {
            Attribute attribute =
attributes.next();
            if
(attribute.getName().toString().equals("name")) {
                if
(attribute.getValue().contains("Hazard: "))
                {
                    System.out.print(name+ " "+
attribute.getValue());
                }
            }
        }
    }
    Iterator<Attribute> attributes2
=xmlEvent.asStartElement().getAttributes();
    while (attributes2.hasNext()) {
        Attribute attribute =
attributes2.next();
        if
(attribute.getName().toString().equals("name")) {
            if
(attribute.getValue().contains("Vulnerability "))
            {
                System.out.print(name+ " "+
attribute.getValue());
            }
        }
    }
    }
    }else if (xmlEvent.isCharacters()){
    }
}

xmlEventReader.close();
}

public static void main(String args[]){
    try{
        XMISearch obj = new XMISearch();
        obj.read();
    }catch(Exception exception){
        exception.printStackTrace();
    }
}
}

```


Appendix B

EPS XMI Code from Block Definition Diagram

```
<?xml version='1.0' encoding='UTF-8'?>

<xmi:XMI xmi:version='2.1' xmlns:uml='http://www.omg.org/spec/UML/20090901' xmlns:xmi='http://schema.omg.org/spec/XMI/2.1'
xmlns:StandardProfileL2='http://schema.omg.org/spec/UML/2.3/StandardProfileL2.xmi' xmlns:sysml='http://www.omg.org/spec/SysML/20100301/SysML-profile'
xmlns:StandardProfileL3='http://schema.omg.org/spec/UML/2.3/StandardProfileL3.xmi'
xmlns:additional_stereotypes='http://www.magicdraw.com/schemas/additional_stereotypes.xmi'
xmlns:customizations_for_traceability='http://www.magicdraw.com/schemas/customizations_for_traceability.xmi'
xmlns:DSL_Customization='http://www.magicdraw.com/schemas/DSL_Customization.xmi'
xmlns:Traceability_customization='http://www.magicdraw.com/schemas/Traceability_customization.xmi'
xmlns:Validation_Profile='http://www.magicdraw.com/schemas/Validation_Profile.xmi'
xmlns:MagicDraw_Profile='http://www.magicdraw.com/schemas/MagicDraw_Profile.xmi'
xmlns:Matrix_Templates_Profile='http://www.magicdraw.com/schemas/Matrix_Templates_Profile.xmi'
xmlns:matrix_templates='http://www.magicdraw.com/schemas/matrix_templates.xmi'>
  <xmi:Documentation exporter='MagicDraw UML' exporterVersion='16.9'/>
  <xmi:Extension extender='MagicDraw UML 16.9'>
    <plugin pluginName='SysML' pluginVersion='16.9 sp1'/>
    <req_resource resourceID='1440' resourceName='SysML' resourceValueName='MD_customization_for_SysML.mdzip'/>
    <req_resource resourceID='1440' resourceName='SysML' resourceValueName='SysML_matrix_templates_module.mdzip'/>
    <req_resource resourceID='1440' resourceName='SysML' resourceValueName='SysML Profile.mdzip'/>
    <req_resource resourceID='1440' resourceName='SysML' resourceValueName='SysML Block Definition Diagram'/>
    <req_resource resourceID='1440' resourceName='SysML' resourceValueName='SysML'/>
    <shareTable shareVersion='-1' standardSystemProfile='false'/>
    <mountTable>
      <module resource='file:/C:/Program%20Files/MagicDraw%20UML/profiles/UML_Standard_Profile.xml' autoloadType='ALWAYS_LOAD'
readOnly='true' loadIndex='false' requiredVersion='-1' version='16.9'>
        <mount mountPoint='magicdraw_uml_standard_profile_v_0001' mountedOn='eee_1045467100313_135436_1'/>
      </module>
      <module resource='file:/C:/Program%20Files/MagicDraw%20UML/profiles/MD_customization_for_SysML.mdzip'
```

