TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Jari Rauhamäki

Designing Functional Safety Systems

A Pattern Language Approach

Julkaisu 1478 • Publication 1478

Tampere 2017

Tampereen teknillinen yliopisto. Julkaisu 1478
Tampere University of Technology. Publication 1478

Jari Rauhamäki

Designing Functional Safety Systems

A Pattern Language Approach

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Festia Building, Auditorium Pieni Sali 1, at Tampere University of Technology, on the 9th of June 2017, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2017

ISBN 978-952-15-3964-0 (printed)
ISBN 978-952-15-3967-1 (PDF)
ISSN 1459-2045

Abstract

Human beings, at least most of us, want to feel and be safe. This is one of the fundamental needs of an organism. However, several of the processes and machines used in current societies introduce hazards that could and can harm us causing unnecessary pain and financial losses. Still, our modern societies need these processes and machines to operate so we cannot really be without them. Fortunately, there are ways to reduce risks introduced by systems around us to a tolerable level.

This thesis considers the design and development of safety-related systems and safety-related parts of control systems referred to as functional safety systems. These systems implement safety functions that reduce risks introduced by machines, processes, and other systems. That is, the functions affect the system under control so that the likelihood of occurrence or severity of consequences are reduced.

The design and development of safety systems is typically regulated by laws and standards. This increases the cost of safety system development and therefore eventually also the product in which it is incorporated. However, from a manufacturer viewpoint, safety in all its forms is also a potential asset for the companies developing, producing, and selling the systems. An increase in efficiency to develop and design safety systems offers the potential for a larger margin or increased sales due to the reduced price.

One way to support design and development efficiency is to apply good design methods and solutions in form of design patterns. In this thesis, a design pattern language for the development and design of functional safety systems is introduced. The purpose of the language is to support the designers in their task to design and implement safety functions in machines and processes. The language considers various aspects of the development and design of safety systems starting from the initial phases of hazard and risk analysis, followed by the selection of the hazard and risk reduction methods, and concluding with the hardware and software structure, functionality, and design principles considerations. Finally, a functional safety system may, and often does, co-exist and co-operate with a control system. Therefore, a part of the pattern language takes this aspect into account.

To compile the design pattern language and the included patterns a design science research approach complemented with grounded theory approach is applied The data to identify the patterns is collected from literature, personal experience, interviews, and discussions with industry representatives and people engaged with the design or use of systems including safety systems or functionality. Like the patterns have evolved during the research, so has the approach to identify, document, and process the patterns.

Preface

The research for this thesis was carried out at the Department of Automation Science and Engineering at the Tampere University of Technology during the years 2010-2016. Several projects, including Improving software architecting practices in machine control systems (Sulava), Safety-critical software in machinery applications (Ohjelmaturva), Reuse of the process management design solutions (ReUse), and Design patterns in functional safety system design for intelligent machines (DPSafe) provided the framework and most of the funding for the research work.

I want to express my gratitude to Professor Seppo Kuikka for inspiration, support, and guidance during the research work and all the work he has put to realize the projects in which I have had the opportunity to work in. The devotion and persistence of him has also encouraged me with my research. My gratitude also belongs to Professor Matti Vilkko for becoming my supervisor, support, and providing a fresh viewpoint.

I want to thank the pre-examiners Prof. Tor Stålhane from Norwegian University of Science and Technology and Dr.Tech. Christian Kreiner from Graz University of Technology for reviewing my thesis. I also want to thank Dr.Tech. Olli Ventä from VTT and Dr.Tech. Kreiner for agreeing to be my opponents in the public defence of the thesis.

During these years, I have had the pleasure to work with inspiring people from, including but not limited to: TUT, VTT, Aalto University, and several companies. Thank you all, it has been a supportive atmosphere, and I have enjoyed each day. Foremost, I want to thank my current and former colleagues in the Automation Software Engineering research group, especially Timo Vepsäläinen, David Hästbacka, Petri Kannisto, Outi Rask, and Pekka Alho.

The results of the thesis would likely look rather different unless I had been introduced to the world of design patterns. I cannot remember how this actually happened, but I would like to thank Seppo and Timo for supporting me in the research and documentation of the patterns. Equally, I would like to thank Veli-Pekka Eloranta and Marko Leppänen for introducing and making me a part of the pattern community and for all their support, ideas, and advice. Besides all the great people in the pattern community with who I have had the opportunity to talk and work with, I especially want to thank all my shepherds and folks in writer's workshops whose input has helped me to improve the patterns.

I want to thank the organizations that have funded the research of this thesis. The Finnish Funding Agency for Innovation (TEKES), Forum for Intelligent Machines (FIMA), and various companies have funded the research projects in which I have worked in. I also thank Automaatiosäätiö for the grants that have provided me with the possibility to participate conferences around Europe.

Finally, I would like to present my sincerest gratitude for my family and friends. I am most deeply grateful for all the support and love I have received from you during these years. I want to thank my parents for reminding me about the importance and uniqueness of the work. I want to thank my darling wife Anne-Mari for her love, support, understanding, and encouragement with research objectives. I am really happy you have walked this path with me. Finally, I want to thank my sons for all the great moments so far and especially providing me with something completely different to think about.

Kangasala, 14 May 2017

Jari Rauhamäki

Contents

Acronyms

DPSafe	Design patterns in functional safety system design for intelligent machines. A research project.
E/E/PE	Electrical/Electronic/Programmable Electronic
EHSR	Essential health and safety requirements
EU	European Union
EUC	Equipment Under Control
EuroPLoP	European Conference on Pattern Languages of Programs
FIMA	Forum for Intelligent Machines. An association of Finnish machinery manufacturers, engineering offices, and subcontractors.
GOF	Gang of Four consisting of Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides
ICT	Information and Communication Technology
IEC	International Electrotechnical Commission
IEC 61508	Functional safety of electrical/electronic/programmable electronic safety-related systems
MTTF	Mean Time To Fail
MTTR	Mean Time To Repair
Ohjelmaturva	Safety-critical software in machinery applications. A research project.
PLoP	Pattern Language of Programs
PLr	Required Performance Level
POSA	Pattern-Oriented Software Architecture
PPE	Personal Protective Equipment
ReUse	Reuse of the process management design solutions. A research project.
SIL	Safety Integrity Level
SIS	Safety Instrumented System

SRESW	Safety-related embedded software
Sulava	Improving software architecting practices in machine control systems. A research project.

List of Publications

1. Rauhamäki, J., Vepsäläinen, T., Kuikka, S. (2013). Patterns in Safety System Development. In Leistner, W. and Lorenz, P. (Eds.), Proceedings of *The Third International Conference on Performance, Safety and Robustness in Complex Systems and Applications, PESARO 2013, April 21-26, 2013, Venice, Italy.* (pp. 9-15). ISSN 2308-3700. ISBN 978-1-61208-268-4. International Academy, Research, and Industry Association (IARIA). Available: `https://www.thinkmind.org/index.php?view=article&articleid=pesaro_2013_1_20_70017`

   The publication provides a brief overview for the domain of the thesis including design patterns, co-existence between the safety and control systems, and safety system development process. Most importantly, the paper outlines a rationale for the usage of design patterns in the context of functional safety system development and discusses the potential drawbacks of the approach.
   The candidate was the responsible author for the publication. The ideas presented in the publication have been produced in collaboration with the second author who also contributed minor part of the content in addition to providing comments to improve the paper with the third author.

2. Rauhamäki, J., Vepsäläinen, T., Kuikka, S. (2015). Towards Systematic Safety System Development with a Tool Supported Pattern Language. In Oberhauser, R., Lavazza, L., Mannaert, H. and Clyde, S. Proceedings of *The Tenth International Conference on Software Engineering Advances, ICSEA 2015, November 15-20, 2015, Barcelona, Spain.* (pp. 341-348). ISSN 2308-4235. ISBN 978-1-61208-438-1. International Academy, Research, and Industry Association (IARIA). Available: `https://www.thinkmind.org/index.php?view=article&articleid=icsea_2015_13_20_10159`

   The publication discusses patterns and their use in safety system development considering potential pattern format, pattern mining, relations with applicable standards, and the modelling of patterns, and their support and use in development environments.
   The candidate was the responsible author for the publication and wrote most of the content in sections I-IV. The idea of the publication originates from the second author as well as most of the content of sections V-VI. The third author provided comments to improve the publication.

3. Rauhamäki, J., Kuikka, S. (2015). Patterns for control system safety. In *Proceedings of the 18th European Conference on Pattern Languages of Program.* [23] (ACM International Conference Proceeding Series). New York: ACM. DOI: 10.1145/2739011.2739034

The publication presents four approaches for control system safety documented in a design pattern format. The patterns consider a software architecture to implement interlock functionality, the design of the system to be safe when de-energized, and to check that the operation of the software produces a desired response in the physical world.

The candidate was the responsible author for the publication. The second author has provided comments to improve the publication.

4. Rauhamäki, J., Kuikka, S. (2014). Strategies for Hazard Management Process. In *Proceedings of the 19th European Conference on Pattern Languages of Programs.* [31] (ACM International Conference Proceeding Series). New York: ACM. DOI: 10.1145/2721956.2721966

   The publication illustrates hazard management methods and suggests a preferred order of consideration for the methods in a format of a pattern language.

   The candidate was the responsible author for the publication. The second author has provided comments to improve the publication.

5. Rauhamäki, J., Kuikka, S. (2015). Patterns to Implement Active Protective Measures. In *Proceedings of the 20th European Conference on Pattern Languages of Programs.* [43] (ACM International Conference Proceeding Series). New York: ACM. DOI: 10.1145/2855321.2855365

   The publication presents patterns on implementing active protective measures. The purpose of an active protective measure is to lower the risk related to a hazard by either reducing the likelihood (the frequency of exposure of) or the consequences of the realization of harm by affecting the functionality of the system.

   The candidate was the responsible author for the publication. The second author has provided comments to improve the publication.

6. Rauhamäki, J., Kuikka, S. (2015). Strategies for Hazard Management Process II. In *Proceedings of the 20th European Conference on Pattern Languages of Programs.* [3] (ACM International Conference Proceeding Series). New York: ACM. DOI: 10.1145/2855321.2855325

   The publication presents two strategies, namely active and passive protective measures, in a design pattern format. In many cases, a passive protective measure should be considered initially and preferred over an active protective measure whenever meaningful, but an active protective measure can provide functionality a passive one cannot.

   The candidate was the responsible author for the publication. The second author has provided comments to improve the publication.

7. Rauhamäki, J., Kuikka, S. (2014). Patterns for Sharing Safety System Operation Responsibilities between Humans and Machines. In *Proceedings of the 8th Nordic Conference on Pattern Languages of Programs VikingPLoP 2014, Sagadi Manor, Estonia, 10.4.-13.4.2014.* (pp. 68-74). [7] (ACM International Conference Proceeding Series). New York: ACM. DOI: 10.1145/2676680.2676687

The publication presents two patterns for sharing the responsibilities between automated safety systems and human operators. The strengths of automated safety systems and human operators can be combined so that the weaknesses of one can be compensated with the strengths of the other.

The candidate was the responsible author for the publication. The second author has provided comments to improve the publication.

8. Rauhamäki, J. (In press). Patterns for Functional Safety System Development. LNCS Transactions on Pattern Languages of Programming.

   The publication compiles and enhances the patterns presented in VikingPLoP 2012 (Rauhamäki et al., 2012) and VikingPLoP 2013 (Rauhamäki et al., 2013) papers considering functional safety system development. The authors of the original papers were the candidate, Timo Vepsäläinen, and Seppo Kuikka.

   The candidate was the responsible author for the publication. The Separated safety and Productive safety patterns are originally the work of Timo Vepsäläinen.

1 Introduction

Safety is a centric aspects of human behaviour including the design and development of systems. Thus, it is appropriate to start with thoughts on safety in general to have the background set up for the suggested outcome of the thesis.

1.1 A Craving for Safety

In Maslow's hierarchy of needs (Maslow, 1943) safety is the second most important need an organism, that is, also a human, thrives to have fulfilled. According to Maslow, only physiological needs are considered having a higher priority in the need hierarchy. That is, right after the physiological needs have been gratified, the seek to fulfil the need for safety is established. From the psychology perspective, safety or the seek to feel safe has a high influence in human behaviour.

In the 19th century and even at the beginning of the 20th century, bad working conditions and methods, young and inexperienced workers, long working days, and other occupational problems combined with machinery were factors in several accidents (Eves, 2014, chap. 1-2) (Rosner and Markowitz, 1987, p. xi). In the United States alone, a large number of people died or were injured in machinery related accidents during this era (Rosner and Markowitz, 1987, p. xii). For instance, threshers contained, among other potential hazard sources, exposed belt drives (Wendel, 2005, p. 218) (Pripps and Morland, 1992, cover page) introducing nip points (OSHA, 2007, p. 8) that were capable of causing harm with consequences resulting from minor injury to death.

The lack of safety aspect in the history of machines can also be seen in safety regulations. According to Macdonald, the regulations in the United Kingdom considering machines and their use in workplaces did not initially take safety directly into account. The early regulations in the United Kingdom from 1802 considered health and welfare instead of safety. It was as late as 1842 the first safety provisions appeared in UK regulations requiring, for instance, 'fencing of certain type of machines'. (Macdonald, 2004, p. 25).

In the 20th century, the awareness and willingness to take safety aspect into consideration increased. For instance, a steep increase of accident related costs due to new accident 'compensation laws and tighter employers' liability' was a major driver for this development in the United States in the early 20th century (Aldrich, 2001, sec. Employers Become Interested in Safety). Later administrative parties, such as Occupational Safety and Health Administration (USA), Health and Safety Executive (UK), and National Institute of Occupational Safety and Health (Japan), appeared to improve occupational safety and health including safety of machinery. Also, regulation considering the safety of machinery emerged including, among others (HSE, 1956; Machinery directive, 2006; OSHA, 1996).

1.2 Viewpoints on Safety

In this thesis, the focus is on machinery and process systems and the development of such systems. Within the engineering discipline, in which devices, machines, and systems are designed, safety can be considered from different viewpoints. At least the following types of safety can be considered:

- Perceived safety
- Substantive safety
- Normative safety

The perceived type of safety has the strongest relation to Maslow's hierarchy of needs. This type of safety is characterized by how people feel about their situation (Ericson, 2011, p. 76) (Bartneck et al., 2009, p. 339). Do they feel or think they are threatened or not? Perceived safety may relate to, for instance, hostile activity in neighbouring countries, crime rate, the initial consideration of the safety of a vehicle or a machine, or, as stated by Bartneck et al. (2009, p. 76), the level of comfort when interacting with a robot. Perceived safety is something a product designer or manufacturer should seek to fulfil as it has an effect when people determine, for instance, whether or not they want to use or purchase a product or a machine. Perceived safety potentially steers the decisions and behaviour of humans, but this type of safety has little value when the safety of a product or machine is assessed against laws and regulations.

The substantive type of safety takes neither into account whether people feel safe and secure, nor considers if an engineered system has been developed according to the applicable laws, regulations, or standards. Instead, substantive or objective safety only takes into account the actual or expected safety performance of the subject under consideration (Ericson, 2011, p. 339) (Stein and Neuman, 2007, p. 8). That is, has the system under consideration produced harm and if so, how severely and how often. The objective form of safety can in some cases be considered a valid input for safety system development. For instance, according to the Functional safety of electrical/electronic/programmable electronic safety-related systems (IEC 61508), the compliance of an element of a safety-related system with a required Safety Integrity Level (SIL) can be demonstrated, under certain conditions, with documented and justified records, indication sufficient capability of the element in existing applications (IEC 61508-2, 2010, sec. 7.4.2.2 c).

The normative type of safety considers whether or not a system meets the standards applicable in its design (Ericson, 2011, p. 339). That is, if one is able to justify that a system, its development process, functionality, and other assessed aspects conform to the applicable standards, directives, and laws, the system can be considered safe. In the context of this thesis, this type of safety is likely the most interesting from the functional safety system development viewpoint as functional safety systems are typically developed to conform with the normative safety approach. That is, laws, regulations, and standards define the requirements, development processes, methods, and techniques related to the functional safety system development. Such standards include, among others, (IEC 61508, 2010) and (ISO 13849-1, 2015).

1.3 Viewpoints to Design

Design is both an activity and an outcome. The purpose of an activity of design is to produce a design that meets its requirements and defined boundaries. These aspects of design can be defined as follows (Merriam-Webster, 2016):

> (noun) : the way something has been made : the way the parts of something (such as a building, machine, book, etc.) are formed and arranged for a particular use, effect, etc.
>
> (verb) to plan and make decisions about (something that is being built or created) : to create the plans, drawings, etc., that show how (something) will be made

A variety of methods and approaches to design systems and objects have been developed and documented. Jones lists design methods from traditional ones such as craft evolution and design-by-drawing (Jones, 1974, chap. 2) to more modern methods including interviewing users, brainstorming, interaction matrix, checklists, and systems engineering (Jones, 1974, part 2).

Building on this view, individual methods can be seen and used as parts of larger development processes and system life cycle models, including, but not limited to: waterfall model (Royce, 1970), spiral model (Boehm, 1986) (Kossiakoff et al., 2011, p. 103, 370), V-model (Forsberg and Mooz, 1992, cited by Buede, 2011, p. 10) (Stevens et al., 1998, sec. 6.4), agile approaches such as Scrum (Schwaber, 1995), and Rapid Object-Orientated Process for Embedded Systems (Douglass, 1999, chap. 4).

Each time a new design is produced or a problem related to a design is solved, resources are consumed, for instance, in the form of designers' time. Reuse of design artifacts, such as software components and design solutions, can potentially increase the productivity and efficiency of design activities (Boehm, 1999). Design patterns provide a potential way to document solutions and promote design artefact reuse at the solution and approach levels. The effect of patterns can be further enhanced by forming the patterns into a whole referred to as a pattern language.

1.4 Objectives and Scope

1.4.1 Objectives

The objective of this thesis is to compile, as well as to document and assess the research process resulting, a pattern language for the development process of safety systems to provide support for developers in their design task in the context of normative safety. In that context, a designer of a safety system may encounter at least the following situations in which support could enhance design performance.

An inexperienced designer potentially benefits from support considering the whole development process. For such a designer, solutions illustrating centric design decisions throughout the development process of the system are potentially beneficial. The related and potentially sequential solutions described in the language build a framework for the safety system development.

A designer of any experience level will likely encounter problems for which they have no direct solutions or solution models available as result of their experience. It is still likely that solutions for the problem in hand have already been considered and applied in similar contexts. Such solutions are typically able to provide at least a starting point when the actual problem in the considered context is being solved.

The development of safety systems is regulated by standards such as (ISO 13849-1, 2015), (IEC 61508, 2010), and Machinery directive (Machinery directive, 2006). The regulations and the standards help the development process by defining methods, techniques, and requirements considering the development process and the produced safety systems. However, there is a gap between the requirements given by the standards and the final designs that are supposed to fulfil these requirements. Providing existing solutions to operationalize the standards and their requirements into a working design could have a potential impact on the performance of a designer.

The awareness that a solution has been successfully applied in safety systems can support making a design decision of applying the solution again. The atmosphere of the safety system standards may appear discouraging to new and novel solutions as well-tried components and traditional methods and languages are typically recommended. Therefore, the awareness that a (novel) solution has been applied by other companies and designers can support a designer to apply the solution even if it may initially appear to conflict with a standard or regulation. This approach can also support one to assess whether or not a solution or an approach can be seen fit from a standard or regulation viewpoint.

The objective of completing the pattern language introduced in this thesis is to concretize tacit knowledge into an explicit format. Designers have applied certain solutions over and over again, but they may not have noticed it and even if they had, the solution has not been explicitly documented anywhere. The tacit knowledge can also exist in the designs where it resides potentially in an implicit format. That is, the designer has not explicitly marked, justified, or rationalized the solution used to solve a certain problem.

1.4.2 Research Questions

The research questions of this thesis are:

RQ1 **How can design patterns support the development of functional safety systems?**

RQ2 **Is there a set of commonly applied solutions for functional safety system designs utilized and known by domain experts and practitioners?** Which topic categories do the solutions contribute and belong to? What purposes do the solutions serve?

RQ3 **How do the design patterns for functional safety system development relate to each other?** What kind of whole emerges from the identified design patterns? What kind of relationships can be used to describe the relations between the design patterns?

RQ4 **How to document emerging solution models and approaches applied in the field of functional safety system engineering into a design pattern format?** Does the design pattern format fit to document the commonly known solutions? Is there a need for a specialized format or elements to be used for the solutions in the functional safety system domain?

1.4.3 Scope

This thesis considers design patterns for functional safety system design and development in the domains of machinery and process systems. The domain of machinery includes both mobile and stationary machines such as harvesters, passenger hoists, guillotine shears, and benders. The domain of process systems includes, among others, paper machines and distillation processes. Both domains consider machines that potentially introduce hazard sources to the users and other people. The thesis excludes hand-held machines and machine-like objects that are not considered machines according to the Machinery Directive (Machinery directive, 2006).

IEC 61508 and EN ISO 13849-1 standards provide the background for the patterns. That is, several, but not all of the known uses for the patterns have been discovered from systems developed according to either or both of the standards. However, it should be noted that application of some or all of the the patterns does not necessarily result in compliance with the mentioned standards as such. These standards are widely applied in the field of functional safety system development and the standards are applicable to many types of machines. The patterns could be potentially beneficial and applicable outside these standards as well.

A common denominator of the systems in the scope of this thesis is the presence of a SAFE STATE (Eloranta et al., 2014, p. 179). In a safe state, the ability of a system to produce harm is minimized. In many cases, considering the systems in the scope the safe state is a halted or de-energized state. That is, there is no movement or active operation. For some systems, such a state does not exist. For instance, for the winglets of a flying aeroplane, there is no safe state where they could be taken for an undefined period of time.

1.4.4 Related Work

Figure 1.1 illustrates the positioning of the work in the context of the design patterns and pattern languages. This thesis primarily considers the subjects marked in grey in the figure, that is, design patterns on safety systems and software-based safety functions. The remainder of this subsection outlines the work on the topics related to this thesis.

As discussed in [P1], design patterns have gained popularity in the domain of software engineering. The works in this domain cover, for example, object-oriented software (Freeman et al., 2004; Gamma et al., 1995), pattern-oriented architecture (Buschmann et al., 1996; Schmidt et al., 2000), enterprise applications (Fowler, 2002; Hohpe and Woolf, 2003), and service-oriented architecture (Erl, 2009). The mentioned works focus on software engineering from several aspects. The patterns are most likely not written with safety systems in mind, but one is still free to apply the patterns also in the safety system domain if found applicable.

The domains and subjects of distributed control systems, and fault tolerance and reliability are, in a natural way, related to the functional safety systems. Machines and processes controlled by distributed control systems are often potential targets for functional safety systems as well. In such a case, a (distributed) control system controls and operates a machine or a system that may introduce hazards to the users of the system. For instance, a guillotine shear introduces a shearing hazard. Thus, a safety system alongside the control system can be used to mitigate the risk related to the hazard. A pattern language by Eloranta et al. (2014) considers distributed control systems mainly from the software

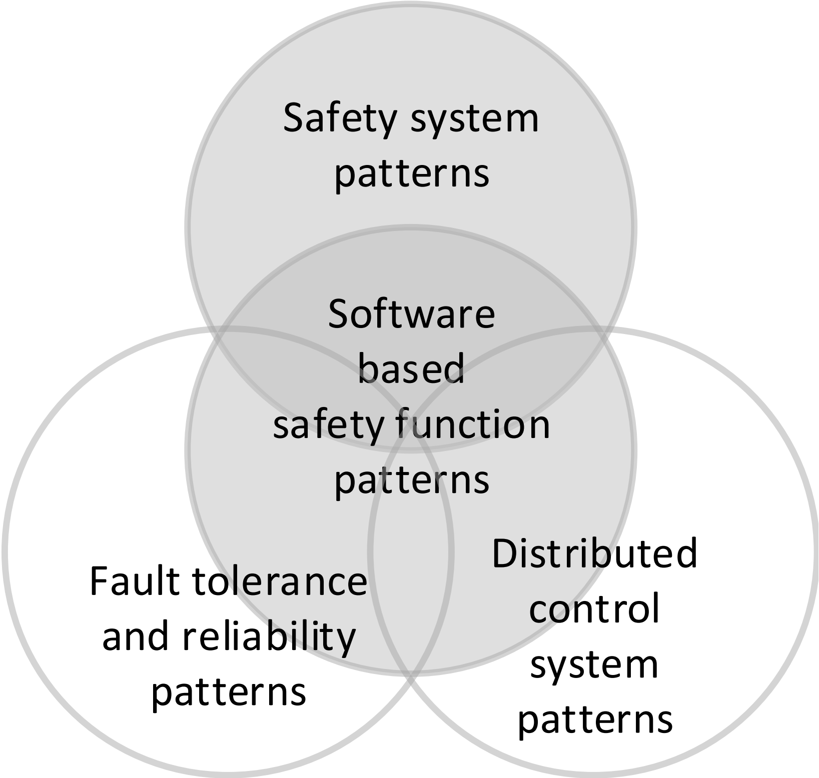

Safety system patterns
Software based safety function patterns
Fault tolerance and reliability patterns
Distributed control system patterns

Figure 1.1: The positioning of the patterns considered in the dissertation

and architecture points of view. The language includes some patterns taking a stand on safety aspects. These patterns such as Safe state and Limp Home provide an anchoring point for the extended safety system and safety function patterns presented in this thesis.

Fault tolerance and reliability are areas from which functional safety systems benefit. Safety systems should be reliable and fault tolerant. In such a case they can produce correct service or operation even when a fault escalates into an error (Avizienis et al., 2004). Another approach to react on the identified faults and errors is to fail safely. In this approach, a system is taken into a safe state when an error is detected (Krishnamurthy and Saran, 2007, p. 16). Patterns and pattern languages by, for instance, Hanmer (2007), Douglass (1999), Armoush (2010) in his dissertation, Alho and Rauhamäki (In press), and Preschern et al. (2015) present patterns for safety-related system design and development. However, in the mentioned design patterns and pattern languages, the focus is on reliability, dependability, and fault tolerance aspects such as redundancy and diversity as well as achievement of these properties with architectural solutions.

Fault tolerance can also be seen from the viewpoint of the resilience of a system. Strigini discusses the concept of resilience and provides multiple interpretations. According to Strigini, in the context of Information and Communication Technology (ICT) systems, traditional approaches on fault tolerance and dependability work well in closed and unchanged systems, whereas resilience approach targets to achieve dependability in open, interconnected and changing systems. The concept of resilience highlights flexibility and management of the unexpected to achieve fault tolerance. (Strigini, 2012, p. 5). In recent discussion occurring in social media, Friedrichsen defines resilience as the ability of a system to recover from erroneous states. He also highlights the availability of the considered system to be achieved by minimizing Mean Time To Repair (MTTR) instead of maximising Mean Time To Fail (MTTF) in context of distributed systems. Friedrichsen approaches resilience from a design pattern perspective. His pattern language introduces patterns to identify errors, restrict their effect, and recover from the errors and failures identified. (Friedrichsen, 2014, 2016) Friedrichsen's patterns can be seen to support the goals of a safety-related system and therefore could be considered providing additional insight also in context of the pattern language presented in this thesis.

In addition to fault tolerance and distributed control, control systems, control engineering, and real-time properties also relate to the work of this thesis. Real-time aspects in safety system development and design emerge from the purpose of a safety function. A safety function executed too late does not achieve its purpose. Works by Douglass (2003), Gomaa (2016, chap. 11), and Zalewski (2001) introduce design patterns for real-time software. Real-time aspects are also present in control systems that implement the algorithms and functions specified by control engineering. Patterns and their usage for control systems and control engineering have been proposed in (Pont, 2001; Sanz and Zalewski, 2003; Zalewski, 2001).

Development process patterns for the IEC 61508-3 (IEC 61508, 2010) based safety system engineering have been proposed by Koskinen et al. (2012). The patterns consider applying and complying with the development process defined in the standard, but lack suggestions for the structural and functional aspects of the safety functions. However, these aspects are considered by (Preschern et al., 2013) in the light of the IEC 61508. The approach utilizes the concept of tactic defined as a 'design decision that influences the achievement of a quality attribute response' (Bass et al., 2012, p. 70).

The scope of this thesis does not cover how patterns are applied in a development process, and neither does it consider any tool support. Instead, Vepsäläinen has studied modelling and tool support of design patterns (Vepsäläinen, 2015, p. 24) and (Vepsäläinen and Kuikka, 2014) and summarized this work also in [P2]. In addition, Preschern et al. (2014) have studied and compared a set of pattern-based approaches and pattern languages used (or proposed to be used) in the design process of safety systems .

1.5 Research Methodology

The research methodology followed in this thesis integrates aspects from both design science research and grounded theory. The design science research methodology is considered as a framework for the whole research to compile and validate a pattern language and the included patterns. The grounded theory methodology is applied within the framework to produce design pattern artifacts, categorize them, and identify pattern relations for the pattern language.

Design science research is a research methodology applied in information systems and information technology. The methodology is described, for instance, by Hevner and Chatterjee (2010); Hevner et al. (2004); Kuechler and Vaishnavi (2008); March and Storey (2008). The foundation of design science lays in the engineering and science of artificial (Simon, 1996, cited by Hevner et al., 2004). Design science is essentially a problem-solving paradigm, which is based on designing, building, and applying artifacts. Design science produces knowledge and understanding about a problem through building and applying an artifact that is produced as a part of the research. (Hevner et al., 2004). These artifacts, resembling human-made constructs (Simon, 1996, cited by Hevner and Chatterjee, 2010), can be constructs, models, methods, instantiations, and better design theories (Hevner and Chatterjee, 2010, p. 6).

Building and evaluating the artifacts are the basic activities of design science. The building phase produces an artifact, the performance of which is measured in the evaluation phase to judge the suitability of the artifact (March and Smith, 1995). The framework of the design science research process as extended by Kuechler and Vaishnavi (2008) includes the following phases as described by Piirainen and Gonzalez (2013).

1. problem awareness
2. finding suggested solutions
3. solution artifact development and testing
4. artifact performance evaluation
5. conclusions and result communication

In this thesis, a design science approach has been applied to construct design patterns and a pattern language as the artifacts considered in design science research. A design pattern approach (see Chapter 3) forms a framework in which the research results are realized and documented. In addition to a usability view of design patterns, a scientific view of design science and grounded theory has been exploited. An evolving pattern mining process has been applied through the research (see Section 5.2 for the initial, 6.2 for an evolved, and 7.2 for the final pattern mining approach). The following list maps the research actions carried out for this thesis to the design science research phases listed above.

1. The first phase of the research process was realized partially in research projects considering control system and safety-critical software development. This work provided insights into the problem area of safety-critical software and control system development. Phase one continued in the actual pattern research phase through seeking problems from standards, literature, and discussions and interviews with industry representatives.

2. Phase two co-occurred partly with phase one. The material acquired in the previous phase also contributed to identifying suggested solutions. Drafts of the solutions were constructed as design pattern draft artifacts.

3. In phase three, the solutions were further developed with feedback from the pattern community, industry, and colleagues. In this phase, another artifact, namely the pattern language, was constructed and developed.

4. In the context of the thesis, the fourth phase can be seen as realized in a set of industry representative interviews and workshops, where the developed artifacts were identified, discussed, criticised, and enhanced. The evaluation of independent pattern artifacts was gained in this phase by identifying known uses for the patterns.

5. The fifth phase of the research is realized in this thesis where the results are communicated and concluded.

Grounded theory is a qualitative research method that has it roots in social sciences (Glaser and Strauss, 1967, p. 1). Research applying grounded theory does not have to begin with questions and hypotheses (Glaser and Strauss, 1967, p. 6&33). Instead, the theories and hypotheses should come from data regarding the subject under research (Glaser and Strauss, 1967, p. 2&3). The theories emerge from data through coding, categorization, and their relations, which are identified during data analysis (Glaser and Strauss, 1967, chap. V) According to Hentrich et al. (2015) Glaserian grounded theory can be applied to pattern mining. Hentrich et al. also propose a process in which the concepts of grounded theory are mapped into a pattern mining process.

In the context of this thesis, the ideas of grounded theory have been primarily applied to identify individual patterns. In the final pattern mining process (see Section 7.2) data was collected from companies without a hypothesis. Although the interviews were semi-structured with a prepared set of questions, no presumption on the results was (intentionally) made, excluding the situations where pre-existing design pattern ideas were shown to the interviewees. The researchers participated in the interviews in the role of interviewers who initiated the discussion and asked prepared questions and follow-up questions reacting and reflecting on the initial answers. The purpose of this was to make the interview a conversational event and to enable adaptation to the expertise of the interviewees.

The interviews did not strictly follow the process described in (Hentrich et al., 2015). The interview questions were not directed to probe dedicated parts of patterns. Instead, the solutions and approaches applied were the primary subjects of interest although the underlying problems, consequences, forces, and relations between the solutions were documented if any were discovered in the interviews.

After the collection phase, the data was analysed to find potential pattern candidates and new known uses for the existing patterns and pattern candidates. This part of the process followed the analysis phase described by Hentrich et al. (2015) rather closely. Interesting findings were formed into sections of patterns including problem, solution, force, or consequence statements, which resembled the codes in the grounded theory. A set of codes produced a concept which emerged in the form of a pattern. Typically, some parts of the patterns needed to be augmented by the researchers to produce a complete pattern as some aspects of the patterns were not always brought up during the interviews. Nevertheless, new patterns emerged as a result of the analysis and subsequent discussions with other industry representatives. Finally, the categories of grounded theory emerged in the form of a pattern language and categorization in the final phases of the research, thus giving the basis for answering the research questions one, two, and three. The fourth research question (of the appropriate pattern format) will be solved based on the various phases of the research process, outlined next in connection with a discussion on design science and grounded theory.

1.6 Contributions

The scientific contribution of this thesis is the following:

Rationale for the Usage of Design Patterns in the Engineering of Safety Systems This contribution describes potential benefits of the application and usage of design patterns in the context of safety system development. Some of the benefits can be considered applicable to design patterns in general and regardless of the application domain. Others, such as the possibility to bridge the gap between the requirements given by standards and their fulfilment in the designs, are more likely to provide benefit in the safety system domain. This contribution is presented in [P1] and [P2] and summarized in Chapter 4.

A Collection of Design Patterns for the Safety System Domain This contribution presents the design patterns and pattern candidates to be used and applied in the design and development of safety systems. The patterns target specifically machinery and process control systems, but they can also be applied in other domains if the context is considered sufficiently similar. The patterns have been collected using the pattern mining processes described in sections 5.2, 6.2, and 7.2. The contribution is presented in Chapters 5, 6, and 7 and their respective publications [P3]...[P8] and (Rauhamäki and Vepsäläinen, 2016).

A Pattern Language for Safety System Development This contribution forms a pattern language of the aforementioned pattern collection. Individual patterns are useful for solving individual problems. However, a pattern may also introduce a set of new problems or benefit from the application or existence of other patterns. Therefore, an illustration of the pattern language emerging from the relations between the individual patterns has been compiled. The pattern language can help a user of the patterns to navigate through the patterns, for instance, to see alternative solutions and consider next patterns. The pattern language is compiled in Chapters 5, 6, and 7.

Categorization of the Patterns and Their Respective Solutions for Safety System Development This contribution is targeted to serve users of the pattern collection. Categorising the patterns according to their topic and purpose helps the users to identify potentially suitable patterns for the problem in hand more quickly as there is supplemental information available on the patterns other than their names. On the other hand, categorizing according to the purpose of the patterns helps to identify potentially suitable patterns to achieve a certain outcome by applying a pattern. The categorizations are described and shown in Chapters 5, 6, and 7.

1.7 Organization of the Thesis

The thesis is organized as a retrospective presenting the research process parts and results obtained in the different parts of the process. The purpose of this is to illustrate the evolution of the research approach, especially in terms of the applied pattern mining process.

The thesis is organized into three main parts: background, result, and discussion and conclusion chapters. The background parts consist of Chapters 2 and 3. Chapter 2 introduces the concepts of safety and safety systems and discusses the development of functional safety systems. Chapter 3 provides an introduction to design patterns and pattern languages. The chapter also considers the representation of patterns in general and how they are formatted in this thesis. Table 1.1 summarizes the publications included in this thesis and the chapters of this thesis where they are discussed.

Table 1.1: The included publications and their appearance and use in the thesis.

Publications	**Thesis Chapter**
[P1] and [P2]	4: On Design Patterns Supporting Safety System Development
[P8] and [P3]	5: Control Systems, Safety Systems, and Their Coexistence
[P4], [P5], [P6], and [P7]	6: Hazard Management Process and Risk Reduction
[P3], [P5], [P7], and [P8]	7: Functional Safety System Development

Chapter 4 summarizes the potential benefits of applying design patterns in functional safety system development and introduces and justifies the structure of the result chapters. Chapters 5, 6, and 7 present the main contributions of the thesis. These chapters define the pattern language and pattern categorization in the related publications. Chapter 5 summarizes patterns for control and safety system development and co-existence. Chapter 6 summarizes the patterns considering a hazard management process. The patterns in this section discuss the development process of a safety system, introduce risk and hazard mitigation methods, and consider the human role in the safety system operation. Chapter 7 summarizes patterns considering the design and development of functional safety systems and augment the pattern categories introduced especially in Chapter 5. Chapter 8 revisits the research questions and discusses both the validity of the research and the potential future work possibilities in terms of open questions.

2 Functional Safety Systems and Their Engineering

This chapter provides backgrounds to safety, safety(-related) systems, and their engineering. Especially the engineering part differs to some extend from the development of a system that has no specific requirements regarding safety.

2.1 Definitions and Terms on Safety Concepts

As the scope of this thesis is on machinery and process applications and their normative safety, the viewpoint to the concepts and definition is adopted from normative safety, which, as illustrated in Section 1.2, differs from perceived and substantive viewpoints by its focus in standards and regulations.

In the context of this thesis **safety** is: 'freedom from unacceptable risk' (IEC 61508-4, 2010, sec. 3.1.11), where **risk** is the: combination of the 'probability of occurrence of harm and the severity of that harm' (IEC 61508-4, 2010, sec. 3.1.6) and **harm** is a 'physical injury or damage to the health or property or the environment' (IEC 61508-4, 2010, sec. 3.1.1).

The definition of risk includes the attribute unacceptable. Consequently, there exists a **tolerable risk**, which is defined as a: 'risk which is accepted in a given context based on the current values of society' (IEC 61508-4, 2010, sec. 3.1.7). This indicates that, in some cases, a residual risk remains regardless of the actions taken to mitigate the risk. For instance, modern cars introduce many approaches to mitigate the risk of severe injury or death in a car accident, but these approaches do not completely remove the risk. Still, most of the people consider the risk tolerable and use cars in traffic.

In summary, to achieve safety one needs to mitigate risks into an acceptable level by affecting either the likelihood of occurrence or the severity of harm. From the perceived safety viewpoint, the sense or feeling of an acceptable risk could suffice. However, from the normative safety viewpoint, there needs to be justified indication that risks have been mitigated to an acceptable level conforming to the relevant regulation.

2.2 Risk Mitigation Approaches

In practice, there are several ways to achieve reduction of a risk. However, regardless of the selected mitigation approach, the risks need to be known, and only then can they be justifiably mitigated.

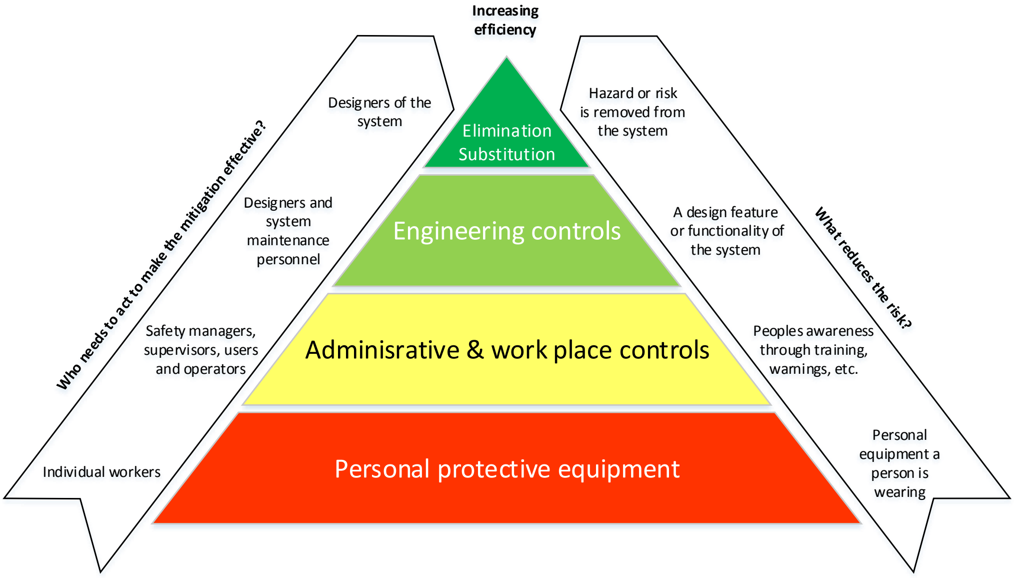

Increasing efficiency
Elimination Substitution
Engineering controls
Adminisrative & work place controls
Personal protective equipment
Who needs to act to make the mitigation effective?
Designers of the system
Designers and system maintenance personnel
Safety managers, supervisors, users and operators
Individual workers
What reduces the risk?
Hazard or risk is removed from the system
A design feature or functionality of the system
Peoples awareness through training, warnings, etc.
Personal equipment a person is wearing

Figure 2.1: Hierarchy of controls. Illustration inspired by (Manuele, 2005; Nix, 2011).

The hierarchy of controls introduces the basic approaches to risk mitigation. Figure 2.1 illustrates the approaches indicating the effectiveness of the approach, the way the risk is reduced, and the main persons and/or roles involved to enable the success of the approach.

The most effective way to reduce risk related to a machine, process and their functions and operation is to apply inherently safer design, that is, to eliminate or substitute the hazard(s) under consideration. This approach

> seeks to remove the hazard at the source, as opposed to accepting the hazard and attempting to mitigate the effects (Center for Chemical Process Safety, 2012, p. 123).

From the system manufacturer point of view, this approach is beneficial as the manufacturer has a high level of control over the risk mitigation. Regardless,

> an inherently safer process should not, however, be considered “inherently safe” or “absolutely safe.” While implementing inherently safer concepts will move a process in the direction of reduced risk, it will not remove all risks. (Center for Chemical Process Safety, 2012, p. 128)

Not all machines and processes can be made inherently sufficiently safe by eliminating or substituting hazards. In such cases, engineering controls also known as **protective measures** as defined in (ISO 12100:2010, 2010, sec. 3.19), can be applied to provide additional risk mitigation to make the risks tolerable. The protective measures include both passive and active methods of reducing the risk. In this thesis, we primarily consider active protective measures. In a protective measure approach, the considered hazard remains in a system, and some of the control over the mitigation is transferred from the system designer or manufacturer to the user of the system. For instance, engineering controls such as guards and Electrical/Electronic/Programmable Electronic (E/E/PE) safety-related systems need to be maintained and inspected periodically to ensure their desired operation.

Finally, risk can be mitigated by administrative controls and Personal Protective Equipment (PPE). In these approaches, the control over risk mitigation is effectively transferred from a designer of a system to the users and user organizations of the system. Consequently, the effectiveness of these controls are lower compared to inherently safer design and engineering controls, as the administrative controls and PPE affect a smaller number of people. For instance, training only affects the ones who have been given the training (although this knowledge can be shared), and PPE only affects the ones who wear the required PPE.

2.3 Functional Safety Systems

Functional safety is one, but not the only, way to achieve the aforementioned state of safety. According to IEC, **functional safety** is:

> the part of the overall safety that depends on a system or equipment operating correctly in response to its inputs and the detection of a potentially dangerous condition resulting in the activation of a protective or corrective device or mechanism to prevent hazardous events arising or providing mitigation to reduce the fight consequence of the hazardous event. (IEC, 2015)

This thesis considers the development and design of **safety-related systems** and **safety-related parts of control systems** that are used to implement safety functions with the purpose of achieving functional safety. The terms are given in (IEC 61508-4, 2010) and (ISO 13849-1, 2015) respectively with the following definitions:

> **safety-related system**: designated system that both implements the required safety functions necessary to achieve or maintain a safe state for the Equipment Under Control (EUC); and is intended to achieve, on its own or with other E/E/PE safety-related systems and other risk reduction measures, the necessary safety integrity for the required safety functions (IEC 61508-4, 2010, sec. 3.4.1)
>
> **safety-related part of a control system**: part of a control system that responds to safety-related input signals and generates safety-related output signals (ISO 13849-1, 2015, sec. 3.1.1)

For consistency reasons regarding the publications included in this thesis, the terms are combined under the **functional safety system** and **safety system** terms. That is, in this thesis, **functional safety system** is considered to be a system conforming to and implementing one or both of the above definitions.

A functional safety system is, as it emerges from the definition in Section 2.2, inherently an active system (IEC, 2015). An active system is one that affects the operation of the considered target system to achieve its purpose. In the context of functional safety, an active system (as functional safety system) would affect the operation of the system under control to achieve safety.

2.4 Safety System Development

In European Union (EU), the Machinery Directive regulates machinery safety by declaring the Essential health and safety requirements (EHSR), which are targeted to ensure the safety of machines. More detailed descriptions of design and use are given in the harmonized standards. One does not necessarily need to comply with the standards, but if a harmonized standard is complied with, the associated EHSRs are considered fulfilled. (Rausand, 2014, p. 15).

The harmonized standards in the field of machinery safety include (European Commission, 2016), among others, (ISO 13849-1, 2015) and (IEC 62061, 2005). In addition, (IEC 61508, 2010) also considers functional safety system development, but does not hold a harmonized standard status. However, (IEC 61508-3, 2010) and (IEC 61508-4, 2010) are normatively referenced from (ISO 13849-1, 2015, chap. 2).

The standards introduce requirements considering the development of functional safety systems. These requirements consider, for instance, the development process (see Section 2.6), the methods and techniques to be applied, and the architecture of the safety system. A manufacturer willing to comply with a standard needs to justifiably show that the methods and techniques are suitably applied (or provide a rationale why they have not been applied) and be able to show that the requirements have been fulfilled. The purpose of this is to provide the manufacturer itself and an assessing party with sufficient confidentiality that the requirements of the regarded standards and regulation are met. To strengthen the confidence, communication between the manufacturer and the assessing party may be initialized already in the development process.

Although the standards provide requirements for the development, they leave room for creativity to fulfil the requirements in the designs and development processes. This introduces a potential problem to manufacturers and developers. Emerging questions may include for instance:

- Can I justifiably use design approach X? Does it cover the requirements given by the standard?

- In what way do I apply a software method or technique? How can it be applied sufficiently well considering the safety requirements?

- How do I manage the hazard and risks? How does the safety system co-exist and operate with a control system? What kind of solutions could benefit functional safety system development?

2.5 Functional Safety System as a Part of the Overall Control of a System

Typically, the main element operating the system under control is referred to as a **control system**. The purpose of a control system is to operate and control the EUC to implement the regarded functionality and outcome. A control system is

> an interconnection of components forming a system configuration that will provide a desired system response (Dorf and Bishop, 2005, p. 2).

In the context of this thesis, a control system can be centralized or distributed. In a centralized control system, all the control logic or software resides in a single processing or logic unit. Another form of a control system is a **distributed control system** where the control logic has been distributed in many interconnected units which thereby only execute a subset of the control loops of the system (Center for Chemical Process Safety, 2010, p. 264). In larger machinery applications, the latter case can be considered a more typical approach.

The definition of the desired system response could also include the idea of safety. That is, a desirably operating machine or process should retain the safety of the people that operate around it. However, safety systems and functions can be and have traditionally been separated from the control system (Sueur and Knobel, 2014, p. 2). Also International Electrotechnical Commission (IEC) acknowledges this approach (IEC, 2016). In this approach, a separated Safety Instrumented System (SIS) exists aside a control system. Currently also integrated control and safety system approaches are becoming available (School and de Groot, 2012). In the integrated approach, the control and safety systems are integrated from the user viewpoint but not necessarily implemented in a single common system. Still, regardless of the integration, the requirements for their development are distinct.

Regardless of the implementation approach, in both cases the purpose of a safety system or function is to reduce the risks to a tolerable level. In addition, safety functions typically do not participate in the productive system control functions. These tasks and functions are left for the control system and include, among others, the feedback control of a motor, the pressure control of a tank, and the path control of a robot arm. Still, both of the systems operate and affect the same system under control. This induces the need to make control and safety systems co-operate at least on some level so that the overall functionality of the system is desirable. An example of safety and control system co-operation and existence is given in the introduction of [P3]. Patterns considering the co-operation and co-existence of control and safety systems are discussed in Chapters 5 and 7.

2.6 On Safety System Development Processes

Figure 2.2 gives an overview of functional safety system development found on the (ISO 13849-1, 2015) and (IEC 61508, 2010). The thesis introduces design patterns considering the underlined parts of the process. The patterns are discussed in Chapters 5, 6, and 7.

The process begins with the definition of the concept and scope of the system to be considered. This includes the boundaries of the system and the system's assessment. In the next step, risk and hazards are assessed to provide a foundation for the risk mitigation work. The system safety requirements are defined and allocated reflecting the risk assessment results. In this phase, the risk mitigation methods are selected. For the scope of this thesis, the mitigation method is primarily considered to be a functional safety system.

The following part of the process is the development of the specified safety functions, that is, the functional safety system which implements the functions. This part of the process begins with the safety function requirement specification to define the function to be developed. Then, the hardware and software for the function are developed. The level of co-operation between hardware and software developers may vary. Regardless, the hardware and software are finally integrated into a functional element, and the performance of the safety function is validated.

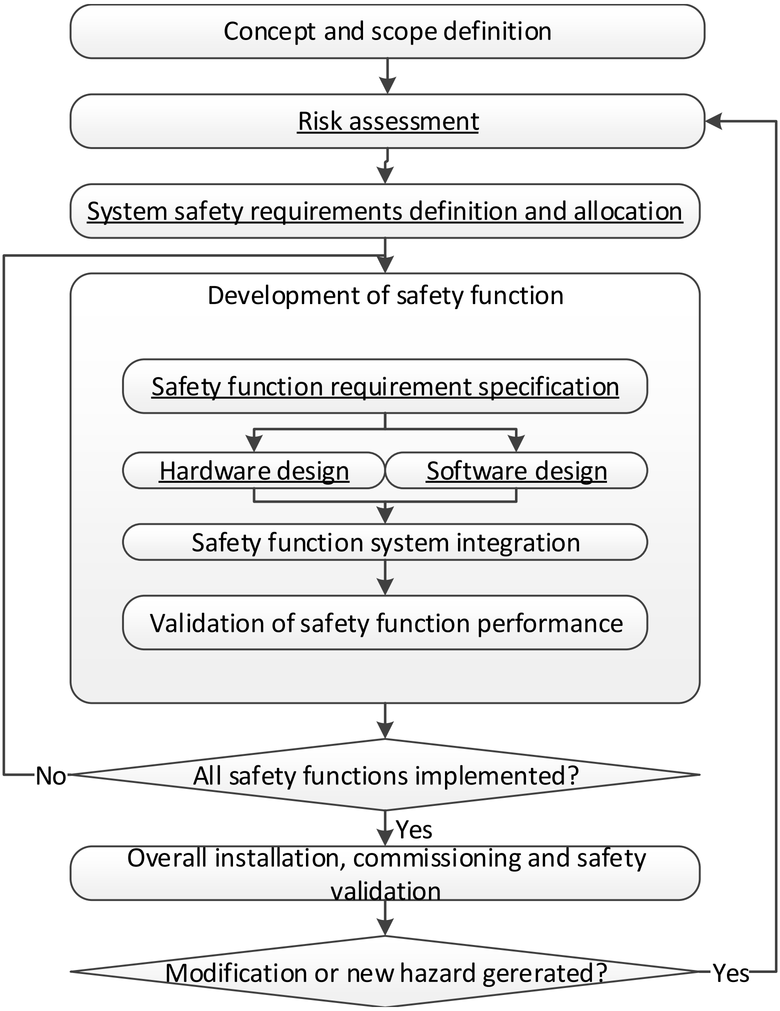

Concept and scope definition
Risk assessment
System safety requirements definition and allocation
Development of safety function
Safety function requirement specification
Hardware design
Software design
Safety function system integration
Validation of safety function performance
All safety functions implemented?
No
Yes
Overall installation, commissioning and safety validation
Modification or new hazard gererated?
Yes

Figure 2.2: A simplified process for safety system development according to (ISO 13849-1, 2015) and (IEC 61508, 2010), derived from [P2].

The development process is carried out for any subsequent safety functions. When all the functions have been developed, the complete system installation, commissioning, and validation can take place. If modifications are needed to the safety system, new hazards may emerge or risks may change. Thus new risk analyses and assurance cases are needed, which again can be based on the design patterns and standards referred by the design patterns of this thesis. Consequently, the process reverts to the risk assessment phase to make appropriate changes to the safety system.

3 Design Patterns

In this chapter, the concept of design patterns is introduced from generic and safety system specific perspectives. The chapter begins with the generic viewpoint and converges into more safety system related aspects at the end of the chapter. The purpose is to present design patterns and pattern languages as a framework, in which design artifacts are produced using the methodology described in Section 1.5.

3.1 Brief History of Patterns

The concept of patterns originates from the field of architecture. The book 'A Pattern Language: Towns, Buildings, Construction', by Christopher Alexander (Alexander et al., 1977) is typically seen as the starting point of patterns as they are considered in the design pattern community. The patterns by Alexander consider, among others, architecture, urban design and community building (Alexander et al., 1977, p. xix-xxxiv).

Thus, the origins of patterns reside in the domain of architecture, and it took some time before the concept of patterns was adopted into other domains of engineering. Eventually, in the late 1980s, the surge was sparked in the domain of software engineering as a result of the work by Beck and Cunningham (1987) presented in OOPSLA'87 (Eloranta et al., 2014, p. 80). This trend was advanced by Erich Gamma in his dissertation (Gamma, 1992, cited by Gamma et al., 2002) and especially the succeeding book 'Design patterns: Elements of Reusable Object-Oriented software' by Gamma, Helm, Johnson, and Vlissides (also known as Gang of Four (GOF)) (Gamma et al., 1995).

Since the 1990s design patterns have been identified and documented in various domains including, but not limited to, security (Schumacher et al., 2005), embedded systems (Douglass, 2010), teaching (Köppe, 2013), learning (Iba et al., 2009), cooking (Isaku and Iba, 2015), and business (Kelly, 2012). The variation of domains where patterns have been applied supports the idea of design patterns as a generic approach to document the solutions and practices of any domain.

3.2 What are Design Patterns and Pattern Languages?

What is a design pattern? Various definitions have been given. Alexander defined:

> Each pattern is a three-part rule, which expresses a relation between a certain context, a problem, and a solution. (Alexander, 1979, p. 247)

This definition represents one view on patterns and captures the concept of what is generally considered the core of a design pattern. Eloranta et al. provide a similar definition:

> A pattern is in essence a design solution to a recurring problem in a specific context. (Eloranta et al., 2014, p. 80)

Both of the definitions share a similar approach, where the lack of recurrence of the solution in Alexander's definition seems to be the main difference. However, according to Coplien, Alexander's definition is further explained after the definition as follows:

> As an element in the world, each pattern is a relationship between a certain context, a certain system of forces which occurs repeatedly in that context, and a certain spatial configuration which allows these forces to resolve themselves.
>
> As an element of language, a pattern is an instruction, which shows how this spatial configuration can be used, over and over again, to resolve the given system of forces, wherever the context makes it relevant. (Alexander, 1979, p. 247)

The further explanation introduces more centric aspects of patterns. Firstly, application of a pattern does not necessarily lead to an identical outcome each time. Instead, depending on the context (the system and its environment) where a pattern is applied, the applier, and other aspects, the result may vary to some extent, but the essence of the resolution stays identifiable. Consequently, a design pattern typically provides a framework of the solution, leaving some of the details to the applier of the pattern.

Secondly, the explanation introduces the concept of language. A **pattern language** is a set of related patterns that form a whole considering the domain or subject of the patterns. Buschmann et al. state the following about pattern languages:

> A pattern language defines a network of patterns that build on one another, typically a tree or direct graph, so that one pattern can optionally or necessarily draw on another, elaborating a design in a particular way, responding to specific forces, taking different paths as appropriate. (Buschmann et al., 2007, p. 13)

The purpose of such a language is to cover a broader domain or area of interest than a single pattern that typically only considers a specific problem. Here, the application of one pattern may introduce new problems that need to be solved. In addition, many (practically all) domains naturally introduce multiple problems to be solved before a functional whole can be reached.

In a pattern language, the patterns can be related to each other with various relationships. Different languages utilize different relationship types. Presumably, the most typical kind of relationship in a pattern language is the 'consider or apply next' type of relationship. This relationship indicates a potential or recommended application order of the patterns so that the patterns build on each other. That is, when one pattern is applied, another one can be or should be considered as it potentially resolves the emerged problems caused by the previous pattern. This type of relationship is applied, for instance, by Eloranta et al. (2014, sec. 4.2), Buschmann et al. (2007, p. 12-15), Noble and Weir (2001, p. 17) and Hanmer (2007, p. xiii, 34-35).

There have been discussions on whether or not a pattern language should cover its domain (or a defined part of it) completely or not. A similar concept **pattern collection** does not assume the collection to represent a whole. Hanmer discusses the aspects by stating the following:

> A pattern language is morphologically complete if the solution space that it describes has no gaps that are not addressed. (Hanmer, 2007, p. xiii)

Here, a morphologically incomplete pattern language is paralleled to the concept of pattern collection. However, it is rather difficult to justify the absence of gaps in the solution space of a pattern language. Thus, in the context of this thesis, a pattern language is referred to without the assumption of a whole or a morphological completeness. That is, the relationships between individual patterns supported by the categorization of the patterns are taken as a pattern language.

In summary, design patterns can be seen as nuggets of wisdom (Kohls and Panke, 2010). They contain documented solutions to recurring problems in defined contexts. Patterns tend to be brief (to earn the nugget title), but the length varies from a couple of lines to dozens of pages. Multiple design patterns can compose a pattern language, where the solutions add up as the patterns of the language are applied sequentially.

3.3 What Makes Pattern a Pattern?

The nature of design patterns is not to present new ideas. Instead, design patterns consider existing and applied solutions and ideas. The solutions and ideas are (or at least should be) known as they have been applied by designers or seen in existing systems.

However, a pattern can be applied without ever noticing it. In such a case, a designer uses tacit knowledge in the form of ideas, experience, and personal skills (Chugh, 2015) (Nonaka and Takeuchi, 1995, p. 60) potentially alongside some explicit knowledge (Nonaka and Takeuchi, 1995, p. 61) to solve a problem. Now, a design pattern discovered and documented can turn this tacit knowledge into new explicit knowledge. This approach takes a step towards a scientific approach and according to Kohls and Panke:

> If science is about the nature of things, then patterns certainly belong to science. In the case of patterns, the things or objects of consideration are artefacts and practices of creating the artefacts. Hence, patterns are a way to investigate the "science of the artificial" (a term coined by Simon (1969)), or the nature of artificial objects. (Kohls and Panke, 2010)

The validity and justification of the existence of design patterns lie in their known uses. In the pattern community, a pattern has been traditionally considered valid only after three independent known uses have been discovered (Holzner, 2006, p. 282) (Kohls and Panke, 2010)(Eloranta et al., 2014, p. 88). The rule of three known uses is also applied in this thesis. That is, the patterns with three known uses are considered **patterns**. If a solution or approach lacks at least three known uses, it is called a **pattern candidate**.

The process of publishing and discussing patterns in dedicated Pattern Language of Programs (PLoP) series conference is seen as valuable from the pattern credibility and quality viewpoints. In these PLoP conferences, patterns are set out for criticism in a two (or three-phase) process. In a PLoP conference, a paper and the included patterns are first inspected by the program committee or a similar entity[1]. Then, the patterns and the paper are enhanced with the help of and under the supervision of a person who is an experienced pattern author. This process is called shepherding (Harrison, 1999). However, the role of a shepherd is not to review the paper as such but to help the author to improve it. Finally, the paper is discussed in a Writers' workshop (Gabriel, 2002) where a group of peers discusses the paper and the included patterns. The target of the discussion is to provide the author of the paper with input to improve the paper. For the pattern papers published in PLoP conferences, this process integrates as a part of the overall pattern quality assurance process.

3.4 Design Pattern Formats

Design patterns are typically documented using a dedicated format. Especially, the patterns by a specific author or patterns belonging to a specific pattern language, collection, or publication unit, such as a book, often use a uniform format to help the reader to understand and follow the patterns.

There are many formats available that differ slightly (Fowler, 2006). The various formats promote various aspects and elements. Ideally, each pattern format has been developed for a certain purpose. That is, the format has been selected to communicate the message of the pattern as clearly, understandably, and efficiently as possible. Another possibility is that the wide selection of formats is a result of divergent likings of various pattern authors. According to Rising (1998a, p. 85), Deugo et al. (1999, slide 19), Fowler (2006), and Appleton (2000), typical pattern formats include among others: AGCS, Alexandrian, GOF, Canonical, Coplien, Portland, and Pattern-Oriented Software Architecture (POSA) formats.

A pattern format may consider the elements of a pattern (Appleton, 2000), the semantics of text formatting, and structuring of a pattern (Rising, 1998a, p. 85). The elements of a pattern represent the section titles that are considered in a pattern. These section titles indicate what information regarding the pattern is located under each section. The structural semantics and formatting consider the visual style of the pattern. These aspects include, among other things, the style of the pattern description. Is it narrative or more structured under defined sections? In addition, the visual formatting may consider the semantics of various highlighting approaches. For instance, SMALL CAPS TEXT often indicates a pattern name when used within the description of a pattern.

According to Appleton (2000), the Canonical format takes the shared (or typical) elements from various other formats. According to Fowler (2006), the Canonical format is a synonym for the Coplien format, which only considers a subset of elements compared with Appleton's listing. In this thesis, Appleton's view is considered the starting point.

[1] The practices vary between the conferences and organizers.

3.5 The Pattern Format Used in this Thesis

The pattern format utilized in the patterns considered in this thesis represents a variation of the Canonical format (Appleton, 2000). The format divides each pattern element under a distinct section. The intelligibility of the format was one of the main reasons to select it initially alongside the fact that it was also the format through which the design patterns were initially introduced to the author. The applied pattern format has slightly changed over the time. The variations have included the additions and removals of single pattern elements.

An overview of the elements of the applied format and their relationships is given in Figure 3.1. The core of the pattern format includes context, problem, and solution elements. These elements are supplemented with forces that discuss and bind the core elements together. A solution induces consequences that often indicate new problems and define new contexts in which new problems need to be solved. The examples and known use elements ground a solution to more intuitive and concrete environments. Related patterns help the reader to navigate in the pattern language by pointing out other relevant patterns to be considered.

The definitions of the elements as considered in this thesis are the following:

Context The boundaries and the environment where the pattern is considered applicable.

Problem An issue to be solved emerging from the context.

Forces Aspects to be taken into account when solving the problem in the context. The purpose of forces is multidimensional. Firstly, they refine the problem by providing insights into the problem. Secondly, they relate to the context and nominate the aspects that are preferred over their counterparts or show which aspects are contradictory in terms of the solution. Thirdly, the previously mentioned aspects direct the solution. In many cases, several solutions could be considerable, but the defined forces direct the solution to the one given in the pattern.

Solution An approach, structure, functionality, or a combination of these that resolves the forces in the context and introduces a framework, that can be applied in the design, which typically requires further refinement to be applied in the context of the system under development.

Consequences A collection of positive, negative, and sometimes neutral effects on the system under design resulting the application of the pattern. The effects can be used to judge whether or not a pattern should be applied. That is, they can be used as a basis for decision making.

Example A case where the pattern can or could be applied. The purpose of the case is to ground the potentially abstract solution to a more easily approachable context. The case is not necessarily from a real world application.

Known uses A case where the pattern has been applied. The case is identified from a real world application, documentation, literature, or designer interview.

Related patterns Patterns that can be considered alongside the current solution. These solutions indicate, for instance, patterns that specialize, conflict, or provide an alternative approach to the current solution.

Some elements that are used previously or rarely in the patterns include:

Intent A short introduction of the pattern including the core solution statement supplemented with a short example case. This element was used in [P5].

Resulting context The environment that exists after the application of a pattern. In practice, the element defines a new context in which new problems may emerge. This element appeared in the early patterns and it can be seen, for instance, in (Rauhamäki et al., 2012).

Related standards A listing of standards or sections of standards that the solution may comply with, support, or conflict with. This element appeared in (Rauhamäki and Vepsäläinen, 2016).

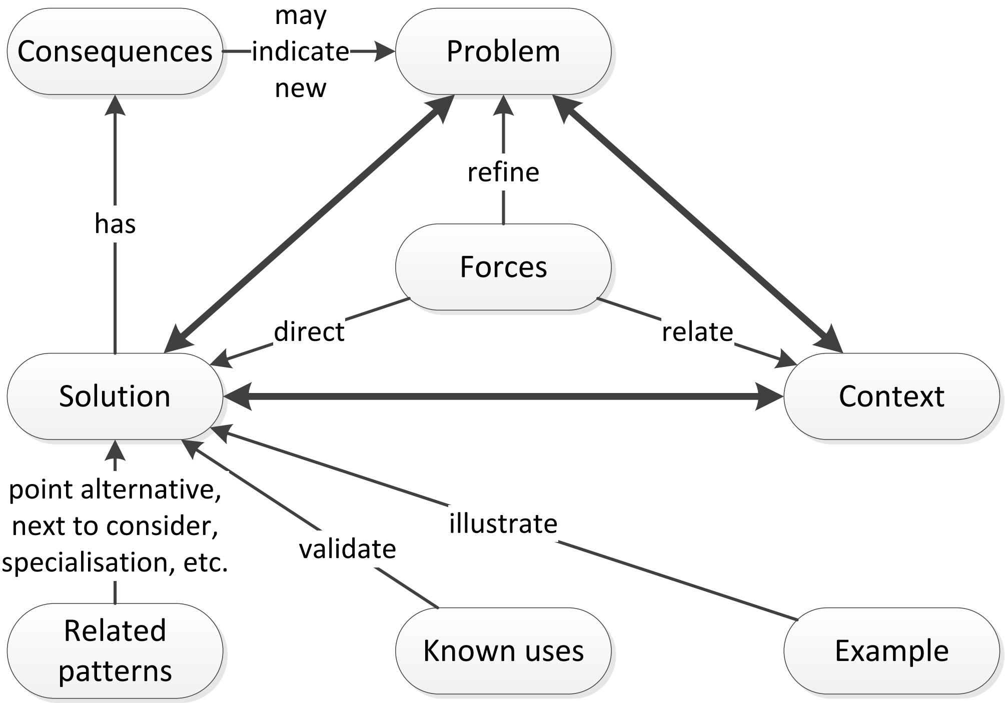


Figure 3.1: The pattern framework utilized in pattern documentation in the context of this thesis illustrating the centric elements of a pattern: context, problem, and solution augmented with the supporting elements and their relations. Derived from [P2].

4 On Design Patterns Supporting Safety System Development

This short chapter summarizes the potential support that application of design patterns may introduce in functional safety system development. In addition, the structure of the following result chapters is briefly presented and justified at the end of the chapter.

4.1 Why Apply Design Patterns in Safety System Development?

Arguably the design pattern approach is not the only one that can be applied in the design of safety systems. However, the usage of design patterns can support the design and development of functional safety systems. The support can be seen to cover all the phases of the development process including requirements specification, design, implementation, testing, and validation phases. Some of the patterns address only one of the phases whereas others influence several. The supporting factors form the rationale to mine, document, and apply design patterns in the domain of safety systems and their development.

Publications [P1] and [P2] discuss the potentially achievable improvements of applying design patterns in functional safety system development and design. Some of the potential benefits can be considered generic to the design patterns of any domain whereas others are more specific to the functional safety system domain.

The main findings are summarized in the following sections. The findings are based on the mentioned publications and they are augmented, when applicable, with findings and experiences from the discussions and comments acquired during the interviews and workshops of the DPSafe project.

Well-Tried Solutions

Standards considering functional safety system development including (IEC 61508-3, 2010; ISO 13849-1, 2015) can be considered preferring well-tried solutions over novel ones. This emerges, for instance, in Table B.2 of 61508-2:2010 and Section 6.2.4 ISO 13849-1:2015. This is understandable as the components, solutions, approaches, and methods that have demonstrated suitability and reliability in the past safety systems and their development processes have already gained the empirical experience, which novel approaches lack.

When mined using an appropriate process, design patterns and the solutions they present can be seen to reflect the concept of well-tried components. A design pattern is typically considered forming when it has been identified three times from independent sources. It is acknowledged that the known uses of a pattern do not suffice or comply with the concept of a well-tried solution or component mentioned in the standards. It is possible that an approach with poor suitability in the context of functional safety systems has been applied widely. This is especially possible if the patterns have been mined and documented by researchers alone without feedback and refining discussions with industry experts.

Path Through Design Decisions

Design patterns can be used to document solutions, approaches, and practices applied commonly in safety system development. Individual solutions are valuable mainly in the context of a single problem to be solved. Although this is valuable when a solution for an individual problem is sought after, other benefits appear when multiple patterns are used together. Patterns, and the solutions they present, often relate to each other in one or more ways. When patterns are arranged and documented so that they form a larger whole and relate to each other, a pattern language is formed.

The relations between the patterns of a pattern language can be seen as one of the strengths of a pattern language from a design viewpoint (Buschmann et al., 2007, p. 15). Typically, the relations form paths through the pattern language which can be seen, for instance, in (Eloranta et al., 2014, p. 85) and (Buschmann et al., 2007, p. 40) and figures 5.5 and 6.5 of this thesis. The language approach supports the designer by indicating, for instance, which patterns could be considered next and which patterns provide alternative approaches to the current pattern. In the latter case, this information can support the process of making a design decision as potential alternative approaches are indicated. The consequences or forces of the individual alternative patterns can then be reviewed to select the most suitable approach or to look for completely different approaches.

Transforming Experience in Explicit Format

Experience is a valuable resource in safety system development as in any engineering discipline. Design patterns make an effort to transform the knowledge accumulated in the heads of designers and existing system documentation into an explicit format. This makes the information accessible also to other designers. In the context of functional safety systems, such experience can add credibility of a solution through the previous known use(s) of the solution. Design patterns (at least give a try to) document the positive and negative consequences of a solution. This information may be obvious to the original applier of the solution. However, the consequences and rationale may not appear, for instance, in the documentation of the system where the approach has been used although, according to van Heesch et al. (2012, sec. 2), such rationale for design decisions should be provided. Design patterns are typically named. This enables and supports the usage of patterns as a part of communication (Riehle, 2011, sec. 3.1). A solution can be referred to with the corresponding pattern name in discussions and potentially in documentation too with appropriate references.

Transfer of Information Between Parties

Patterns and pattern languages act as media that transmit information between parties such as safety system development practitioners, researchers, and companies. Naturally, there is a number of other ways to achieve the same effect. For instance, books, websites, and other documentation media offer similar capabilities. Nevertheless, during the final phases of the research, it was also noticed that the communal processing of patterns in the workshops motivated companies and their representatives to share their knowledge for the benefit of all the participants. Especially the workshops (see Section 7.2) organized in the final phases of the research supported this aspect of pattern mining work.

During the DPSafe project, both workshops and interviews were organized. The circulation and sharing of information regarding approaches and practices during these events was considered one of the main benefit of the project. During the events, the participants shared opinions, views, approaches, and solutions they had used or encountered during their daily work. This happened both in the workshops with people from different companies as well as in interviews within specific companies.

The interviews and discussions resulted from the applied pattern mining approach. Similar results could be achieved without pattern mining in mind. However, in the case of the DPSafe project, pattern mining seemed to build grounds for the discussion, especially between companies. This is most likely due to the purpose of the project that was to gather and document information to benefit all the participants and their companies. One of the factors for successful information sharing can also be seen in the project setup which enabled the information sharing. The researchers acted and were considered as an abstraction layer, that enabled making the initial pattern prototypes so that the company providing the information could be abstracted away leaving only the pattern and the solution to be discussed with all the participants.

Increasing Awareness of Potential Approaches

According to discussions with industry representatives, the development of functional safety systems is partly seen as problematic due to the uncertainty related to the standards such as (IEC 61508, 2010; ISO 13849-1, 2015). The issues seem to raise from the uncertainty regarding the interpretation of the standards and their requirements. This is caused partly by the degree of freedom that is built into the standards. That is, the standards provide room for distinct, but justified, approaches. In such an environment it may be hard to obtain sufficient self-assurance to suggest or apply certain approaches or solutions due to the uncertainty if there is any doubt regarding the compatibility of the approach or solution with the considered standards. As a result, a designer or a designer group may be left asking questions. What kinds of techniques, solutions, or approaches comply with the requirements? Is the taken approach compatible with a standard or a standard requirement?

For these issues and uncertainties, design patterns can offer a potential solution. When design patterns are gathered from industry representatives and materials, they reflect the approaches, architectures, solutions, and methods applied in the industry. In such cases, the company or designer, from which a pattern originates, has already used or is aware of the use of the pattern and the solution suggested by it. For other companies and their workers, the obtained approach may be unknown or it may be considered uncertain in the context of a standard to be applied. In such cases, the awareness that an approach or a solution has already been applied in a similar context may open new possibilities and remove uncertainty regarding what is acceptable. In the context of the DPSafe project, the spread of awareness and confidence was most likely boosted by the communal pattern workshop approach in which industry representatives had the possibility to discuss the patterns directly with each other.

Alleviating Bureaucracy

The standards and regulations considering the development of functional safety systems introduce long lists of requirements considering the development process, methods and techniques to be used, and structures to be applied. The usage of certain methods and approaches is practically mandatory, and the standards may require more documentation and more thorough testing than would be carried out for a regular control system or application. Thus, typically the development process of a functional safety system tends to be costlier than a similar control system without the safety-related parts or need to comply with a similar standard.

Due to the increased cost structure, in the most cases, it would be cheaper to reduce the scope and/or size of the safety system to the minimum. From the author's perspective, the standards considering functional safety system development provide sparingly help in this regard. However, the industry has identified ways to mitigate the scope of the safety system and move some of the tasks out of the safety system scope to the control system as indicated in Chapter 7. In this regard, the pattern approach has been used to identify and document approaches to mitigate the unwanted effects of functional safety system standards.

In addition to the safety system scope reduction and the solutions and approaches documented, patterns can help to bridge a gap between the requirements given by standards and the design that should comply with the standards. For instance, (ISO 13849-1, 2015) defines in Section 4.6.2:

> For Safety-related embedded software (SRESW) for components with Required Performance Level (PLr) c or d, the following additional measures shall be applied:
>
> ...
>
> modular and structured programming, separation in non-safety-related software, limited module sizes with *fully defined interfaces*, use of design and coding standards;
>
> ...

The highlighted part can be interpreted in many ways and provides some freedom regarding the actual implementation. In this case, the Units of measurement pattern candidate, suggesting a compile time unit management of program variables, provides an insight to achieve the full definition of the interfaces of a module.

4.2 Presentation in the Result Chapters

Chapters 5, 6, and 7 illustrate the main results of the research, namely the design patterns and the related pattern language. The chapters are organized as follows.

At the beginning of each chapter, a short introduction to the main categories and topics of the discussed patterns is provided. In addition, a short overview of the origin of the patterns is provided to ground the patterns discussed in the chapter on the timeline of the research. The patterns discussed in Chapter 5 present older work whereas the patterns discussed in Chapter 7 represent more recently documented patterns.

The reason for the approach lies in the evolution of the pattern mining process that has been applied during the research. The patterns in each chapter have been mined using a slightly different mining process. The process has evolved from the initial approach described in Section 5.2 to the one described in Section 7.2. Therefore, the applied pattern mining approach is described in each result chapter.

The actual result section follows the pattern mining description. The main categories of patterns discussed in the chapter are introduced. This introduction provides the reader with an idea of the topics that the patterns consider. The category descriptions are followed by patlets of the patterns belonging to the chapter. A patlet is a short description of a pattern representing the core problem statement followed by the core solution statement.

To assist the developers of safety systems, some of the patlets are completed with references to related standards. The purpose of this is to indicate when a pattern supports, conflicts or otherwise relates to a standard or its section to give the applier an idea of the effect of a pattern in context of a standard. A related standard reference was added to the patlets of such patterns that included a suitable reference to a standard in the pattern description. The complete pattern descriptions can be found in the referenced articles or report.

The patlets are given in table format. The name of the pattern is followed by the publication where the pattern has been previously published. The rightmost column of the table provides the status of the pattern. A pattern with three or more known uses is marked with **P** to indicate a **pattern** and a pattern with less than three known uses is marked with **PC** to indicate a **pattern candidate**. In Chapter 7, more detailed status indicators are used.

Following the pattern descriptions, the patterns are sorted into the categories discussed above. The categorization considers both the topic and purpose of the patterns. Following the categorization, an illustration of the pattern language part is given to show the relations between the presented patterns. Finally, a discussion section summarizes the patterns, topics, and applied pattern mining process.

5 Control Systems, Safety Systems, and Their Co-existence

5.1 Introduction

Architecture considers the fundamental structural and behavioural aspects of a functional safety system. The topic of architecture is centric in a sense that it relates to and affects nearly all the other aspects of the system. According to (Kruchten, 2004, p. 9-10):

> architecture encompasses significant decisions about the following:
>
> - The organization of a software system
> - The selection of structural elements and their interfaces by which the system is composed
> - Their behaviour, as specified in the collaboration among those elements
> - The composition of these structural and behavioral elements into progressively larger subsystems
> - The architectural style that guides this organization: these elements and their interfaces, their collaborations, and their composition

Architecture can also be considered as

> the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution. (IEEE 1471:2000, 2000, cited by Carnegie Mellon University, 2015)

In the context of this thesis, many of the presented patterns can be considered to relate to architecture in one way or another. Some patterns contribute to the structure, some the operation, and some all the potential aspects of architecture in terms of functional safety systems. One aspect where architectural decisions are made considering functional safety development is the design of co-operation between safety and control systems.

Although safety and control systems have a similar form of operation, they have a distinct purpose. A safety system tries to retain the safe operation of a system. A control system also contributes to this, but its primary concern is to control the system to produce its output with an optimal outcome. The optimal outcome may be, for instance, the achieved production rate. In a such case, higher speeds, forces, and concentrations are typically required. However, from the safety point of view, lower speeds, forces, and concentrations would be preferable as often these lead to more inherently safe design and operation. Consequently, the objectives of the safety and control systems may be conflicting.

The co-operation and co-existence of safety and controls systems were one of the main categories that emerged during the research for safety system patterns. Safety and control systems both operate and affect the system under control. In addition, they both typically utilize a rather similar operation principle. The system under control is measured or observed to acquire information on the system state. The acquired information is used to execute logic considering how the system should be controlled to achieve the purpose of the controlling element (that is, the purpose of a control or safety system). Finally, the system under control is affected by a set of actuators operated by control and safety systems.

5.2 Initial Pattern Mining Approach

The pattern research of this thesis was initially started by searching for patterns from the (IEC 61508, 2010). The standard status of the IEC 61508-3 was considered as a sufficient proof of applicability of the solution. In this phase, the documented patterns were design solutions to the requirements of the IEC 61508-3 standard. However, when familiarity with the standard increased, it became obvious that it is hard to provide added value by just looking at the standard. Instead, more relevant aspects could be found between the standard and the actual designs. Thus, a new approach was incorporated.

The patterns presented in this chapter represent the first patterns documented during the research and they are also the first ones outside the scope of the immediate IEC 61508-3 contexts. The work began around 2011. During the preceding (and forthcoming) years, a set of projects considering control system development, architecture, and safety-related software had been active. In these projects, a set of discussions with industry members had occurred and views to the safety and control system development had been exchanged with the industry representatives. These discussions had no direct purpose to serve pattern mining. However, they were used as an inspiration to the first patterns of the work.

In this phase of the research work, the pattern ideas, solutions, and approaches were crafted to the pattern format without further analysis. The pattern mining process applied in the initial phases of the research work is illustrated in Figure 5.1. The centric aspect and the basis of the process was the identification of promising solutions and approaches and the documentation of these ideas into prototype patterns. The ideas for the prototype patterns emerged from discussions with industry representatives and considerations of researchers. The industry representatives came from the domains of control system engineering including control, safety-critical, and safety-related systems.

After the pattern prototype was written, it was internally reviewed by researchers and revised if needed. Some of the pattern prototypes went through a mini workshop with external academic participants before publishing.

In this phase of the research, and regardless of the source of the pattern idea, the lack of three known uses of the solution or the approach was not considered an obstacle to document and publish the patterns. Instead, it was considered more important to document the interesting solutions in the pattern format with the extended information regarding the context, forces, and consequences related to the solution and to obtain

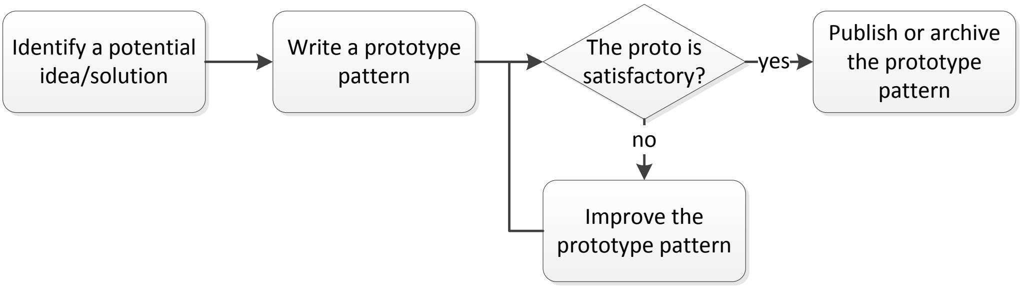

Identify a potential idea/solution
Write a prototype pattern
The proto is satisfactory?
yes
Publish or archive the prototype pattern
no
Improve the prototype pattern

Figure 5.1: The pattern mining process applied in the initial phases of the research.

feedback on the prototype patterns from the pattern community through PLoP conferences. The review process of the conferences and constructive criticism received in the workshops is considered here as the quality control of the suggested patterns.

5.3 Results

The categories of the patterns considered in this chapter include architecture and co-operation related aspects in terms of control and safety system and their development. The categories (on the right-hand side) and the topics within the categories and their relations are depicted in Figure 5.2.

The architecture category considers the structure and behaviour of the system, and the principles regarding the operation and the design of the system. The *System architecture* patterns include *Hardware architecture* as well as software architecture aspects, which again are divided into structural and behavioural aspects. The *Software functionality* topic considers the behavioural aspect of the software.

The co-operation category considers the roles of control and safety systems and their separation, the co-existence of the systems, and finally the need to override a control system if a safety system decides so. The included patterns build on the idea of separation between control and safety systems, but some of them can be potentially applied also in case the safety and control systems are integrated.

The *Safety and control system separation* topic is the starting point in the category. The patterns in the topic consider separation and consequent responsibilities between safety and control systems. When the safety and control systems are separated, the need to operate and control the same system with distinct objectives arises. The *Co-existence with control system* topic provides ideas and design solutions for such a situation. Finally, a safety system must be able to decide whether a system operation, for instance, movement, start-up, or energization is safe or not and consequently dictate whether the operation is allowed or not. For this purpose, a set of *Control system override* patterns has been documented. The purpose of these patterns is to provide a safety system with an ability to override a control system and bring the system under control in a safe state.

It should be noted that all the patterns could be somehow categorized under the architecture category. They all tend to consider either the structure or the functionality of a system. In addition, this may be implemented in either software or hardware.

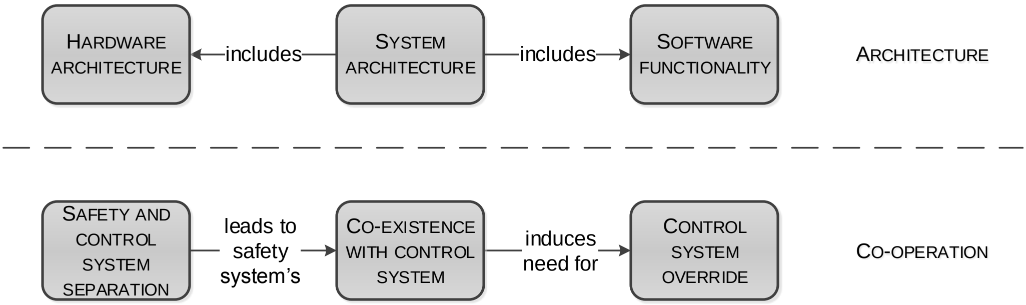


Figure 5.2: High level categorization of the patterns in this chapter.

5.3.1 Patterns for Control and Safety Systems and their Co-existence

Table 5.1 summarizes the patterns for control and safety systems and their co-existence presented in this thesis. The table gives the pattern name **pattern**, the short description of the pattern **patlet**, and the **status** of the pattern, where **P** equals a pattern (with at least three known uses) and a **PC** equals a pattern candidate, with less than three known uses. Table 5.2 summarizes the patterns referenced in the patlets of Table 5.1.

Table 5.1: Patterns for control and safety systems and their co-existence. The patlets have been reproduced here from the referenced publications.

Pattern	Patlet	Status
CHECK PHYSICAL RESPONSE, [P3]	Operations and commands executed in software may succeed or fail in the physical world though software continues execution. Therefore, use sensors to ensure that operations executed in software have the presumed effect in the physical world.	P
CONTROL SIGNAL BLOCKING, [P8]	A safety system needs to have the ability to drive the system under control or the process variable of interest into a SAFE STATE regardless of the operations of a control system. Therefore, override a control signal produced by a control system with a suitable blocking device controlled by a safety system.	P
CONTROL SYSTEM NOTIFICATION, [P8]	The operation of a control system is disturbed when a safety system overrides or restricts the operation of the control system, which may cause the unexpected behaviour of the control system. Therefore, make a control system aware of the state changes of a safety system so that the control system can react accordingly.	P

Table 5.1 – continued from previous page

Pattern	Patlet	Status
Co-operative safety related actuation, [P8]	How to increase the consistency between the operation of a safety system and a control system during situations where the safety system overrides the control system partly or completely? Let a safety system drive a control system into a Safe state whenever a safe state needs to be obtained (according to the safety system).	P
De-energized safe state, [P3]	Power supply for the safety system and the control system as well as the system under control cannot be guaranteed, which might inflict a hazardous state during blackout or power loss in (part of the) safety system. Therefore, design the safety system (and control system as well if applicable) to take the Safe state when power is lost. Related standards: (IEC 61508-7, 2010, section A.1.5)	P
Hardwired safety, [P3]	Development of safety-related application software is costly and provides no real benefit in the context of the considered safety function. Therefore, use a hardware-based safety system instead of application software to implement the safety functionality. Related standards: (IEC 61508, 2010)	P
Indirect response check, [P3]	Adding dedicated hardware for checking that operations executed in software really occurred in the physical world is costly and increases the complexity and the spatial properties of the system. Therefore, check operation success by indirect indication.	PC
Output interlocking, [P3]	Implementing protective functions in control algorithms makes the control algorithms complex. Therefore, use an interlock element alongside each control actuator output in the control system.	P
Productive safety, [P8]	A system under control should be kept in an operational region for as long as possible to avoid the activation of safety functionality while the functionality of a safety system should be kept minimal in comparison to control functionality. Therefore, implement corrective functions in a control system and use the simplest approach for the safety system functionality.	P
Safety limiter, [P8]	A safety system needs to have the ability to drive a system under control or a process variable of interest into a Safe state regardless of the operations of a control system. Therefore, let the safety system manipulate the control signal before directing the control signal to the actuator.	PC

Table 5.1 – continued from previous page

Pattern	Patlet	Status
Separated override, [P8]	A safety system needs to have the ability to drive the system under control or one of its process variables into a Safe state regardless of the operations of a control system. Therefore, use separate actuators for safety and control systems.	P
Separated safety, [P8]	Designing a whole control system according to safety standards is a costly, bureaucratic, and slow process. Therefore, divide and separate the system control functionality into two separate entities: a control system and a safety system. Related standards: (IEC 61508, 2010; ISO 13849-1, 2015)	P
Shared safety actuator, [P8]	Providing each subsystem with a dedicated safety actuator when the same input variable is used by multiple subsystems increases the number of needed safety actuators in the system. Therefore, use a shared safety actuator for all the subsystems.	P

Table 5.2: External patterns referenced in the aforementioned sources. The patlets have been reproduced here from the referenced publications.