```

autoloadType='ALWAYS_LOAD' readOnly='true' loadIndex='true' requiredVersion='-1' version='16.9'>
    <mount mountPoint='_12_0EAPbeta_be00301_1156851270584_552173_1' mountedOn='eee_1045467100313_135436_1'>
    <mount mountPoint='_16_8beta_2104050f_1262918510515_114803_6875' mountedOn='eee_1045467100313_135436_1'>
    </module>
    <module resource='file:/C:/Program%20Files/MagicDraw%20UML/profiles/SysML_matrix_templates_module.mdzip'
autoloadType='ALWAYS_LOAD' readOnly='false' loadIndex='true' requiredVersion='-1' version='16.9'>
    <mount mountPoint='_16_8_2104050f_1270804267770_52342_3174'
mountedOn='_12_0EAPbeta_be00301_1156851270584_552173_1'>
    <mount mountPoint='_10_5_1_8f90291_1141933276109_541328_27'
mountedOn='_12_0EAPbeta_be00301_1156851270584_552173_1'>
    </module>
    <module resource='file:/C:/Program%20Files/MagicDraw%20UML/profiles/SysML%20Profile.mdzip' autoloadType='ALWAYS_LOAD'
readOnly='true' loadIndex='true' requiredVersion='-1' version='16.9'>
    <mount mountPoint='_11_5EAPbeta_be00301_1147434586638_637562_1900' mountedOn='eee_1045467100313_135436_1'>
    </module>
    </mountTable>
</xmi:Extension>
<uml:Model xmi:id='eee_1045467100313_135436_1' name='Data' visibility='public'>
    <xmi:Extension extender='MagicDraw UML 16.9'>
    <moduleExtension ignoredInModule='true'>
    </xmi:Extension>
    <ownedComment xmi:type='uml:Comment' xmi:id='_12_1_8740266_1173856349343_83760_98' body='Author:mehrpouy.&#10;Created:7/11/11 6:10
PM.&#10;Title:.&#10;Comment:.&#10;'>
    <annotatedElement xmi:idref='eee_1045467100313_135436_1'>
    </ownedComment>
    <packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310434395380_549645_12068' name='VoltSenE242' visibility='public'>
    <packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313637315795_131603_12956' visibility='public'>
    <informationSource xmi:idref='_17_0_2_f4d037a_1310434436678_450278_12112'>
    <informationTarget xmi:idref='_17_0_2_f4d037a_1310436033703_359759_12472'>
    </packagedElement>
    <packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313636960470_162006_12812' visibility='public'>
    <informationSource xmi:idref='_17_0_2_f4d037a_1310434395380_549645_12068'>
    <informationTarget xmi:idref='_17_0_2_f4d037a_1310435061119_657986_12371'>
    </packagedElement>
    <packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313653172795_667697_13995' visibility='public'>
    <informationSource xmi:idref='_17_0_2_f4d037a_1310436933537_766003_13616'>
    <informationTarget xmi:idref='_17_0_2_f4d037a_1310500418054_330653_14154'>

```