Pattern	Patlet	Status
Control system, (Eloranta et al., 2014)	The productivity of a work machine cannot be increased any further using traditional ways of building the machine – using hydraulics, electronics and mechanics. Therefore, implement control system software that controls the machine and has interfaces to communicate with other machines and systems.	P
Limit number of retries, (Rauhamäki et al., 2015)	A retry approach to achieve fault tolerance may lead to an infinite loop. Therefore, limit the number of retries per occurred fault to a reasonable level within time tolerances.	PC

Table 5.2 – continued from previous page

Pattern	Patlet	Status
Safe state, (Eloranta et al., 2014)	When the control system tries to control a part of the machine that is malfunctioning, the machine may respond in an unpredictable way. Consequently, the machine may harm the operator, machine or surroundings. These kinds of situations should not take place. Therefore, design a safe state that can be entered if the control system encounters a malfunction that cannot be handled autonomously. The safe state is such that it prevents the machine from causing harm. The safe state is device and functionality dependent and is not necessarily the same as the unpowered state.	P
Small subsystem fault detection, (Rauhamäki et al., 2015)	It is problematic to transfer substantial amounts of information to high-level subsystems considering faults. Therefore, apply fault detection on as low a subsystem level as possible and aggregate fault information for higher-level subsystems.	PC

5.3.2 Pattern Topic Categorization

Figure 5.3 categorizes the patterns given in Table 5.1 under the categories defined in Figure 5.2. The figure also indicates the relations between the E/E/PE safety system pattern (see Table 6.1) and the topics considered here, to show the relations between patterns discussed in different chapters. The topics describe the main area of concern regarding the included patterns. That is, for instance, the topic of the Hardwired safety pattern is *Hardware architecture.*

5.3.3 Pattern Purpose Categorization

Figure 5.4 illustrates the purpose of the patterns. Here, the purpose describes the target outcome or result of the application of the pattern. In the context of the patterns discussed in this chapter, the following purpose categories were defined. It can be argued that each or most of the patterns illustrated in Figure 5.4 could be included in more than one purpose category, but for the sake of clarity, such more refined categorization is omitted here.

Control and safety system co-existence The main objectives of control and safety systems differ substantially and a safety system is typically able to override a control system. Thus, it is beneficial for both systems that they operate in a coherent way.

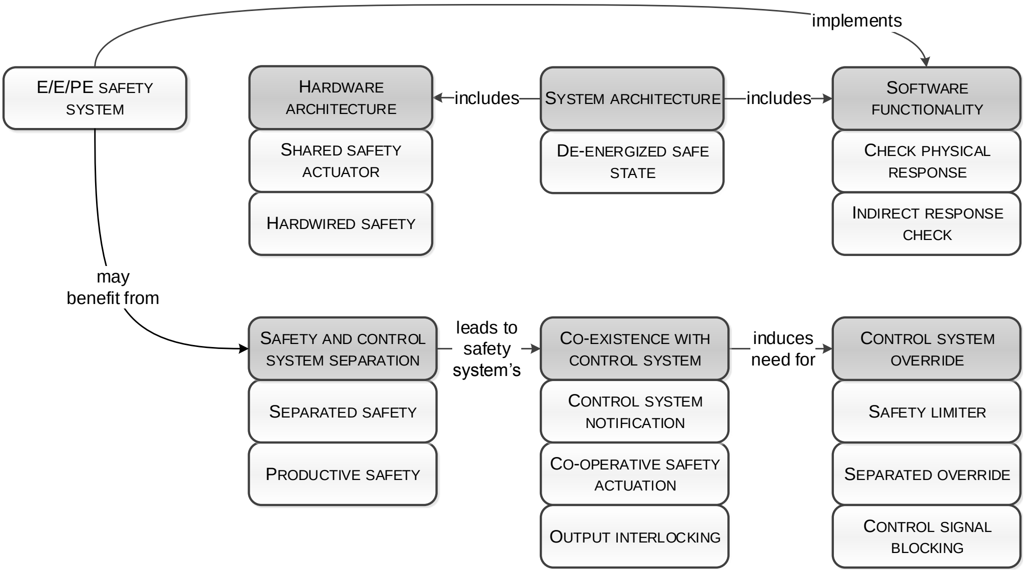


Figure 5.3: Control, safety and co-existence patterns arranged according to their topic

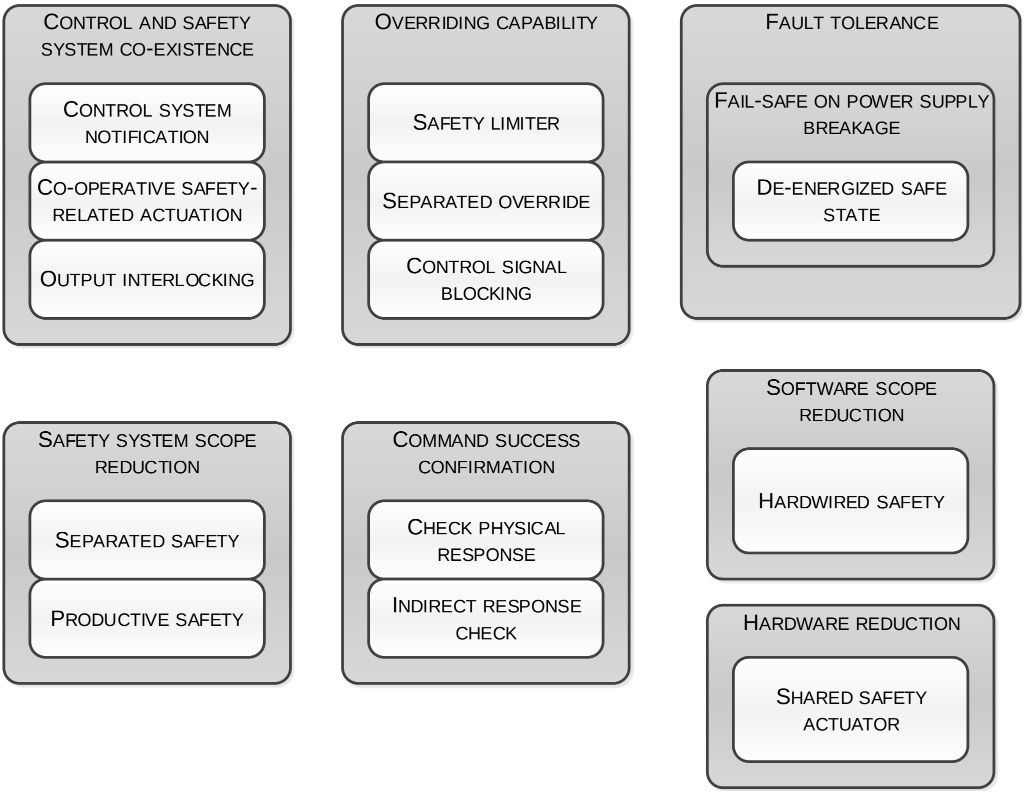


Figure 5.4: Control, safety and co-existence patterns arranged according to their purposes

Overriding capability

A safety system typically requires the ability to override a control system to reach and retain the safe state considering the system under control. The patterns in this purpose category describe means to achieve such capability.

Fault tolerance

These patterns support the ability of a safety system to retain safety in the presence of faults. Typically, this means reverting the system under control into a safe state rather than actually tolerating the fault through recovery.

Command success confirmation

A system controlling a physical entity does not know whether or not a command or operation executed in the software domain has produced the desired outcome in the physical world unless the outcome is checked.

Safety system scope reduction

The smaller the scope of a safety system is, the less work the development of the safety system likely involves.

Software scope reduction

The development of safety-related software in the IEC 61508 context involves several phases, documentation, high testing effort, and the application of suitable methods and tools. Thus, mitigating or eliminating the need for safety-related software can save development effort and costs.

Hardware reduction

The amount of hardware required is a considerable factor in high volume systems, in which spatial, weight, and cost requirements are key factors of the competitive strength.

5.3.4 Pattern Language Part

Figure 5.5 illustrates the part of the pattern language composed out of the patterns given in Table 5.1. Correspondingly, the patterns with dash and dash dot dot outlines are given in Table 5.2. The categories are omitted from this figure for clarity. The language part is compiled based on similar sub-illustrations given in [P3] and [P8].

The centric pattern of this part of the language is the Separated safety pattern. It is considered as a starting point or a prerequisite for the subsequent patterns. That is, one is usually expected to start with this pattern and then apply others as needed. For instance, after Separated safety has been applied, one could consider reducing the functionality of a safety system by applying the Productive safety pattern and to provide the safety system with a way to override the control system by applying the Separated override pattern.

5.4 Discussion

This chapter introduced the first set of patterns documented and published during the research. The patterns consider topics related to control systems, safety systems, and

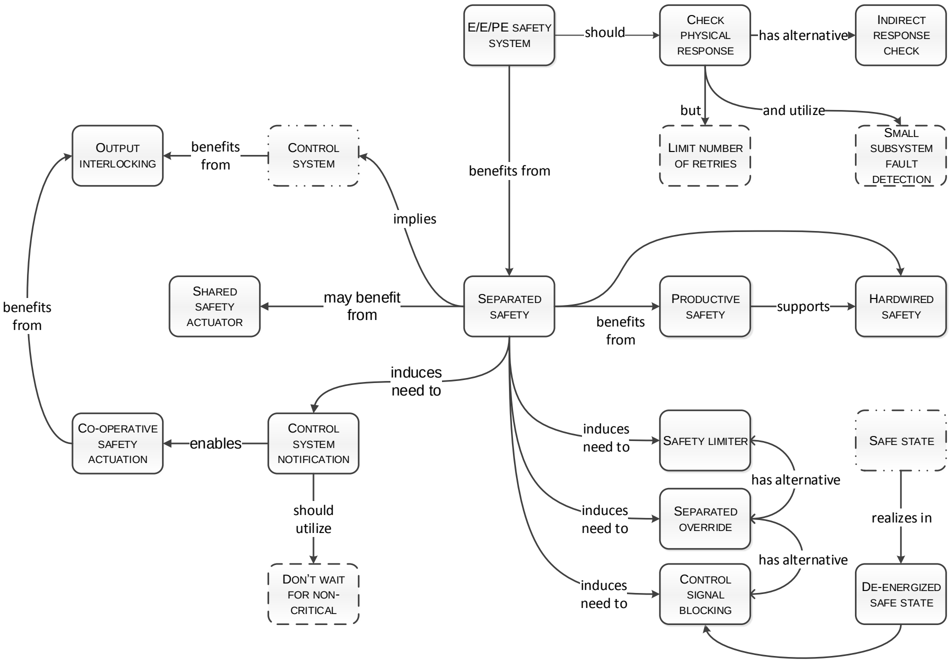


Figure 5.5: Pattern language part of the control, safety and co-existence patterns

their co-existence. All the patterns discussed in this chapter can be considered belonging to one of these topics. The rather broad topic definition reflects the explorative nature of the initial steps of the research work.

The pattern mining approach applied in this part of the research is not sufficiently credible as such. For some of the patterns, the initial idea emerged from discussions with industry representatives, but for others, the initial idea was formulated by the researchers. It is acknowledged that at least the latter approach does not represent a completely sound approach to design pattern mining. However, for most of the patterns and pattern candidates, the credibility through three known uses was gained in the later phases of the research. Therefore, the patterns identified and documented in this phase proved to be a solid part of the pattern language as the research proceeded. This reflects the evolution of the research approach.

6 Hazard Management Process and Risk Reduction

6.1 Introduction

Before any decisions on potential or possible risk reduction methods can be made, hazards and corresponding risks need to be identified. Unless the risks and hazards are known, mitigation methods cannot be justifiably selected and applied. In this chapter, design patterns for the early phases of safety system development are considered. In some cases, it may appear that actually no functional safety systems are needed to reduce risks. However, this does not release the system developers from hazard and risk analysis because to arguably decide not to apply a functional safety system, one has to justify the decision.

Standards such as (IEC 61508, 2010) and (ISO 13849-1, 2015) specify a hazard and risk analysis as an important part of the development of a safety-related system. The process begins with a system scope definition followed by a hazard and risk analysis as also discussed in [P2]. Only after these steps, one goes to the selection of the risk mitigation methods. Furthermore, the application of a functional safety system should not be the primary approach; one should first consider hazard elimination and a passive safety system instead of an active functional safety system (Center for Chemical Process Safety, 2012, p. 123-124).

The hazard management process focuses on engineering approaches to achieve required risk reduction from the system developer point of view. However, as presented by Leveson (2009, p. 6), also system operation and operators participate in the overall safety control structure. Therefore, human aspects should not be left outside the scope of hazard and risk management.

6.2 Evolved Pattern Mining Approach

The initial pattern mining process applied in the first phases of the research was arguably not sufficient as such to obtain credible patterns. A considerable trigger to change the pattern mining and documentation process occurred in the European Conference on Pattern Languages of Programs (EuroPLoP) 2013 where it was stated that the patterns are well written, but they are prototype patterns lacking the three known uses. This was seen as a drawback of the patterns and mining process so far. The dearth of known uses was acknowledged, and the pattern mining process was corrected accordingly.

The evolved pattern mining process is depicted in Figure 6.1. The main difference with regard to the initial pattern mining process (described in Section 5.2) is the introduction

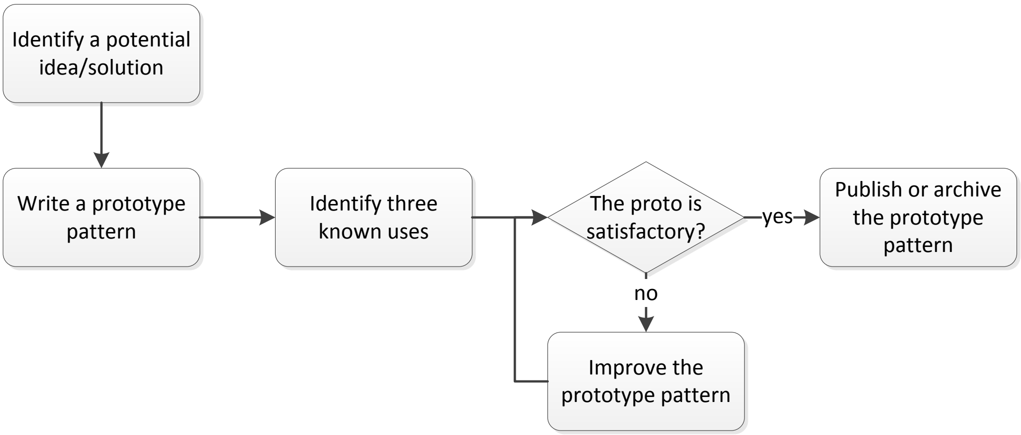

Identify a potential idea/solution
Write a prototype pattern
Identify three known uses
The proto is satisfactory?
yes
Publish or archive the prototype pattern
no
Improve the prototype pattern

Figure 6.1: The evolved pattern mining process applied in the research

of a step to identify three known uses for the solutions described in the pattern.

At least three known uses were searched for the solutions presented in the patterns before they were submitted and published in PLoP conferences. These three known uses were not directly obtained from industry representatives or system documentation. Instead, the known uses were collected using literature sources or experiences from real machines applying a user point of view. Although this cannot be considered an ideal approach from the practitioner point of view, it still provides the patterns with more credibility compared to a situation with no known uses at all.

6.3 Results

The category of patterns considered in this chapter includes *Hazard management* in terms of control and safety system and their development. The category with the included subcategories is depicted in Figure 6.2.

The patterns in this category discuss how the safety aspect can be taken into account in the design and development process of a system. The consideration includes processes, strategies, and functionality approaches. The abstraction level of the patterns varies between the topics. The *Hazard management process* patterns illustrate an abstract approach to hazard management. The patterns in this topic suggest a set of basic approaches to detect and mitigate hazards. In the *Hazard management strategy* topic, abstract approaches are made more concrete, indicating how the basic approaches can be implemented in systems. Finally, in the *Safety functionality approach* topic, actual functions to mitigate certain hazards in certain systems are described. The rationale for the categorization is to provide the user with the core idea or principle of the hazard management approach as well as a more practical illustration of the approach.

Human-Machine responsibilities are also considered in the category of *Hazard management* patterns. A human being is both the protected target and a potential co-operator or factor in achieving safe operation of the system under consideration. Therefore, the human viewpoint should be considered alongside the technical methods in the risk reduction process.

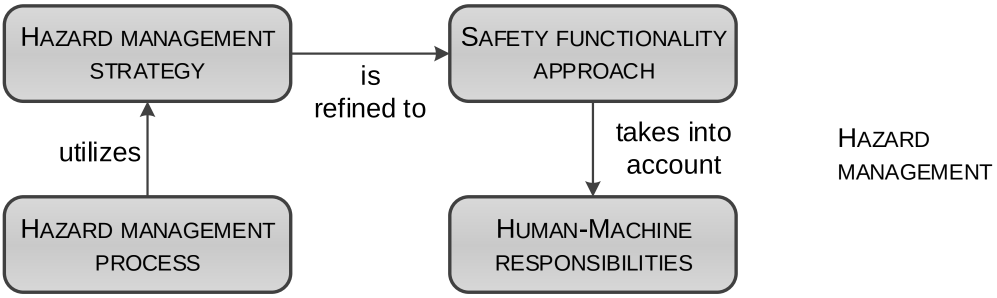

HAZARD MANAGEMENT STRATEGY
is refined to
SAFETY FUNCTIONALITY APPROACH
utilizes
takes into account
HAZARD MANAGEMENT
HAZARD MANAGEMENT PROCESS
HUMAN-MACHINE RESPONSIBILITIES

Figure 6.2: High level categorization of the patterns in this chapter

6.3.1 Patterns for Hazard Management Process

Table 6.1 summarizes the patterns for hazard management process presented in this thesis. The table gives the pattern name **pattern**, the short description of the pattern **patlet**, and the **status** of the pattern where **P** equals a pattern (with at least three known uses) and a **PC** equals a pattern candidate, with less than three known uses.

Table 6.1: Patterns for hazard management. The patlets have been reproduced here from the referenced publications.

Pattern	**Patlet**	**Status**
SAFETY RISK IDENTIFICATION, [P4]	To make conscious decisions considering hazard and risk management, information on these aspects needs to be available. Therefore, use structured and/or systematic method(s) in order to identify the hazards and associated risks introduced by the system. Related standards: (IEC 61508, 2010; ISO 12100:2010, 2010; ISO 13849-1, 2015)	P
ELIMINATE HAZARD, [P4]	You want to maximize the likelihood that a hazard introduced by a system cannot cause harm in any part of the system lifecycle. Therefore, eliminate the hazard completely by removing the component introducing the hazard from the system.	P
SUBSTITUTE HAZARD, [P4]	A hazard needs to be eliminated from the system. Therefore, substitute the hazardous element with a non-hazardous or at least less hazardous element.	P
PASSIVE PROTECTIVE MEASURE, [P6]	A hazard or a hazardous element remains in the system, but the related risk needs to be mitigated. Therefore, use a passive protective measure to mitigate the risk by non-functional design solutions, equipment, or system design features that mitigate the risk related to the hazard.	P

Table 6.1 – continued from previous page

Pattern	Patlet	Status
Active protective measure, [P6]	A hazard or a hazardous element remains in the system and the related risk needs to be mitigated. Therefore, use an active protective measure to mitigate the risk by affecting system operation through a defined functionality so that the risk is reduced.	P
Isolate hazard, [P4]	A hazardous element needs to remain in the system and, with the current exposure profile, the related risk is intolerable. Therefore, isolate the hazard by physically isolating the hazardous element from the environment and people.	P
E/E/PE safety system, [P4]	An active protective measure of relatively complex functionality needs to be implemented. Therefore, use an electric, electronic or programmable electronic (E/E/PE) safety system to implement the protective measure functionality. Related standards: (IEC 61508, 2010; ISO 13849-1, 2015)	P
Interrupted hazardous action, [P5]	A user operated system element may enter a hazardous operating range due to actions initiated by an operator, but without the operator noticing this. Therefore, interrupt the hazardous system operation before the hazardous operation range is reached and force the operator to acknowledge this.	P
Interruptible hazardous zone, [P5]	The hazardous zone or element needs to be easily accessible, but hazardous conditions within the zone may exist in defined situations. Therefore, implement an interlocking guard over the hazardous element or zone.	P
Locked hazardous zone, [P5]	A hazardous zone or element needs to be easily accessible, but hazardous conditions within the zone may exist in defined situations. Therefore, implement a locking guard to cover the hazardous element or zone.	P
Automated safety system, [P7]	Humans are slow and unreliable decision makers and cannot deterministically react on random events. Therefore, avoid humans as a part of the functionality of a safety system .	P
Manual safety ability, [P7]	E/E/PE safety systems have limited possibilities to observe and interact with their environment and system. Therefore, provide the human operators with the ability to manually drive the system into a safe state.	P

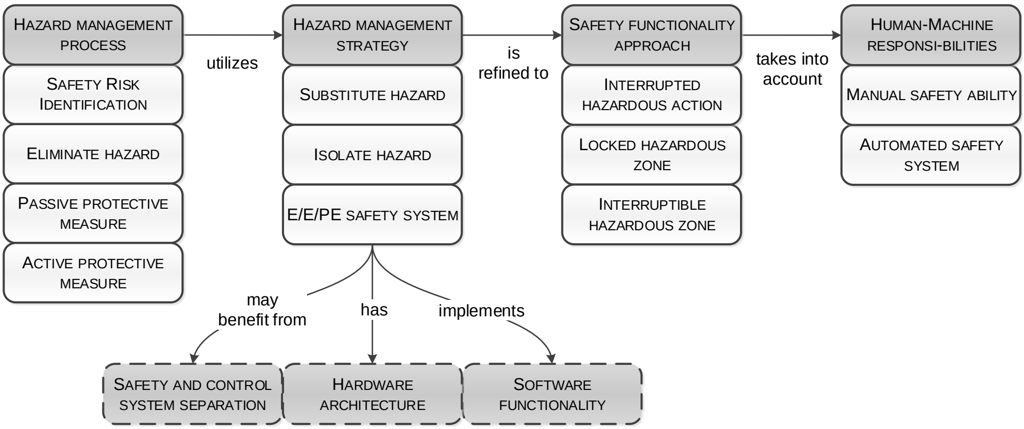

Hazard management process
Safety Risk Identification
Eliminate hazard
Passive protective measure
Active protective measure
utilizes
Hazard management strategy
Substitute hazard
Isolate hazard
E/E/PE safety system
is refined to
Safety functionality approach
Interrupted hazardous action
Locked hazardous zone
Interruptible hazardous zone
takes into account
Human-Machine responsi-bilities
Manual safety ability
Automated safety system
may benefit from
has
implements
Safety and control system separation
Hardware architecture
Software functionality

Figure 6.3: Hazard and risk management patterns arranged according to their topic

6.3.2 Pattern Topic Categorization

Figure 6.3 categorizes the patterns given in Table 6.1 under the topics defined in Figure 6.2. The topics describe the main area of concern regarding the included patterns. For instance, the topic of the Substitute hazard pattern belongs to *Hazard management strategy*. The figure also indicates the relations between the E/E/PE safety system pattern (see Table 6.1) and the topics introduced in Chapter 5.

6.3.3 Pattern Purpose Categorization

Figure 6.4 illustrates the purpose of the patterns presented in this chapter. Here, the purpose describes the target outcome or result of the application of the pattern. In the context of the patterns discussed in this chapter, the following purpose categories were defined. It can be argued that each or most of the patterns illustrated in Figure 6.4 could be included in more than one purpose category, but for the sake of clarity, such categorization is omitted here.

Foundation for risk management

To be able to make justifiable decisions on the risk mitigation methods and approaches, the hazards and corresponding risks need to be known. Unless the risks are known, excessive or inadequate mitigation is likely to occur. Both of the cases come with cost overhead. The former case introduces additional development and system costs as too high a SIL is complied with or unnecessary safety measures are implemented in the system. In the latter case, costs are saved during development time, but the risks are more likely to be realized in the later phases of the system life cycle.

Risk elimination

The patterns in this category aim to eliminate, that is, to remove the hazard or hazard source completely

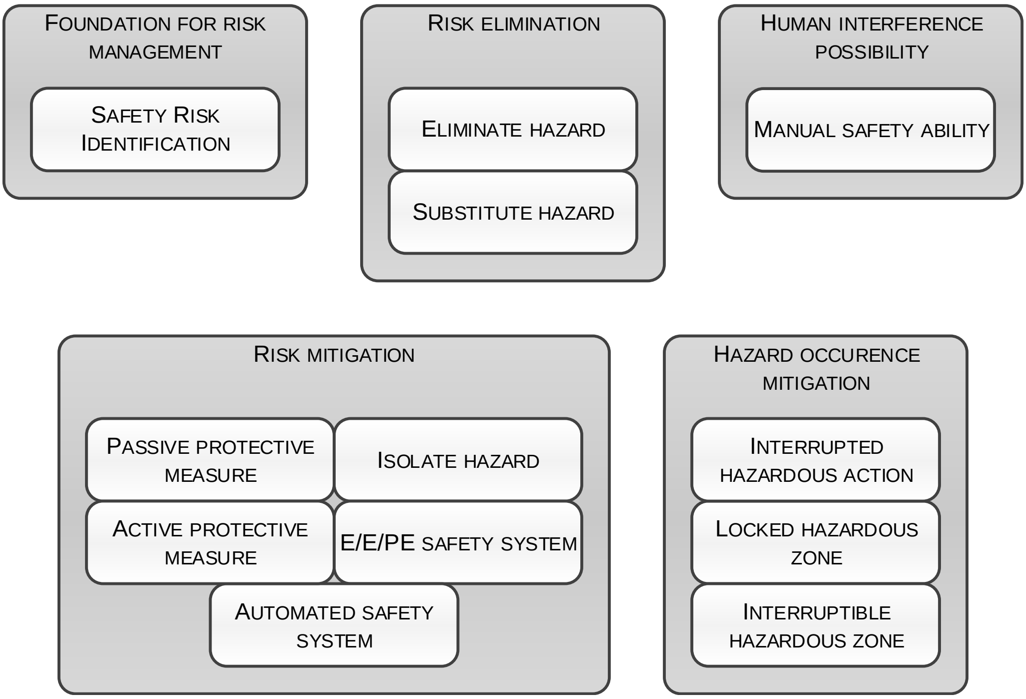

Foundation for risk management
Safety Risk Identification
Risk elimination
Eliminate hazard
Substitute hazard
Human interference possibility
Manual safety ability
Risk mitigation
Passive protective measure
Isolate hazard
Active protective measure
E/E/PE safety system
Automated safety system
Hazard occurence mitigation
Interrupted hazardous action
Locked hazardous zone
Interruptible hazardous zone

Figure 6.4: Hazard and risk management patterns arranged according to their purposes

from the system under development. The eliminated hazards introduce no risks throughout the system life cycle. Therefore, the approach is effective. For instance, if asbestos is eliminated from the system, it cannot introduce risks. However, a substitute or alternative approach can introduce similar or different risks.

Risk mitigation

In some cases, hazards cannot be completely eliminated from the system. In such cases, the related risks need to be mitigated. The patterns in this category aim to reduce the risk by decreasing the likelihood of the hazard occurrence or severity of the consequences.

Hazard occurrence mitigation

The patterns in this category aim specifically at reducing the realization likelihood of a hazard. The hazard remains in the system under consideration, but this is recognized, and the realization likelihood is targeted.

Human interference possibility

Machines are superior to people in some tasks. For example, a reaction time of a human (approximately 200 millisecond (Kosinski and Cummings, 1999, p. 79)) is typically relatively high compared to computer reaction time and human factors are involved in many incidents in process and related industries (Broadribb, 2012). However, this does not indicate that people should not be able to take the system into a safe state. Instead, machines

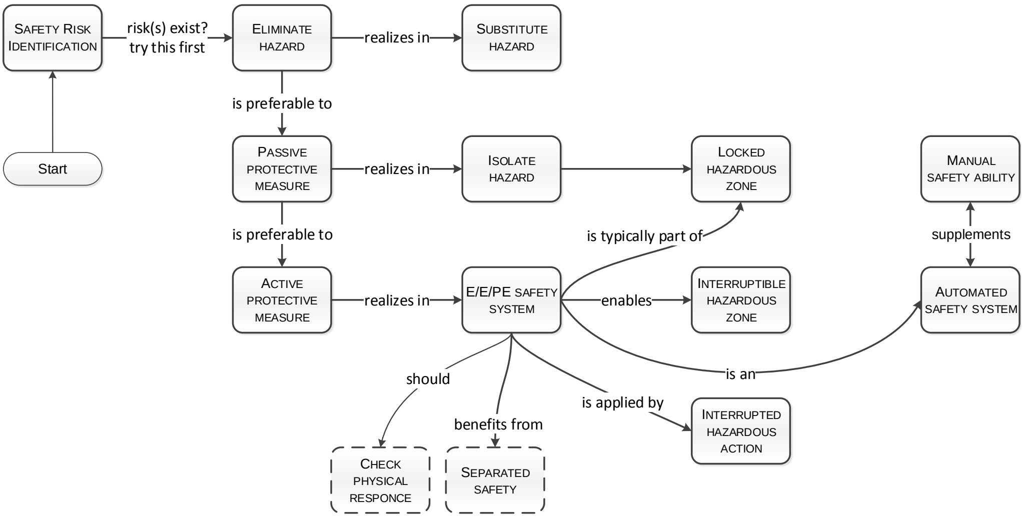


Figure 6.5: Pattern language part of the hazard and risk management patterns

and control systems have intrinsic weaknesses, too, and people can remedy (at least partially) these weaknesses.

6.3.4 Pattern Language Part

Figure 6.5 illustrates the part of the pattern language composed of the patterns given in Table 6.1. The patterns with dashed outlines (Separated safety and Check physical response) are given in Table 5.1. Pattern categories are omitted from this figure for clarity. The language part is compiled based on similar sub-illustrations given in [P4], [P5], and [P6].

The root pattern of the whole pattern language and consequently also this part of the language is the Safety risk identification pattern. The pattern suggests the usage of systematic methods of identifying the risks and hazards related to the system in design. Only after the hazards and risks are known, justifiable argumentation on possible risk mitigation methods can be considered.

The subsequent patterns Eliminate hazard and Substitute hazard propose removing a risk by the removal of the related hazard. If the hazard cannot occur, the risk is also removed. If the hazard cannot be removed, one should consider a Passive protective measure or an Active protective measure to mitigate the risk. These measures do not remove the hazard but decrease the likelihood of the occurrence or the severity of the consequences of the hazard. The Isolate hazard and the E/E/PE safety system are corresponding approaches to implement the aforementioned measures. Finally, the Locked hazardous zone, Interruptible hazardous zone, and Interrupted hazardous action build on the aforementioned and more abstract approaches.

The Automated safety system and the Manual safety ability patterns consider the human perspective on safety system operation. These patterns are connected to the E/E/PE safety system pattern and provide an additional insight into the design of such a system.

6.4 Discussion

The patterns in this section considered hazard and risk management. The patterns were categorized into one main category and four topics. Each topic had a different viewpoint and abstraction level considering hazard and risk management.

The *Hazard management process* topic introduces a set of engineering or development time methods and approaches for risk mitigation. These methods are under the control of the system designer or manufacturer and they are therefore considered preferable to other approaches. Hazards can also be managed through administrative means and personal protective devices (NIOSH, 2015), (Spellman and Bieber, 2009, p. 16). The administrative measures include, among others, training, instructions, warning signs, and shift systems. In these measures, an employer or other organization needs to provide the measures. The PPE include protective measures that are worn by employees and people operating in the hazardous zone. PPEs include, among others, hearing protectors, eye shields, boiler suits, and helmets.

The main problem with both the aforementioned approaches is the absence of control from designer and manufacturer perspectives. The system manufacturer cannot force an organization to comply with the administrative measures or force people to wear the required PPE. Therefore, this thesis focuses on the measures the designer and manufacturer of the system can control.

The patterns discussed in this chapter were obtained by applying an evolved pattern mining process. However, the process lacked direct co-operation with industry and known uses from automation systems. Still, the known uses from literature and experience were obtained to the patterns to improve their credibility.

7 Functional Safety System Development

The domain of functional safety system development is broad and includes several aspects. The patterns in this chapter regard architecture, co-operation and development aspects of functional safety systems and their development. The patterns augment the patterns, categories, and topics introduced in chapters 5 and 6. The pattern mining approach applied to obtain the patterns in this chapter also produced new known uses for the patterns discussed in chapters 5 and 6.

7.1 Introduction

The final version of the pattern mining process was applied in the DPSafe project. During the project, a solid connection with functional safety system development practitioners was established. Total of 18 interviews were carried out in 13 companies. The main target of the project was to acquire new design patterns for the participating companies. During the project new known uses for the previously obtained and documented patterns (discussed in chapters 5 and 6) were searched and found from real world applications and developers. Consequently, most of the pattern candidates documented beforehand were validated during the project, in which the known uses were discovered.

In the DPSafe project, the pattern mining process was not primarily carried out from an academic viewpoint. The purpose of the project was to spread practices and solutions applied in functional safety systems and their development between FIMA members. The known uses were seen as a valuable asset for the patterns, but spreading the ideas was considered more valuable. Therefore, also pattern candidates with less than three known uses were documented and reported.

7.2 Final Pattern Mining Process and Validation Approach

The final pattern mining process is described in the following subsections and an outline of the process is depicted in Figure 7.1. The research process was split into three main phases: the pattern mining phase, the workshop phase, and the finalizing phase. In the pattern mining phase, data was obtained through interviewing companies considering their practices, solutions, and principles regarding the development of functional safety systems. Pattern candidates were sketched according to the obtained data and material. Then, selected patterns were discussed in workshops. Finally, the patterns and pattern candidates were categorized and compiled into a pattern language in the finalization phase.

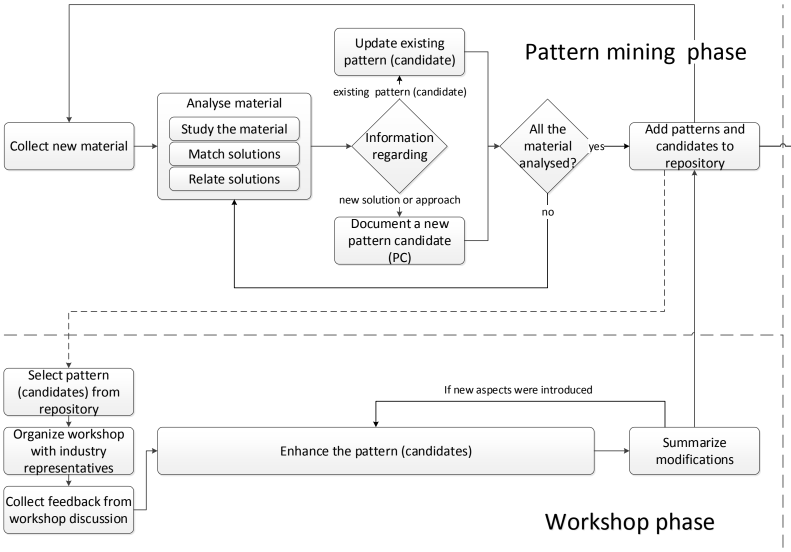

Pattern mining phase
Collect new material
Analyse material
Study the material
Match solutions
Relate solutions
Update existing pattern (candidate)
existing pattern (candidate)
Information regarding
new solution or approach
Document a new pattern candidate (PC)
All the material analysed?
yes
no
Add patterns and candidates to repository
Pattern language
Purpose categorization
Topic categorization
Finalizing phase
Select pattern (candidates) from repository
Organize workshop with industry representatives
Collect feedback from workshop discussion
If new aspects were introduced
Enhance the pattern (candidates)
Summarize modifications
Workshop phase

Figure 7.1: The final pattern mining process applied in the research

The new patterns and pattern candidates mined using this process were documented as a technical report (Rauhamäki and Vepsäläinen, 2016) produced for FIMA. The following subsections are reproduced here based on the report with major extensions, clarifications, and modifications.

7.2.1 Pattern Mining Phase

The first step of the pattern mining phase was the material collection effort. The main method of collecting material for the study was interviews with participating companies. The aim of the interviews was to gather data considering safety system development. The interviews were carried out applying a semi-structured interview approach (Edwards and Holland, 2013, p. 29) (Harrell and Bradley, 2009, p. 27). The researchers had a question list prepared that was used as a basis for the interview. However, the interviews were kept purposefully free-form. That is, the question list was not obeyed if the discussion was flowing freely under the subject of interest. Probe questions were used to gain further information on specific subjects.

The interview method is one of the methods suggested for pattern mining (Rising, 1998b, p. 46, 91-92) (Kerth and Cunningham, 1997, p. 56). For most of the interviews, two interviewers participated, which was found beneficial for the interview as also suggested by Hove and Anda (2005, sec. 4.1.2). The interviews were not recorded. Instead, notes were taken. This removed the possibility to go through the interview again, but also eliminated the amount of work needed to transcript the recorded interview. However, the notes still needed some post-interview editing such as typing error corrections, filling in missing words, and generally making the text more readable. After the editing work, the interview materials were compared and combined. When the interview notes had been processed and the pattern candidates had been documented, the interviewees had the possibility to review the notes and the pattern candidates. In some cases, misunderstood notes and pattern candidates were corrected as a result of this approach.

The second step in the pattern mining phase was to analyse the collected material. In this step, the acquired material was studied to identify interesting pieces of information. The primary focus was to find solutions and approaches used in the development of safety systems. Both new and already emerged solutions (pattern matching) were considered desirable findings. In addition, other items such as rationale, context, consequences, and examples of the solutions were under interest as they supported composing new pattern candidates as well as enhancing the existing ones.

Identifying relations between the patterns and solutions was a part of the pattern language building effort. Relationships and structure between patterns were already considered during pattern mining process. For instance, should a general idea be presented on a separate pattern and the more practical implementation patterns in their own? Pattern matching work was carried out alongside the material analysis and relation building steps. The purpose was to find new instances or uses of existing patterns. The importance of this step is high as known uses validate the pattern candidates.

When a new solution or approach was identified, it was documented as a pattern candidate in a design pattern format (see Section 3.5). The pattern candidates were sketched based on the obtained material and personal experience. If a new known use or new additional information regarding an existing pattern or pattern candidate was observed, the existing pattern was updated accordingly. When the material had been analysed, the obtained pattern candidates and patterns were moved to a repository.

7.2.2 Workshop Phase

The workshop phase resembles the workshop phase applied in PLoP conferences (see Section 3.3).

The intent of the workshop phase was to refine the documented patterns and pattern candidates with a broad set of comments from different companies. The workshop process was as follows: patterns and pattern candidates to be discussed were selected by the researchers from the repository. Each workshop participant was assigned a pattern to be read and commented before the workshop. The purpose of this was to ensure at least one of the workshop participants had read the pattern (candidate) to be discussed thoroughly. Discussion about each pattern was started with a five minutes reading period so that all participants would have time to familiarize themselves with the pattern to be discussed. After this, the assigned reader started the discussion by providing the initial comments. Typically, after this, the discussion went on and each participant had a chance to comment and discuss. The researchers wrote down the comments and discussion. Finally, researchers might ask clarifying questions and summarize the main findings.

On average, six pattern candidates were discussed in a single workshop day. After the workshop, a refined version of the patterns was written according to the comments given in the workshop. The workshop discussions and feedback yielded more insight into the patterns and several new known uses also emerged. On the other hand, some pattern candidates received justified criticism which either resulted in major changes in content or a change in the pattern candidate status.

The results of the previous workshop were briefly summarized at the beginning of the next workshop where the participants had a chance to provide additional comments. The

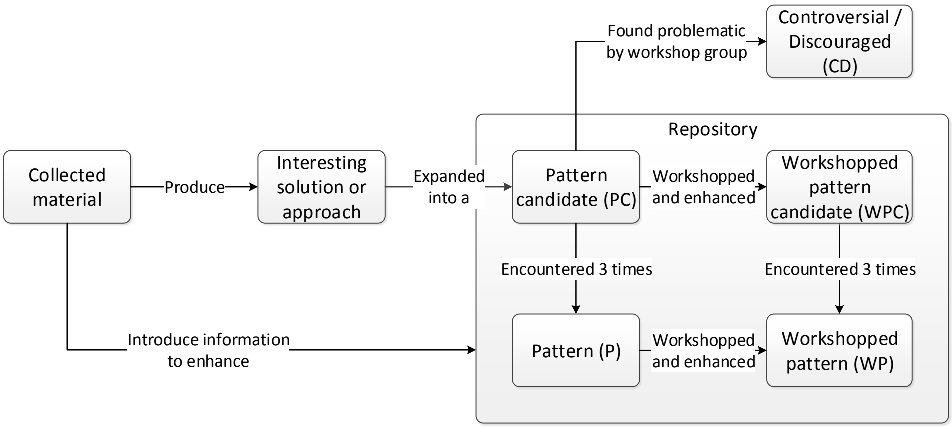


Figure 7.2: The potential states of patterns during the pattern mining process

comments were taken into account before updating the repository with an enhanced version of each pattern (candidate).

7.2.3 Finalization Phase

In the finalization phase, patterns and pattern candidates were categorized, and a pattern language was compiled based on the relationships of the patterns. The patterns were categorized according to their purpose and the parts of the system their application is likely to affect. In this thesis, the latter categorization is considered to be the topics of the patterns. The finalization tasks were carried out at the end of the project. Some of the pattern (candidates) were not discussed in a workshop during the DPSafe project. However, workshops continued in a follow-up project.

Figure 7.2 illustrates the potential states of patterns and pattern candidates as a result of the pattern mining approach described in this section. The pattern (candidates) started as ideas emerging from interview materials. Each potential pattern candidate (PC) was documented. If three known uses were identified for a candidate, it was promoted to a pattern (P). Both patterns and pattern candidates were discussed in workshops providing them with workshopped pattern (WP) and workshopped pattern candidate (WPC) statuses respectively. A couple of patterns were considered controversial or discouraged by a workshop discussion. These pattern candidates were given the controversial/discouraged (CD) status.

7.3 Results

The categories of patterns considered in this chapter include co-operation, architecture, and development related topics in terms of control and safety systems and their development. The categories (on the right-hand side) and the included topics and their relations are depicted in Figure 7.3. In addition, the categories and topics also indicate the expected part or scope of the system, which is likely affected by the application of a pattern.

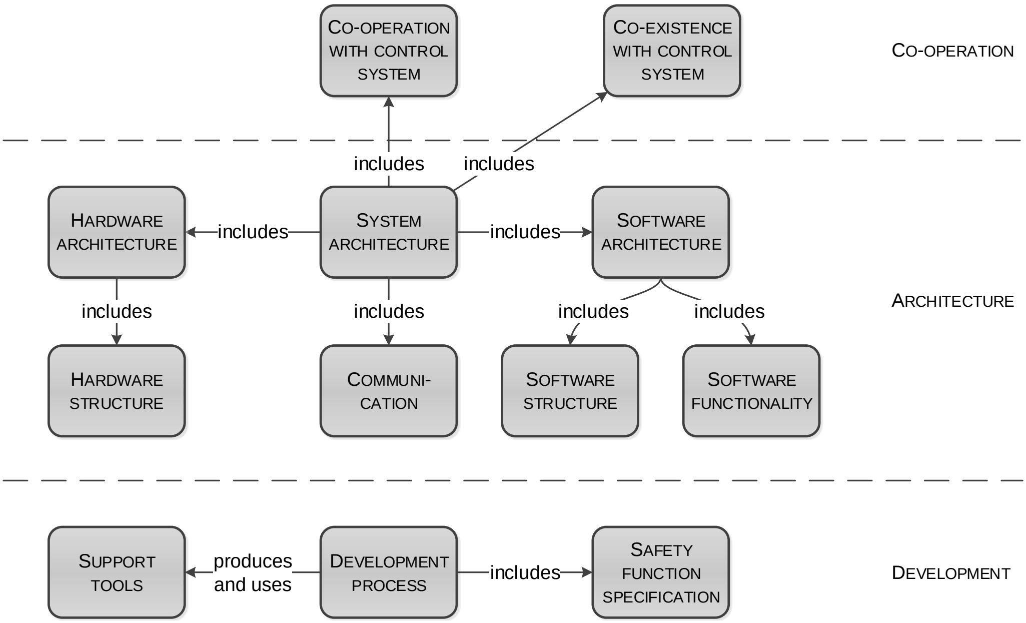


Figure 7.3: High level categorization of the patterns in this chapter

The *Co-operation* category represents the same category as introduced in Section 5.3. In the context of this chapter, the category is extended with the *Co-operation with control system* topic. The patterns in this topic consider how a safety system can co-operate and potentially benefit from the control system. For instance, some tasks that are not directly included in the safety system core functionality can be transferred to the control system, thus increasing the simplicity of the safety system. In addition, new patterns are introduced in the *Co-existence with control system* topic.

The *Architecture* category represents the same category as introduced in Section 5.3. In the context of this chapter, the category is extended with new topics and new patterns belonging to the topics already introduced in Section 5.3. The topic of *Software architecture* is added. This topic includes both the *Software structure* and *Software functionality* topics. The *Communication* topic suggests approaches to organizing the communication within the nodes or processing units of the safety system. The *Hardware architecture* topic introduced in Section 5.3 is extended with the *Hardware structure* topic, which considers the organization of the hardware components of a functional safety system.

The *Development* category considers aspects that are related to the development and manufacturing process of the safety system instead of the functionality and structure of the system. The development process may produce or use certain *Support tools* alongside the process. These tools may include, among others, programming devices, software, and compilers. A *Safety function specification* is a part of the development process where the functions are defined and assigned to nodes.

7.3.1 Patterns for Functional Safety System Development

Table 7.1 summarizes the patterns for functional safety system development presented in this thesis. The table gives the pattern name **pattern**, the short description of the pattern

patlet, and the **status** of the pattern. The information given is based on (Rauhamäki and Vepsäläinen, 2016). However, the statuses of the pattern and pattern candidates resemble the current situation. The available statuses are described in Figure 7.2.

Table 7.1: Patterns for functional safety system development. The patlets have been reproduced here with updates from (Rauhamäki and Vepsäläinen, 2016).

Pattern	Patlet	Status
Black channel	A message bus is a part of a safety function and it needs to conform to the applicable safety standards. Therefore, implement a black channel safety features over the message bus. In practice, set up a safety protocol between the standard bus protocol and the application layer. Related standards: (IEC 61508-2, 2010) referring to to (IEC 61784-3, 2016; IEC 62280, 2014)	WP
Clock-pulse synchronization	A synchronized action should take place in distributed nodes attached to a message bus. Therefore, define a synchronization message that is used to trigger a synchronized action in all the nodes attached to the message bus.	CD
Certified processing hardware	A design and development process to produce processing hardware compatible with standards considering functional safety system development likely results high development costs. Therefore, use certified processing hardware in a safety system implementing a safety function. In many cases, this eases the development process of the safety function considerably as the development team does not have to take the processing hardware itself into account. Related standards: (IEC 61131-6, 2012; IEC 62061, 2005; ISO 13849-1, 2015)	WP
Control system OK condition	A control system should not try to operate a system under control when a safety system is not available, operational, or working as specified. Therefore, let the OK-status information of the safety system be a condition to the normal operation of the control system.	WP
Core problem focus	Limited resources of programmers and developers should be directed to ensure that a safety function to be developed operates as desired. Therefore, select and favour programming languages that reduce the need of the programmers to focus on the computer routine driving in favour of solving the actual problem.	WPC

Table 7.1 – continued from previous page

Pattern	Patlet	Status
Cross monitoring	In multi-channel safety function architecture individual nodes should make sure that they are in a consistent state with the other nodes. Therefore, arrange a communication channel between the nodes and send state/status information between the channels so that each node has sufficient information to deduce itself being in a consistent state with others. Related standards: (ISO 13849-1, 2015)	P
Externalized parametrisation	A change in a coded parameter within a safety-related software part results in a change in the software part and introduces a need to follow an applicable software modification process which produces additional work and costs. Therefore, externalize the factors (parameters) from the safety-related part of software to enable changes to the functionality and configuration without changes in the software itself. Related standards: (IEC 61508-3, 2010; ISO 13849-1, 2015)	WP
Flashed parameters	A set of parameters needs to be stored into a non-volatile structure outside the software. Therefore, store the parameters in an electrically erasable and reprogrammable, non-volatile memory such as Flash memory to separate the parameters from the software code. Related standards: (IEC 61508-3, 2010)	WP
Generic processing hardware	Safety-related software needs to be run on a processing unit, but ready-made and certified safety system processors are relatively expensive hardware units. Therefore, use a generic processor hardware in the safety function instead of a ready-made and certified hardware unit, but also develop all the required diagnostic functionalities and compile required documentation to comply with the followed standard. Related standards: introduces challenges with (IEC 61131-6, 2012; Machinery directive, 2006)	WPC
I/O output grouping	How to arrange the de-energization of actuators and subsystems without using a relay for each of them? Map the outputs to the I/O devices so that the output channels that need to and can be de-energized simultaneously are in same I/O module.	WPC

Table 7.1 – continued from previous page

Pattern	Patlet	Status
Independent output block	Although an output monitoring functionality is a part of the overall safety function logic, it does not represent the core functionality of the safety function. Therefore, implement the output monitoring activities in the (software) output blocks that produce the outputs and communicate with the actuators. Related standards: (ISO 13849-1, 2015)	WPC
Interference blocking platform	Non-interference between software elements needs to be established and justifiably demonstrated. Therefore, use a premade framework, operating system, firmware, platform, etc. that provides the non-interference between the software elements.	WPC
Layered safety architecture	How to divide responsibilities of the layers related to checking the safety of commands so that the resulting system (whole) is safe although the system can be large and complex and the layers developed by different teams? Each layer in the hierarchy performs safety checks for both the commands that they send to lower levels and for commands that they receive from upper levels.	WPC
Local utility logic	Non-safety-related functionality should be included into a multichannel safety system architecture. Therefore, add a dedicated local utility logic/processor to the system, which takes care of all the non-safety-related aspects that do not directly influence the safety function itself. Related standards: (IEC 61508-3, 2010; ISO 13849-1, 2015)	WP
Monitored safety limit	A safety system has to be able to dynamically restrict the values of the process variables of a machine to safe ranges without the need to implement whole control loops in the safety system. Therefore, make a control system of the machine subordinate to the safety system so that the control system tries to keep the process variables within safe ranges, and the safety system intervenes if necessary.	WP
Open loop limp home	In a case of a sensor failure, the sensor data cannot be trusted, and thus the produced information cannot be used as a basis for closed-loop control of a system. Therefore, by-pass the closed-loop controls, for example, machine velocity control, and use direct controls from an operator and drive the system using an open loop control scheme.	WPC

Table 7.1 – continued from previous page

Pattern	Patlet	Status
Outsourced display interface	Information including the state, operation, and parameter values regarding a safety system should be presented to the user, but adding a display unit in the scope of the safety system increases complexity, cost, and development effort. Therefore, use a control system (or other similar resource that is external to the safety systems) to provide users with the data and information concerning the safety system.	WP
Outsourced failure analysis	The core functionality of a safety system is to produce the decision: whether or not to activate the safety function. In several cases the system and its users would benefit from additional analysis functions such as analysis of failures in the safety system. However, addition of functionality in the safety system scope increases its complexity and consequently the development effort of the safety system. Therefore, transfer the responsibility for the analyses of data, including the analysis of fault situations, to a control system which is outside the safety system scope.	WP
Outsourced safety calculations	Deploying complex or resource intensive calculations within the safety system software is hindered due to restrictions resulting from the development or execution environment, or the requirements of a followed standard. Therefore, execute the algorithms requiring complex calculations and/or functionality outside the safety system scope. Related standards: (IEC 61508, 2010)	WPC
Parametrized limp-mode	A limp-mode requires distinct functionality or operation and this need to be implemented alongside normal system control functionality. Therefore, on activation of the limp-mode, load limp-mode dedicated parameters to the control software to alter the operation of the system accordingly to the mode.	PC
Progressing safety restrictions	Activation of a safety function will decrease the potential of a machine to cause harm, but the activation of a safety function also likely causes an interruption in use that is typically an unwanted side-effect. Therefore, react to evolving hazardous situations by restricting, for instance, power and speed of the machine, and warn users so that appropriate actions are taken in time and shutting down the machine is avoided.	WP

Table 7.1 – continued from previous page

Pattern	Patlet	Status
Reference point switch	A continuous measurement (sensor) can drift from the actual value due to, for example, wearing, aging, heating, and catching dirt in the measuring element. Due to the drift, the measurements provided can compromise safety functions. For example, a measure could never reach a zero point. Therefore, in addition to the continuous measurement, add a (secondary) binary measurement that is used to provide a reference point to calibrate the continuous measurement.	WP
Safety adapter	Including the possibility to interface with all the possible data providers would increase the complexity of the safety system in terms of software and hardware. Therefore, implement interfacing, pre-processing, and filtering of the measurement (or other kinds of data) in a safety adapter component that plugs into the safety system.	WP
Safety documentation templates	Producing an acceptable documentation for a safety system, and achieving increased confidence of compliance with a followed standard is not a trivial task initially. Therefore, develop and apply documentation templates for (the software parts of) the safety system development to support the certification and development processes. Related standards: (IEC 61508-3, 2010, Annex A and B)	WP
Single packet tactic	When a failed element recovers into a normal operational state, the communication between the element and other connected elements should need to continue in a normal way. Specifically, the recovery should happen without the need to explicitly setup the communication (i.e. handshake etc.). Therefore, implement a stateless communication scheme between the communicating elements so that each sent and received package includes all the information required by the consumer.	WPC
Software fuse	After the safe state has been entered, it must be made sure that the state is maintained until appropriate actions are taken by the operator. Therefore, when entering the safe state, transfer the safety system into an inactive state from which it cannot be recovered except through a restart.	WPC

Table 7.1 – continued from previous page

Pattern	Patlet	Status
State change permission	Transitions between states of a system may have undesired consequences if taken when components and subsystems have inconsistent states, e.g., when some components and elements are in inappropriate ranges. Therefore, check that a state transition is safe and will not cause danger before initiating the transition.	WP
Synchronized control device actuation	A system should ensure that controls are actually affected by the user as intended. Therefore, implement a checking functionality to ensure that a control is not permanently forced (e.g. taped) into a certain state that enables usage in single hand mode. Related standards: (ISO 13851, 2002)	WP
Traceability-driven change management	How to minimize the re-work related to changes in existing safety systems of machines when changes are required or when all future configurations are not known beforehand so that parametrisation could be used? Start by identifying the first artefacts in the traceability chain that require modifications because of the changes. Related standards: (IEC 61508-3, 2010, sections 7.4, 7.8.2, Annex A)	WP
Two-level safety functions	A complete shutdown of a machine including electronics, hydraulic, motors, etc., is typically able to take the machine into a Safe state (Eloranta et al., 2014), but such an approach can also decrease the availability of the machine and increase the time to revert back to operational state. Therefore, apply two kinds of stopping approaches. The primary one tries to stop the machine in a controlled manner whereas the secondary one relies on shutting down the machine completely.	WP
United integrity level	In order to deploy software elements of different integrity levels (PL/SIL) on a single execution environment, the separation between the software elements needs to be shown and documented. This increases complexity of the system and the development effort, e.g., in terms of documentation. Therefore, elevate the integrity level of each element deployed on a shared execution environment to match the highest required integrity level of all the software elements. Related standards: (IEC 61508-3, 2010, section 7.4.2.9)	WP

Table 7.1 – continued from previous page

Pattern	Patlet	Status
Units of measurement	Applying different units of measurement may cause logical mistakes in a software operation and result in severe consequences in the context of safety functionality. Therefore, take units of measurement (e.g., length, time, or mass) and a unit prefixes (e.g., mega, nano, or pico) as a part of a strong type system. Related standards: (ISO 13849-1, 2015, section 4.6.2)	WPC
Tedious safe state by-pass	An existence of a way to overcome restrictions introduced by a safety (or similar) system may encourage to use a system to an unintended purpose even though such functionality should only be used to transfer the system from an acute hazardous state to more suitable one. Therefore, make the safety function bypass possible, but ensure that nobody wants to use the by-pass mode for anything but the emergency situation.	WP
User-warning hazard indication	How to assist a user of a system to achieve safe operation of the system in presence of underlying hazards for which it is not necessary to develop safety functions? Inform the user about the detected hazardous situation and leave the responsibility of appropriate reaction to the user.	WP
Validate non-trusted input data	Data needs to be fed into a safety system outside its scope, for instance, through a non-safety-related part of the system which effectively invalidates the integrity and credibility of the data from safety system point of view. Therefore, ask for user confirmation that the data is correct.	WP
XOR-monitoring	An AND gate produces correct output in case of input combinations of 0 0, 0 1 or 1 0 -> Control = 0, but it cannot indicate fault states of the inputs (1 0 and 0 1). Therefore, add an XOR gate parallel to the AND gate and provide it with the same inputs as the AND gate to produce an alert signal for further use.	WPC

7.3.2 Pattern Topic Categorization

Figure 7.4 categorizes the patterns given in table 7.1 under the topics defined in Figure 7.3. The topics describe the main area of concern regarding the included patterns. That is, for instance, the topic of the Monitored safety limit pattern belongs to *System architecture.*

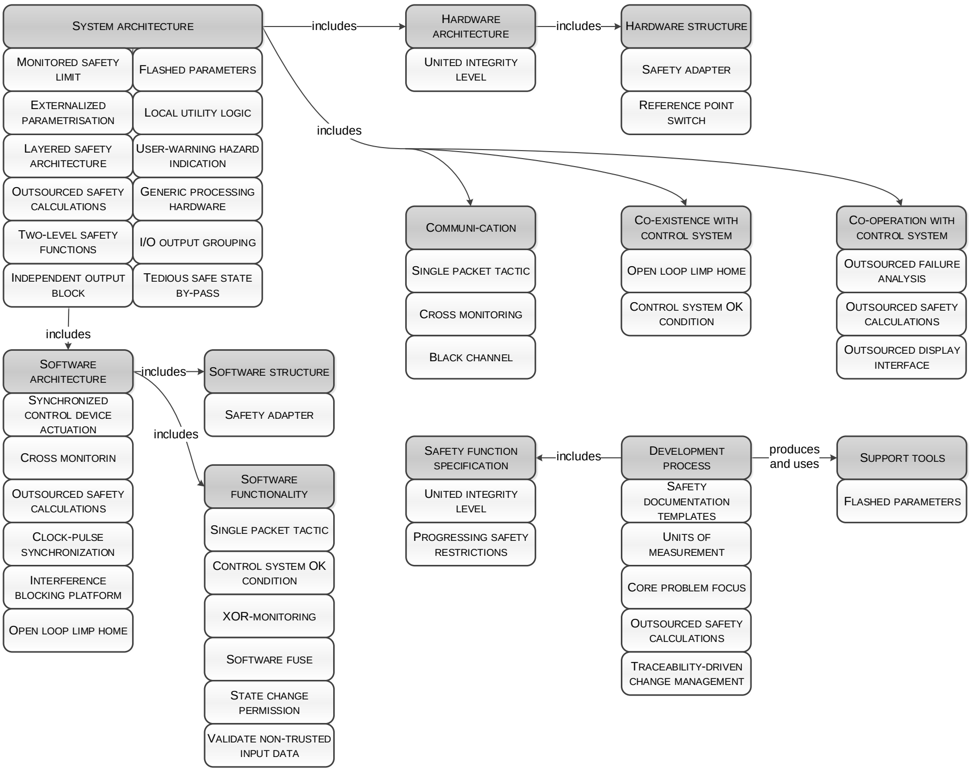

System architecture
includes
Hardware architecture
includes
Hardware structure
Monitored safety limit
Flashed parameters
United integrity level
Safety adapter
Externalized parametrisation
Local utility logic
Reference point switch
includes
Layered safety architecture
User-warning hazard indication
Outsourced safety calculations
Generic processing hardware
Communi-cation
Co-existence with control system
Co-operation with control system
Two-level safety functions
I/O output grouping
Single packet tactic
Open loop limp home
Outsourced failure analysis
Independent output block
Tedious safe state by-pass
Cross monitoring
Control system OK condition
Outsourced safety calculations
includes
Black channel
Outsourced display interface
Software architecture
includes
Software structure
Synchronized control device actuation
Safety adapter
includes
Safety function specification
includes
Development process
produces and uses
Support tools
Cross monitorin
Software functionality
United integrity level
Safety documentation templates
Flashed parameters
Outsourced safety calculations
Single packet tactic
Progressing safety restrictions
Units of measurement
Clock-pulse synchronization
Control system OK condition
Interference blocking platform
Core problem focus
XOR-monitoring
Open loop limp home
Outsourced safety calculations
Software fuse
Traceability-driven change management
State change permission
Validate non-trusted input data

Figure 7.4: Functional safety system development patterns arranged according to their topic. Reproduced from (Rauhamäki and Vepsäläinen, 2016).

7.3.3 Pattern Purpose Categorization

Figure 7.5 illustrates the purpose of the patterns. Here, the purpose describes the target outcome or result or the aspect enhanced, supported, or increased as a result of the application of the pattern. Some of the patterns illustrated in Figure 7.5 have been included in multiple categories. Some of the patterns could be potentially included in a number of other categories too, but for the sake of clarity, such a detailed categorization is omitted here. The following list defines the purpose categories in which the patterns presented in this chapter belong to. Where named identically, the categories extend the purpose categories introduced in Section 5.3.3.

Control system co-existence A control system can supplement the safety system and potentially take some of the tasks that should be otherwise implemented by the safety system. The latter approach likely reduces the development cost of the safety system so there is a rationale to minimize the scope of the safety system in terms of functionality.

Quality attribute enhancement A set of quality attributes can be used to reflect the quality of a system. Application of a design pattern typically promotes some of the attributes but may hinder others. In this classification, the most dominant attribute has been

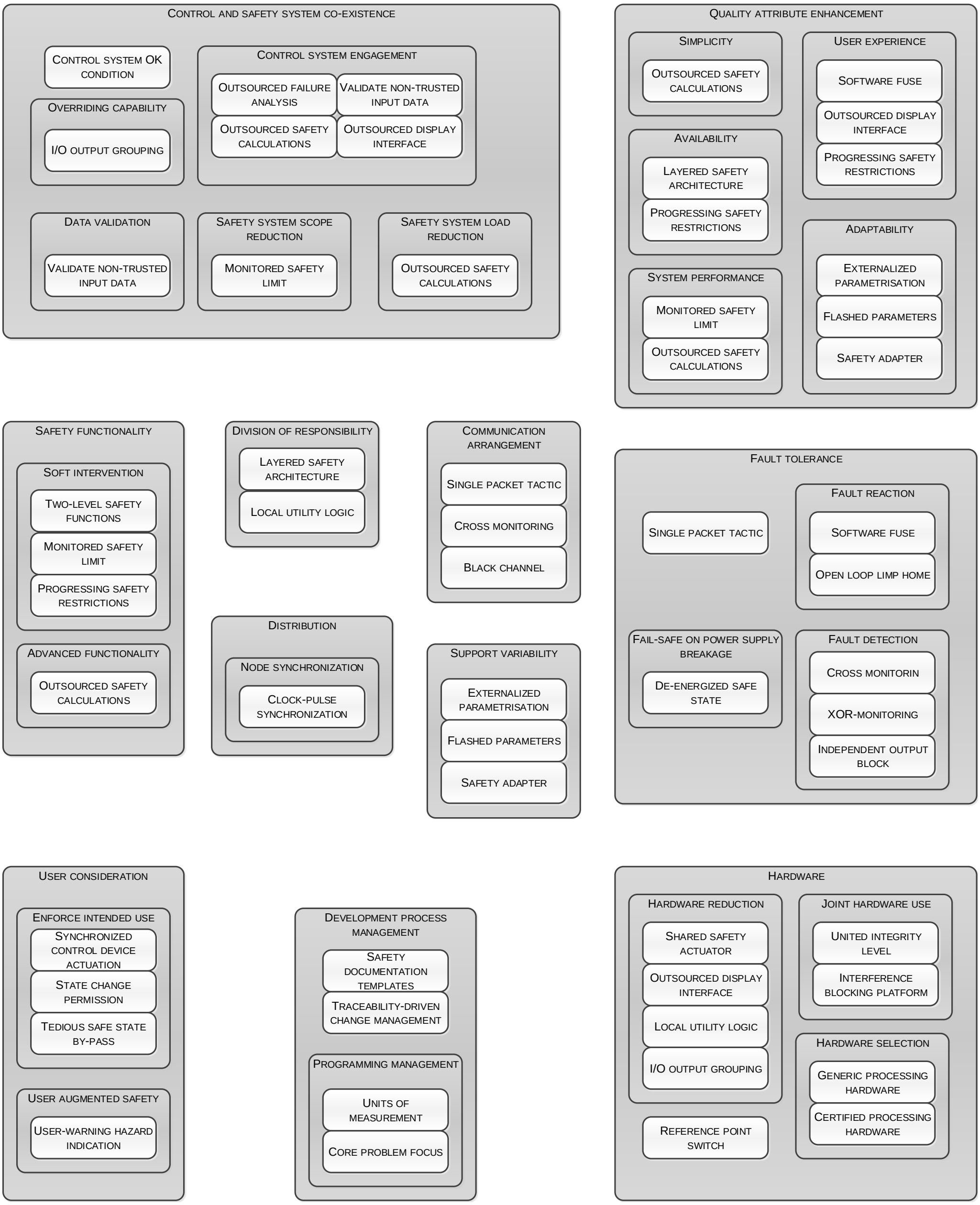

Control and safety system co-existence
Control system OK condition
Overriding capability
I/O output grouping
Control system engagement
Outsourced failure analysis
Validate non-trusted input data
Outsourced safety calculations
Outsourced display interface
Data validation
Validate non-trusted input data
Safety system scope reduction
Monitored safety limit
Safety system load reduction
Outsourced safety calculations
Quality attribute enhancement
Simplicity
Outsourced safety calculations
Availability
Layered safety architecture
Progressing safety restrictions
System performance
Monitored safety limit
Outsourced safety calculations
User experience
Software fuse
Outsourced display interface
Progressing safety restrictions
Adaptability
Externalized parametrisation
Flashed parameters
Safety adapter
Safety functionality
Soft intervention
Two-level safety functions
Monitored safety limit
Progressing safety restrictions
Advanced functionality
Outsourced safety calculations
Division of responsibility
Layered safety architecture
Local utility logic
Distribution
Node synchronization
Clock-pulse synchronization
Communication arrangement
Single packet tactic
Cross monitoring
Black channel
Support variability
Externalized parametrisation
Flashed parameters
Safety adapter
Fault tolerance
Single packet tactic
Fault reaction
Software fuse
Open loop limp home
Fail-safe on power supply breakage
De-energized safe state
Fault detection
Cross monitorin
XOR-monitoring
Independent output block
User consideration
Enforce intended use
Synchronized control device actuation
State change permission
Tedious safe state by-pass
User augmented safety
User-warning hazard indication
Development process management
Safety documentation templates
Traceability-driven change management
Programming management
Units of measurement
Core problem focus
Hardware
Hardware reduction
Shared safety actuator
Outsourced display interface
Local utility logic
I/O output grouping
Reference point switch
Joint hardware use
United integrity level
Interference blocking platform
Hardware selection
Generic processing hardware
Certified processing hardware

Figure 7.5: Functional safety system development patterns arranged according to their purposes

considered when the patterns have been assigned to the categories.

Safety functionality

In some cases, a relatively straightforward functionality for a safety system is sufficient. However, in some cases, more elaborated functionality is desirable due to, among other things, performance or user experience reasons. The patterns in this category support achieving advanced safety functionality.

Division of responsibility

In a typical case, the elements of a safety system need to be partitioned and modularized. The patterns in this category suggest an approach to assign responsibilities to the modules and parts of the system. Both hardware and software aspects need to be considered.

Distribution

In some cases, a safety system may utilize multiple nodes operating in collaboration to implement a safety function. The patterns in this category aim to enable the distribution.

Communication arrangement

This category is related to the distribution category. When the nodes or elements of a system are distributed, they need to communicate in order to implement the desired functionality.

Fault tolerance

These patterns support the ability of a safety system to retain safety in the presence of faults. Typically, this means reverting the system under control into a safe state when a failure or error is identified.

Variability support

Changing a safety system in terms of software or hardware is not a trivial or easy task. Therefore, such a situation should be avoided. One way to achieve this is to add variability to the safety system so that it can adapt to the current and potentially future uses and environments.

Hardware

Although safety system logic is implemented in software, some hardware is still needed. The selection and structure of hardware typically affect the development process and the price of the safety system in terms of hardware. The patterns in this category suggest approaches to various situations to select a suitable hardware configuration for the system in design.

User consideration

Users can both augment and hinder the operation of a safety system. Therefore, user actions should be considered in safety system design by providing the user with possibilities to retain the system in its safe operating range and on the other hand trying to enforce the users to operate the system as it was designed.

Development process

In the context of safety system development, the development process typically generates a considerable amount of costs. Approaches to minimize and ease the process help to reduce the development cost similarly.

7.3.4 Pattern Language Part

Figure 7.6 illustrates the pattern language related to the patterns and pattern candidates presented in this chapter (solid line) and documented in (Rauhamäki and Vepsäläinen, 2016). The language has been augmented with selected patterns of author's previous work (dash line) including patterns and pattern candidates from (Rauhamäki et al., 2015). To supplement the language even further, patterns from external sources have also been included to provide connections between pattern languages. The external patterns (dash dot dot line) are referenced from (Eloranta et al., 2014; Freeman et al., 2004; Hanmer, 2007).

The language part shown here is made from the point of view of the patterns and pattern candidates presented in this chapter. Therefore, it does not directly reflect the pattern language parts given in Figures 5.5 and 6.5. Nevertheless, Separated safety can be seen as a centric pattern also in this language part. It provides grounds for various other patterns expanding into larger pattern groups in the language.

7.4 Discussion

The patterns presented in this chapter do not necessarily increase the overall safety of the target system as such. Instead, they help the system designer and manufacturers to implement the safety functions in a cost efficient and development efficient way. However, this can also be seen as a viewpoint to increase safety. It is not always necessary to put every potential element of the safety-related parts of the control system to the safety system. This increases development effort as the standards considering functional safety system (software) development typically require tasks that are not necessary for normal control application development. The saved development effort can then be invested in new safety functions or other issues such as the removal of a hazard, which also promote the overall safety properties of the system in design.

The pattern candidate approach worked well. It was found efficient to present the idea of a solution or an approach in the form of a pattern candidate to the industry representatives. This helped to gain feedback and new known uses regarding the candidates.

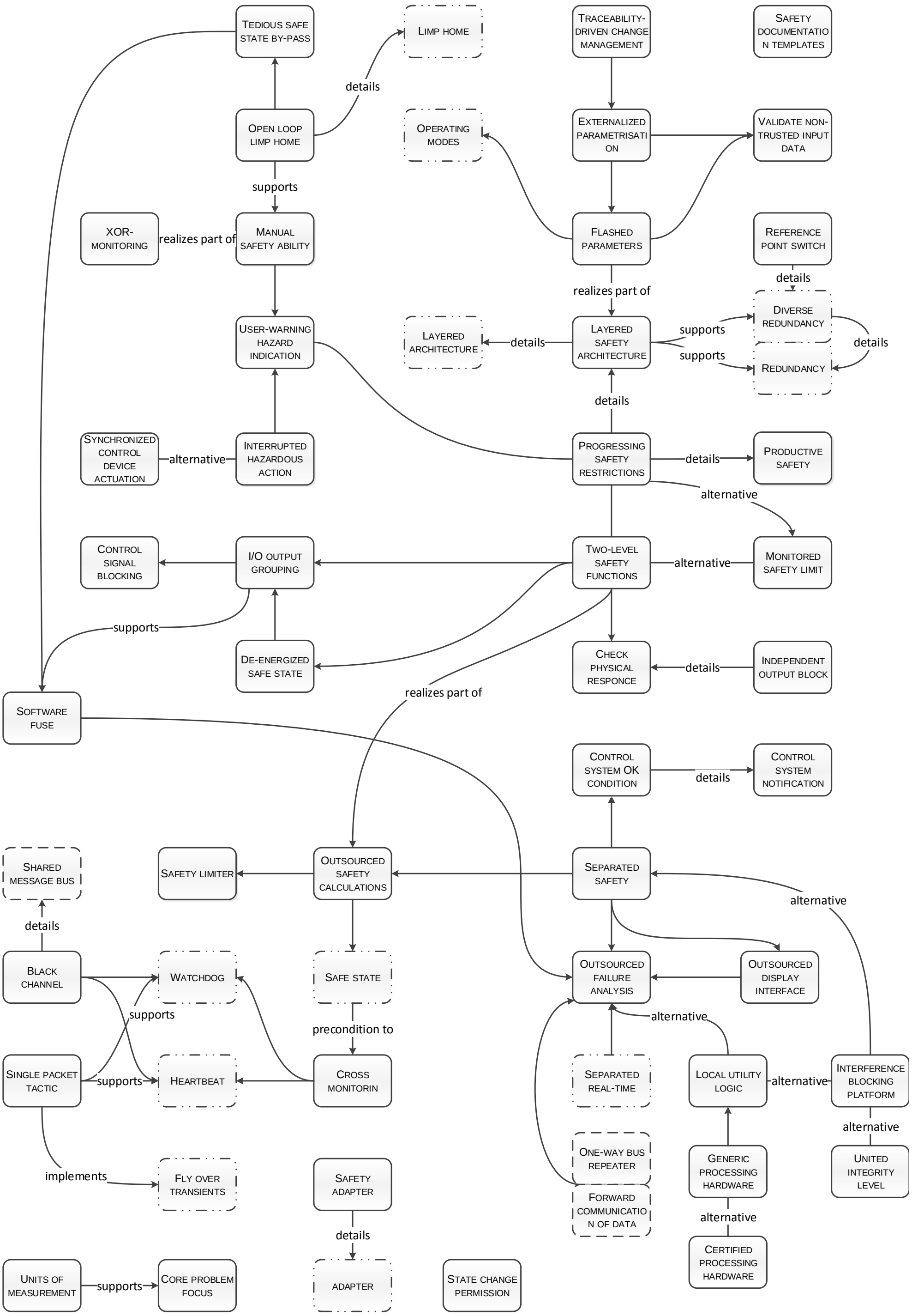


Figure 7.6: Pattern language part of the functional safety system development patterns. Reproduced from (Rauhamäki and Vepsäläinen, 2016).

8 Summary and Discussion

8.1 Research Questions Revisited

RQ1: How can design patterns support the development of functional safety systems?

When mined with an appropriate approach, design patterns present solutions and approaches that have been successfully applied in the design and development of functional safety systems. In such cases, patterns can be considered to reflect the concepts of well-tried solutions or components preferred by standards such as (IEC 61508, 2010; ISO 13849-1, 2015). When patterns are arranged in the format of a pattern language, the design process can at least partially become a decision-making process where the relations between the patterns indicate the suggested order and other inter-pattern aspects. In the context of functional safety systems, design patterns can help to alleviate the bureaucracy by indicating approaches to fulfil certain requirements or sections of standards or directives. Another way to cope with the standards and their requirements is to find ways to minimize the scope of functional safety systems, for which the patterns in the context of this thesis were also identified.

In addition to patterns and the pattern languages, a pattern mining process is also a resource that can be used to support the development of functional safety systems. A mining process aims to transfer tacit knowledge from the designers' experience and the documentation of existing systems into the explicit format of the design patterns and pattern languages. Patterns help to transfer information between parties, partly because of the information becoming explicit. In addition, if a pattern mining process involves multiple industrial participants and workshops, discussions between the parties help to spread knowledge and ideas in the industry. Finally, the same aspects can help to increase the awareness of the applied solutions. The increased awareness can encourage one to apply a similar approach and thus increase the available solution space from which to select the most suitable approaches to the design in hand.

RQ2: Is there a set of commonly applied solutions for functional safety system designs utilized and known by domain experts and practitioners? Which topic categories do the solutions contribute and belong to? What purposes do the solutions serve?

Solutions for functional safety systems, their structure, functionality, and development applied commonly in various machines and systems were identified. For many of the solutions and approaches, at least three independent known uses were found during the research. The solutions and approaches having at least three known uses were considered patterns in the context of this thesis whereas the solutions which lack at least three known uses were considered pattern candidates. In addition to the three known uses, the patterns and pattern candidates were exposed to criticism and discussion in pattern workshops. The discussions supported the existence and applicability of several of the patterns and pattern candidates. However, some of the solutions and approaches documented in the patterns received justified criticism, and therefore the usage of the corresponding patterns was discouraged.

The identified patterns considered several topics including architecture, co-operation and co-existence with a control system, hazard management, and development processes regarding functional safety systems. The patterns promoted and enhanced, among others, fault tolerance, safety system scope reduction, co-operation, hazard and risk mitigation, functionality, and user consideration. Most of the patterns emerged from machinery applications, but some of them were recognized to also apply in process control applications.

RQ3: How do the design patterns for functional safety system development relate to each other? What kind of whole emerges from the identified design patterns? What kind of relationships can be used the describe the relations between the design patterns?

The pattern language and the relations between the patterns have been formed incrementally through the research process. The pattern language parts presented in chapters 5, 6, and 7 link to and complement each other and patterns from other pattern languages to build a broader whole than the individual parts of the language alone. An illustration of the whole pattern language based on the patterns presented in this thesis is given in Figure 8.1.

The patterns can, among other things, support, detail, benefit from, require the presence of, utilize, enable, realize in, and provide an alternative to other patterns. In addition, the patterns can be arranged to a typical or preferred order of application. The relations between the patterns form the resulting pattern language. The pattern language format helps a user of the patterns to select and apply suitable patterns. The relations between the patterns form paths through the language that supports especially designers with less experience in functional safety system design and development.

RQ4: How to document emerging solution models and approaches applied in the field of functional safety system engineering into a design pattern format? Is the generic design pattern format suitable for documenting the commonly known solutions? Is there a need for a specialized format or elements to be used for the solutions in the functional safety system domain?

The identified solutions were documented in a design pattern format applying a modified version of the canonical pattern format. The applied format utilizes the fields introduced by the canonical format, excluding the resulting context and rationale fields. The resulting context field was considered to be redundant with the context emerging from the solution description. Similarly, the rationale was reflected in the consequences field, stating the pros and cons of the approach.

The only safety system specific field used is the related standards field. This field provides links to the standards and their sections that the described solution may, among others, implement, support, hinder or conflict with. The field was used whenever a clear relation to a certain standards or preferably a standard section was identified. When a certain standard is followed in the development of a functional safety system, the developer needs to ensure and justify that all the requirements of the standard are fulfilled. When a pattern is applied, the link makes it easier to trace which standard section the solution covers completely or partly. This eases the burden of the developer as the relevant standard sections can be identified with less effort.

During the pattern workshops (described in sections 3.3 and 7.2.2) it was noted that, in most cases, the participants were able to understand the patterns and read the approach as intended. This aspect supports the pattern approach as a format to document the solutions applied in the development of functional safety systems.

8.2 Discussion of the Study

8.2.1 Pattern Mining Approach

The patterns and the pattern language described in this thesis are targeted for the domain of functional safety systems. The goal for the research work carried out for this study has been to provide support for the designers of functional safety systems. This has affected the applied pattern mining process. The development of functional safety systems is heavily based on standards. The standards provide requirements and recommendations considering the work and design flows, architecture and methods to be used in the development process of functional safety systems. Therefore, the applicable standards can also be seen as a possible foundation for design patterns considering functional safety systems and their development.

As a general guideline in the pattern community, a credible pattern is expected to have three known uses (Holzner, 2006, p. 282) (Kohls and Panke, 2010) (Eloranta et al., 2014, p. 88). This aspect forms a guideline for pattern mining as a mark of credibility of the pattern. Still, a design solution without three known uses can be considered useful. The solution or approach can, regardless of the number of the known instances, solve a problem encountered by a designer. In the context of this thesis, three known uses have been pursued for all the patterns. However, the lack of three known uses has not been taken as a restriction to propose pattern if it has had the potential to support a designer.

In the initial pattern mining process (see Section 5.2), the guideline of three known uses was overridden by the potential to provide the designers with ideas and approaches to be applied in the development of the functional safety systems. This made it possible to introduce the pattern candidates for public discussion in an early state of the research. Later, known uses from the industry and literature were obtained for the patterns.

The evolved version of the pattern mining process (see Section 6.2) gave more thought on the three known uses required for patterns. Consequently, three known uses were obtained before a pattern was published. Regardless, this phase of the study leaned on literature sources rather than known uses from the practitioners or the documentation of real world applications. Still, the resulting patterns align well with the applicable standards and practices considering functional safety system development and achieving safety.

The final pattern mining approach (see Section 7.2) was the closest match with the process described by Eloranta (2015, p. 39). In this phase of the study, co-operation with practitioners from industry was established to first identify new applied solutions and then learn new known uses for them and previously identified patterns. Although interviews were a centric data collection method in this phase of research, pattern candidates were also used. That is, patterns without three known uses were also shown to industry representatives. It is acknowledged that such an approach to gather known uses can be problematic as the interviewee could try to match a more or less different solution to the pattern. However, the same problem can also occur when a researcher matches the statements of interviewees to existing patterns. During the interviews, the industry representatives had the opportunity to decline that the approach is known or used or leave the question unanswered.

The early documentation and publication of the patterns also had some apparent benefits. The most visible benefit was the ease of discussion with practitioners. It was relatively easy to begin a discussion with industry representatives considering a solution or approach by showing an initial concept, draft, or picture of the solution. During the DPSafe project, many of the known uses were found by showing and discussing a pattern draft. One of the factors may relate to the fear of giving away a valuable piece of information. However, as the pattern is already documented, it may encourage the interviewees to share their knowledge as it is already apparent that the researcher knows the pattern. In such a case, the interviewed persons do not disclose unique organizational secrets by providing or sharing the solution with the researchers.

8.2.2 Applicability of the Patterns

The patterns for which known uses from industry have been identified have also been implicitly applied in industrial project environments. These patterns include at least the ones mentioned in Functional Safety System Designer's Handbook - Design Patterns for Safety System Development (Rauhamäki and Vepsäläinen, 2016), Chapter 7, and many of the ones presented in Chapter 5 for which new known uses were identified during the DPSafe project. However, the documented patterns and the pattern language have not been tested in a new real-world project environment after the patterns and the pattern language have been compiled. In discussions with industry representatives, the patterns and the pattern language have been mentioned as having a good potential and value. The value was seen in the mined patterns, the pattern language, and the pattern mining process. For example, a DPSafe project participant mentioned that 'the project implied a very good value for the invested resources.' The participants have considered the patterns to provide new ideas and support the use of the approaches as some demonstration of previous usage has been given. However, no organized test to validate the patterns and their suitability in practical design work has been established. In the context of this thesis, the validity of each pattern lies on the known uses and the workshop discussions regarding the patterns.

The produced pattern language cannot be considered to cover all the aspects related to the development of functional safety systems. Arguably, some categories and topics are omitted and missing from the language. However, patterns and pattern languages are artifacts that should be considered to be constantly evolving. This, alongside an open community of industry representatives to be approached with the pattern mining process described in Section 7.2, provides a good premise for extending and enhancing the pattern language as well as the individual patterns of the language.

8.3 Future Work

The future work related to the thesis includes the extension of the pattern language as well as the validation of the patterns. The need to extend the language emerges from the fact that the pattern language and the patterns do not fully cover the field of functional safety system development. Most likely, new patterns to augment the current language could be discovered by organizing more interviews or obtaining documents etc. of functional safety systems to be researched and studied. The patterns and the pattern language provide a core collection of the results that have been obtained so far rather than a final truth or state of the matters. This also reflects the nature of design patterns. They are constantly evolving artifacts that should never be considered finished, instead of being once written and cemented ever after.

Another work not considered in this thesis is the validation of the patterns. This can be seen to include two viewpoints: gathering more known uses for the patterns and trying to apply them in industrial development projects. Known uses can be, for instance, obtained by arranging interviews on functional safety systems and their development as done in the final pattern mining approach of this thesis. When a new known use is identified, it is quite straightforward to augment the known use in the pattern description.

The latter aspect, that is, the validation of the patterns in industrial development projects, is intentionally left outside the scope of the thesis. Arranging such a validation would require a completely different research approach and experimentation compared with the approach in the context of this thesis. The interesting questions here would include, among others:

- What kind of effect do a design pattern approach provide on development efficiency?
- Do design patterns help to increase the quality of functional safety systems in terms of, for example, cost efficiency, dependability, or new safety functions?
- Is it easier for a novice designer to start with or without patterns?
- How would the patterns and the pattern language be applied in practical designs? Would they be used separately or sequentially as suggested by the language?