```

</packagedElement>
<packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313653331060_629880_14058' visibility='public'>
  <informationSource xmi:idref='_17_0_2_f4d037a_1310436948803_713125_13660'>
  <informationTarget xmi:idref='_17_0_2_f4d037a_1310500827932_728300_14261'>
</packagedElement>
<packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313631904106_92043_11903' visibility='public'>
  <informationSource xmi:idref='_17_0_2_f4d037a_1310433556281_867835_11463'>
  <informationTarget xmi:idref='_17_0_2_f4d037a_1310434279692_962656_11830'>
</packagedElement>
<packagedElement xmi:type='uml:Package' href='MD_customization_for_SysML.mdzip#_12_0EAPbeta_be00301_1156851270584_552173_1'>
  <xmi:Extension extender='MagicDraw UML 16.9'>
    <referenceExtension referentPath='MD Customization for SysML' referentType='Package'>
  </xmi:Extension>
</packagedElement>
<packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313631822624_244848_11775' visibility='public'>
  <informationSource xmi:idref='_17_0_2_f4d037a_1310434279692_962656_11830'>
  <informationTarget xmi:idref='_17_0_2_f4d037a_1310434349083_50537_11949'>
</packagedElement>
<packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310433556281_867835_11463' name='CurrSenIT240' visibility='public'>
<packagedElement xmi:type='uml:Association' xmi:id='_17_0_2_f4d037a_1310433595047_45243_11496' visibility='public'>
  <memberEnd xmi:idref='_17_0_2_f4d037a_1310433595047_720560_11497'>
  <memberEnd xmi:idref='_17_0_2_f4d037a_1310433595047_398870_11498'>
  <ownedEnd xmi:type='uml:Property' xmi:id='_17_0_2_f4d037a_1310433595047_398870_11498' visibility='private'
type='_17_0_2_f4d037a_1310433075013_79097_11320' association='_17_0_2_f4d037a_1310433595047_45243_11496'>
</packagedElement>
<packagedElement xmi:type='uml:Profile' href='Matrix_Templates_Profile.xml#_10_5_8f90291_1141412747796_552778_939'>
  <xmi:Extension extender='MagicDraw UML 16.9'>
    <referenceExtension referentPath='Matrix Templates Profile' referentType='Profile'>
  </xmi:Extension>
</packagedElement>
<packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310434205395_579859_11682' name='Ground' visibility='public'>
<packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313686472379_15180_14196' visibility='public'>
  <informationSource xmi:idref='_17_0_2_f4d037a_1310500418054_330653_14154'>
  <informationTarget xmi:idref='_17_0_2_f4d037a_1310434205395_579859_11682'>
</packagedElement>
<packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313637523036_146890_13154' visibility='public'>
  <informationSource xmi:idref='_17_0_2_f4d037a_1310436236752_919309_12589'>

```



```

        <informationTarget xmi:idref='_17_0_2_f4d037a_1310434205395_579859_11682'/>
    </packagedElement>
    <packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313686414428_328288_14149' visibility='public'>
        <informationSource xmi:idref='_17_0_2_f4d037a_1310500827932_728300_14261'/>
        <informationTarget xmi:idref='_17_0_2_f4d037a_1310434205395_579859_11682'/>
    </packagedElement>
    <packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313653033245_25033_13898' visibility='public'>
        <informationSource xmi:idref='_17_0_2_f4d037a_1310436033703_359759_12472'/>
        <informationTarget xmi:idref='_17_0_2_f4d037a_1310436948803_713125_13660'/>
    </packagedElement>
    <packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313634356268_602346_12313' visibility='public'>
        <informationSource xmi:idref='_17_0_2_f4d037a_1310434216801_798185_11726'/>
        <informationTarget xmi:idref='_17_0_2_f4d037a_1310434205395_579859_11682'/>
    </packagedElement>
    <packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310436948803_713125_13660' name='RelayE275' visibility='public'/>
    <packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313643132141_377608_13638' visibility='public'>
        <informationSource xmi:idref='_17_0_2_f4d037a_1310436344080_649693_12738'/>
        <informationTarget xmi:idref='_17_0_2_f4d037a_1310434205395_579859_11682'/>
    </packagedElement>
    <packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310436344080_649693_12738' name='VoltSenE281' visibility='public'/>
    <packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313643343806_161172_13743' visibility='public'>
        <informationSource xmi:idref='_17_0_2_f4d037a_1310500924214_62150_14402'/>
        <informationTarget xmi:idref='_17_0_2_f4d037a_1310434205395_579859_11682'/>
    </packagedElement>
    <packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310435061119_657986_12371' name='CurrSenIT281' visibility='public'/>
    <packagedElement xmi:type='uml:Usage' xmi:id='_17_0_2_f4d037a_1310433220857_272445_11400' visibility='public'>
        <supplier xmi:idref='_17_0_2_f4d037a_1310433220810_711651_11398'/>
        <client xmi:idref='_17_0_2_f4d037a_1310433075013_79097_11320'/>
    </packagedElement>
    <packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310434216801_798185_11726' name='VoltSenE240' visibility='public'/>
    <packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310500827932_728300_14261' name='FANFAN416' visibility='public'>
        <ownedAttribute xmi:type='uml:Property' xmi:id='_17_0_2_f4d037a_1310500879432_822390_14340' name='Vulnerability Over Current'
visibility='public'/>
        <ownedAttribute xmi:type='uml:Property' xmi:id='_16_9_39a017d_1313686346319_567793_14115' name='Hazard:Null' visibility='public'/>
        <ownedAttribute xmi:type='uml:Property' xmi:id='_16_9_39a017d_1313686365645_869499_14122' name='Carr:Electrical Current'
visibility='public'/>
    </packagedElement>

```

```

<packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313612773790_290628_10921' name='Electric' visibility='public'>
  <informationSource xmi:idref='_17_0_2_f4d037a_1310433890190_553579_11547'>
  <informationTarget xmi:idref='_17_0_2_f4d037a_1310433107372_936347_11353'>
</packagedElement>
<packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310436864974_400360_13446' name='RelayEY284' visibility='public'>
<packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310436933537_766003_13616' name='RelayE272' visibility='public'>
<packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313632002135_518017_11983' visibility='public'>
  <informationSource xmi:idref='_17_0_2_f4d037a_1310433556281_867835_11463'>
  <informationTarget xmi:idref='_17_0_2_f4d037a_1310434216801_798185_11726'>
</packagedElement>
<packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313631674627_75364_11707' visibility='public'>
  <informationSource xmi:idref='_17_0_2_f4d037a_1310433890190_553579_11547'>
  <informationTarget xmi:idref='_17_0_2_f4d037a_1310433556281_867835_11463'>
</packagedElement>
<packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313634579627_724438_12598' visibility='public'>
  <informationSource xmi:idref='_17_0_2_f4d037a_1310434436678_450278_12112'>
  <informationTarget xmi:idref='_17_0_2_f4d037a_1310434395380_549645_12068'>
</packagedElement>
<packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313634250633_354420_12197' visibility='public'>
  <informationSource xmi:idref='_17_0_2_f4d037a_1310434279692_962656_11830'>
  <informationTarget xmi:idref='_17_0_2_f4d037a_1310434216801_798185_11726'>
</packagedElement>
<packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310436236752_919309_12589' name='VoltSenE265' visibility='public'>
<packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313652069821_36404_13840' visibility='public'>
  <informationSource xmi:idref='_17_0_2_f4d037a_1310436033703_359759_12472'>
  <informationTarget xmi:idref='_17_0_2_f4d037a_1310436933537_766003_13616'>
</packagedElement>
<packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310434349083_50537_11949' name='PosSenESH244A' visibility='public'>
<packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310433075013_79097_11320' name='Battery' visibility='public'>
  <ownedAttribute xmi:type='uml:Port' xmi:id='_17_0_2_f4d037a_1310433168201_673590_11384' name='unnamed1' visibility='public'
aggregation='composite' type='_17_0_2_f4d037a_1310433075013_79097_11320'>
  <ownedAttribute xmi:type='uml:Port' xmi:id='_17_0_2_f4d037a_1310433291577_154829_11413' visibility='public' aggregation='composite'>
  <ownedAttribute xmi:type='uml:Property' xmi:id='_17_0_2_f4d037a_1310433517156_332774_11450' visibility='private'
type='_17_0_2_f4d037a_1310433107372_936347_11353' association='_17_0_2_f4d037a_1310433517156_92474_11449'>
  <ownedAttribute xmi:type='uml:Property' xmi:id='_17_0_2_f4d037a_1310433595047_720560_11497' visibility='private'
type='_17_0_2_f4d037a_1310433556281_867835_11463' association='_17_0_2_f4d037a_1310433595047_45243_11496'>
  <ownedAttribute xmi:type='uml:Property' xmi:id='_17_0_2_f4d037a_1310433791298_867870_11513' name='Over Current' visibility='public'

```