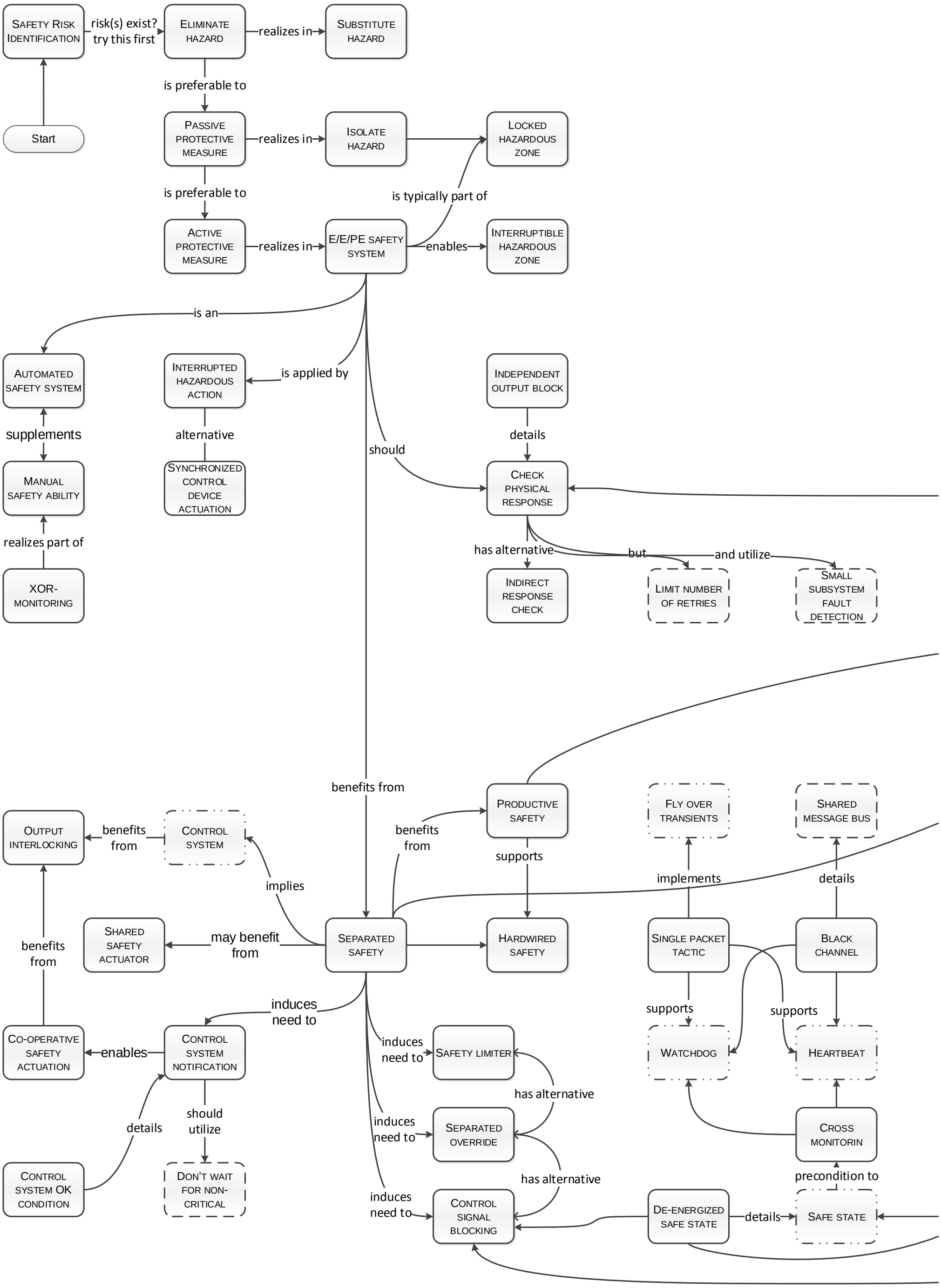
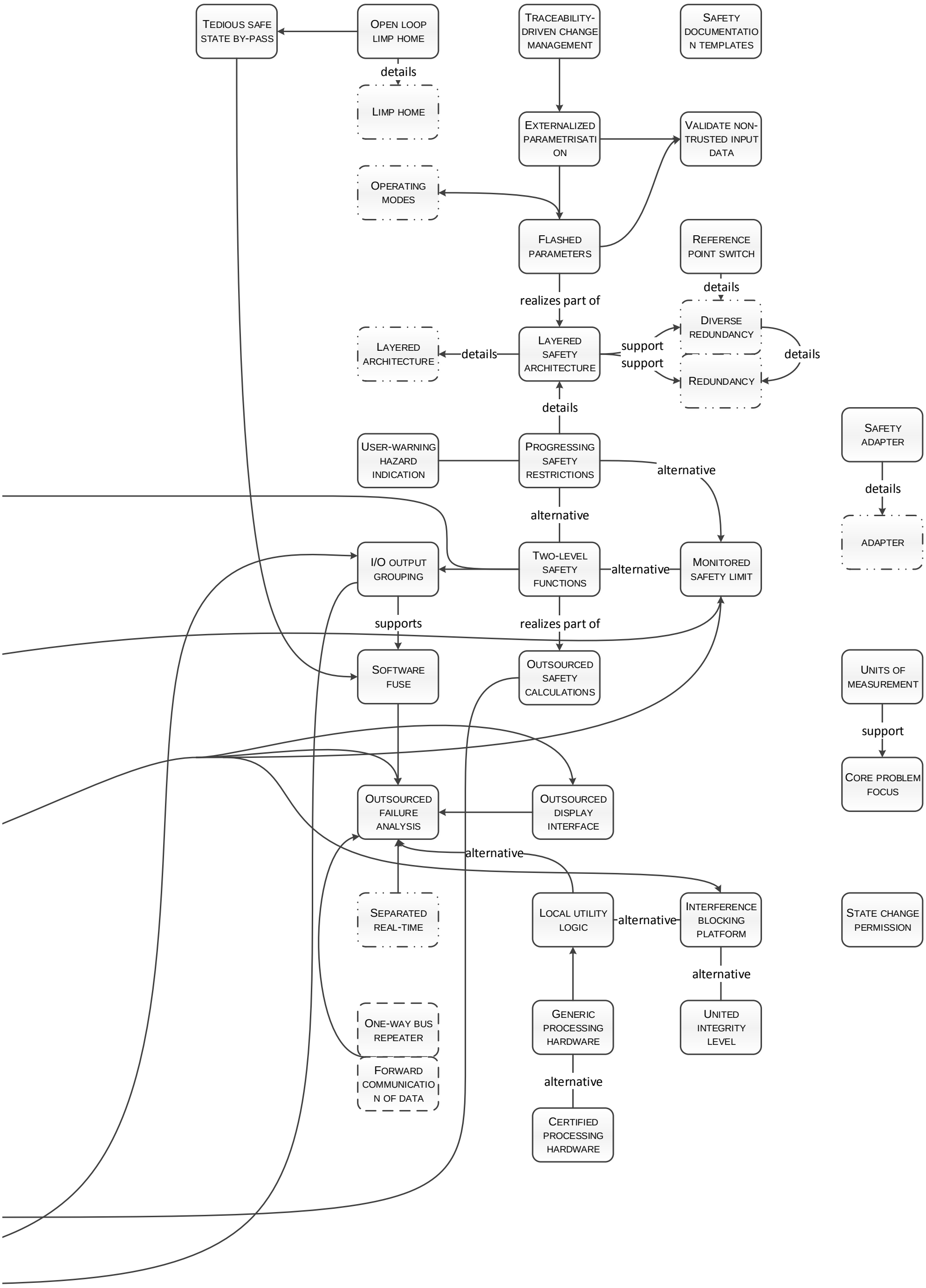


Figure 8.1: Pattern language for functional safety system design and development


Tedious safe state by-pass
Open loop limp home
Traceability-driven change management
Safety documentation templates
details
Limp home
Externalized parametrisation
Validate non-trusted input data
Operating modes
Flashed parameters
Reference point switch
details
realizes part of
Diverse redundancy
Layered architecture
details
Layered safety architecture
support
support
details
Redundancy
details
Safety adapter
User-warning hazard indication
Progressing safety restrictions
alternative
details
alternative
Adapter
I/O output grouping
Two-level safety functions
alternative
Monitored safety limit
supports
realizes part of
Software fuse
Outsourced safety calculations
Units of measurement
support
Core problem focus
Outsourced failure analysis
Outsourced display interface
alternative
Separated real-time
Local utility logic
alternative
Interference blocking platform
State change permission
alternative
One-way bus repeater
Generic processing hardware
United integrity level
Forward communication of data
alternative
Certified processing hardware

Bibliography

Aldrich, M. (2001, Aug) History of workplace safety in the united states, 1880-1970. EH.Net Encyclopedia. [Online]. Available: http://eh.net/encyclopedia/history-of-workplace-safety-in-the-united-states-1880-1970/

Alexander, C., *The Timeless Way of Building*, ser. Center for Environmental Structure Berkeley, Calif: Center for Environmental Structure series. Oxford University Press, 1979, no. 8. [Online]. Available: https://books.google.fi/books?id=H6CE9hlbO8sC

Alexander, C., Ishikawa, S., and Silverstein, M., *A Pattern Language: Towns, Buildings, Construction*, ser. Center for Environmental Structure Berkeley, Calif: Center for Environmental Structure series. OUP USA, 1977. [Online]. Available: https://books.google.fi/books?id=hwAHmktpk5IC

Alho, P. and Rauhamäki, J., "Patterns for light-weight fault tolerance and decoupled design in distributed control systems," *LNCS Transactions on Pattern Languages of Programming*, In press.

Appleton, B. (2000, Feb) Patterns and software: Essential concepts and terminology. [Online]. Available: http://www.sci.brooklyn.cuny.edu/~sklar/teaching/s08/cis20.2/papers/appleton-patterns-intro.pdf

Armoush, A., "Design patterns for safety-critical embedded systems," Dissertation, Embedded Software Laboratory - RWTH Aachen University, June 2010, aIB-2010-13. [Online]. Available: http://aib.informatik.rwth-aachen.de/2010/2010-13.pdf

Avizienis, A., Laprie, J.-C., Randell, B., and Landwehr, C., "Basic concepts and taxonomy of dependable and secure computing," *IEEE Trans. Dependable Secur. Comput.*, vol. 1, no. 1, pp. 11–33, Jan. 2004. [Online]. Available: http://dx.doi.org/10.1109/TDSC.2004.2

Bartneck, C., Kulić, D., Croft, E., and Zoghbi, S., "Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots," *International Journal of Social Robotics*, vol. 1, no. 1, pp. 71–81, 2009. [Online]. Available: http://dx.doi.org/10.1007/s12369-008-0001-3

Bass, L., Clements, P., and Kazman, R., *Software Architecture in Practice*, 3rd ed. Addison-Wesley Professional, 2012.

Beck, K. and Cunningham, W., "Using pattern languages for object oriented programs," in *Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*, 1987. [Online]. Available: http://c2.com/doc/oopsla87.html

Boehm, B., “Managing software productivity and reuse,” *Computer*, vol. 32, no. 9, pp. 111–113, Sep 1999.

———, “A spiral model of software development and enhancement,” *SIGSOFT Softw. Eng. Notes*, vol. 11, no. 4, pp. 14–24, Aug. 1986. [Online]. Available: http://doi.acm.org/10.1145/12944.12948

Broadribb, M. P., “It’s people, stupid!: Human factors in incident investigation,” *Process Safety Progress*, vol. 31, no. 2, pp. 152–158, 2012. [Online]. Available: http://dx.doi.org/10.1002/prs.11475

Buede, D., *The Engineering Design of Systems: Models and Methods*, 2nd ed., ser. Wiley Series in Systems Engineering and Management. Wiley, 2011.

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., and Stal, M., *Pattern-Oriented Software Architecture - Volume 1: A System of Patterns.* Wiley Publishing, 1996.

Buschmann, F., Henney, K., and Schmidt, D. C., *Pattern-Oriented Software Architecture, Volume 4: A Pattern Language for Distributed Computing.* Chichester, UK: Wiley, 2007.

Carnegie Mellon University. (2015) Published software architecture definitions. [Online]. Available: http://www.sei.cmu.edu/architecture/start/glossary/published.cfm

Center for Chemical Process Safety, *Guidelines for Engineering Design for Process Safety.* Wiley, 2010. [Online]. Available: https://books.google.fi/books?id=mDgJjpmj950C

———, *Guidelines for Engineering Design for Process Safety*, 2nd ed. Wiley-AIChE, May 2012.

Chugh, R., “Do australian universities encourage tacit knowledge transfer?” in *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 2015, pp. 128–135.

Deugo, D., Kendall, E., and Weiss, M. (1999, Nov) Agent patterns. [Online]. Available: http://people.scs.carleton.ca/~deugo/Patterns/Agent/Presentations/AgentPatterns/index.htm

Dorf, R. and Bishop, R., *Modern Control Systems.* Pearson Prentice Hall, 2005.

Douglass, B., *Real-time Design Patterns: Robust Scalable Architecture for Real-time Systems*, ser. Addison-Wesley object technology series. Addison-Wesley, 2003. [Online]. Available: https://books.google.fi/books?id=drlsKjcw3xQC

Douglass, B. P., *Doing Hard Time: Developing Real-time Systems with UML, Objects, Frameworks, and Patterns.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.

———, *Design Patterns for Embedded Systems in C: An Embedded Software Engineering Toolkit*, 1st ed. Newton, MA, USA: Newnes, 2010.

Edwards, R. and Holland, J., *What is Qualitative Interviewing?*, ser. The ’What is?’ Research Methods Series. Bloomsbury Publishing, 2013. [Online]. Available: https://books.google.fi/books?id=GdCOAQAAQBAJ

Eloranta, V.-P., *Techniques and Practices for Software Architecture Work in Agile Software Development*, ser. Tampere University of Technology. Publication;1293. Tampere University of Technology, 5 2015.

Eloranta, V.-P., Koskinen, J., Leppänen, M., and Reijonen, V., *Designing Distributed Control Systems: A Pattern Language Approach.* Chichester, United Kingdom: John Wiley & Sons, Ltd., 2014.

Ericson, C., *Concise Encyclopedia of System Safety: Definition of Terms and Concepts.* Wiley, 2011. [Online]. Available: https://books.google.fi/books?id=uousK00QAREC

Erl, T., *SOA Design Patterns*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2009.

European Commission. (2016, May) Summary list of titles and references harmonised standards under directive 2006/42/ec for machinery. Machinery Safety 101. [Online]. Available: http://ec.europa.eu/growth/single-market/european-standards/harmonised-standards/machinery/index_en.htm

Eves, D. (2014, April) A brief history of the origins, development and implementation of health and safety law in the united kingdom, 1802−2014. British Safety Council. [Online]. Available: http://www.historyofosh.org.uk/brief/

Forsberg, K. and Mooz, H., "The relationship of systems engineering to the project cycle," *Engineering Management Journal*, vol. 4, no. 3, pp. 36−43, 1992. [Online]. Available: http://dx.doi.org/10.1080/10429247.1992.11414684

Fowler, M., *Patterns of Enterprise Application Architecture.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.

−−−. (2006, Aug) Writing software patterns. [Online]. Available: http://www.martinfowler.com/articles/writingPatterns.html

Freeman, E., Freeman, E., Sierra, K., and Bates, B., *Head First Design Patterns*, ser. Head first series. O'Reilly Media, 2004. [Online]. Available: https://books.google.fi/books?id=GGpXN9SMELMC

Friedrichsen, U. (2014, Nov) Patterns of resilience - a small pattern language. codecentric AG. [Online]. Available: https://www.slideshare.net/ufried/patterns-of-resilience

−−−. (2016, Apr) Resilience reloaded - more resilience patterns. codecentric AG. [Online]. Available: https://www.slideshare.net/ufried/resilience-reloaded-more-resilience-patterns

Gabriel, R. P., *Writer's Workshops and the Work of Making Things.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.

Gamma, E., *Objektorientierte Software-Entwicklung am Beispiel von ET++: Design-Muster, Klassenbibliothek, Werkzeuge.* Springer-Verlag Berlin Heidelberg, 1992. [Online]. Available: http://www.springer.com/gp/book/9783540560067

Gamma, E., Helm, R., Johnson, R., and Vlissides, J., *Design Patterns: Elements of Reusable Object-oriented Software.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995.

−−−, *Design Patterns: Abstraction and Reuse of Object-Oriented Design.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 701−717. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-59412-0_40

Glaser, B. and Strauss, A., *The Discovery of Grounded Theory: Strategies for Qualitative Research*, ser. Observations (Chicago, Ill.). Aldine Publishing Company, 1967. [Online]. Available: https://books.google.fi/books?id=oUxEAQAAIAAJ

Gomaa, H., *Real-Time Software Design for Embedded Systems.* Cambridge University Press, 2016. [Online]. Available: https://books.google.fi/books?id=6xh-CwAAQBAJ

Hanmer, R., *Patterns for Fault Tolerant Software.* Wiley Publishing, 2007.

Harrell, M. C. and Bradley, M. A., *Data collection methods : Semi-Structured Interviews and Focus Groups.* RAND Santa Monica, Calif, 2009.

Harrison, N., "The language of shepherds: A pattern language for shepherding," in *Proceedings of the 6th Annual Conference on the Pattern Languages of Programs (PLoP)*, Monticello, Illinois, USA., August 15-18 1999. [Online]. Available: http://jerry.cs.uiuc.edu/~plop/plop99/proceedings/harrison/shepherding4.pdf

Hentrich, C., Zdun, U., Hlupic, V., and Dotsika, F., "An approach for pattern mining through grounded theory techniques and its applications to process-driven soa patterns," in *Proceedings of the 18th European Conference on Pattern Languages of Program*, ser. EuroPLoP '13. New York, NY, USA: ACM, 2015, pp. 9:1−9:16. [Online]. Available: http://doi.acm.org/10.1145/2739011.2739020

Hevner, A. and Chatterjee, S., *Design Research in Information Systems: Theory and Practice*, 1st ed. Springer US, 2010.

Hevner, A. R., March, S. T., Park, J., and Ram, S., "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75−105, Mar 2004. [Online]. Available: http://dl.acm.org/citation.cfm?id=2017212.2017217

Hohpe, G. and Woolf, B., *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003.

Holzner, S., *Design Patterns For Dummies*, ser. −For dummies. Wiley, 2006. [Online]. Available: https://books.google.fi/books?id=uLU9k-mKvzoC

Hove, S. E. and Anda, B., "Experiences from conducting semi-structured interviews in empirical software engineering research," in *Proceedings of the 11th IEEE International Software Metrics Symposium*, ser. METRICS '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 23−. [Online]. Available: http://dx.doi.org/10.1109/METRICS.2005.24

HSE, "Agriculture (safety, health and welfare provisions) act 1956," Health and Safety Executive, Tech. Rep., 1956.

Iba, T., Miyake, T., Naruse, M., and Yotsumoto, N., "Learning patterns: A pattern language for active learners," in *16th Conference on Pattern Languages of Programs (PLoP)*, 2009. [Online]. Available: http://hillside.net/plop/2009/papers/People/Learning%20Patterns%20A%20Pattern%20Language%20for%20Active%20Learners.pdf

IEC. (2015) Functional safety. [Online]. Available: http://www.iec.ch/functionalsafety/explained/page1.htm

−−−. (2016) IEC 61508 FAQ - Edition 2.0 - Key concepts - Give me example architectures for the different modes of operation. [Online]. Available: http://www.iec.ch/functionalsafety/faq-ed2/page5.htm

IEC 61131-6, "Programmable controllers - part 6: Functional safety," International Electrotechnical Commission, IEC 61131-6, Dec 2012.

IEC 61508, "Functional safety of electrical/electronic/programmable electronic safety-related systems," European Committee for Electrotechnical Standardization, IEC 61508, May 2010.

IEC 61508-2, "Functional safety of electrical/electronic/programmable electronic safety-related systems − Part 2: Requirements for electrical/electronic/programmable electronic safety-related systems," European Committee for Electrotechnical Standardization, IEC 61508-2, May 2010.

IEC 61508-3, "Functional safety of electrical/electronic/programmable electronic safety-related systems − Part 3: Software requirements," European Committee for Electrotechnical Standardization, IEC 61508-3, May 2010.

IEC 61508-4, "Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 4: Definitions and abbreviations," European Committee for Electrotechnical Standardization, IEC 61508-4, May 2010.

IEC 61508-7, "Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 7: Overview of techniques and measures," European Committee for Electrotechnical Standardization, IEC 61508-7, Apr 2010.

IEC 61784-3, "Industrial communication networks − Profiles − Part 3: Functional safety fieldbuses − General rules and profile definitions," International Electrotechnical Commission, IEC 61784-3, May 2016.

IEC 62061, "Safety of machinery - functional safety of safety-related electrical, electronic and programmable electronic control systems," European Committee for Electrotechnical Standardization, IEC 62061, Jan 2005.

IEC 62280, "Railway applications - Communication, signalling and processing systems - Safety related communication in transmission systems," International Electrotechnical Commission, IEC 62280, Feb 2014.

IEEE 1471:2000, "IEEE Recommended Practice for Architectural Description for Software-Intensive Systems," IEEE Computer Society, IEEE Standard 1471:2000, 2000.

Isaku, T. and Iba, T., "Creative cocooking patterns: A pattern language for creative collaborative cooking," in *Proceedings of the 20th European Conference on Pattern Languages of Programs*, ser. EuroPLoP '15. New York, NY, USA: ACM, 2015, pp. 1−17. [Online]. Available: http://doi.acm.org/10.1145/2855321.2855366

ISO 12100:2010, "Safety of machinery. General principles for design. Risk assessment and risk reduction," European Committee for Standardization, ISO 12100:2010, Oct 2010.

ISO 13849-1, "Safety of machinery. Safety-related parts of control systems. Part 1: General principles for design (ISO 13849-1:2015)," European Committee for Electrotechnical Standardization, ISO 13849-1, Dec 2015.

ISO 13851, "Safety of machinery – Two-hand control devices – Functional aspects and design principles," International Organization for Standardization, ISO 13851, Mar 2002.

Jones, J. C., *Design methods : seeds of human futures.* London: Wiley, 1974.

Kelly, A., *Business Patterns for Software Developers*, ser. Wiley Series in Software Design Patterns. Wiley, 2012. [Online]. Available: https://books.google.fi/books?id=45KY0fepzsoC

Kerth, N. L. and Cunningham, W., "Using patterns to improve our architectural vision," *IEEE Software*, vol. 14, no. 1, pp. 53–59, Jan. 1997. [Online]. Available: http://dx.doi.org/10.1109/52.566428

Kohls, C. and Panke, S., "Is that true...?: Thoughts on the epistemology of patterns," in *Proceedings of the 16th Conference on Pattern Languages of Programs*, ser. PLoP '09. New York, NY, USA: ACM, 2010, pp. 9:1–9:14. [Online]. Available: http://doi.acm.org/10.1145/1943226.1943237

Köppe, C., *A Pattern Language for Teaching Design Patterns.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 24–54. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-38676-3_2

Kosinski, B. and Cummings, J., "The scientific method: An introduction using reaction time," *Tested studies for laboratory teaching*, vol. 20, pp. 63–84, 1999. [Online]. Available: http://www.ableweb.org/volumes/vol-20/3-kosinski.pdf

Koskinen, J., Vuori, M., and Katara, M., *Safety Process Patterns: Demystifying Safety Standards*, ser. IEEE International Conference on Software Science, Technology and Engineering. IEEE Computer Society, 2012, pp. 63–71.

Kossiakoff, A., Sweet, W., Seymour, S., and Biemer, S., *Systems Engineering Principles and Practice*, ser. Wiley Series in Systems Engineering and Management. Wiley, 2011.

Krishnamurthy, N. and Saran, A., *Building Software: A Practitioner's Guide*, ser. Applied Software Engineering Series. CRC Press, 2007. [Online]. Available: https://books.google.fi/books?id=epPUYuw2X3AC

Kruchten, P., *The Rational Unified Process: An Introduction*, ser. The Addison-Wesley object technology series. Addison-Wesley, 2004. [Online]. Available: https://books.google.ca/books?id=RYCMx6o47pMC

Kuechler, B. and Vaishnavi, V., "On theory development in design science research: anatomy of a research project," *European Journal of Information Systems*, vol. 17, no. 5, pp. 489–504, 2008.

Leveson, N. G., *Safety-Critical Systems: Problems, Process and Practice: Proceedings of the Seventeenth Safety-Critical Systems Symposium, Brighton, UK, 3–5 February 2009.* London: Springer London, 2009, ch. The Need for New Paradigms in Safety Engineering, pp. 3–20. [Online]. Available: http://dx.doi.org/10.1007/978-1-84882-349-5_1

Macdonald, D., *Practical Machinery Safety*, ser. Practical professional books from Elsevier. Elsevier Science, 2004. [Online]. Available: https://books.google.fi/books?id=Dtn6Ey_ItvUC

Machinery directive, "Directive 2006/42/EC of the European parliament and of the council," Official Journal of the European Union, May 2006.

Manuele, F. A., "Risk assessment & hierarchies of control," *Professional Safety*, vol. 50, no. 5, p. 33, 2005.

March, S. T. and Smith, G. F., "Design and natural science research on information technology," *Decis. Support Syst.*, vol. 15, no. 4, pp. 251–266, Dec. 1995. [Online]. Available: http://dx.doi.org/10.1016/0167-9236(94)00041-2

March, S. T. and Storey, V. C., "Design science in the information systems discipline: An introduction to the special issue on design science research," *MIS Q.*, vol. 32, no. 4, pp. 725–730, Dec. 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=2017399.2017404

Maslow, A. H., "A theory of human motivation," *Psychological Review*, vol. 50, no. 4, pp. 370–396, 1943.

Merriam-Webster. (2016) design. [Online]. Available: http://www.merriam-webster.com/dictionary/design

NIOSH. (2015) Hierarchy of controls. "The National Institute for Occupational Safety and Health". [Online]. Available: http://www.cdc.gov/niosh/topics/hierarchy/

Nix, D. (2011, Feb) Undertanding the hierarchy of controls. Machinery Safety 101. [Online]. Available: http://machinerysafety101.com/2011/02/28/understanding-the-hierarchy-of-controls/

Noble, J. and Weir, C., *Small Memory Software - Patterns for systems with limited memory.* Addison-Wesley, 2001.

Nonaka, I. and Takeuchi, H., *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation.* Oxford University Press, 1995. [Online]. Available: https://books.google.fi/books?id=tmziBwAAQBAJ

OSHA, "Occupational safety and health standards - machinery and machine guarding," Occupational Safety and Health Administration, Tech. Rep., 1996, 1910 Subpart O.

———, "Safeguarding equipment and protecting employees from amputations," Occupational Safety and Health Administration, Tech. Rep., 2007, OSHA 3170-02R. [Online]. Available: https://www.osha.gov/Publications/osha3170.pdf

Piirainen, K. A. and Gonzalez, R. A., *Design Science at the Intersection of Physical and Virtual Design: 8th International Conference, DESRIST 2013, Helsinki, Finland, June 11-12, 2013. Proceedings.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, ch. Seeking Constructive Synergy: Design Science and the Constructive Research Approach, pp. 59–72. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-38827-9_5

Pont, M. J., *Patterns for Time-triggered Embedded Systems: Building Reliable Applications with the 8051 Family of Microcontrollers.* New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2001.

Preschern, C., Kajtazovic, N., and Kreiner, C. J., “Catalog of safety tactics in the light of the iec 61508 safety lifecycle,” in *Proceedings of VikingPLoP 2013 Conference*, 2013, pp. 79–95.

Preschern, C., Kajtazovic, N., Höller, A., and Kreiner, C., “Pattern-based safety development methods: Overview and comparison,” in *Proceedings of the 19th European Conference on Pattern Languages of Programs*, ser. EuroPLoP ’14. New York, NY, USA: ACM, 2014, pp. 28:1–28:20. [Online]. Available: http://doi.acm.org/10.1145/2721956.2721958

Preschern, C., Kajtazovic, N., and Kreiner, C., “Building a safety architecture pattern system,” in *Proceedings of the 18th European Conference on Pattern Languages of Program*, ser. EuroPLoP ’13. New York, NY, USA: ACM, 2015, pp. 17:1–17:55. [Online]. Available: http://doi.acm.org/10.1145/2739011.2739028

Pripps, R. N. and Morland, A., *Threshers: History of the Separator Threshing Machine Reaper and Harvester.* Motorbooks, 1992.

Rauhamäki, J. and Vepsäläinen, T., “Functional safety system designer’s handbook - design patterns for safety system development,” Tampere University of Technology, Project report, Feb 2016, unpublished.

Rauhamäki, J., Vepsäläinen, T., and Kuikka, S., *Functional safety system patterns*, ser. Nordic Conference of Pattern Languages of Programs. Tampere University of Technology, 2012, pp. 48–68.

———, *Patterns for safety and control system cooperation*, ser. Tampere University of Technology. Department of Pervasive Computing. Report. Tampere University of Technology, 2013, pp. 96–108.

———, “Design patterns for safety-related control applications,” Tampere University of Technology, Working report, 2015, unpublished.

Rausand, M., *Reliability of Safety-Critical Systems: Theory and Applications.* Wiley, 2014. [Online]. Available: https://books.google.fi/books?id=AaoEAwAAQBAJ

Riehle, D., *Transactions on Pattern Languages of Programming II: Special Issue on Applying Patterns.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, ch. Lessons Learned from Using Design Patterns in Industry Projects, pp. 1–15. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-19432-0_1

Rising, L., *The Patterns Handbook: Techniques, Strategies, and Applications*, ser. SIGS Reference Library. SIGS, 1998. [Online]. Available: https://books.google.fi/books?id=DbsKW7Yzwt8C

———, *Patterns Mining.* Taylor & Francis, 1998. [Online]. Available: https://books.google.fi/books?id=sN0J6dNnrkkC

Rosner, D. and Markowitz, G., *Dying for Work: Workers’ Safety and Health in Twentieth-century America*, ser. Interdisciplinary studies in history. Indiana University Press, 1987. [Online]. Available: https://books.google.fi/books?id=I-7VfHOQ-Q0C

Royce, W. W., “Managing the development of large software systems: concepts and techniques,” *Proc. IEEE WESTCON, Los Angeles*, pp. 1−9, August 1970, reprinted in Proceedings of the Ninth International Conference on Software Engineering, March 1987, pp. 328−338. [Online]. Available: http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf

Sanz, R. and Zalewski, J., “Pattern-based control systems engineering - using design patterns to document, transfer, and exploit design knowledge,” *IEEE Control Systems Magazine*, vol. 23, no. 3, pp. 43−60, Jun. 2003. [Online]. Available: http://dx.doi.org/10.1109/mcs.2003.1200245

Schmidt, D. C., Stal, M., Rohnert, H., and Buschmann, F., *Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects*, 2nd ed. New York, NY, USA: John Wiley & Sons, Inc., 2000.

School, J. and de Groot, E., “Integrated dcs and sis - the right solution improves plant availability, safety,” *InTech*, Mar/Apr 2012. [Online]. Available: https://www.isa.org/standards-and-publications/isa-publications/intech-magazine/2012/april/system-integration-integrated-dcs-and-sis/

Schumacher, M., Fernandez, E., Hybertson, D., and Buschmann, F., *Security Patterns: Integrating Security and Systems Engineering.* John Wiley & Sons, 2005.

Schwaber, K., “Scrum development process,” in *Proceedings of the 10th Annual ACM Conference on Object Oriented Programming Systems, Languages, and Applications (OOPSLA*, 1995, pp. 117−134.

Simon, H., *The sciences of the artificial*, ser. Karl Taylor Compton lectures. M.I.T. Press, 1969. [Online]. Available: https://books.google.fi/books?id=0ONEAAAAIAAJ

Simon, H. A., *The Sciences of the Artificial (3rd Ed.).* Cambridge, MA, USA: MIT Press, 1996.

Spellman, F. R. and Bieber, R. M., *Occupational Safety and Health Simplified for the Chemical Industry.* Government Institutes, 2009. [Online]. Available: https://books.google.fi/books?id=73O_rl2QLgkC

Stein, W. J. and Neuman, T. R., “Mitigation strategies for design exceptions,” CH2M HILL, Inc., Report, Jul 2007.

Stevens, R., Brook, P., Jackson, K., and Arnold, S., *Systems Engineering: Coping with Complexity.* Prentice Hall, 1998. [Online]. Available: https://books.google.com.au/books?id=PPBp2RwMFWwC

Strigini, L., “Fault tolerance and resilience: meanings, measures and assessment,” in *Resilience Assessment and Evaluation of Computing Systems*, Wolter, K., Avritzer, A., Vieira, M., and van Moorsel, A., Eds. Berlin, Germany: Springer, 2012. [Online]. Available: http://openaccess.city.ac.uk/1275/

Sueur, G. L. and Knobel, P., “Integrated control and safety: Assessing the benefits, weighing the risks,” White paper, 2014. [Online]. Available: http://oreo.schneider-electric.com/flipFlop/599765608/files/docs/all.pdf van Heesch, U., Avgeriou, P., and Hilliard, R., “A documentation framework for architecture decisions,” *Journal of Systems and Software*, vol. 85, no. 4, pp. 795 – 820, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0164121211002755

Vepsäläinen, T., *Model-Driven Development of Control Applications: On Modeling Tools, Simulations and Safety*, ser. Tampere University of Technology. Publication. Tampere University of Technology, 6 2015.

Vepsäläinen, T. and Kuikka, S., *Design Pattern Support for Model-Driven Development.* SCITEPRESS - Science and Technology Publications, 2014, pp. 277−286.

Wendel, C., *150 Years of JI Case*, ser. Classic American tractors. Krause Publications, 2005. [Online]. Available: https://books.google.fi/books?id=MNaaSJMHPHEC

Zalewski, J., “Real-time software architectures and design patterns: fundamental concepts and their consequences,” *Annual Reviews in Control*, vol. 25, pp. 133 – 146, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1367578801800018

Publications

Publication I

Rauhamäki, J., Vepsäläinen, T., Kuikka, S. (2013). Patterns in Safety System Development. In Leistner, W. and Lorenz, P. (Eds.), Proceedings of *The Third International Conference on Performance, Safety and Robustness in Complex Systems and Applications, PESARO 2013, April 21-26, 2013, Venice, Italy.* (pp. 9-15). ISSN 2308-3700. ISBN 978-1-61208-268-4. International Academy, Research, and Industry Association (IARIA). Available: `https://www.thinkmind.org/index.php?view=article&articleid=pesaro_2013_1_20_70017`

Patterns in Safety System Development

Jari Rauhamäki, Timo Vepsäläinen and Seppo Kuikka
Department of Automation Science and Engineering
Tampere University of Technology
Tampere, Finland
jari.rauhamaki|timo.vepsalainen|seppo.kuikka@tut.fi

***Abstract*—Development of safety systems for modern industrial control applications is challenged on the one hand by ever growing systems and on the other hand by increasing cost pressures. That is, design process efficiency is a crucial aspect. How to efficiently utilize existing engineering knowledge and document suitable approaches to the common problems of the domain? Design patterns provide a design process with solutions. Design patterns can represent existing knowledge from past projects or illustrate solution blueprints inspired indirectly, e.g., by safety standards. Thus, they provide a designer with support for design decisions during a development process.**

Keywords-design pattern; safety; control system; engineering

I. INTRODUCTION

Safety awareness is constantly increasing across engineering disciplines, regulative governing bodies as well as customers. This trend results in an increasing demand for higher safety integrity levels as well as broadens the product spectrum in which safety systems are deployed. On the other hand, safety system engineering has a constantly increasing need to make the design process efficient in terms of schedule and cost. These issues lead to pressure to increase the efficiency of the safety system development process.

Engineering industry produces vast amounts of tacit and explicit knowledge during customer and R&D projects. This knowledge is a valuable resource that can be used to increase efficiency when available in a suitable format. Explicit project knowledge is typically left as is, i.e., produced knowledge is archived, but it is not indexed or otherwise edited to be easily accessible. Engineers can access the information, but they need to know exactly the project id, subsystem, diagram etc. to locate the existing solution to the problem they are working with. In the context of safety system development explicit existing knowledge could be, for example, a solution to arrange communication between safety-critical and non-safety-critical subsystems according to a safety standard. Tacit knowledge is another source of valuable engineering knowledge. Tacit knowledge is knowledge of individuals or organizations, not available in explicit documented format. In a context of safety system development tacit knowledge could be for instance a solution model of an engineer to a certain problem.

Development of a safety system is bureaucratic and costly, typically regulated by legislation, regulations and standards, which set requirements for the development process. Typical requirements are sets of certain safety functions that need to be implemented in the system and collections of methods and techniques that need to be utilized to achieve sufficient safety integrity levels of the safety functions to reduce risks into a tolerable level. The standards, legislation and regulations require various matters, but give little to no solutions on how these requirements can be fulfilled not to mention guidance for practical safety function implementation.

Our proposed solution for the problems above is application of design patterns in the field of safety system development. Design patterns document solutions to problems commonly encountered and they have proven their value in engineering disciplines such as software engineering [1]. In software engineering large amounts of patterns have been identified and documented.

The contribution of this article is to show how design patterns could benefit the engineering process also the in domain of safety system development. We indicate the rationale to use design patterns in the safety system engineering domain, which is not similar to traditional software engineering though some reasons of use are obviously the same. Problematic issues related to patterns in context of safety system engineering are discussed to provide a broader viewpoint. This also provides a premise and rationale for further studies considering the topic.

The article is organized as follows. In section II we provide background information on design patterns and the domain. Section III presents related work and positions the research. In section IV we present a generalized model of a development process in which safety related aspects are involved and illustrate pattern usage in such a process. In section V the justification for the usage of design patterns in context of the safety system development is discussed in detail. In section VI, the challenging issues of design pattern usage in the domain are pointed out. Sections VII and VIII discuss future work and conclude the article respectively.

ISBN: 978-1-61208-268-4

II. Background

A. Patterns in engineering

The concept of design patterns originates from Christopher Alexander's book: A Pattern Language: Towns, Buildings, Construction [2]. The book illustrates 253 patterns considering architecture, urban design and community habitability. Thus, the roots of design patterns are originated in a domain that has been studied and used for hundreds of years. This illustrates the original nature of design patterns, which is to document solutions identified from real world applications.

Alexander defines design patterns as abstracted solutions to recurring design problems in a given context [2]. This definition is also adopted by the Design Patterns: Elements of Reusable Object-Oriented Software [3], which considers design patterns in the domain of software engineering. The definition includes the three main elements of a design pattern: context, problem and solution. Patterns illustrate solutions to problems that can be applied, in a suitable context, many times but never end up with completely identical solutions.

An analogy for pattern solution application can be found in interior furnishing. An apartment building may have dozens of apartments with the same floor plan, but none of the apartments is similar in interior decoration. When residents move in an apartment, they furnish it, i.e., let us assume they apply an imaginary "Furnish for habitability" pattern. The context of the pattern is an unfurnished and empty apartment, the problem is the low habitability of an unfurnished apartment and the solution is to furnish the apartment with furniture, textiles and other decoration elements to improve habitability. As none of the apartments have identical furnishing the "Furnish for habitability" pattern has been applied multiple times but ending up with a distinct outcome each time.

Process patterns illustrate processes used to complete a task. The purpose is to divide the execution of a task into steps and provide instructions how to execute the steps to complete the whole task. [4]. Process patterns also represent the context, problem, solution paradigm.

B. Two kinds of control systems

Safety systems often, though not always, co-exist and sometimes also co-operate with ordinary control systems. A control system is a system consisting of sensor(s), logic(s) and actuator(s). In this sense, a control system is similar to an E/E/PE (Electrical/Electronic/Programmable Electronic) safety system.

The purpose of a control system is, however, different from the purpose of a safety system. The main purpose of a control system is to control a machine or a process to produce a desired output, e.g., rolls of paper or printed circuit boards. The purpose of a safety system is to ensure the safety of humans, environment and machinery itself, i.e., the main concern of a safety system is to prevent the realization of hazards.

The problem is that the tasks of these two systems differentiate in purpose. A safety system tries to retain the system in a safe operation state whereas a control system tries to maximize the output of the system. To carry out the tasks the same process variables need to be considered. The situation is even worse as the systems often have opposite preferences considering the state of the system.

C. Some patterns for functional safety system development

In our recent research projects, we have focused on safety system principles and architectures. It was noted that patterns for safety systems are not available although patterns for the related domains are considered (see section III). However, we see potential in patterns in the domain of safety system development. During the recent projects, we have identified and developed patterns for the development of safety systems. Table I summarizes the patterns published in VikingPlop'12 [5].

The patterns consider various aspects of safety systems. The Separated safety and the Productive safety patterns consider the co-existence and distribution of liabilities between the safety and main control systems. The Separated override, De-energized override and Safety limiter patterns illustrate approaches to override the main control system with a safety system. The approaches have distinct redeeming features and downsides. For instance, the Separated override pattern emphasizes separation between the systems whereas the Safety limiter pattern allows cooperation between the systems and reduces the amount of needed hardware. The Hardwired safety pattern proposes usage of a hardwired safety system instead of a software based solution in a suitable context.

TABLE I. FUNCTIONAL SAFETY SYSTEM PATTERNS [5]

Pattern	Description
Separated safety	Development of a complete system according to safety regulations is a bureaucratic and slow process. Therefore, divide the system into basic control and safety systems and develop only the safety system according to safety regulations.
Productive safety	A control system utilizes advanced and complex corrective functions to keep the controlled process in the operational state. These functions are very hard to implement in a safety system. Therefore, implement the corrective functions in a basic control system and use simple(st) approach for the safety system.
Separated override	A safety system must be able to override a basic control system whenever systems control same process quantities. Therefore, provide the safety system with a separate actuator to obtain a safe state.
De-energized override	A safety system must be able to override a basic control system whenever systems control same process quantities. Therefore, let the safety system use de-energization of the basic control system's actuator(s) to obtain a safe state.
Safety limiter	A safety system must be able to override basic control system whenever systems control same process quantities. Therefore, disengage the basic control system completely from the actuator and let the safety system control the actuator. Route the output of the basic control system to the safety system and let the safety system treat the control value so that safe operation is ensured.
Hardwired safety	Development of safety-related application software for simple safety function is bureaucratic, time consuming and costly. Therefore, instead of a software-based solution, use a hardware-based safety system.

III. RELATED WORK

Design patterns have been studied and documented in the field of software engineering extensively covering for example object-oriented software [3] and [6], Pattern-oriented architecture [7] and [8], enterprise applications [9], [10], and service-oriented architecture [11]. These books concentrate on software engineering for desktop and server-side applications and architectures. Though these patterns may be usable in safety system development they are not focused on safety aspects.

Fault tolerance is a part of safety system design as safety systems should preferably be fault-tolerant to be able to operate under fault conditions and ensure safety. However, fault-tolerance is not a sufficient condition for safety. Fault-tolerant software can be hazardous in a safety system if the functionality of the software is hazardous, e.g., due to erroneously set requirements. Design patterns for fault-tolerant software systems have been introduced, for example, by Hanmer [12].

E/E/PE safety systems include both hardware and software components. Armoush [13] and Douglass [14] introduce design patterns covering software and hardware aspects of safety systems. The presented patterns are focused on redundancy, which, again, is an approach to increase reliability and fault-tolerance of a system.

Eloranta, Koskinen, Leppänen and Reijonen [15] have studied distributed machine control systems and documented patterns for the design of such systems. Some of the patterns are also related to functional safety aspects. The application domain of the above patterns is closely related to our design patterns considering safety system development and architecture [5].

Koskinen, Vuori and Katara have studied and developed process patterns for the application of the IEC 61508-3 standard. In their article [4] they stated that process patterns can speed up the training of inexperienced engineers and remove ambiguities typically related to safety standard application. This provides additional support for the usage of patterns in the domain of safety systems.

Riehle [1] properly points out three main usage areas of design patterns in current software industry practice. These areas are communication, implementation and documentation. In this article, we consider how these usage areas are transferable into safety system engineering.

IV. GENERALIZED SAFETY SYSTEM DEVELOPMENT PROCESS

In this section, a generalized process for the development of safety-related E/E/PE systems is illustrated. The purpose is to provide an idea about how pattern usage can relate to such a process. The illustrated process is inspired by the IEC 61508-1 overall safety lifecycle [16] and eight steps to safety [14].

Development of a safety system begins with the definition of the overall *scope* of the EUC (Equipment Under Control) and the concept of the system. In this phase understanding about the system and its environment is built. In this context process patterns can be used to identify EUC related aspects (e.g., what are typical characteristics of, for example, bending machines) and typical machinery concepts.

Hazard and risk analysis follows the scope definition. Hazard and risk analysis forms a significant part of the development process as the results directly impact on the coverage of the safety system and selection of safety measures and safety integrity levels. Patterns can be used to identify typical hazards related to specific systems (machinery type) and processes (operations executed by the machinery). Process patterns can be used to describe and interpret the phases of the hazard and risk analysis as required by the followed standard.

When the risks are defined, the *requirements* for the risk mitigation methods are documented in the requirement specification phase. This includes the definition of the risk mitigation methods, safety functions, and the non-functional requirements and safety integrity levels related to them. Patterns can be used to document typical approaches to mitigate risks with the positive and negative effects related to the approaches thus providing support for decision making. The requirement specification phase can also be supported with process patterns. For instance, the Software Safety Requirements Specification pattern in [4] illustrates a requirement specification process mined from the IEC 61508 to provide help and document the sub phases of this development phase.