```

aggregation='composite'/>
    <ownedAttribute xmi:type='uml:Property' xmi:id='_17_0_2_f4d037a_1310433827674_155209_11517' name='unnamed2' visibility='public'
aggregation='composite'/>
    </packagedElement>
    <packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310500924214_62150_14402' name='DC ResistorDC485' visibility='public'>
        <ownedAttribute xmi:type='uml:Property' xmi:id='_17_0_2_f4d037a_1310501623250_331799_14512' name='Vulnerability Over Current'
visibility='public'/>
        <ownedAttribute xmi:type='uml:Property' xmi:id='_16_9_39a017d_1313650979730_854569_13791' name='Hazard:Null' visibility='public'/>
        <ownedAttribute xmi:type='uml:Property' xmi:id='_16_9_39a017d_1313650997868_472768_13798' name='Carr:Electrical Current'
visibility='public'/>
    </packagedElement>
    <packagedElement xmi:type='uml:Interface' xmi:id='_17_0_2_f4d037a_1310433220810_711651_11398' visibility='public'/>
    <packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310433890190_553579_11547' name='SourceofPower&#10;' visibility='public'>
        <ownedAttribute xmi:type='uml:Property' xmi:id='_16_9_39a017d_1313641212043_802391_13415' name='Vulnerability:Null'
visibility='public'/>
        <ownedAttribute xmi:type='uml:Property' xmi:id='_17_0_2_f4d037a_1310434162129_729781_11663' name='Hazard: Over Current'
visibility='public'/>
        <ownedAttribute xmi:type='uml:Property' xmi:id='_16_9_39a017d_1313641248463_484328_13422' name='Carr:Electrical Current'
visibility='public'/>
        <ownedOperation xmi:type='uml:Operation' xmi:id='_17_0_2_f4d037a_1310433936737_522046_11649' name='unnamed1' visibility='public'/>
    </packagedElement>
    <packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310434279692_962656_11830' name='RelayEY244' visibility='public'/>
    <packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313643231272_651848_13689' visibility='public'>
        <informationSource xmi:idref='_17_0_2_f4d037a_1310436864974_400360_13446'/>
        <informationTarget xmi:idref='_17_0_2_f4d037a_1310500924214_62150_14402'/>
    </packagedElement>
    <packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310433107372_936347_11353' name='TmpSenTE228' visibility='public'>
        <ownedAttribute xmi:type='uml:Port' xmi:id='_17_0_2_f4d037a_1310433197888_403639_11391' name='unnamed1' visibility='public'
aggregation='composite' type='_17_0_2_f4d037a_1310433107372_936347_11353'/>
        <ownedAttribute xmi:type='uml:Port' xmi:id='_17_0_2_f4d037a_1310433298873_287943_11429' visibility='public' aggregation='composite'/>
    </packagedElement>
    <packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313632019477_500597_12018' visibility='public'>
        <informationSource xmi:idref='_17_0_2_f4d037a_1310434216801_798185_11726'/>
        <informationTarget xmi:idref='_17_0_2_f4d037a_1310434279692_962656_11830'/>
    </packagedElement>
    <packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313634599633_680820_12622' visibility='public'>
        <informationSource xmi:idref='_17_0_2_f4d037a_1310434395380_549645_12068'/>

```

```

        <informationTarget xmi:idref='_17_0_2_f4d037a_1310434205395_579859_11682'/>
    </packagedElement>
    <packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313637477240_447797_13123' visibility='public'>
        <informationSource xmi:idref='_17_0_2_f4d037a_1310436033703_359759_12472'/>
        <informationTarget xmi:idref='_17_0_2_f4d037a_1310436236752_919309_12589'/>
    </packagedElement>
    <packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313634041474_361279_12137' visibility='public'>
        <informationSource xmi:idref='_17_0_2_f4d037a_1310434279692_962656_11830'/>
        <informationTarget xmi:idref='_17_0_2_f4d037a_1310434436678_450278_12112'/>
    </packagedElement>
    <packagedElement xmi:type='uml:Package' href='MD_customization_for_SysML.mdzip#_16_8beta_2104050f_1262918510515_114803_6875'>
        <xmi:Extension extender='MagicDraw UML 16.9'>
            <referenceExtension referentPath='QUDV Library' referentType='Package'/>
        </xmi:Extension>
    </packagedElement>
    <packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313643114654_927013_13607' visibility='public'>
        <informationSource xmi:idref='_17_0_2_f4d037a_1310436864974_400360_13446'/>
        <informationTarget xmi:idref='_17_0_2_f4d037a_1310436344080_649693_12738'/>
    </packagedElement>
    <packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310434436678_450278_12112' name='InverterINV2' visibility='public'/>
    <packagedElement xmi:type='uml:Association' xmi:id='_17_0_2_f4d037a_1310433517156_92474_11449' visibility='public'>
        <memberEnd xmi:idref='_17_0_2_f4d037a_1310433517156_332774_11450'/>
        <memberEnd xmi:idref='_17_0_2_f4d037a_1310433517156_136793_11451'/>
        <ownedEnd xmi:type='uml:Property' xmi:id='_17_0_2_f4d037a_1310433517156_136793_11451' visibility='private'
type='_17_0_2_f4d037a_1310433075013_79097_11320' association='_17_0_2_f4d037a_1310433517156_92474_11449'/>
    </packagedElement>
    <packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310436748614_269513_13278' visibility='public'/>
    <packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310436586301_205_13080' name='Over Current' visibility='public'/>
    <packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313641105836_742123_13366' visibility='public'>
        <informationSource xmi:idref='_17_0_2_f4d037a_1310435061119_657986_12371'/>
        <informationTarget xmi:idref='_17_0_2_f4d037a_1310436864974_400360_13446'/>
    </packagedElement>
    <packagedElement xmi:type='uml:Profile' href='http://www.omg.org/spec/SysML/20100301/SysML-profile#_0'>
        <xmi:Extension extender='MagicDraw UML 16.9'>
            <referenceExtension referentPath='SysML' referentType='Profile'/>
        </xmi:Extension>
    </packagedElement>

```

```

        <packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310500418054_330653_14154' name='AC ResistorAC483' visibility='public'>
            <ownedAttribute xmi:type='uml:Property' xmi:id='_17_0_2_f4d037a_1310500464773_267357_14204' name='Vulnerability Over Current'
visibility='public'/>
            <ownedAttribute xmi:type='uml:Property' xmi:id='_16_9_39a017d_1313653267172_306687_14033' name='Hazard:Null' visibility='public'/>
            <ownedAttribute xmi:type='uml:Property' xmi:id='_16_9_39a017d_1313653282902_708666_14040' name='Carr:Electrical Current'
visibility='public'/>
        </packagedElement>
        <packagedElement xmi:type='uml:InformationFlow' xmi:id='_16_9_39a017d_1313637007306_714684_12857' visibility='public'>
            <informationSource xmi:idref='_17_0_2_f4d037a_1310435061119_657986_12371'/>
            <informationTarget xmi:idref='_17_0_2_f4d037a_1310434205395_579859_11682'/>
        </packagedElement>
        <packagedElement xmi:type='uml:Class' xmi:id='_17_0_2_f4d037a_1310436033703_359759_12472' name='CurrSenIT267' visibility='public'/>
        <packagedElement xmi:type='uml:Package' href='UML_Standard_Profile.xml#magicdraw_uml_standard_profile_v_0001'>
            <xmi:Extension extender='MagicDraw UML 16.9'>
                <referenceExtension referentPath='UML Standard Profile' referentType='Package'/>
            </xmi:Extension>
        </packagedElement>
    </uml:Model>
    <xmi:Extension extender='MagicDraw UML 16.9'>
        <mdOwnedDiagrams

```


På svenska

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under en längre tid från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida <http://www.ep.liu.se/>

In English

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: <http://www.ep.liu.se/>