As the requirements for safety measures and functions are defined the process can continue on to the *realization* of safety system. The phase consists of design and implementation of the safety system. In this phase of development process, patterns have value as the level of abstraction suits well to describe solutions to design and implementation problems. The patterns provide designers with documented solutions to commonly encountered safety design problems. However, the patterns also provide information about consequences related to application of them. This enables an engineer to select the most suitable solution by justifying the consequences. For instance, the three override patterns described in Table I illustrate different approaches to a design problem where a safety system should be able to override a control system. Each of the solutions has their own consequences and the designer can choose the one that is the best fit for the system under development. Process patterns can support the realization process by, e.g., providing support to carry out the recurring phases of development such as the modification or architectural design of the software [4].

The implementation part of the *realization* phase can be supported with design patterns as in this phase engineers encounter a large number of common problems where design patterns are able to provide solutions. The patterns applied in the implementation phase often represent a lower level of abstraction and provide focused solution models to lower level implementation problems.

The rest of the development process relate to *validation, verification, testing, installation and maintenance* aspects. Process patterns for validation and verification document and help to follow the processes. For instance, patterns for validation and verification in context of the IEC 61508 are provided in [4]. Maintenance of long life cycle systems benefits from the usage of design patterns as known solution models are used.

V. RATIONALE FOR DESIGN PATTERN USAGE IN SAFETY SYSTEM ENGINEERING

As illustrated in Section II/A design patterns have redeeming features in context of the software engineering domain. However, the software engineering domain, at least desktop software engineering, is different from safety system engineering. Of course, software systems are a part of modern safety systems, but the nature of a pure software system is distinct from a safety system. In the following subsections rationale for the usage of design patterns in safety system engineering is discussed.

A. Ability to avoid physical damage

Normal application/desktop software, run on a personal computer, mobile device, server or similar device, has limited possibilities to interact with its environment. Potential physical risks associated with such devices and applications are, for example, overheating, electric shock, battery malfunctions and fans none of which are directly controllable by the application software ran in the system. That is not to say that application software cannot be critical. For example, a failure of banking, insurance or other large scale business system may inflict massive losses to its owners in form of revenue or work contribution losses and is thus considered critical. However, no (direct) human, environmental or machinery related hazards exist in such cases.

Systems in which safety systems are deployed are able to cause hazardous situations for humans, environment and hardware by their nature (if not, no safety control system would be required). Industrial and machinery control systems operate actuators (e.g., fans, valves, and heaters) process devices (e.g., conveyors, robots, and guillotines) and substances (e.g., toxic chemicals or hot fluids) that are hazardous for humans, environment and the systems itself. As the safety systems are dedicated to mitigate risk related to such machinery they are expected and required to have certain level integrity to carry out the safety functions.

Design patterns document good approaches, practices and solutions common in safety system development. This provides designers with tried solutions to problems as well as removes the need to reinvent the wheel thus resulting in a more productive development process as well as solutions with a justified approach. The development burden is decreased and the designers can focus on details as patterns describe the main solution model.

B. Experience as a valuable resource in safety system development

In the field of safety system engineering, well-tried solutions are welcome as they have additional empirical data to back up applicability. By identifying patterns from existing projects and designs and making the solutions explicit in patterns, experience can be transferred from one engineer to another. Design patterns support the illustration of experience in explicit format by requiring the pattern writer to consider different aspects of the solution. This work is carried out in consideration on the context in which the solution can be used, consequences and the resulting context related to the solution. Patterns document (or at least they should document) also negative consequences, preconditions and assumptions related to pattern application. This provides engineers with a foundation to use or not to use certain solutions and compare them against each other to select the best approach for the problem under consideration.

A good approach is to document the proven solutions of past projects into patterns to be used in forthcoming projects. In this way, the patterns are directly related to the domain, they can be written to solve a dedicated problem and the consequences are known. That is not to say one should limit to such patterns only. Third-party patterns may provide fruitful insight into other kinds of solution models and open new kinds of approach possibilities to solve a certain kind of problem with more desirable consequences.

Experience illustrated in format of patterns, also provides a name for the solutions and approaches. This enables the usage of patterns as a part of communication [1], but requires that the patterns have reached awareness of the engineering community using them. When this point is achieved, patterns can be used in communication to illustrate the solutions and approaches described in design patterns. For example, safety system engineers could discuss about how to override a control system with a safety system: “I think separated override [5] would be a good approach in this situation.”, “I disagree; I find separated override an excessive action as it would require an additional safety actuator. Maybe we should consider de-energized override [5] instead”, “That is true, de-energized override is a more cost-effective approach in this case.”

C. Alleviating bureaucracy

Development of safety systems is regulated by directives, legislation and standards such as [17], [18], [19]. Such documents are written partly from a legislative point of view, are too generic to cover various applications and domains, and do not (want to) strictly enforce a certain approach. These aspects restrict the documents from providing solution models. Rather such documents require various techniques, methods, and processes to be used in the development of safety systems, but give minor importance on examples or other guidelines for any specific implementation. In addition, the documents are massive, often hundreds of pages long, which makes finding solutions difficult. This does not mean standards etc. are useless; they just have a different view to safety systems compared with patterns. The standards provide a framework that is applied in a certain way to develop the system. The framework describes methods and techniques to develop safety systems and, e.g., define what to verify and validate when a system is being developed. This is certainly a valuable aspect in safety system development.

The purpose of patterns in this context is to supplement the standards and document the solution models and approaches compliant with the given requirements. The IEC 61508-3 [20], for instance, illustrates a number of techniques and measures to be used in the development of safety-critical software. However, little information on how these techniques shall be used and what kind of solution models they (may) produce is given. Especially safety-related standards can prove hard for a person with limited experience in the development of safety systems [4]. With design patterns, solutions and approaches to implement techniques and measures required in standards can be documented, which illustrates the usage of design patterns as a source of implementation [1]. Process patterns can be used to capture the recurring tasks in the development of a safety system [4].

In context of safety system development the value of patterns is fully established when patterns are mined from a system that has already found compliant with a safety standard. This adds confidence in the solution model validity in context of the considered safety standard. Such patterns can increase development process efficiency as the solution model can be used in other systems with a fairly good confidence as long as the context described in the pattern matches the context in which the pattern is applied. The solution, approach or method once approved in a certification process or assessment for standard compliance, for instance, is useful as it provides at least the main solution framework for the problem under consideration.

Development of a safety system also requires extensive documentation. This is required, e.g., to illustrate compliance with a standard considering development of a safety system or an informal document illustrating the safety foundation of a system, which can be used as a part of safety assessment of a system. Design patterns can be used for documentation purposes [1]. The applied patterns and roles of the patterns can be marked in a document (e.g., in a diagram). For an experienced pattern user this quickly indicates the type of solution used (described by the pattern). The need for reading textual representations decreases as the reader can obtain the information on the roles of the system elements directly from the diagram. In an informal supplementary documentation usage of well-known safety related patterns can be justified. The reader is able to identify the patterns applied and assess their suitability in context of the safety system under consideration. However, in context of the legal safety system documentation, the usage of patterns in documentation does not remove the need for textual representations as the usage of pattern notation in the documentation does not cover the whole functionality and all the aspects of the applied solution.

D. Co-existence of control and safety systems

A safety system often co-exists with a main control system as stated in section II. Although safety and control systems are designed to be separated, they often need to be connected some way (e.g., to share state and operation information). This aspect further increases the amount of work needed to design an operational entity consisting of safety and control systems.

Integration of safety and main control systems is sometimes, especially in context of larger processes, a unique design. The operation and responsibilities of the safety and control systems need to be defined and fitted to operate in harmony. If such a system is repeatedly designed from scratch, a great amount of design work needs to be redone. In such situations the design process may greatly benefit from the reuse of templates [21], model libraries and similar ways of reusing existing designs developed in a specific development environment. However, templates and library solutions as such are not a good fit to document solution models and approaches on a generic level. This is due to the fact that solutions are bound to the implementation environment: the solutions are described in terms of the implementation environment/tool. Such an approach complicates the understanding about the solution on a higher level of abstraction.

Contrarily, patterns provide a format to document solutions on a platform independent level. This enables the documentation of solutions, which can be used in different implementation environments as long as the context and other prerequisites are considered. The benefit of a pattern approach is that one is able to take the idea from a pattern and adapt the principle of the solution to solve the problem in hand, thus increasing the efficiency of the design process.

1) A case for pattern usage in design of safety and control system co-existence

This section illustrates a case for usage of design patterns is design of system in which safety and control system operate the system under control. The functional safety system patterns introduced in Table I illustrate solutions for safety and control system co-existence. The patterns describe approaches to arrange the responsibilities of the systems and override of the control system. The idea is to divide the responsibilities so that the development of the safety system is as lightweight as possible, but the safety system still is retaining full control over the machinery.

A potential design decision flow to utilize the patterns is illustrated in Fig. 1. The figure illustrates the pattern relations of the patterns in Table I. The Separated safety pattern is applied first to the system under development. This decision results separated safety and control systems and only the safety system has to be developed according to safety standards, which decreases the development burden considerable as the control system (which is typically a larger entity than the safety system) can now be developed

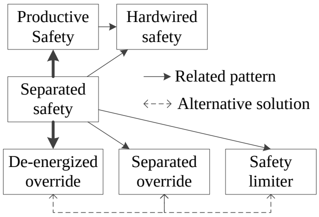


Figure 1. Design flow using functional safety system patterns [5]

without safety standard conformance.

When the system is divided into safety and control system, the Productive safety pattern can be considered. It suggests that the control system implements the basic interlock mechanisms that (try to) keep the system in a normal operational state as far as possible. The interlock mechanisms can be as complex as needed as they do not need to conform to the safety standards. The actual safety functions are implemented on the safety system and they can be rather simple because the control system keeps the machinery in the normal operational state. The safety system can, for instance, only implement an emergency shutdown of the machinery when the control system has failed to retain the normal operational state. This approach simplifies, and thus potentially lowers the cost of, the safety system development and implementation.

As the safety system must be able to drive the system into a safe state regardless of the control system state, the designer needs to implement such functionality. The three override patterns provide three distinct approaches how the safety system can override the control system on the actuator level. The designer can compare the suggested approaches and select the one with most desirable consequences regarding the system under control. For instance, if separation between the safety and control system is the main concern, the Separated or De-energized override pattern is the most appropriate. However, if there is a need to lower the amount of actuators or use advanced safety functionality, the Safety limiter pattern may be a better alternative.

The above workflow illustration also depicts the potential of pattern language utilization. The designer uses a pattern language as a framework and selects the most appropriate patterns to design the system. The pattern language supports the design process by defining relationships between the patterns. The relationships illustrate, e.g., patterns that are applicable after a certain pattern has been applied, conflicting patterns or patterns that solve problems, which may arise when a pattern is applied.

E. Maintainability of common solution models

An important feature of control systems, especially in an industrial domain, is long life-cycles. As safety systems are part of control structures of a system, they also have long life-cycles in similar applications. The maintenance phase of a system may contribute considerably to a large part of system design and development costs when the whole life-cycle costs of the system are considered. Thus the maintainability of a safety system is an important aspect to be ensured during the initial development process of the safety system.

Maintenance of a system is easier if the system is intelligible. Usage of design patterns can improve intelligibility through common vocabulary. If design patterns are used in system development and documentation [1], the maintenance team can more easily understand the system concepts and execute maintenance operations to the system. Naturally this requires that both the developer and the maintenance team know and understand the used patterns. This, in practice, requires either company’s internal patterns or widely adopted patterns related to the domain.

VI. CHALLENGES IN DESIGN PATTERN USAGE IN SAFETY SYSTEM ENGINEERING

Patterns have qualities that justify their usage in the development of safety systems. However, some challenging issues can be identified as well. To provide ample insight into patterns the issues of patterns are discussed in this section.

Patterns are not exact. As mentioned, patterns (typically) describe solution on a relatively high abstraction level so that they can be used and implemented in multiple ways. In safety system development exactness and completeness are considered virtues that patterns can, but often do not, provide.

Usage of patterns may lead to *inconsistent understanding* between system developers. A pattern can be implemented in many ways and each person has a unique mindset about a pattern. Thus patterns are not applicable as safety documentation as such. However, when a set of patterns has been used extensively, the patterns may become a part of a communication language that clarifies the ideas shared between individuals [1] and thus may act as a supporting form of documentation.

A developer may *misunderstand pattern solutions* or use them in contexts not suitable for the pattern. A similar issue is naturally related to all situations when documented solutions are applied. One can also misunderstand solutions illustrated in a book, journal article or data of a preceding project.

Patterns are not meant to be detailed illustrations of the solution (though some patterns indeed illustrate details). Instead, they typically provide a *generic framework of the solution*, which the designer can apply in the environment in which the problem is considered. This is one of the strengths of patterns, but it is also a potential issue. A pattern author may have accidentally or intentionally left out some information that would be needed to be fully able to consider all the side-effects of the pattern.

If a pattern reader is *unfamiliar with the domain* the patterns consider, an incorrect overall picture could be adopted. Though patterns consider various aspects of the solution, they cannot take into account all the relevant aspects. In the domain of safety system engineering artefacts relate to each other in complex manners. A single pattern cannot consider all these aspects as it would shift the focus of the pattern. Thus the reader should regard patterns with a healthy sense of criticism when they are applied.

Patterns may encourage designers to *stick with existing solutions*. Often the reuse of solutions is a productive way to go and well-tried solutions are valuable in the field of safety system engineering. However, this should not mean that reuse of solutions is the only way to go. New, more efficient, simpler, and better approaches cannot be developed if old solutions are constantly used. It has to be identified if the design benefits from the reuse of solutions and when one needs to focus on creating a better, novel approach to the problem in hand.

VII. Future work

Our future effort is expansion of our safety system development related pattern collection [5] and development of tool support for a semantic search of patterns. The target of the pattern collection expansion is to construct a pattern language that could serve safety system developers. Another aspect is to study the effects of pattern usage in practical development processes. Empirical studies on pattern usage in the development processes of safety systems would provide insight into widening the usage of patterns.

The semantic search for patterns eases pattern discovery. Semantic relations between pattern data are being developed. This enables the search of patterns supported with a semantic deduction engine to identify patterns with similar features and consequences as given in the original search.

VIII. Conclusion

In this article, we have illustrated rationale for using design patterns in the development of safety systems. The foundation of usage of patterns lies in the idea of providing a way to document tacit and existing knowledge into an explicit format. When experience is formatted as a design pattern, it can become common knowledge that can serve in documentation, implementation and communication purposes.

Safety systems are parts of critical systems that are able to cause physical damage. The sole purpose of a safety system is to prevent the hazardous situations leading to physical damage. Well-tried solutions and approaches documented in patterns can help in the development of a dependable and cost-effective safety system. Development of safety systems is heavily regulated by standards and legislation, which require methods, techniques and processes to be used, but provide few practical solutions. With design patterns practical solutions can be documented into an intelligible format while providing room for modifiability.

Cooperation between a control and a safety system can prove to be a burdensome task especially if it is made from scratch. This may occur in larger control system projects for large scale unique plants. In such cases patterns provide a valuable engineering resource as they describe solution and approaches on an abstract level. This enables a designer to apply the approach in a suitable way considering the system.

Design patterns also have some drawbacks in context of safety system development. They are not exact and accepted as documentation or proof of compliance. Still patterns can help to improve development process and share knowledge.

References

[1] D. Riehle, "Lessons Learned from Using Design Patterns in Industry Projects," in Transactions on Pattern Languages of Programming II, vol. 6510, J. Noble, R. Johnson, P. Avgeriou, N. Harrison, and U. Zdun, Eds. Springer Berlin / Heidelberg, 2011, pp. 1-15.

[2] C. Alexander, S. Ishikawa, and M. Silverstein, A pattern language: Towns, buildings, construction. New York: Oxford University Press, 1977.

[3] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design patterns, Elements of reusable object-oriented software. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc, 1995.

[4] J. Koskinen, M. Vuori, and M. Katara, "Safety Process Patterns: Demystifying Safety Standards," Proc. 2012 IEEE International Conference on Software Science, Technology and Engineering (SWSTE), IEEE Computer Society, 2012, pp. 63-71.

[5] J. Rauhamäki, T. Vepsäläinen, and S. Kuikka, "Functional Safety System Patterns," Proc. VikingPLoP 2012 Conference, 2012, pp. 48-68, http://URN.fi/URN:ISBN:978-952-15-2944-3 [retrieved: February, 2013].

[6] E. Freeman, E. Freeman, B. Bates, and K. Sierra, Head first design patterns. O' Reilly & Associates, Inc, 2004.

[7] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, Pattern-oriented software architecture, volume 1: A system of patterns. Chichester, UK: Wiley, 1996.

[8] D. C. Schmidt, M. Stal, H. Rohnert, and F. Buschmann, Pattern-oriented software architecture: Patterns for concurrent and networked objects. 2nd ed.,New York, NY, USA: John Wiley & Sons, Inc, 2000.

[9] M. Fowler, Patterns of enterprise application architecture. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc, 2002.

[10] G. Hohpe and B. Woolf, Enterprise integration patterns: Designing, building, and deploying messaging solutions. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc, 2003.

[11] T. Erl, SOA design patterns. 1st ed.,Upper Saddle River, NJ, USA: Prentice Hall PTR, 2009.

[12] R. S. Hanmer, Patterns for fault tolerant software. Chichester, England ; Hoboken, NJ: John Wiley, 2007.

[13] A. Armoush, Design Patterns for Safety-Critical Embedded Systems. 2010.

[14] B. P. Douglass, Doing hard time: Developing real-time systems with UML, objects, frameworks, and patterns. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc, 1999.

[15] V. Eloranta, J. Koskinen, M. Leppänen, and V. Reijonen, A Pattern Language for Distributed Machine Control Systems. 2010. http://practise.cs.tut.fi/project.php?project=sulake [retrieved: February, 2013].

[16] International Electrotechnical Commission, Functional safety of electrical/electronic/programmable electronic safety-related systems, Part 1: General requirements, IEC, 2010.

[17] European Parliament and of the Council, Directive 2006/42/EC of the european parliament and of the council, vol. L 157/24, 2006.

[18] International Electrotechnical Commission, Functional safety of electrical/electronic/programmable electronic safety-related systems, IEC, 2010.

[19] European Committee for Standardization, Safety of machinery, safety-related parts of control systems, Part 1: General principles for design, 2008.

[20] International Electrotechnical Commission, Functional safety of electrical/electronic/programmable electronic safety-related systems, Part 3: Software requirements, IEC, 2010.

[21] M. Kairaila, Domain-Specific Template-Based Visual Language and Tools for Automation Industry. 2010.

Publication II

Rauhamäki, J., Vepsäläinen, T., Kuikka, S. (2015). Towards Systematic Safety System Development with a Tool Supported Pattern Language. In Oberhauser, R., Lavazza, L., Mannaert, H. and Clyde, S. Proceedings of *The Tenth International Conference on Software Engineering Advances, ICSEA 2015, November 15-20, 2015, Barcelona, Spain.* (pp. 341-348). ISSN 2308-4235. ISBN 978-1-61208-438-1. International Academy, Research, and Industry Association (IARIA). Available: `https://www.thinkmind.org/index.php?view=article&articleid=icsea_2015_13_20_10159`

Towards Systematic Safety System Development with a Tool Supported Pattern Language

Jari Rauhamäki, Timo Vepsäläinen and Seppo Kuikka
Department of Automation Science and Engineering
Tampere University of Technology
Finland
Email: {jari.rauhamaki, timo.vepsalainen, seppo.kuikka}@tut.fi

Abstract—**Design patterns illustrate qualities and features that would suit well in current understanding of safety system development, design and documentation. However, though a number of design patterns for safety system development have been proposed, the focus has been on individual quality attributes such as fault tolerance and reliability. The systematic use of design patterns in the development process has received less attention. In this paper, we discuss and illustrate extended usage possibilities for design patterns as part of safety system development. We discuss a design pattern language that we are developing to cover, e.g., safety system architecture, scope minimization and co-operation with basic control systems. Use of patterns for documentation purposes, tool support for using patterns, and rationale for the pattern approach are discussed as well.**

Keywords-safety system; software; design pattern; safety standard; tool support

I. Introduction

Design patterns are a means to systematically promote the re-use of design and proven solutions to recurring problems and challenges in design. Each design pattern represents a general, reusable solution to a recurring problem in a given context. Triplets of problems, contexts and solutions are also the essential pieces of information in patterns. In addition, pattern representation conventions can include, among others, relations to other patterns. With such relations describing, for example, rational orders to use patterns, patterns can be combined to collections and to pattern languages. Depending on patterns, the natures of their solution parts can vary too, for example, from source code templates to text and Unified Modeling Language (UML) illustrations.

Software safety functions are software parts of usually multi-technical systems, the purpose of which is to ensure the safety of controlled processes and plants. Unlike many other software systems, safety systems are developed according to standards. The standards govern the development lifecycle activities, as well as techniques and applicable solutions of such systems. However, although design patterns have been specified also for safety system development, their systematic use has not been researched in the domain. This is surprising because the use of patterns could facilitate both design and documentation activities, which are equally important in safety system development.

In this paper, we address the aforementioned issues. The contributions of the paper are as follows. We rationalize how and why design patterns, which have already shown their value in software development, in general [1], could be especially useful in safety system development. We discuss a design pattern language for safety systems, which has been developed and published iteratively and is to be finalized during DPSafe project in collaboration with Forum for Intelligent Machines (FIMA) in the machinery domain. Lastly, we discuss and rationalize the role of tool support in facilitating the use of patterns and in benefitting from patterns.

The rest of this article is organized as follows. Section 2 reviews work related to design patterns and the use of design patterns in safety system development. Section 3 presents a view on the development of software safety systems and rationalizes why and how design patterns could be beneficial. In Section 4, we discuss a design pattern language for safety system development that has been developed at the Tampere University of Technology. Before conclusions, Section 5 discusses the role of tool support when trying to benefit from patterns.

II. Related Work

The design pattern concept was originally presented by Alexander [2][3] in the building architecture domain to refer to recurring design solutions. In software development, design patterns begun to attract interest after the publication of the Gang of Four (GoF) patterns [4]. Thereafter, collections of design patterns have been gathered and used for various purposes in various domains. Results from their use have included, among others, improvements in quality of code, as well as improved communication through shorthand concepts [1].

Design patterns have also been developed for special purposes and application domains, including critical [5] and distributed [6] control systems. In the functional safety domain, especially, patterns already cover many solutions and techniques that are recommended by standards, such as IEC 61508 [7] and ISO 13849 [8]. For example, related to architecture design in [7], there are patterns to implement redundancy [9] and recovery from faults [10].

Pattern languages, on the other hand, aim to provide holistic support for developing software systems by using and weaving patterns and sequences of patterns [11]. For embedded safety system development, for example, a large collection of (both software and hardware) patterns for various problems is listed in [5]. However, the multi-technical collection is not regarded as a pattern language, per se.

Partially because of reasons to be discussed in the next section, documentation is of special importance in safety system development. A developer of a software safety system needs to be able to prove the compliance of the application to standards. Otherwise, the application cannot be used in the safety system. However, certifiable safety applications are not made by coincidences but by designing the systems and applications systematically, with certifiability in mind. As such, also the software parts need to be specified (modeled) prior to their implementation. On the other hand, the suitable solutions (patterns) that are used in the applications should already be visible in the models. Otherwise, the use of the patterns would not be documented in the models and valuable information could be lost.

It is thus clear that the systematic use of design patterns in safety application development requires tool support for the patterns already in the modeling phase. This is regardless of whether or not the models can be used in producing (automatically) executable code as, e.g., in Model-Driven Development (MDD). Using and applying patterns in UML, which is currently the de-facto software modeling language, has been addressed in several publications. For example, work has been published to specify patterns in a precise manner [12], to apply patterns to models [13, 14], to detect pattern instances [15, 16] and to visualize pattern instances in models and diagrams [17]. However, without extensions the support for patterns is still weak in UML [18].

III. Patterns in Safety System Development

The development of safety functions is governed by standards, such as IEC 61508 [7], IEC 62061 [19], and EN ISO 13849-1 [8]. These standards guide the development of safety systems involving electric, electronic and programmable electronic control systems in their operation. Regardless of the variety of standards, we outline a generic development process for safety systems common to the aforementioned standards. The simplified process is illustrated in Figure 1.

The development process begins by the definition of the concepts and scope of the system to be developed. This includes forming an overall picture of the system and defining the boundaries of the system/machine to be analyzed or made safe. The next step is to carry out a hazard analysis and risk assessment. The role of this phase is centric as only known risks can be consciously mitigated. Otherwise risk mitigation measures have no justification. Typically, risk assessment includes hazard identification, risk estimation and evaluation. The former provides an indicator for the risk and the latter assess the impact of the risk, that is, is the risk tolerable or not. Intolerable risks need to be mitigated or made tolerable otherwise.

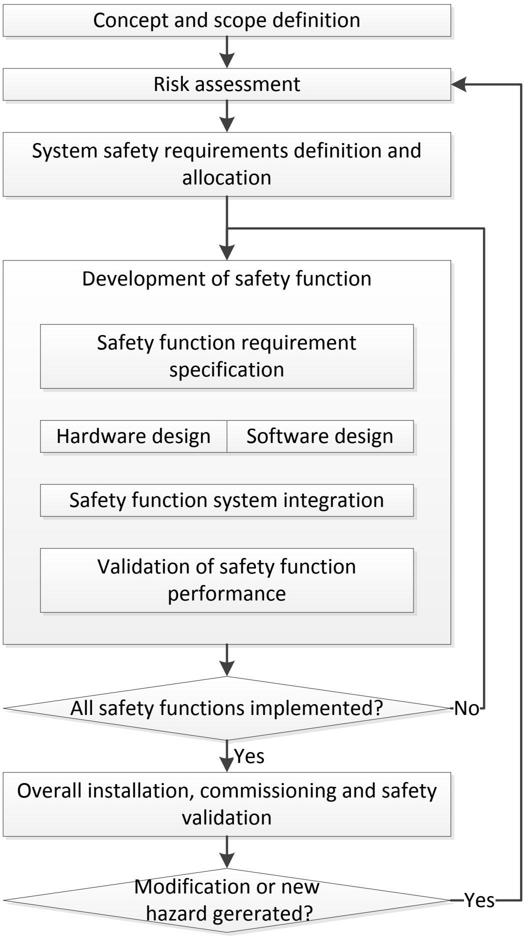

Concept and scope definition
Risk assessment
System safety requirements definition and allocation
Development of safety function
Safety function requirement specification
Hardware design
Software design
Safety function system integration
Validation of safety function performance
All safety functions implemented?
No
Yes
Overall installation, commissioning and safety validation
Modification or new hazard gererated?
Yes

Figure 1. Simplified safety system development process according to EN ISO 13849-1 [8] and IEC 61508 [7]

As the risks are assessed, the requirements considering the system safety can be justifiably made. In this phase, suitable risk reduction methods are selected and their requirements are documented. In the context of this paper it is assumed that the risk reduction method is a protective measure depending on a control system to implement the required functionality. In addition, the allocation of the measures is done. That is, to allocate the measures for dedicated functions.

The next phase is the development (realization in IEC 61508 terminology) of the safety functions allocated in the previous phase. The development process starts with compiling a requirement specification for the safety functions. The specification should include both functional descriptions, what the functions need to do, and non-functional descriptions, how or within which restrictions the functions need to operate.

Quite often, the non-functional descriptions include the specification of performance or integrity levels for the functions. When the requirement specification is completed, the hardware and software design can begin. In this state the hardware and software parts of the safety function are designed, potentially with separation between the design teams. Thus, hardware and software integration needs to take place along the design process. At this point, a functional entity can be constructed including both the hardware and software to be used in the final system. Finally, the results of the safety function development are verified to match the safety function requirements and required performance/integrity levels. If unimplemented safety functions exist, the development process is reinitialized for the next safety function.

A. *Utilization of patterns in safety system development*

In the context of safety system development and design, design patterns can be used to capture and provide solution models for techniques and applicable solutions that are recommended and/or required by applicable standards. In this case, a design pattern captures the solution that is used in order to fulfill the requirements and recommendations of a standard. Such design patterns can be linked to the parts of the standards for which the design patterns provide a complete or partial fulfillment or help to achieve to fulfill the standard requirements. This kind of approach also supports building the libraries of named solutions. That is, the patterns support the awareness and usage of the solutions.

One can justifiably argue that standard solutions to recurring problems have been applied in safety system development and other domains of engineering for years – without necessarily calling them patterns. However, their unconscious use may not have eased the task of documenting the systems. Since design patterns provide names for solutions, they can be used in communication, too [1]. Though initially applicable to discussions and face-to-face communication, design patterns can be used as a part of written and diagrammatic documentation. This is achieved by referring to the solution illustrated by a pattern with the name of the pattern that should be both illustrative and related to the application context.

The documentation aspect can be achieved by marking the patterns in, e.g., diagrams that are used as a part of the system documentation. This can enhance traceability between the standard solutions and their practical applications in systems. For a pattern-aware person, this may increase the understandability and traceability of the design decisions, too. To take further advantage of this setup, statistics could be gathered to see which patterns are used the most and in which kind of situations. It can also be noted that the quality attributes understandability and traceability are similarly components of systematic integrity acknowledged by IEC 61508 [7].

Other viewpoints supporting the utilization of design patterns in safety system development include for instance [20]:

- Patterns document well-tried solutions and thus condense experience on proven solutions, which is of special importance in the domain. The approach resembles, for instance, the proven in use concept defined by IEC 61508.
- Patterns can alleviate bureaucracy by providing practical solutions and approaches to fulfil requirements given to safety system development in, for example, standards. Bridging the gap between the requirements and design and implementation eases the burden of designers.
- Patterns create the vocabulary of solutions to domains. Assuming that the patterns are known by both the developer and maintainer of a system, patterns can help to communicate the structural and operational principles of the system. This aspect thus improves the communicability and maintainability of the system.

B. *Safety system patterns*

In the context of this paper, we are especially interested in design patterns for safety system development, called safety system patterns here. These patterns are, or at least they are meant to be, most useful in the development of (functional) safety systems. This does not indicate that the patterns could not be used for other purposes as well. However, the contexts of the patterns relate them to the safety system development. It is up to the readers or appliers of the patterns to judge whether the solutions are applicable outside the indented contexts of the patterns, too.

It should be noted that a pattern does not necessarily illustrate the cleverest or the most innovative solution or approach to the defined problem. Instead, the preferable approach is to provide proven solutions and approaches that have been utilized successfully in practice, in real projects and systems. This is, on one hand, targeted to provide assurance on the applicability of the solution, for instance, in the eyes of an inspector. On the other hand, the most innovative solutions might promote other quality attributes than simplicity, which is one of the most important driving qualities behind a safety system development.

So, which parts does a safety system pattern consist of? In our work, we have used a slightly modified canonical pattern format [21]. That is, each pattern documents the context, problem and solution. They are complemented with forces, consequences, example, known usages and related patterns, see Figure 2. The triplet of context, problem and solution provides the main framework for the patterns. These aspects should provide sufficient information to apply a given pattern. However, the other aspects, for instance, support the selection of the most suitable pattern and help to identify other potentially applicable patterns. The former aspect is achieved through the definition of forces and consequences. Forces relate to the context, refine the problem, and direct the solution to the one selected to be illustrated on the pattern. On the other hand, consequences

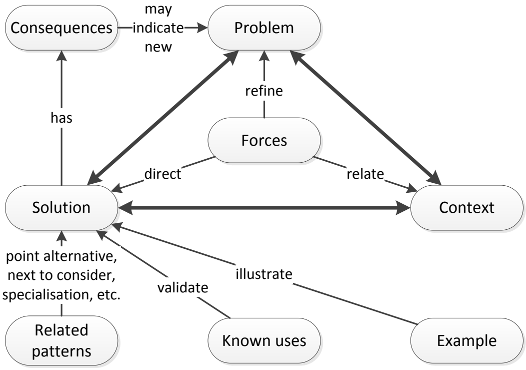


Figure 2. The pattern structure used in our safety system patterns.

provide hints to select a solution proposed by a certain pattern. Presumably one wants to select a pattern or a solution that has the most positive consequences and/or the least negative consequences produced by the solution.

In addition to the mentioned pattern aspects, safety system patterns could be complemented with an aspect indicating the applicable performance level (PL), safety integrity level (SIL), or similar quantity. This is to indicate for which purposes or levels (as defined in standards) the pattern can be used. [21]. For certain patterns or solutions such indicators can be given directly and for others such indicators are indirect or cannot be given at all. For instance, a pattern implementing cyclic execution behavior could be recommended or highly recommended on all safety integrity levels (as defined on IEC 61508-3:2010 table A.2 [7]).

How and where can design patterns then be obtained? Foundationally, design patterns document recurring solutions. The basic assumption is that at least three known usages for a solution need to be obtained to call a solution a design pattern [22]. Keeping this in mind at least the following pattern mining approaches can be considered.

As standards, such as the mentioned IEC 61508 and EN ISO 13849-1, provide requirements considering safety system design and development, they are potential candidates as source information. One potential approach is to take requirement clauses or required techniques or methods and search and provide practical solutions to fulfil the requirements. Depending on the standard and case, the standard may or may not provide instructions on how to actually apply and use required methods, techniques and clauses. Thus treating such elements as problems yields a way to found similar solutions and format them as patterns. For instance, one could consider graceful degradation, which is at least recommended on all SIL levels (as defined by IEC 61508-3:2010 table A.2), and mine patterns to design and implement graceful degradation on software. Using this approach, the integrity (or performance or similar quantity) levels can be directly linked to the patterns.

Literature and similar sources provide a feasible source for pattern mining. Solutions found from different literature sources can be considered pattern input. However, potentially the most credible sources for pattern mining are existing systems and their documentation. In the context of safety system patterns, such sources would be safety systems, their documentation and developers. To provide additional credibility for the mined safety system patterns (at least from the standard point of view), the patterns should be mined from inspected and approved systems. Such merit supports the patterns as the solution has been used as a part of an approved system. It should be noted, however, that a pattern originating from an inspected system does not directly implicate that the new system in which the pattern is applied, would be automatically approved. Nevertheless, such a pattern provides support and trust to believe that the solution is approvable in similar context.

Thus, ideally safety system patterns are mined from existing, inspected, and approved safety systems. As such, the solutions should be applicable on similar integrity level systems and also on lower levels although this is not always the case. Actually, by looking for instance IEC 61508-3 Annex A, this is not always the case. There are methods and techniques highly recommended, e.g., on SIL 3-4 and only recommended on SIL 1-2. Apparently the method or technique is still applicable, but it may be considered too heavy-weight or expensive for the lower integrity levels. To complement this approach, the inspection process and results could be systematically used to document the approved solutions in the form of patterns. During the process, the inspector approves and declines some of the solutions, approaches, and design decisions, which should be considered valuable input for future work. In the end, the inspections cost money and other resources to the customer so it is rational to try to minimize the process and to learn from mistakes and successful designs. Such work would support one of the purposes of patterns in the first place, that is, the systematic reuse of solutions.

IV. A PATTERN LANGUAGE FOR SAFETY SYSTEMS?

First of all, what do we mean by a pattern language? A pattern language is in our case a set of patterns that consider the same domain and are interconnected through relations. According to Eloranta et al., a pattern language is a concept “guiding the designer in building a coherent whole using patterns as building blocks” [6]. In this context, building block mindset, pattern relations and shared domain context between the patterns is seen centric to form the grammar to use the patterns. In practice, the pattern language defines restrictions, rules and suggestions on how to compose the designs of the provided building blocks. [6]. A collection of patterns, in contrast to a pattern language, does not have to have grammar or relations between the patterns.

The relations promote co-usage of the patterns as they guide a designer through the language by providing her with links indicating patterns that can be considered next, alternative, specialized and incompatible solutions related to the pattern that has been recently applied. Although the described approach may ease decision making, it may also narrow the designer viewpoint. A pattern language cannot include all possible solutions and the ones that are included, do not necessarily introduce the best alternative for a problem or situation under consideration.

One way to utilize the pattern language in design work was described above. The mentioned pattern relation based language walkthrough approach is a rather optimistic view at least if a large context is considered. Safety system development as well as other system development is a process consisting of multiple phases. Covering all of these with a single language of patterns is a large scale problem itself not to mention how to parse a meaningful language by establishing the pattern relations and interconnections. Still, patterns can provide pinpointed solutions to encountered problems and the related patterns may offer ideas during the design process. From our perspective, this is a more feasible use case for a safety system pattern language. To support the usage of the language, the patterns should be, however, grouped so that they resemble the corresponding design phases. That is, architectural patterns would benefit architecture design phase issues and implementation patterns (or idioms) the implementation phase issues.

The safety system design pattern language developed at the Tampere University of Technology has currently some 50 patterns and/or pattern candidates and some of them have been discussed in the workshops of patterns conferences [23]-[27]. (Pattern candidates are initial pattern ideas that do not yet have three known uses, that is, they are under construction. We have found writing pattern candidates an excellent way to communicate the ideas and find new known usages for the pattern candidates.)

In its current state, relations have not been specified for all the patterns of the language, but there are relations between the individual patterns. For example, patterns can specialize more general solutions in stricter contexts. Thus one could say the language lies somewhere between a pattern language and a collection of patterns at the moment. However, our purpose is to develop a full pattern language for safety system development.

We started the work in 2010 and the patterns have been collected, developed and published under various projects such as SULAVA, ReUse, and currently under DPSafe project. In the DPSafe project, we are working with several companies involved one way or another in safety systems design and development in the context of machinery applications. The target of the project is to mine and document design patterns considering software based safety functions and systems as well as gain new known uses for the existing patterns and identified pattern candidates. The participating companies include machinery producers, engineering offices, as well as software houses so there is potential to have different relevant views on the subject.

The patterns are targeted to safety system development. Currently, the language includes patterns and pattern candidates considering, for instance:

- development process
- risk mitigation strategies
- architecture and principles in terms of
  - software
  - hardware
  - system
- co-existence with control system
- scope reduction

In contrast to, for example, redundancy, diversity and other fault tolerance related matters, the sub domains mentioned above seemed to have less attention by pattern community. Thus our purpose is to extend the pattern approach to cover larger part of the safety system development outside the fault tolerance aspect. According to our work carried out in the DPSafe project, there seems to be a clear need for such an approach.

V. On Tool Support for Design Patterns

Whereas some of the benefits of patterns described in Section 3 could be achievable in any case, it is clear that tool support for patterns could increase their benefits significantly. For example, even without tool support, pattern names can become a part of the developer vocabulary [1]. Without a doubt, recurring solutions have also been used in the domain. However, using patterns to improve the traceability of standards solutions, for instance, would certainly benefit from automated functions already during the specification and modeling of the applications. Unfortunately, the support for patterns is in current software modeling tools restricted, at best. The purpose of this section is to discuss opportunities and challenges related to pattern tool support in safety system development. When appropriate, lessons learned from the previous work of the authors [18] will also be provided.

A. On Pattern Modeling

As mentioned, tool support for patterns is currently weak. For example, the pattern concepts of UML, structured collaborations [28], restrict patterns to describe the contents of the UML classifiers only. Thus, elements such as components and packages that would be useful in describing architectural patterns (for instance) cannot be used in patterns in UML [18]. The variety of published patterns in literature, however, covers problems on different levels of design and for various purposes. It cannot be said that all the patterns would be related to classifiers (classes) when all patterns are not even related to software systems. The origin of the (pattern) concept is in building architectures [2, 3] and there are also, for example, multi-technical pattern collections (such as [5]) with both software and hardware aspects. It is thus clear that the UML pattern concepts are currently too restricting, by nature.

With respect to the modeling of multi-technical patterns mentioned above, they could be used in SysML models, which are not restricted to software. However, the use of patterns would not have to be limited to modeling languages at all. For example, patterns could be equally useful in, for example, Computer Aided Design (CAD) tools and software Integrated Development Environments (IDE), in aiding practical design and programming work. Similarly to software engineering, also other engineering disciplines most certainly have recurring problems with known solutions.

While acknowledging this, in our work [18] the focus in developing tool support has been on safety systems and their UML and Systems Modeling Language (SysML) based modeling in a Model-Driven Development (MDD) context. With new pattern modeling concepts and by integrating them into both UML and SysML, the aim has been to support hardware aspects in addition to software and UML modeling. Safety systems are also systems that are developed and approved as a whole. Good practices and documentation are needed not only for software parts but for all parts of the systems, regardless of their implementation technologies. However, while the developed approach [18] currently allows pattern definitions and instances to consist of practically any modeling elements, the approach suffers from the drawback of not being easily portable to standard tools.

B. *On Pattern Instances*

In addition to (more or less) formal approaches, e.g., that of UML, modeling tools could support patterns also in an informal manner. Informal support has been developed into, e.g., MagicDraw that enables instantiating patterns from libraries by copying modeling elements. This functionality is not restricted to classifiers as is the case with standard UML. However, copying patterns (informally) can support mainly the aspect of using the solutions and not necessarily using the information about the use of the solutions. Copying model elements may not enable storing information about the elements being part of a pattern instance so that the information could be used for, e.g., documentation purposes.

There is existing research, e.g., [15] and [16], on detecting pattern instances in design models by searching for model structures that are similar to pattern definitions. However, it is questionable whether the use of such work would be an appropriate solution in safety system development. A developer does not use a design pattern by a coincidence. Instead, developers decide to apply patterns because they are facing challenges that they aim to solve with the solutions of the patterns. As such, it is natural that the decisions, which are architectural decisions, should be documented. Why should one try to guess whether a pattern has been applied when the decision could have been explicitly marked in the model when applying the pattern?

Identifying pattern instances based on markings could also be more reliable by nature than trying to detect instances with, for example, the mentioned comparison techniques. When patterns are used in design, they are applied to contexts in which it is feasible to use context specific names and to include additional properties. For example, a non-trivial subject (in an Observer [4] instance) should probably have properties (etc.) that the observer would be interested in. With context specific names, properties and surroundings (in the model), the results of comparisons could be less reliable. However, by marking pattern instances explicitly, the information should be as reliable as documentation is in general. In the end, it would be about the reliability of the developer that marks the pattern instances.

It is thus clear that the information on pattern occurrences should be stored (i.e., the pattern occurrences marked) when they are created. This is also the case in the approach of the authors [18]. Patterns, however, could be in general instantiated both manually and in a tool-assisted manner and the initiatives (to instantiate patterns) could come from either a developer or a tool.

C. *On Instantiating Patterns*

In a simple, conventional case, pattern instances can be assumed to be always created manually. In this case, it is natural to assume the markings (about the pattern instances) to be created manually, too. Otherwise, a tool would need to – somehow - know about a pattern being applied although the task would be performed by a developer. A tool could also include support for marking the pattern instances - without assisting in the pattern application task itself. However, also in this case the responsibility over the (possibly easily forgotten) marking task should be taken by the developer who knows about the pattern being applied.

Assuming that the pattern application process would be assisted by the tool, also the markings could be on the responsibility of the tool because the tool would know about the application. This thinking has also been used in our work [18]. When patterns are created with an interactive wizard, a developer can justifiably expect the tool to handle the markings. However, markings can be edited (and created) also manually. For example, functions to manually edit markings are needed when deleting or editing a pattern instance.

D. *On Initiatives to Instantiate Patterns*

In order to *actively* suggest a design pattern to be applied, the tool should have the ability to identify both the context and the problem at hand (in the design task) and to notice that they correspond to the context and problem of the pattern. If the active party was the developer, the tool would not necessarily need to have all the abilities. A set of suggested patterns, to be shown as a response to a user activity for example, could be narrowed down from all possible patterns based on the identification of context or problem. Naturally, with less information, not all the suggestions could be appropriate. However, it would still be up to the developer to make the decision.

Detecting a context of a pattern to match that at hand could be done based on a graph or semantic techniques, for example. However, there could still be challenges in formalizing contexts of many existing patterns that have been defined mainly with text. Identifying a problem, *what the developer would like the system to be like*, could be even more difficult to automate, and prone to errors.

If the active party to initiate an activity to apply a pattern would be the developer, also key words and search functions could be used to filter suggested patterns. This would not be possible if the active party would be the tool, so that the initiative would come prior to any user activity, i.e., prior to typing the key words. In addition, with the key words would come the problem of using different words to describe similar aspects. Nevertheless, key words could provide a sufficiently practical solution for suggesting patterns.

When suggesting patterns to use, a tool could also take advantage on information included - not in the patterns themselves - but in the pattern languages and collections that the patterns appear in. For example, when noticing a pattern to follow a recently used pattern in a pattern language and the problem of the pattern to match the context at hand, the pattern could be (at least) raised in a list of suggested patterns. Similarly, relations in pattern languages that indicate patterns solving the resulting problems of other patterns could be used in an automated manner to facilitate the work of developers.

In our work [18], pattern suggestions currently based on comparing the patterns that are used in models to collections of patterns that have been formed to correspond to the recommendations of standards. In the domain, this is meaningful since the standards govern and restrict the practical solutions that can (or should) be used by developers. However, the patterns are not yet suggested in any specific phase and the initiative to use patterns comes always from the developer. On the other hand, suggestions do not rely on the identification of either context or problem at hand. This could, however, be a possible future research direction.

In the domain, there can be also competence requirements for developers. As such, it can be assumed that appropriate solutions (patterns) are known by developers and that tool support for suggesting patterns would not even be a necessity. Nonetheless, automated functions can be useful in gathering information on the use of the patterns when there is reliable information about their presence available.

E. On Using Pattern Instances

When pattern instances are reliably detected (marked), the information can be collected from models for analysis purposes or to present it in a tabular, compact form. Especially this can be used to support traceability between solutions and their use, as demonstrated in [18]. Traceability is also a good example property in the (safety) domain because it is a property of systematic integrity and required from safety system development. As discussed in Section 3, the development process of software safety systems and applications consists of phases during which developers should apply appropriate techniques and measures that are to ensure the quality of the applications. Documentation is, though, needed to indicate how and where the techniques and measures have been used.

With pattern marks, it is also possible to automate different kinds of consistency checks, in addition to supporting traceability. For example, it can be made sure that patterns are appropriate for the safety levels required from the safety function or application. Naturally, this requires information on the applicability of the solutions to different levels of safety.

VI. Discussion and Conclusions

This paper has discussed the role of design patterns in facilitating the development of software safety systems and applications. Design patterns, which are essentially triplets of contexts, problems and solutions, are a means to systematically re-use design and proven solutions to recurring problems and needs. Their systematic use in the safety system development, however, has not been researched extensively although the re-use of recommended solutions is a general virtue in the domain.

Reasons why design patterns could, in general, benefit safety system development are various. Patterns document proven solutions, which provide designer support on selecting the solution to be used in the safety system under design. Known usages and ideally known usages from inspected and approved systems build this support. Patterns can illustrate practical approaches and solutions to alleviate the requirements considering safety system development given in standards, etc. This eases the burden of the designer by bridging the gap between standards and safety system design and implementation. In relation to this, patterns can be used as a part of documentation.

To provide designers with the patterns to be used in safety system design and development, we have mined and documented a set design patterns and pattern prototypes. The patterns consider various aspects of the safety system design including the development process, architecture, co-existence with basic control systems and scope minimization aspects. The work considering the pattern collection is in progress and current effort is to extend the collection to software based safety functions. New known usages for the existing patterns and pattern candidates are also being collected.

The development of safety systems is a systematic process that is governed by standards. Phases of the process build on information produced in the previous phases so that, for example, safety function requirements are specified to treat previously identified hazards and their associated risks. In the implementation phases of the process, developers are required to apply solutions, techniques and measures that are recommended by the standards and can be assumed to result in sufficient quality. However, in safety system development, it is not enough to apply the required techniques and solutions. Developers need to be able to prove the compliance of the applications to standards. This is where appropriate documentation - including information on the usage of the solutions - is needed.

Clearly, certifiable software parts of safety systems are not built by coincidences but by designing them systematically, with the use of appropriate solutions and techniques. As such, the applications need to be specified prior to their implementation, which usually includes at least their partial modeling. Unfortunately, the support for patterns is in UML, the de-facto software modeling language, restricted at best.

When developing pattern modeling approaches, however, patterns should be specified with dedicated modeling concepts and pattern instances marked in the models. In this way, reliable information on patterns could be used for documentation purposes and to automate consistency checks. In the future, tool support could be developed also for assisting developers in selecting patterns to use. However, this task should perhaps consider not only information included in the patterns themselves but also the information included in pattern languages and collections of patterns. Such collections could then be developed with the requirements of safety standards in mind.

REFERENCES

[1] K. Beck, et al., "Industrial experience with design patterns," in Proceedings of the 18th International Conference on Software Engineering, 1996, pp. 103-114.

[2] C. Alexander, S. Ishikawa, and M. Silverstein, Pattern languages. Center for Environmental Structure, vol. 2, 1977.

[3] C. Alexander, The timeless way of building. Oxford University Press, 1979.

[4] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994.

[5] A. Armoush, Design Patterns for Safety-Critical Embedded Systems. Ph.D. thesis, Aachen University, 2010. Available http://darwin.bth.rwth-aachen.de/opus3/volltexte/2010/3273/pdf/3273.pdf [referenced 25.6.2015].

[6] V. Eloranta, J. Koskinen, M. Leppänen, and V. Reijonen, Designing Distributed Control Systems: A Pattern Language Approach. Wiley Publishing, 2014.

[7] IEC, 61508: functional safety of electrical/electronic/programmable electronic safety-related systems. International Electrotechnical Commission, 2010.

[8] ISO, 13849-1:2006 Safety of machinery - Safety-related parts of control systems - Part 1: General principles for design. International Organization for Standardization, 2006.

[9] B. P. Douglass, Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems. Addison-Wesley, 2003.

[10] R. Hanmer, Patterns for Fault Tolerant Software. John Wiley & Sons, 2013.

[11] F. Buschmann, K. Henney, and D. Schimdt, Pattern-Oriented Software Architecture: On Patterns and Pattern Language. John Wiley & Sons, 2007.

[12] R. B. France, D. Kim, S. Ghosh, and E. Song, "A UML-based pattern specification technique", Software Engineering, IEEE Transactions On, vol. 30, 2004, pp. 193-206.

[13] P. Kajsa and L. Majtás, "Design patterns instantiation based on semantics and model transformations", in SOFSEM 2010: Theory and Practice of Computer Science, Springer, 2010, pp. 540-551.

[14] R. France, S. Chosh, E. Song and, D. Kim, "A metamodeling approach to pattern-based model refactoring," IEEE Software, vol. 20, 2003, pp. 52-58.

[15] A. Pande, M. Gupta, and A. K. Tripathi, "A new approach for detecting design patterns by graph decomposition and graph isomorphism," in Contemporary Computing, Springer, 2010, pp. 108-119.

[16] N. Tsantalis, A. Chatzigeorgiou, G. Stephanides, and S. T. Halkidis, "Design pattern detection using similarity scoring," Software Engineering, IEEE Transactions on, vol. 32, 2006, pp. 896-909.

[17] D. Jing, Y. Sheng, and Z. Kang, "Visualizing design patterns in their applications and compositions", Software Engineering, IEEE Transactions on, vol. 33, 2007, pp. 433-453.

[18] T. Vepsäläinen and S. Kuikka, "Safety patterns in model-driven development," The 9th International Conference on Software Engineering Advances (ICSEA 2014), Nice, France, 2014, pp. 233-239. ISBN: 978-1-61208-367-4.

[19] IEC, 62061: Safety of machinery - Functional safety of safety-related electrical, electronic and programmable electronic control systems. International Electrotechnical Commission, 2005.

[20] J. Rauhamäki, T. Vepsäläinen, and S. Kuikka, "Patterns in safety system development", The Third International Conference on Performance, Safety and Robustness in Complex Systems and Applications (PESARO 2013), 2013, pp. 9-15.

[21] B. Appleton, “Patterns and software: Essential concepts and terminology”, Object Magazine Online, vol. 3, no. 5, 1997, pp. 20-25.

[22] C. Kohls and S. Panke, “Is that true...?: thoughts on the epistemology of patterns”. In Proceedings of the 16th Conference on Pattern Languages of Programs (PLoP '09). ACM, New York, NY, USA, Article 9, 2009, 14 pages. http://doi.acm.org/10.1145/1943226.1943237.

[23] J. Rauhamäki and S. Kuikka, Strategies for hazard management process. The 19th European Conference on Pattern Languages of Programs (EuroPLoP 2014), 9.-13.7.2014, Irsee, Germany, ACM New York, NY, USA 2014. Article 31. DOI: 10.1145/2721956.2721966. ISBN: 978-1-4503-3416-7.

[24] J. Rauhamäki and S. Kuikka, Patterns for Sharing Safety System Operation Responsibilities between Humans and Machines. The VikingPLoP 2014 Conference, 10.-13.4.2014, Vihula, Estonia, 2014. ACM New York, NY, USA, 2014, pp. 68-74.

[25] J. Rauhamäki and S. Kuikka, Patterns for control system safety. The 18th European Conference on Pattern Languages of Program, EuroPLoP 2013, Irsee, Germany, July 10-14, 2013. ACM, 2013, Article 23. DOI: 10.1145/2739011.2739034, ISBN 978-1-4503-3465-5.

[26] J. Rauhamäki, T. Vepsäläinen, and S. Kuikka, Patterns for safety and control system cooperation. In: Eloranta, V.-P., Koskinen, J. & Leppänen, M. (eds.). Proceedings of VikingPLoP 2013 Conference, Ikaalinen, Finland 21.3. - 24.3.2013.Tampere University of Technology. Department of Pervasive Computing. Report 2, 2013, pp. 96-108.

[27] J. Rauhamäki, T. Vepsäläinen, and S. Kuikka, Functional safety system patterns. In: Eloranta V.-P., Koskinen, J., Leppänen M. (eds.). Proceedings of VikingPloP 2012 Conference, 17.-20.3.2012. Tampere University of Technology. Department of Software Systems. Report. Nordic Conference of Pattern Languages of Programs vol. 22, Tampere, Tampere University of Technology. 2012, pp. 48-68. Available: http://URN.fi/URN:ISBN:978-952-15-2944-3.

[28] OMG, Unified Modeling Language Specification 2.4.1: SuperStructure. Object Management Group, 2011.

Publication III

Rauhamäki, J., Kuikka, S. (2015). Patterns for control system safety. In *Proceedings of the 18th European Conference on Pattern Languages of Program.* [23] (ACM International Conference Proceeding Series). New York: ACM.

DOI: 10.1145/2739011.2739034

Patterns for control system safety

JARI RAUHAMÄKI and SEPPO KUIKKA, Tampere University of Technology

The main purpose of a control system is to operate a system under control so that it functions as desired. However, when a control system for a plant, process or device is being designed, safety-related aspects also need to be considered. In this article four design patterns for control system safety are illustrated. The patterns consider software architecture to implement interlock mechanism, design of the system to be safe when de-energized, and to check that operation in software has the desired response in the physical world. The patterns are applicable to safety systems and to control systems with safety-related aspects.

Categories and Subject Descriptors: Design patterns; Process control systems; Safety critical systems, Software system structures.

Additional Key Words and Phrases: Control system, safety

ACM Reference Format:

Rauhamäki, J. and Kuikka, S. 2015. Patterns for control system safety . EuroPLoP '13: Proceedings of the 18th European Conference on Pattern Languages of Program, Article 23 (July 2013) , 11 pages. ACM.

1. INTRODUCTION

Control systems are nowadays used to operate various systems from miniature to large scale. The purpose of a control system is to control the machinery so that it operates in a desirable manner. Part of desirable operation of a system is that the system under control does not cause an intolerable risk level to people, environment or the machine itself. Typically a dedicated safety system is employed to reduce the risk related to a hazard into a tolerable level. However, the control system itself should also take a stand for safe operation, especially in terms of safety of the machine itself. In this article we present four patterns related to control system safety. The patterns are applicable in control systems to implement safety related functions which, however, are not actual safety functions. Such safety related functions are sometimes referred to as SIL 0 (Safety Integrity Level 0) functions opposed to actual safety functions with SIL 1-4. In addition, the patterns can be applied in actual safety functions and systems as well.

The patterns presented here are a part of a larger collection of patterns. Six of the patterns of the collection have been published in VikingPLoP 2012 (Rauhamäki et al. 2012). The patterns of this article partly relate to the formerly published patterns.

The patterns are not directly discovered from real systems or applications. Instead, our approach is constructive: the patterns are sketched and documented based on our vision of a potential pattern and information gathered from standards related to safety system development, literature, and discussions. The approach includes both inductive and deductive parts as illustrated by Kohls and Panke (Kohls and Panke 2009). The core ideas of the patterns are mostly inducted from discussions, standards or literature, but the solution models are mostly result of deductive approach so far. Our goal is, however, to identify real world applications for all the patterns.

EuroPLoP '13, July 10 - 14, 2013, Irsee, Germany

DOI: http://dx.doi.org/10.1145/2739011.2739034

1.1 Elements of a control system

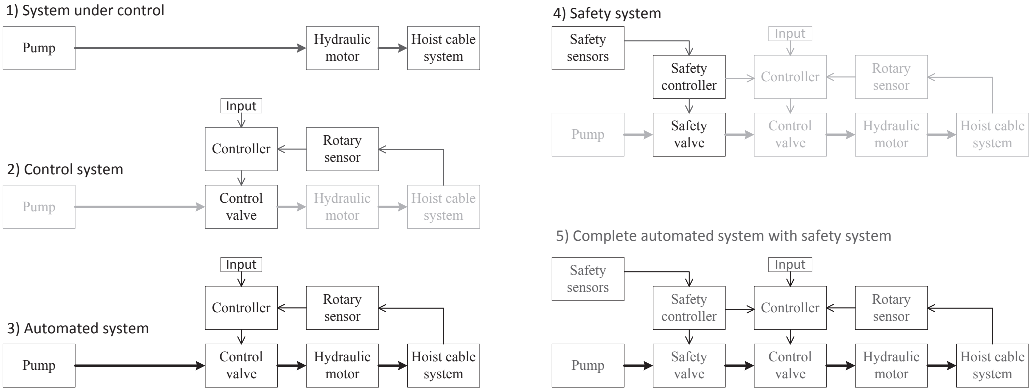


Figure 1: Illustration elements of a control system completed with a safety system.

Figure 1 illustrates basic concepts and elements of a control system operating alongside a dedicated safety system. In the figure a simplified structure of a hydraulic motor control application is sketched. The figure includes five parts each illustrating a logical part of the system. The considered application can be, for instance, a hoist cable system control of a crane (mobile or stationary). Notice that the presented system is simplified and many aspects have been left out for the sake of intelligibility.

The first illustration 1) presents the *system under control*. The system consists of a hydraulic pump and motor and a hoist cable system including transmission, cable, reels etc. In illustration 2), a *control system* is added to the system (the system to be controlled is greyed out). The control system consists of the following elements: controller, control valve and rotary sensor. The controller uses operator input to control the motor. Illustration 3) presents the *automated system* (control system + system to be controlled).

In the illustration 4) a *safety system* is given. The safety system consists of a safety controller, safety system sensors and a safety valve. Through the safety valve the safety system is able to actuate safety functions such as prevent lifting when support legs have no sufficient ground contact or overload is detected. Finally illustration 5) provides the whole system combining the automated system and the safety system. The safety system is connected to the automated system so that the safety system can communicate its state to the control system (and vice versa) (Rauhamäki et al. 2013).

1.2 Pattern overview

Figure 2 provides an illustration of the relations between the patterns described in this paper. The bold lined boxes represent the patterns considered in detail in this article. The arrows between the patterns represent a potential application order of patterns. For example, the OUTPUT INTERLOCKING pattern can be applied after the CO-OPERATIVE SAFETY ACTUATION. Table 1 provides short descriptions of the patters illustrated and referenced in this paper.

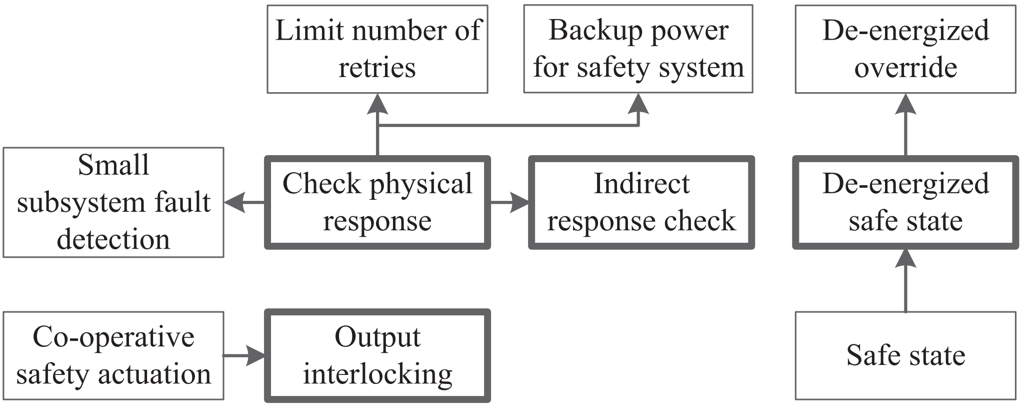


Figure 2: Relations of the patterns

Table 1: Short descriptions (patlets) of the patterns considered and referenced in this article

Pattern	Patlet
Check physical response	Operations and commands executed in software may success or fail in physical world though software continues execution. Therefore, use sensors to ensure that operations executed in software have presumed effect in physical world.
De-energized safe state	Power blackout or power supply failure causes malfunction of the safety system and introduces possible hazardous state of the system. Therefore, use power to keep system in operational state and let the system take safe state autonomously using stored potential energy.
Indirect response check	Adding dedicated hardware for checking that operations executed in software really occurred in physical world is costly and increases complexity and spatial properties of the system. Therefore, check operation success by indirect indication.
Output Interlocking	The control system must protect machinery, environment and humans from being damaged. Implementing protective interlocking functions in control algorithms makes the algorithms complex and hinders reusability of the algorithms. Therefore, use an interlock element alongside each control actuator output in the control system to separate the interlock logic from control logic.

Co-operative safety actuation (Rauhamäki et al. 2013)	How should the control system react on a safe state notification from the safety system? Let the safety system also drive the control system into a safe state whenever safe state needs to be obtained (according to the safety system).
De-energized override (Rauhamäki et al. 2012)	A safety system must be able to override the control system whenever systems control same process quantities. Therefore, let the safety system use de-energization of the basic control system's actuator(s) to obtain safe state.
Limit number of retries	Retry fault tolerance may lead to an infinite loop. Therefore, limit the number of retries per occurred fault to a reasonable level within time tolerances.
Productive safety (Rauhamäki et al. 2012)	A control system utilizes advanced and complex corrective functions to keep the controlled process in the operational state. These functions are very hard to implement in a safety system. Therefore, implement the corrective functions in a control system and use simple(st) approach for the safety system.
Safe state (Eloranta et al. 2010)	"If something potentially harmful occurs all nodes should enter a predetermined safe state." (Eloranta et al. 2010)
Small subsystem fault detection	It is problematic to transfer substantial information to high-level subsystems considering faults. Therefore, apply fault detection on as small subsystem level as possible and aggregate fault information for higher-level subsystems.

In the following sections we describe the four patterns for control system safety.

2. OUTPUT INTERLOCKING

Context

The PRODUCTIVE SAFETY pattern has been applied. A safety system implements fundamental safety functions, but safety-related and protective functions are also implemented into a control system to keep the system under control in a productive state.

Problem

Implementing protective functions in control algorithms makes the control algorithm complex.

Forces

- The control system needs to take into account safety-related aspects as they are used in keeping the system in operative region
- It should be possible to interlock a control actuator from multiple sources that may have different priorities in terms of the interlock functionality → interlock logic may become complex
- The main concern of a control loop and the control algorithm is to control the process variable of interest as instructed by the set point provided to the control algorithm considering the process variable

Solution

Use an interlock element alongside each control actuator output in the control system. The purpose of an interlock element is to implement the logic used to retain the system in a specified operation region. In contrast, the purpose of the control algorithm is to implement the logic according to which the process variable is controlled. That is, apply the principle of separation of concerns (Hürsch and Lopes 1995) to distinct the interlock logic from the control logic. The interlock element restricts actuator operation (e.g. through a control output element) so that the system remains in its specified operation region. For instance, an interlock element may force an actuator into a closed state, regardless of the control algorithm primarily operating the actuator. The interlock element gathers information considering the system under control and uses it to determine whether or not, and if so, how the control actuator output should be interlocked. Interlock element concentrates the logic into a single, well defined place and frees the control algorithm elements from interlocking related aspects.

The principle of control output interlocking is illustrated in Figure 3. The upper row in the control loop illustrates a basic control structure with a measurement input element (on the left-hand side), a control algorithm element (in the middle), and a control output element (on the right-hand side). The control algorithm reads the measurement input, calculates a control value, and operates the control actuator with the control value through a control output element. An interlocking element is added to the loop to manage control output interlocking. The interlocking element observers the measurement value and according to that determines whether or not the control output should be interlocked. Thus, the control algorithm element doesn't have to take a stand on interlocking issues.

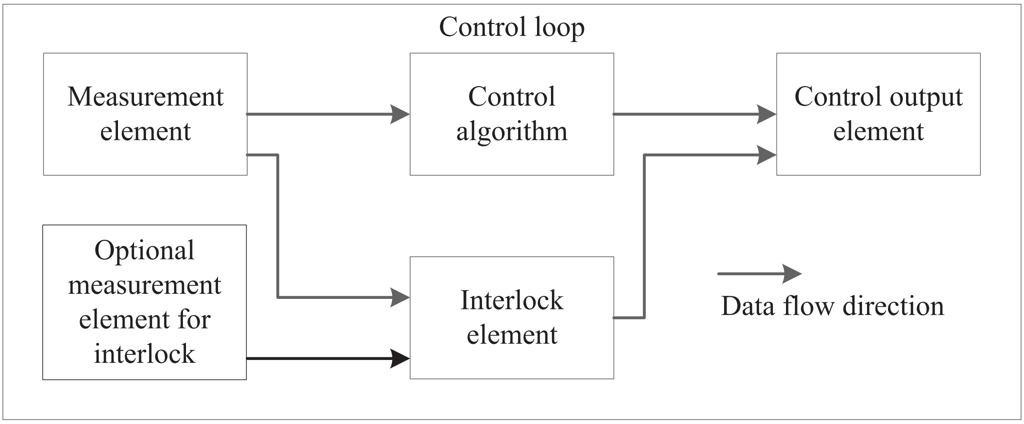


Figure 3: Basic principle of output interlocking

The interlock element can be a simple or complex entity. The number of inputs and outputs of an interlock element may vary in large scale. Largest interlock elements may have several even dozens of inputs. The inputs may consist of any signals used in the system such as measurement, control value, or interlocking signals. A control output may have more than one interlock elements attached to it if necessary and they can be connected in arbitrary ways to implement the desired functionality. In some cases it is beneficial to split the logic of complex elements into smaller entities, for example, to enable usage of readymade logic components.

To be able to function properly, interlock elements must have a way to override the control outputs into a suitable state. That is, an interlock element is able to force a control output element into a certain state regardless of the control value produced by the control algorithm. In an opposite situation, the interlock element typically releases a control output element under the control of the control algorithm. In some development environments and restricted modeling/development languages there is a possibility to use dedicated types of ports or signals to achieve such functionality. If such an environment is not used, (e.g. full variability programming language such as C is used) the developer needs to take care of this aspect. To make things even more complex, the safety control system must have override capability of the control system. On the other hand, the interlocking element can be used to transmit safety control system override and/or notification information to force the control system into a certain state.

Consequences

+ Control algorithms do not need to implement interlocking functionality
+ Increased expandability, maintainability and testability of control loop (elements)
+ Increased reusability of interlock and control logic
+ Opens a potential way to enable co-operative safety actuation
− Increased complexity of control output elements
− Increased workload due to additional element to be considered in control loops.

Example

Consider a hoisting crane's hoisting motor controller. The controller drives the motor attached to the cable to lift and lower the crane hook. Cranes have a maximum load they can handle due to their mechanical structure, platform support properties etc. In this case the crane measures the load through the torque of the hoisting motor. If the maximum load is exceeded the motor, cable, crane boom structures etc. may suffer damage. Thus it is necessary to prevent hoisting of overload. This is naturally a safety function, but

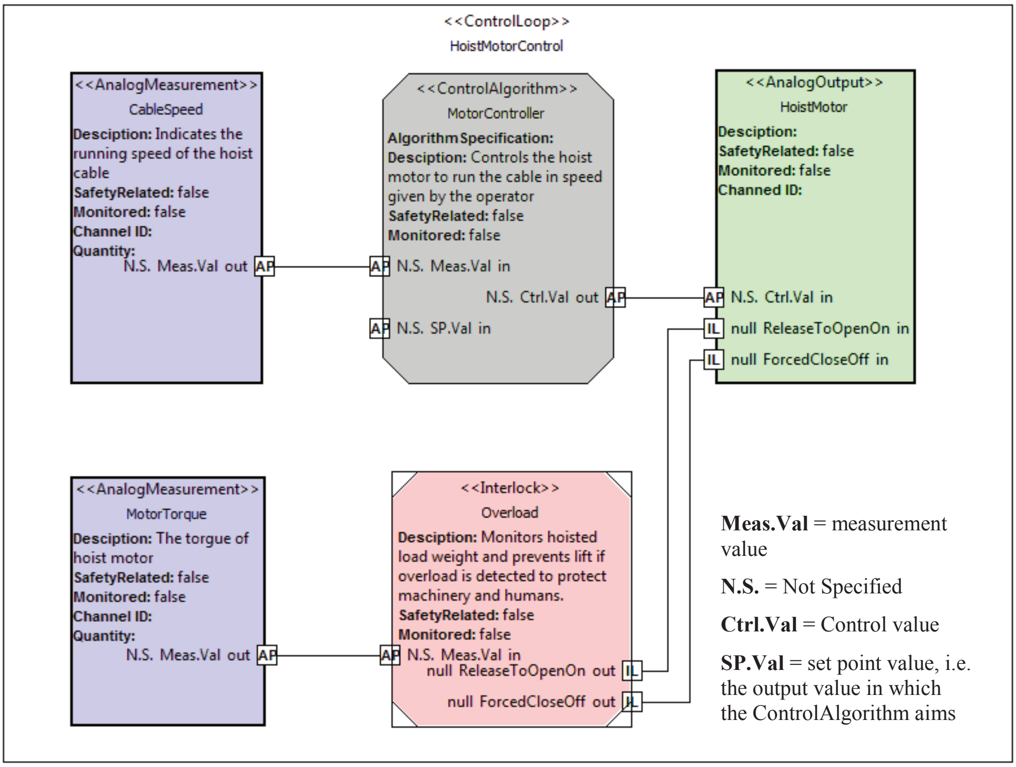


Figure 4: Interlock element in crane hoist motor control loop

high economic losses are also related to a breakdown of the machine. Thus, a protective function is added in the control system.

The protection is obtained with an interlock element in the hoist motor control loop. The loop is illustrated in Figure 4 using the Control Structure Diagram of the UML Automation Profile (Hästbacka et al. 2011). The stereotypes attached to the elements describe the nature of the element. The AP and IL ports attached to the elements illustrate Automation Ports (to transfer generic control system data) and Interlock Ports (to transfer interlock function related data) respectively. The connections illustrate data connection between the ports attached by the connection.

The upper row forms the motor control part with measurement, control algorithm and control output (AnalogOutput) elements. The lower row forms the interlock part of the loop. The overload interlock element includes the interlocking logic for overload protection. The element observes hoist motor torque. Normally the interlock is in ReleaseToOpen state in which the control output passes the MotorController control value to the hoist motor. If a specified torque limit is exceeded, the interlock forces the control output to a close-off state. Now regardless of the MotorController signal the output is closed (i.e. the actual motor is stopped).

It can be seen that the interlocking system is easy to maintain, expand, or modify if needed, because the interlocking logic is separated from the control algorithm.

Related patterns

The pattern may be used in conjunction with the CO-OPERATIVE SAFETY ACTUATION pattern to provide a way for the safety control system to force the control system actuators into a certain state.

3. DE-ENERGIZED SAFE STATE

Context

The system has a well-defined SAFE STATE (Eloranta et al. 2010), i.e. a predefined state that can be always obtained which minimizes risks for humans directly or indirectly through environment or devices related to the system.

Problem

Power supply for the safety and the control system as well as the system under control cannot be guaranteed, which might inflict a hazardous state during blackout or power loss in (part of the) safety system.

Forces

- Safety must be ensured in all reasonably foreseeable situations including power cut-offs
- The system itself and the safety system as well use energy to operate
- Usage of a reliable backup power source is typically expensive and sometimes not an option at all to ensure safety system operation
- No power source or energy transfer arrangement is infallible

Solution

Design the safety system (and control system as well if applicable) to take the safe state when power is lost. By following this principle, the probability that the system reverts into safe state, if there is no energy to keep it in operative state, is increased[1]. The principle can be applied in both safety-critical as well as non-safety-critical devices and systems.

The actual implementation of the de-energized safe state principle depends on the system, safe state conditions, devices, etc. The only generic approach is to use potential energy of some kind to apply to the safe state. The potential energy used to actuate the safe state generates a force that is overcome by a force generated with energy used to operate the system when the control system has power available. The potential energy can be stored e.g. in springs, pressure accumulators, batteries, etc. When power is cut-off from the control system(s) or actuator(s) the potential energy will position the actuator(s) and thus the system in a safe state.

The following steps provide an outline for designing a system to obtain safe state when de-energized:

(1) Define the safe state of the (sub)system under consideration (This can be only found for certain types of systems. For instance, an airplane has no implicit safe state when airborne.)

(2) Validate that the safe state can be obtained by de-energizing the (sub)system. Certain type of safe state such as: “active cooling by circulating water” are hard to obtain in de-energized state

(3) Define which actuators of the system are needed to enter and retain the safe state.

(4) For each required actuator, define the position (typically open or closed) for the safe state.

(5) Verify that there are no conflicts or hazards in the system when N actuators enter safe state due to power loss (e.g. cabling breakage).

(6) For each required actuator, choose an actuator type that takes the defined state when de-energized (e.g. normally open or normally closed) or design a similar functionality.

Consequences

+ Safe state can be obtained when power is lost from (a part of) the system
+ Reduces or eliminates need for backup power arrangements
− Wastes energy as power is used to keep the actuators in operative positions
− Decreases alternatives of suitable actuators and their positioners
− Possibly increases the cost of safety and control system actuators
− Increases testing effort due to test effects of de-energization of actuators
− Increases maintenance effort due to test of operation of the system during power loss

Example

A good example of usage of the de-energized safe state is a spring loaded hydraulic valve. Figure 5 illustrates a hydraulic system with de-energized safe state principle applied. Both valves are spring loaded to take safe state whenever power is lost. When power is lost the magnetic elements can no longer push

[1] The measures that return a system into a safe state (or actuators required to obtain safe state) may also fail.

the valves open as there is no electric power to generate force to overcome the spring loading. Notice that the principle is applied in both safety and control systems, which increases the likelihood of successfully taking safe state when power is lost.

As discussed earlier the measures to return the system into a safe state when unpowered do not provide fully guaranteed operation. In the hydraulic system discussed above a potential problem in operation of the de-energized safe state are, for example, related to breakdown of the spring loading or impurities within the hydraulic fluid that may block valve.

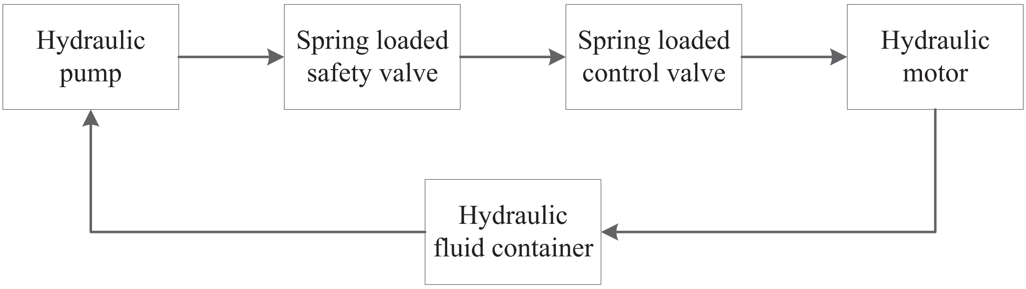

Hydraulic pump
Spring loaded safety valve
Spring loaded control valve
Hydraulic motor
Hydraulic fluid container

Figure 5: De-energized safe state applied on hydraulics

Known use

Pneumatic parking brakes used in heavy vehicles etc. use the principle of de-energized safe state. The brakes are spring loaded and they are operated with pressurized air. Pressurized air released into a brake unit (when the parking brake is removed) applies an opposed force to the spring which detaches the brake pad from the brake disk. Now, if the air compressor fails, loses power or the piping fails for instance, the spring applies the parking brakes.

The principle of de-energized safe sate is also suggested by the Machinery Directive (European Parliament and of the Council 2006). In clause 1.2.6 of the directive states “The interruption, the re-establishment after an interruption or the fluctuation in whatever manner of the power supply to the machinery must not lead to dangerous situations.” (European Parliament and of the Council 2006) By applying the de-energized safe state, interruption of power supply transfers the system (or the affected part of it) into the SAFE STATE (Eloranta et al. 2010). The IEC 61508-7, in section A.1.5: Idle current principle, also suggests usage of the de-energized safe state approach (International Electrotechnical Commission 2010).

Related patterns

The BACKUP POWER FOR SAFETY SYSTEM pattern describes a possible approach to circumvent the principles of the DE-ENERGIZED SAFE STATE pattern and to decrease waste of energy and other resources.

The DE-ENERGIZED OVERRIDE pattern (Rauhamäki et al. 2012) describes how the principle of de-energization can be used by a safety system to override a control system.

4. CHECK PHYSICAL RESPONSE

Context

Control logic software is being developed. The logic controls physical actuator elements the operation of which is critical from the safety aspect.

Problem

Operations and commands executed in control logic may succeed or fail in physical world unrecognized by the logic.

Forces

- The control logic needs to be aware whether the operation executed correctly in the physical world
- Additional hardware can be tolerated in the system (e.g. in terms of cost, space and complexity, etc.)

Solution

Use sensors to ensure that operations executed in control logic have the presumed effect in the physical world. When operation is executed in control logic, it is confirmed that the corresponding action in the physical world also occurred. Provide the system with suitable sensor elements if none of the existing ones can directly sense effects of the executed operations. In control logic it is easy to execute commands and presume they were successfully executed. This is often due to omission of return value checks or presuming success if no exceptions were caught.

A generic principle of operation checking is illustrated in Figure 6. First, an operation to affect a physical world quantity is executed. Then, control logic checks if the quantity was physically affected. This may take some time (e.g. valve opens and allows flow rise in a certain rate, not instantly (from time perspective of a computer). If the operation was successful execution can continue. If not, the quantity is checked again. To prevent an infinite loop, the number of maximum checks is specified (see LIMIT NUMBER OF RETRIES pattern). If no success is achieved, an error is detected and suitable actions can be taken. For example, manual actuation of a valve can be requested.

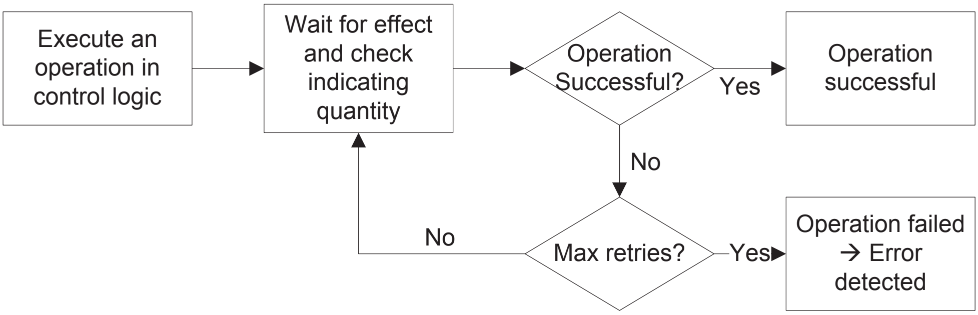

Execute an operation in control logic
Wait for effect and check indicating quantity
Operation Successful?
Yes
Operation successful
No
No
Max retries?
Yes
Operation failed → Error detected

Figure 6: Checking operation success in control logic

To be able to identify changes in the physical world, sensors are required. A control system has (typically) some sensors monitoring the physical world. However, these sensors do not necessarily monitor the quantity from which the success or failure of an operation could be identified directly. In such cases, an additional sensor needs to be added in the system. This increases cost, complexity, and spatial properties of the system.

The primary sensor candidate to be used in determining successful operation is the sensor that triggers the safety function if one exists. For instance, the best sensor to indicate that overfill prevention has succeed is the level indicator that originally indicated the overfill situation. However, in some cases such indicators are too slow or don’t exist so they cannot be directly used as indicators of successful operation.

Consequences

+ Operation of physical world devices can be ensured in the control logic
+ System is added with a fault detection method
+ Direct measurement of the quantity of interest provides direct information on the operation success
− May require additional (sensor, IO-board, wiring, etc.) hardware
− Increases complexity of the system
− Decreases performance of the control logic as additional checks (with potential waiting) are executed

Example

Consider a steam operated heating process. A safety system is implemented to disable steam flow to prevent over temperature of the heated container. When the safety valve is closed, the safety system software ensures that steam flow is actually stopped. The safety system measures steam flow. If it appears that the steam flow did not stop, an error is detected and suitable actions can be taken.

Related patterns

The SMALL SUBSYSTEM FAULT DETECTION pattern illustrates where to deploy fault detection functionality in software considering physical world occurrences.

The INDIRECT RESPONSE CHECK pattern describes a way to check effect of a control logic operation to the physical world indirectly, e.g. in case of non-existing direct sensor element or slow real world operations. In some cases the safety-related piece of software cannot stay waiting for direct response when the observed quantity changes slowly after command.

5. INDIRECT RESPONSE CHECK

Context

Control logic software is being developed. There is no sensor for directly measuring a quantity indicating success of an operation executed in control logic. The system includes sensing elements that can indirectly indicate success of an operation.

Problem

Operations and commands executed in control logic may succeed or fail in physical world unrecognized by the logic.

Forces

- The control logic needs to be aware that the operation executed correctly in the physical world
- Adding a dedicated sensor in the system to indicate whether or not a operation was successful is costly and increases system complexity
- Direct measurement of the effect of the operation is suboptimal due to process dynamics, e.g. slow response

Solution

Check operation success by indirect indication. Typical control systems use multiple sensing elements to measure the state of the system under control. Some of these sensors may provide indirect indication that the operation executed in software has had a sensible effect in the physical world. Indirect indicators can be used alongside direct indicators to support decision making.

The first problem to apply the approach is to identify a suitable indirect indicator. Unfortunately no generic approach can be given. One need to consider what should happen when an action is taken and which process quantities are affected. To catch the idea of indirect indication, consider the following cases:

- Closing a valve in a steam supply line may for example stop the steam flow, increase pressure in the pipeline or increase steam flow in another location (pressure relief line)
- Starting a hydraulic pump may for example increase electric current through the pump, increase pressure and flow, generate more heat (slowly), increase sound pressure and vibration level in presence of the pump

The emerging issue in the approach is unreliability. Especially in complex systems it is difficult to ensure correct cause and effect relations of executed operations. That is, it may be hard to demonstrate that an indicator really indicates success (or failure) of an operation. In some cases the indirect measurement may indicate success of the operation, but the operation may have failed due to other reasons. For instance current flowing into an electric fan does not necessarily indicate that the motor is actually rotating or providing air flow to the target application. The problem gets worse the more indirect the used indicator is. A good primary approach is to use the least indirect indicator.

Consequences

+ No need for additional sensor hardware for checking purposes
+ Provides a “free” method to enable detection of operation success
+ In some cases may provide a better approach (e.g. in terms of sensing speed) to direct measurement of the effect

− Less reliable method than direct measurement
− Though success of an operation is indicated by an indirect indicator, the desired effect may have not succeeded

Example

Cooling fans are turned on by control logic. The purpose of cooling fans is to lower temperature of the element under cooling. The cooling effect is, however, relatively slow. Thus the software uses other means to detect that the fans actually started. Potential methods include e.g. current consumption of the fans, measurement of air flow, or generated noise for instance.

Related patterns

The CHECK PHYSICAL RESPONSE pattern describes the main idea of checking that commands executed in control logic have the desired impact in physical world using direct measurement of the process variable of interest.

6. ACKNOWLEDGEMENT

The authors would like to greatly thank the workshop group members in the EuroPLoP event and especially our shepherd Christopher Preschern for their valuable feedback and input to improve the patterns and the paper.

7. REFERENCES

Eloranta V.-P., Koskinen J., Leppänen M., and Reijonen V. 2010. *A Pattern Language for Distributed Machine Control Systems*. Tampere University of Technology, Department of Software Systems, Tampere, 2010. Retrieved April 2013, from Patterns @TUT: http://patterns.cs.tut.fi/publications.html

European Parliament and of the Council. 2006. Directive 2006/42/EC of the European Parliament and of the Council, vol. L 157/24.

Hürsch, W., Lopes, C. 1995. *Separation of Concerns*. Technical Report NUCCS-95-03. College of Computer Science, Northeastern University, Boston, MA. Retrieved November 2013 from Citeseerx: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.125.2723&rep=rep1&type=pdf

Hästbacka D., Vepsäläinen T. and Kuikka S. 2011. Model-driven development of industrial process control applications, *Journal of Systems and Software*, 84, 7, 1100--1113.

International Electrotechnical Commission. 2010. Functional safety of electrical/electronic/programmable electronic safety-related systems. Part 7: Overview of techniques and measures, IEC.

Kohls C. and Panke S. 2009. Is that true...?: thoughts on the epistemology of patterns. In *Proceedings of the 16th Conference on Pattern Languages of Programs (PLoP '09)*. ACM, New York, NY, USA, Article 9,14 pages. http://doi.acm.org/10.1145/1943226.1943237

Rauhamäki, J., Vepsäläinen T. and Kuikka S. 2012, Functional safety system patterns. In *Proceedings of the VikingPloP 2012 Conference.* Tampere University of Technology, Department of Software Systems, Tampere, 48--68. http://URN.fi/URN:ISBN:978-952-15-2944-3

Rauhamäki, J., Vepsäläinen, T. and Kuikka, S. 2013. Patterns for safety and control system cooperation. In *Proceedings of VikingPLoP 2013 Conference*, Tampere University of Technology. Department of Pervasive Computing. Tampere, 96--108. http://URN.fi/URN:ISBN:978-952-15-3167-5

Publication IV

Rauhamäki, J., Kuikka, S. (2014). Strategies for Hazard Management Process. In *Proceedings of the 19th European Conference on Pattern Languages of Programs.* [31] (ACM International Conference Proceeding Series). New York: ACM.

DOI: 10.1145/2721956.2721966

Strategies for Hazard Management Process

Jari Rauhamäki
Tampere University of Technology
Korkeakoulunkatu 3
33720 Tampere, Finland
+358401981164
jari.rauhamaki@tut.fi

Seppo Kuikka
Tampere University of Technology
Korkeakoulunkatu 3
33720 Tampere, Finland
+358408494540
seppo.kuikka@tut.fi

ABSTRACT

When a system is being designed, the hazards and corresponding risks introduced by the system must be identified. Mitigation of risks is required if they are found intolerable. To mitigate risk there are multiple valid possibilities, but some are more preferable than other. Hazard elimination is the most preferable approach but it is not always applicable. In such case substitution, isolation and active protective measures in form of electric, electronic and programmable electronic systems need to be considered. In this paper, we illustrate some hazard management methods and provide the suggested order of consideration for the methods in format of strategy collection.

Categories and Subject Descriptors

D.2.11 **[Software Architectures]**: Patterns; K.4.1 **[Public Policy Issues]**: Human Safety; J.7 **[Computers In Other Systems]**: Industrial control Process control

General Terms

Design

Keywords

Safety systems, risk mitigation, and hazard management

1. Introduction

When a system, whether a machine or a process, is being designed, the hazards introduced by the system need to be identified. In addition it is equally important to assess the risks associated with the hazards and decide if they are intolerable. In this context we shall use the following definitions:

- Harm: "physical injury or damage to the health of people or damage to property or the environment" [6]
- Hazard: "potential source of harm" [6]
- Risk: "combination of the probability of occurrence of harm and the severity of that harm" [6]
- Protective measure: "measure intended to achieve risk reduction" [2]

In this paper, a set of strategies considering the methods for hazard management is introduced. The purpose of the paper is to format a potential approach to begin the process of safety system design into a set of strategies. The strategies form a chain of actions and design decisions considering how risks and hazards related to the system are identified and mitigated on an architectural design level.

We begin with the root action of nearly any safety system development process, which is hazard and risk identification. One cannot mitigate risk or remove a hazard if one does not know what are the hazards related to the system and what are the corresponding risks. Therefore hazard and risk assessment has to be carried out. When the hazards and related risks are known, the controls for intolerable risks are considered. The strategies illustrate potential mitigation approaches also known as the hierarchy of hazard control and indicate the recommended consideration order of the actions as provided in the hierarchy. Figure 1 illustrates the strategies introduced in the paper and the preferable order of application, that is, one should initially start with safety risk identification and to mitigate the risk, first consider hazard elimination. Table 1 provides short descriptions of the strategies. The strategies illustrated with dashed outline on Figure 1 are not included in the paper. However, they may be included in a paper to be published in future.

The strategies have been mined from standards considering the development of safety system and literature sources. If a method is required by a widely applied standard, for instance IEC 61508 [6], it provides, from our perspective, sufficient proof for the method. For the latter approach, the methods have been discovered from official guidelines and other sources such as literature and authority guidelines. The main application domain of the guidelines is workplace safety but the same principles can be applied in machinery and process system domains as well. The strategies belong to a collection of patterns from which parts have been previously published [13], [14] and [12].

Table 1. Short descriptions of the patterns mentioned in the paper

SAFETY RISK IDENTIFICATION	To make conscious decisions considering hazard and risk management, information on these aspects need to be available. Therefore,

EuroPLoP '14, July 09 - 13, 2014, Irsee, Germany
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3416-7/14/07…$15.00
http://dx.doi.org/10.1145/2721956.2721966

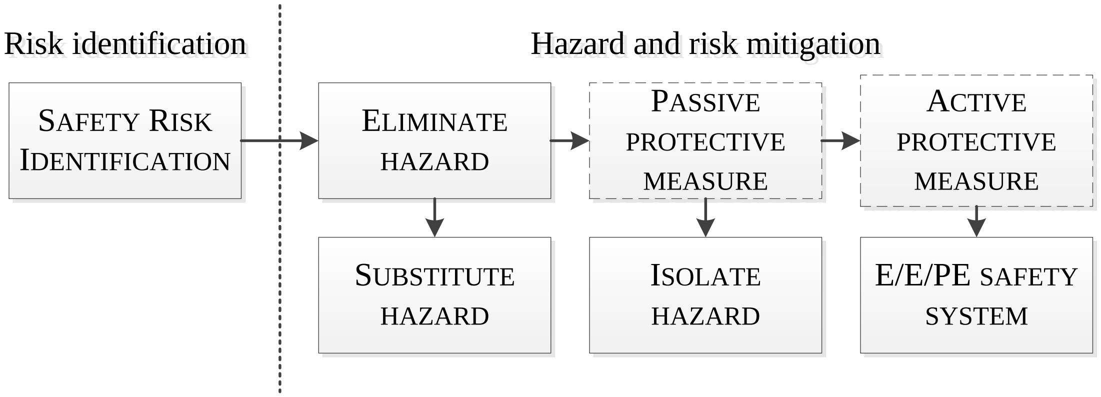

Risk identification
Hazard and risk mitigation
SAFETY RISK IDENTIFICATION
ELIMINATE HAZARD
PASSIVE PROTECTIVE MEASURE
ACTIVE PROTECTIVE MEASURE
SUBSTITUTE HAZARD
ISOLATE HAZARD
E/E/PE SAFETY SYSTEM

Figure 1. Strategy - pattern map

	use structured and/or systematic method(s) to identify the hazards and associated risks introduced by the system.
ELIMINATE HAZARD	You want to maximize the likelihood that a hazard introduced by a system cannot cause harm in any part of the system lifecycle. Therefore, eliminate the hazard completely by removing the component introducing the hazard from the system.
SUBSTITUTE HAZARD	A hazard is wanted to be eliminated from the system. Therefore, substitute the hazardous element with a non-hazardous or at least less hazardous element.
PASSIVE PROTECTIVE MEASURE	The system introduces a hazardous element of intolerable risk, which cannot be eliminated from it. Therefore, use a passive protective measure of a fixed nature to mitigate the risk.
ISOLATE HAZARD	A hazardous element needs to remain in the system and with the current exposure profile, the related risk is intolerable. Therefore, isolate the hazard by physically isolating the hazardous element from the environment and people.
ACTIVE PROTECTIVE MEASURE	Passive protective measures are unable to achieve desirable secondary quality attributes in context of the considered protective measure. Therefore, use an active protective measure that is able to monitor the state of the system and/or its environment and affect the operation of the system or itself accordingly.
E/E/PE SAFETY SYSTEM	Active protective measure of relatively complex functionality needs to be implemented. Therefore, use an electric, electronic or programmable electronic (E/E/PE) safety system to implement the protective measure functionality.

2. Safety risk identification

2.1 Context

A system, for instance an industrial process or a machinery construction, is being designed and the development project has limited resources. The system under development may introduce safety risks, but they are not (completely) known beforehand. Hazards and their associated risks potentially introduced by the system are wanted to be managed and mitigated in a justifiable manner. That is, on one hand, consciously manage all the risks introduced by the system, and on the other hand, deploy meaningful management measures against each risk and hazard. Information regarding the system under development exists (the type of the system, initial design, target user group, etc.)

2.2 Problem

To make conscious decisions considering hazard and risk management, information on these aspects need to be available.

2.3 Forces

- Realization of, especially, a high severity risk causes financial losses, severe injury or loss of life. That is, all the foreseeable risks should be mitigated into a tolerable level.

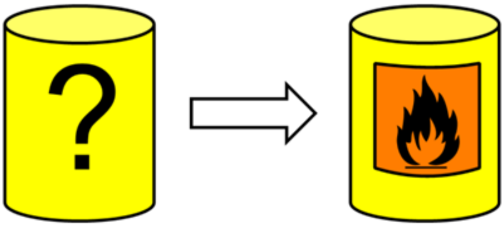
- Over-mitigation of a risk adds little value, but adds a considerable amount of cost as each risk mitigation measure increases development, component, manufacturing, and maintenance cost of the system. The cost realizes in the design, construction and maintenance phases of system lifecycle.

- Under-mitigating a risk leads to realization of the risk in higher probability or with more severe consequences compared to meaningfully mitigated option. That is, the risk may realize more frequently or have more severe consequences than is acceptable. There is also the cost of negative publicity. All the costs realize when the risk realizes. Before this, money can be saved compared to the case in which hazards and risks are identified and mitigated into tolerable level during development process.

- Deploying protective measures without identifying and analysing the risks, that is, taking a haphazard approach, is likely to lead to either under- or over-mitigation of the risks. Both are likely to increase the system lifecycle cost.

2.4 Solution

Use structured and/or systematic method(s), such as Failure Mode and Effect Analysis (FMEA), Fault Tree Analysis (FTA), or Hazard and Operability Study (HAZOP), to identify the hazards and associated risks introduced by the system. Only hazards and risks which are identified and assessed can be justifiably mitigated. If protective measures are added without knowledge of the hazard and the related risk, there is no foundation for the applied measures.



The key to success is to identify hazards and associated risks as early as possible in the system lifecycle. The later the hazard and risk is identified, the more costly the mitigation will be. To take most out of the hazard and risk assessment, it should be carried out several times during the development process. Each iteration will have more detailed input information so the potential to identify hazards and assess the associated risks meaningfully is increased. Also, keep in mind that different hazard and risk identification methods characteristically suit in different applications. Thus, it is typically a good idea to apply different methods for the purposes they fit rather than trying to apply one method to all purposes.

The safety risk identification is divided into two parts: hazard analysis and risk analysis. In hazard analysis phase possible hazards related to the system are identified. Hazards can be identified using various methods such as failure mode and effect analysis or hazard and operability study. These methods may also reveal information about how the hazard occurs. This information is valuable while risk mitigation methods are considered and selected.

The risk analysis is based on the hazard analysis. The risks are defined for the identified hazards. A hazard is considered and the risk related to the hazard is defined by the probability of hazard realization and the consequences of realization of the hazard. Qualitative or quantitative value for the hazard probability and consequence is defined. The actual risk is decided as a combination of the probability and consequence of the considered hazard. Again, there are numerous methods to decide the final risk reference value, such as the multiplication of the risk factors or utilization of a risk table.

Consequences

+ Risks are assessed and mitigation needs can be evaluated. Thus, the resources can be allocated to aspects that are most relevant from safety perspective.

+ Risks can be mitigated into tolerable level as there is knowledge available on them.

+ The hazards and risks identified can be mitigated during development process in which case it is typically cheapest to mitigate them.

+ The hazard identification may have produced valuable information on the reasons leading into harm which can be used when mitigation methods are selected.

+ From legal point of view, it is valuable to have documented evidence that hazards and risk were considered using structured method, in case a legal issue rises afterwards.

− Hazard and risk analysis requires resources and thus adds design and development cost. However, almost in any case this investment is many times cheaper compared to mitigations designed and applied later in the system lifecycle. For instance, consider costs of a few days of additional time spent in hazard and risk analysis compared to retrofitting protective measures in existing products around the market area.

− Unreal sense of control over hazards and risks may have been achieved during analysis process.

2.5 Example

Consider the simple electronic low-pass filter circuit illustrated on Figure 2. Our task is to identify and analyse potential failures of the low-pass. In this simplified example case the failure is carried out using fault tree analysis. We shall limit the analysis on typical failure modes of the electrical components of the circuit, namely the resistor R and the capacitor C. Wires, input voltage source Vin, and other aspects are not considered in the analysis.

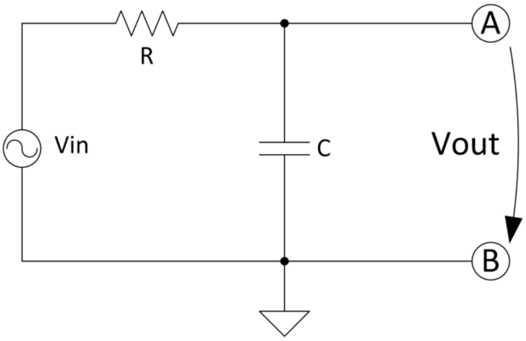


Figure 2. Schematic figure of a simple electric low-pass filter

On Figure 3 a fault tree analysis for the low-pass filter failure is illustrated. Based on the analysis there are three main causes for the low-pass filter failure, which are caused by different combinations of failure modes of the capacitor C and resistor R. For instance No or very low output is occurs if either capacitor C fails short (circuit) or resistor R fails open (circuit). Note that the analysis stops on this level. For more detailed analysis one could consider why the resistor fails open, for example.

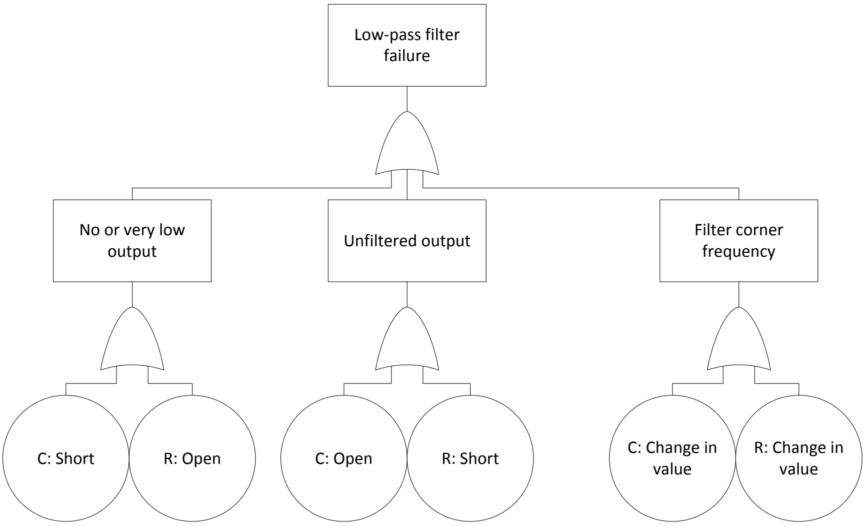


Figure 3. Fault tree analysis of low-pass filter circuit

The fault tree analysis indicated the failure modes of the low-pass filter. Now the related risks need to be assessed and this analysis is illustrated on Table 2. The likelihood estimates are constructed according to component specific failure mode probabilities (Table 3), assuming, for the sake of simplicity, the same failure rate per million hours for both capacitor and resistor components. The severity is estimated according to the function of the circuit in its context, which is out of the scope of this example.

2.6 Known use

The EN ISO 12100 [2] and IEC 61508 2010 [6] require hazard and risk analysis as a part of the development process of any safety-related system regardless of their implementation (software or hardware).

Table 2. Risk analysis of low-pass filter failure modes

Failure mode	Likelihood	Severity	Risk
No or very low output	Very High	High	Very high
Unfiltered output voltage	Very Low	Medium	Low
Wrong filter corner frequency	Medium	Low	Medium

Table 3. Failure mode distribution of the low-pass filter components according to [9, p. 440-444]

Device type	Failure mode	Mode probability
Capacitor, ceramic type	Short	0.49
	Change in value	0.29
	Open	0.22
Resistor, film type	Open	0.59
	Parameter change	0.36
	Short	0.05

2.7 Related patterns

When a hazard introducing an intolerable risk is identified, the ELIMINATE HAZARD strategy should be considered initially to remove the hazard so that it cannot occur.

3. Eliminate hazard

3.1 Context

Risk analysis for the system under development has been carried out and a hazard has been identified. The risk level related to the hazard is intolerable without mitigation actions. That is, either the consequence or the probability of the realization of the hazard needs to be mitigated.

3.2 Problem

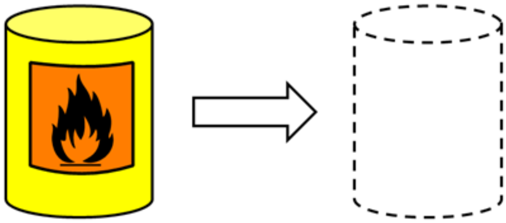
You want to maximize the likelihood that a hazard introduced by a system cannot cause harm in any part of the system lifecycle.

3.3 Forces

- No other hazard mitigation method has been considered yet. The approach should be the primary approach to mitigate hazards.
- Hazard elimination does not intolerably decrease productivity of the system.
- Other quality attributes can be sacrificed in preference of safety. For instance, usability or performance of the system may be able to be compromised if necessary.
- Hazard elimination does not introduce intolerable increase in costs of usage of the system in context of constructability, main operation execution, maintainability or other aspects affecting life cycle costs.

3.4 Solution

Eliminate the hazard completely by removing the component introducing the hazard from the system. Designing the hazard out this way is the most effective measure to prevent the realization of the considered hazard. This is due to fact that if hazard is eliminated, it is not able to realize anymore (unless the hazard is reintroduced in the system).



Elimination of the hazard should be considered as the primary means of risk reduction related to a hazard. The hazard is not only eliminated from the system in normal operation, but also during special situations such as the maintenance and deconstruction of the system. Going even further, the hazard may be eliminated also in larger scale, in best case throughout the whole supply chain. That is, hazard is not only eliminated on the site considered, but for next usage sites of the considered product, device or system.

Eliminate the hazard by removing the cause or sufficient subset of the causes leading to a hazard of the system. The root cause or a sufficient subset of the causes can be identified using, for instance, a fault tree analysis, but any hazard identification method revealing the path to the hazard is applicable. However, removing a critical path to the hazard is prone to human error in the hazard analysis phase. For instance, even though a large amount of resources is put to hazard analysis for a nontrivial system the analysis outcome is typically incomplete in terms of coverage of the hazards and potential paths leading to the hazards. Thus, eliminating a critical path is only a secondary approach. The only reliable way to eliminate a hazard is to remove the root cause of the hazard. For example, to eliminate a drowning the only reliable way is to eliminate any liquid from the system.

Even though elimination of the hazards is the preferred approach, it has to be carefully considered if the elimination of a hazard introduces new, possibly even worse, hazards. There is also a possibility that the originally eliminated hazard is reintroduced in the system in some other part of the design.

Though an effective measure, hazard elimination cannot always be used. This may relate to numerous reasons. One major reason is that elimination of a hazard would prevent the operation the machine is intended to in the first place. For example, the dismissal of a falling hazard in an elevating work platform would require that platform is not lifted from ground or completely closing the platform which prevents the usage of the system for the purpose it was first intended to and still the platform itself might fall. Another example is a cutting machine. In such machine, there is a risk of the operator being cut by the machine. However, the hazard cannot be eliminated as the machine is supposed to cut, so the cutting element and functionality has to be retained in the system. Elimination of a hazard can also prove impractical for other reasons, such as operating cost. For example, a toxic chemical could be replaced with a non-toxic one, but this implies raise in cost due to decrease in the productivity or quality of the product.

3.5 Consequences

+ Hazard is completely eliminated diminishing the risk related to it.

+ Hazard cannot occur either in normal or abnormal conditions (e.g. unplanned maintenance operations).

+ There is no maintenance cost for the safety system as there is no need for a safety system to mitigate the eliminated hazard, which also follows the safe principle of simplicity. This aspect is especially relevant in systems to be operated long time periods.

− Eliminating one hazard may introduce another one [11]. It must be carefully analysed whether new hazards are introduced and determine that the risk levels related to the new hazards are lower than the risk of the original hazard.

− Some of the system's quality attributes such as usability, weight, spatial requirements, or productivity may be degraded.

− Typically hazard elimination introduces larger design, construction, or assembly cost [1].

3.6 Example

Flammable and potentially toxic solvent is used as a carrier in a paint product. The flammability and potential toxicity introduce fire and health related hazards. The risks related to the hazards are identified intolerably high and therefore need to be mitigated. The solvent is eliminated by substituting it with water based solution (see SUBSTITUTE HAZARD). The elimination of the solvent eliminates the hazards related to flammability and intoxication hazards introduced by the solvent. This applies throughout the product supply chain including the end user. [5].

3.7 Known use

Elimination of hazard is given as the primary means to hazard control for example in the following sources [4], [16], [7, p 672], [10], [11], [15, p 197-200], [9] and [17]. The CCPS [1 p. 79] mentions reorganization and elimination of the hazard in the design phase as the ideal way to mitigate hazard potential (in context of ignition sources).

3.8 Related patterns

The SUBSTITUTE HAZARD strategy describes a potential approach for hazard elimination. If hazard cannot be eliminated, then one should consider PASSIVE PROTECTIVE MEASURE approach. In this approach the original hazard or hazardous element may remain in the system, but the risk is mitigated.

4. Substitute hazard

4.1 Context

The system under consideration introduces a hazardous element, which introduces a risk that requires mitigation in order to be tolerable. The ELIMINATE HAZARD strategy is being considered as a management approach. The hazard is introduced by a system property, functionality, element, or substance that can be changed without intolerable degradation in required performance and/or operation of the system.

4.2 Problem

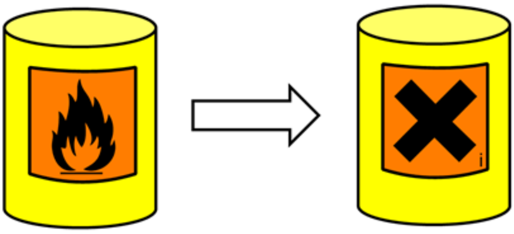
A hazard is wanted to be eliminated from the system.

4.3 Forces

- Hazardous element cannot be completely removed from the system as such due to, for example, cost, performance, usability or maintenance reasons. However, the risk must be mitigated.
- There are considerable alternatives for the source of the hazard available, which do not hinder other quality attributes or operation of the system.
- The most effective substance or system property from the operation efficiency point of view is also often one of the most dangerous alternatives. For instance, chemical reactions tend to be more effective with higher concentration, pressure and temperature as well as working machines' power tools tends to be more effective with higher voltage, speed and acceleration properties.
- Dangerous substances or system properties may be a reason for hazards introduced by the system.

4.4 Solution

Substitute the hazardous element with a non-hazardous or at least less hazardous element. In this case the hazardous element refers to a substance such as strong acid or a system property such as high voltage or pressure capable to introduce a hazard. For example, substitute an extremely flammable substance with a less flammable one, such as ethanal with ethanol, or high voltage with less high voltage, such as of 100 V with 24 V.



Depending on case, substitution of the hazard may eliminate the considered hazard, if the substitute is completely non-hazardous, the substitute may be better in terms of multiple hazards, or the substitute decreases the severity of consequences. However, other hazardous properties related to the original hazard source may still remain. For example, substituting strong acid with weaker or less concentrated one will decrease the severity of consequences for direct skin contact (in case of similar amount and time of exposure), but, for instance, the hazards of potential corrosive damage and leaking of the substance still exist. Similarly, substituting toxic liquid with non-toxic one does not eliminate leakage hazard of the liquid.

The substitution approach can be most typically applied in plant context to a mitigate hazard related to toxic substances. The typical approach is to select a less toxic substance or a less hazardous form of the substance, for example a solid form instead of a powdery form of utilized substance if the substance is hazardous to the respiratory organs. In context of machinery applications substitution can be typically considered in system properties such as voltages and pressures used in the system or the components of the system. For instance, lowering applied hydraulic pressure or using more robust components may mitigate risks related to these substituted system properties and components.

4.5 Consequences

+ The risk related to the considered hazardous element is decreased by either reducing the severity and/or the likelihood of exposure.

+ In best case the hazard is eliminated by substitution, but some related hazards may still remain.

− Typically some of the system quality attributes such as usability, performance, or productivity etc. may be degraded as the substitute may be less effective from the system or process point of view.

− Hazards common to both original and substitute still remain. For instance, if liquid is substituted with another liquid it can still leak or one can drown in it (if stored in open containers).

4.6 Example

Consider a mobile machinery application requiring a battery to operate. Initially, a flooded lead-acid battery was chosen for the application. However, this battery type produces hydrogen when charged. As it was noted that the released hydrogen may induce an explosion hazard, the battery type was substituted with a valve-regulated lead-acid battery that is prone to produce less hydrogen than the flooded lead-acid type.

Another typical application of substitution is related to the state (solid, liquid, or gas) and particle size (solid, pellet, granule, powder etc.) of the substance. Often reactiveness of a substance is dependent on these properties, gas and powder typically being more reactive than solid etc. Thus, substituting the substance state or particle size with more favourable one is a way to eliminate or decrease the risk related to a hazard. This approach is particularly applicable in context of process systems.

4.7 Known use

Substitution of hazard is given as a means to hazard control for example in the following sources [4], [16], [10], [11], and [15]. Substitution is also one of the ways to achieve inherently safer design [1].

4.8 Related patterns

The SUBSTITUTE HAZARD strategy can be used to implement the ELIMINATE HAZARD strategy.

5. Isolate hazard

5.1 Context

The system under consideration introduces a hazard, which is, according to risk assessment, intolerable. This is partly due to likelihood of occurrence due to exposure to the hazard. The ELIMINATE HAZARD approach has been considered, but so far no way to remove the hazardous element or elements has been identified. Therefore, the PASSIVE PROTECTIVE MEASURE is being considered to reduce the risk. The hazardous element is or originates, for example, from:

- inherently hazardous substances, such as toxic materials, flammable liquids, or oxidizers

- high energies or forces introduced by the system, such as moving machine parts, high-intensity laser beams, high-velocity cutting chips, sparkles, or vibrating elements
- system properties, such as high voltages or pressures, or extreme temperatures.

5.2 Problem

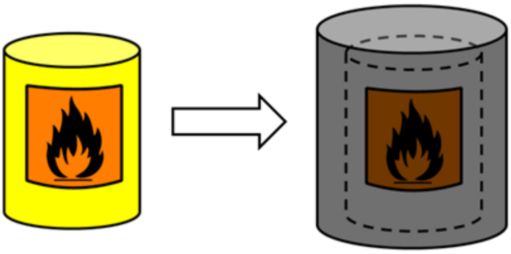
A hazardous element needs to remain in the system and with the current exposure profile the related risk is intolerable.

5.3 Forces

- Reducing the exposure to the hazard reduces the risks even though the severity of the hazard remains the same.
- Reduction of severity of the harm is desirable but not the first order priority.
- Passive protective measures tend to require less ongoing maintenance effort than their active counterparts [1].

5.4 Solution

Isolate the hazard by physically isolating the hazardous element from the environment and people. Isolation needs to ensure that the hazardous substance or process part cannot access environment and the hazard cannot be accessed from environment at least in normal operation conditions. Isolation can be, for example, an enclosure on the hazardous process part or fence or similar barrier to prevent access to the hazardous zone. Hazard isolation does not remove the original hazard from the system, but it reduces the risk by decreasing the exposure to the hazard (the hazard cannot be accessed) or the severity of the consequences (e.g. damage to finger instead of whole arm).



Isolation is an effective risk mitigation method against a hazard occurring during normal system operation. However, typically as special situation occurs, the isolation approach provides limited or no protection. Such situations may occur, for example, during maintenance, solving blockage, or inserting or removing a machineable or a tool. In such cases a person may need to access the hazard source. Nevertheless, hazard isolation is effective as it decreases the number of people having access to the hazardous element as well as limiting the exposure and potentially the severity of the harm caused by the hazard.

The isolation requires periodic maintenance and inspections. The isolating element may, for instance, wear out, break on impacts, be removed intentionally or unintentionally, installed incorrectly after intentional removal, lose effect in case of environment change, etc. Some of the cases can be managed applying an ACTIVE PROTECTIVE MEASURE, such as an E/E/PE SAFETY SYSTEM, to monitor the state of the isolation and alert operators or trigger a safety function if the isolation is lost. The latter case can be addressed by designing the isolation so that it does not depend on environment of the system. For example, isolation should not assume positioning of the system near wall so that the wall acts as an isolating element.

5.5 Consequences

+ Hazard is isolated in normal operating conditions. That is, the hazard cannot be or is very unlikely accessed in normal operation conditions, which reduces the exposure to the hazard and thus also the risk related to the hazard.

+ No need to alter the hazardous element itself, which might degrade the quality attributes of the system in case of elimination or substitution of the hazard.

− The hazard remains in the system.

− No or limited protection under special conditions such as maintenance, repair of hazardous element, or isolation breakage is provided.

− Some quality attributes may still degrade as the isolation is established. For instance, the accessibility to the other components within the isolated space is decreased.

− System lifecycle cost increases as the isolation needs to be designed, manufactured, and installed to the system. In addition the isolation needs to be inspected and maintained throughout the system lifecycle.

5.6 Example

A machine power transmission includes a belt drive. The belt requires regular maintenance as it needs to be changed when a certain number of operations hours are exceeded. To make belt change easy the belt is located outside the machine body and thus easily accessible by humans and foreign objects[1]. When the machine is operational, the belt drive is hazardous as human body part may crush between the belt and the wheels or foreign object may jam the power transmission causing havoc in the machine.

The hazard can be isolated by establishing an isolating enclosure over the belt drive. The solution degrades maintainability as the enclosure must be securely fixed and is not thus easily removable. Actually, the risk could be potentially more effectively mitigated by changing the power transmission into something that (is supposed to) last the machine life-cycle such as shaft drive and locate it completely inside the machine body.

5.7 Known use

In electronic products, a double-shield enclosure isolates high voltage parts from the environment and thus prevents the electric shock hazard in normal conditions. However, during maintenance or repair the high voltage parts are exposed. This also applies if the isolation is broken from the enclosure.

In machineries, fans are isolated from users by establishing a grill, mesh, solid cover, or barrier around or in front of the fan blades. The cover prevents user from accessing the fan (that is, decreases the exposure to the hazard) which could result in, e.g., an injury.

[1] Such belt drives are nowadays rare, but many old fashioned systems use such power transmission.

In case a finger can reach the blades through the isolating element, the isolation still acts to decrease the severity of the harm. That is, the harms are restricted to damage to finger instead of arm or another larger body part.

In laboratories, in which hazardous bacteria, viruses, etc. are studied, isolation is established to protect people. The bacterium etc. is handled in an enclosed chamber of which ventilation is also isolated from the ventilation of the building and arranged so that the air cannot (or should not) flow from the chamber to the room.

Isolation of a hazard is given as a means to hazard control for example in the following sources [4], [16], [7, p 672], [10], [11], [15 p. 194], [9] and [17].

5.8 Related patterns

The E/E/PE SAFETY SYSTEM strategy describes how the risks of hazards are mitigated with electric, electronic and programmable electronic systems. Also, isolation can be combined with E/E/PE SAFETY SYSTEM to produce more versatile protective measure such as interlocked barrier.

6. E/E/PE safety system

6.1 Context

The system under consideration introduces a hazard, which is, according to risk assessment, intolerable. To mitigate the risk, a protective measure is being designed as the hazard could not be eliminated through design (see ELIMINATE HAZARD). There are changes and events occurring in environment and/or the system, which affect the desired operation of the protective measure. That is, the protective measure needs to take the changes and the events into account to successfully reduce the risk. Therefore, the safety system needs to be able to sense the changes to operate accordingly and ACTIVE PROTECTIVE MEASURE strategy is applied.

6.2 Problem

Active protective measure of relatively complex functionality needs to be implemented.

6.3 Forces

- The safety function implements relatively complex logic or functionality. For instance, to implement safety function successfully the safety system needs to obtain information from multiple sources, do reasoning considering the data and control the system in relatively complex and timely precise way to ensure successful operation.

- The implementation of the safety function requires or benefits from cooperation of distributed elements. That is, the elements may be located in different locations considering the system.

- From maintenance and safety point of view it is beneficial if the protective measure can monitor (diagnose) its state and even better if it can communicate potential problems to other systems or people.

- An option for wireless data communication is wanted to be reserved or required for the current system.

6.4 Solution

Use an electric, electronic or programmable electronic (E/E/PE) safety system to implement the protective measure functionality. An E/E/PE system is inherently an ACTIVE PROTECTIVE MEASURE. E/E/PE system observers the system under control for hazardous states and applies active actions to achieve the safety of the users, environment, and machinery itself. An E/E/PE safety system has considerable more possibilities to retain safety as it can affect the operation of the system.

Figure 4 illustrates the typical E/E/PE safety system elements and information flow direction between them. The system consists of sensor element(s), logic element(s), and actuation element(s). There may be multiple instances of each element depending on the architecture of the safety system. In some cases, some elements may be left out or they can physically exist embedded in other elements. For example, the logic element can be embedded into a sensor element in some cases or it can exist in cross connection between sensor elements and actuation elements. The information flow can be arranged applying most appropriate approach. Typical alternatives are point-to-point wires and communication busses (e.g. CAN, FlexRay, ProfiNet, etc.). Wireless communication can be established, if implemented (and applicable) according to adhered law, standard, or guideline. Optical cables can be used in electromagnetically hostile environments.

The sensor elements measure and observe the system under control (typically a process variable) and its environment. The sensor element does not necessarily have to be a dedicated sensor device. Instead, it can be any information source although the integrity of the information need to be sufficient to comply with rest of the safety function implementation and required integrity (compare to IEC 61508 [6]). The information is transmitted to the E/E/PE logic, which may be for example a relay, an electronic circuit, or a microcontroller. The logic element produces control signal for actuation elements that can physically affect the system in order to execute the protective measure.

Although simplicity is desired property in safety-related systems, sometimes more advanced functionality is required. Using E/E/PE logic it is relatively easy, especially in context of programmable devices, to implement complex functionality to the safety function. Information from multiple sensors or information sources can be used in decision making applying advanced algorithms when necessary. This characteristic feature of E/E/PE systems also supports application of diagnostic functionality to monitor the operation and state of the safety functionality and elements of the system.

With E/E/PE system many of the shortcomings of passive protective measures can be circumvented while implementing required safety function to reduce risk into a tolerable level. One of the problems with passive protective measures is that they cannot react in any changes or affect the operation of the system. Using E/E/PE safety system one is able to measure system and its environment. Consequently, the safety system can react on changes in and the state of the system and its environment. However, the development of E/E/PE safety system is considerable more burdensome than mechanical protective measures or guards. Mechanical protective measures and E/E/PE

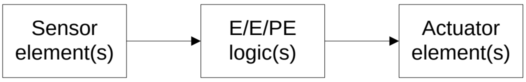


Figure 4. Typical E/E/PE safety system elements and information flow direction.

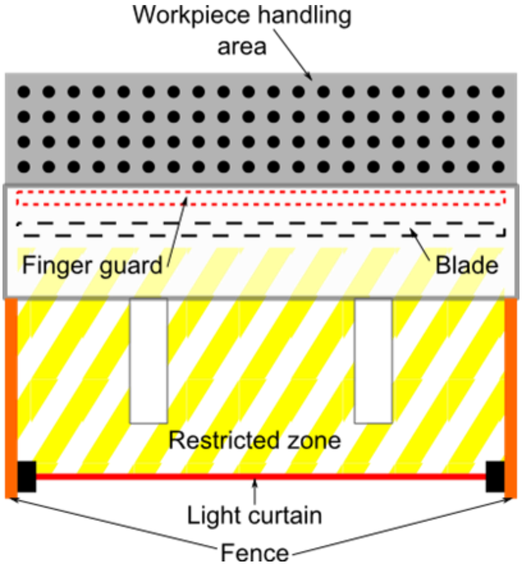


Figure 5. Illustration of steel cutting machine restricted area shutdown (depicted from above the machine).

safety systems can (and usually have to) be used in combination to obtain optimal performance.

6.5 Consequences

+ Increased amount of possible safety function becomes available, as the protective measure can be aware of changes and event in the system and its environment and the system under control can be affected by the safety system.

+ The safety system becomes more flexible due to increase expandability and modifiability.

+ Possible performance (and other secondary quality attribute) gains in system as protective measures can be optimized. For example, replacing passive finger protection barrier of a fan with light curtain and a stop function can improve air flow.

− The complexity of the safety system increases as all the elements, including logic in form of electronics or program code, need to be implemented in the safety system.

− Development process of an E/E/PE safety system is considerable more burdensome than its passive counterpart if applicable laws and regulations are followed (e.g. the IEC 61508 requires vast amount of techniques and measures to be followed to develop software for an E/E/PE safety system)

− E/E/PE safety systems are typically more expensive than passive ones (or inherently safer design concept including eliminating or substituting the hazard) in terms of maintenance effort. E/E/PE systems require periodic and potentially reactive checking and maintenance especially in terms of potential sensor and actuator elements.

− Spurious trips caused by E/E/PE system may decrease productivity of the system.

6.6 Known use

Airbags deployed in most new cars are a well-known E/E/PE safety system. The E/E/PE system consists of various sensor elements that sense collision situations, an airbag control unit / electronic control unit (ACU/ECU), which observers the sensors, and the airbag unit(s) that actuates the airbag(s). The ACU operates the airbag units to inflate the airbags in case of collision. [18] [3]. Airbags are E/E/PE safety systems that are used to improve the safety of passengers. They complement the passive protective measures of a car such as the flexible collision regions of the car body and safely shaped interior decrease the risk of injury and death in car accidents.

A steel cutting machine utilizes an E/E/PE safety system to prevent cutting operation in case a person (or an object) enters the back side of the machine. The machine is depicted on Figure 5. The machine has open backside so the blade is directly accessible from the backside of the machine. The restricted zone illustrates the area in which persons are not allowed to reside during machine operation. The front side (work piece handling area) is protected by partly fixed and partly opening barrier, which interlocks the blade movement if opened. The sides of the machine's backside are fenced. Optic sensors (light curtain) are used to notice the objects entering the working area of the machine and stopping the cutting blade in such a case. Whenever the light curtain is broken the machine is stopped and continuation of the operation requires operator to acknowledge that the restricted area is cleared.

An E/E/PE safety system is given as a means to hazard control for example in the following sources [4], [16], [10], [11], [9] and [17]. The CCPS [1, p. 125] mentions E/E/PE safety systems in context of the active safeguarding strategies.

6.7 Related patterns

The ISOLATE HAZARD strategy describes how risk can be mitigated without an E/E/PE system. In addition E/E/PE protective measures can be combined with isolation methods by adding E/E/PE safety system to monitor and ensure the isolation. For example, a limit switch indicates if isolation is removed from hazard and the safety system drives the system into a safe state, e.g., stops moving parts under the isolating barrier.

7. Acknowledgement

The authors would like to greatly thank the workshop group members in the EuroPLoP 2014 event and especially our shepherd Jason Yip for their valuable feedback and input to improve the patterns and the paper.

8. References

[1] Center for Chemical Process Safety. 2012. Guidelines for Engineering Design for Process Safety. 2nd Edition. Wiley New York, NY, USA. 9781118266670. p. 438.

[2] EN ISO 12100 2010. Safety of machinery - General principles for design - Risk assessment and risk reduction (ISO 12100:2010)

[3] Geitner, H. and Ferraresi, M. 2009. Airbag Electronics: from Single Building Blocks to Integrated Solutions. In Advanced Microsystems for Automotive Applications 2009, eds. Meyer, G., Valldorf, J., Gessner, W., Springer Berlin Heidelberg, 978-3-642-00744-6, 10.1007/978-3-642-00745-3_28, pp. 423-434.

[4] Health and Safety Authority. Hazard and Risk. http://www.hsa.ie/eng/Topics/Hazards/. Retrieved [7.2.2014].

[5] Hendershot, D. C. 2011. Inherently Safer Design - An Overview of Key Elements. ProfessionalSafety. February 2011. Available: http://www.asse.org/professionalsafety/pastissues/056/02/048_055_f2hendershot_0211z.pdf. Retrieved 23.5.2014. p. 8.

[6] IEC 61508 2010. Functional safety of electrical/electronic/programmable electronic safety-related systems, International Electrotechnical Commission, 2010.

[7] Lehto, M., Lesch, M., Horrey, W. 2009. Safety Warnings for Automation. In Safety Warnings for Automation, eds. Nof, S. Y., Springer Berlin Heidelberg. 978-3-540-78830-0. http://dx.doi.org/10.1007/978-3-540-78831-7_39. pp 671-695.

[8] MIL-HDBK-338B 1998. Military Handbook - Electronic Reliability Design Handbook. p. 1046.

[9] Miller, M. and Duta, M. 2010. SafeDesign: Safeguarding Techniques. Rockwell Automation. http://discover.rockwellautomation.com/Files/SafeDesignSafeguardingTechniquesOverviewfinal.pdf [Retrieved: 10.2.2014].

[10] New York Committee for Occupational safety & Health. Hierarchy of Hazard Controls. http://nycosh.org/index.php?page=Hierarchy-of-Hazard-Controls Retrieved 7.2.2014.

[11] Nix, D. 2011. Understanding the Hierarchy of Controls. Machinery Safety 101. http://machinerysafety101.com/2011/02/28/understanding-the-hierarchy-of-controls/. Updated February 28 2011. Retrieved 7.2.2014.

[12] Rauhamäki, J. and Kuikka, S. 2013. Patterns for control system safety. In Proceedings of the EuroPLoP 2013 Conference.

[13] Rauhamäki, J., Vepsäläinen T. and Kuikka S. 2012, Functional safety system patterns. In Proceedings of the VikingPloP 2012 Conference. Tampere University of Technology, Department of Software Systems, Tampere, 48--68. http://URN.fi/URN:ISBN:978-952-15-2944-3

[14] Rauhamäki, J., Vepsäläinen, T. and Kuikka, S. 2013. Patterns for safety and control system cooperation. In Proceedings of VikingPLoP 2013 Conference, Tampere University of Technology. Department of Pervasive Computing. Tampere, 96--108. http://URN.fi/URN:ISBN:978-952-15-3167-5

[15] Roughton, J. E. and Mercurio, J. J. 2002. Developing a Hazard Prevention and Control System, In Developing an Effective Safety Culture, edited by James E. Roughton and James J. Mercurio, Butterworth-Heinemann, Woburn, 2002, Pages 190-226, ISBN 9780750674119, http://dx.doi.org/10.1016/B978-075067411-9/50015-3. (http://www.sciencedirect.com/science/article/pii/B9780750674119500153)

[16] SafeWork SA 2007. Hierarchy of control measures. http://www.safework.sa.gov.au/contentPages/EducationAndTraining/HazardManagement/Machinery/TheAnswers/machAnswerHierarchy.htm. Last updated 7-11-2007. Retrieved 7.2.2014.

[17] Titus, J. B. 2011. Machine guarding and the hierarchy of measures for hazard mitigation. Control Engineering. October 25, 2011. http://www.controleng.com/blogs/machine-safety/blog/machine-guarding-and-the-hierarchy-of-measures-for-hazard-mitigation/03d5431c9a66505d5fa6f3d2ea2b65d1.html [Retrieved 10.2.2014].

[18] Volkswagen 2014. Airbag Control Unit. http://en.volkswagen.com/en/innovation-and-technology/technical-glossary/airbag-steuergeraet.html. [Retrieved 14.2.2014].

Publication V

Rauhamäki, J., Kuikka, S. (2015). Patterns to Implement Active Protective Measures. In *Proceedings of the 20th European Conference on Pattern Languages of Programs.* [43] (ACM International Conference Proceeding Series). New York: ACM.

DOI: 10.1145/2855321.2855365

Patterns to Implement Active Protective Measures

JARI RAUHAMÄKI, Tampere University of Technology
SEPPO KUIKKA, Tampere University of Technology

There are various ways to protect people, environment and other systems from harm caused by machines and system. In this paper, patterns on implementing protective measures applying an active approach are given. The purpose of a protective measure is to lower the risk related to a hazard by either reducing the likelihood (the frequency of exposure of) or the consequences of a realization of harm. The protective measures can, for example, protect user from harms introduced by hazards in the system or operations of the system operator.

Categories and Subject Descriptors: **K.4.1 COMPUTERS AND SOCIETY]**: Public Policy Issues—*Human safety*; **K.6.1 [MANAGEMENT OF COMPUTING AND INFORMATION SYSTEMS]**: Project and People Management—*Systems development*

General Terms: Design

Additional Key Words and Phrases: guard, interlock, lock, hazard, zone

ACM Reference Format:

Rauhamäki, J. and Kuikka, S. 2015. Patterns to Implement Active Protective Measures. In proceedings of 20th European Conference on Pattern Languages of Programs, EuroPLoP 2015. 13 pages.

1. INTRODUCTION

Hazard elimination is a desirable approach in risk reduction as it completely prevents a hazard from causing harm (as the hazard does not exist).In some cases, this applies in many or all phases of system lifecycle from manufacturing to disposal. For instance, if asbestos is eliminated from the system in the design phase, all the hazards and problems caused by it, are eliminated from the system lifecycle. However, in many cases hazards cannot completely be eliminated. In such cases, other protective measures are needed to mitigate the remaining risk related to the hazard.

In case a hazard cannot be eliminated, a potential approach is to apply the ISOLATE HAZARD APPROACH, a variant of a PASSIVE PROTECTIVE MEASURE (Rauhamäki & Kuikka 2014). However, in certain cases simple isolation is not a viable option. For instance, if a recurring access is required to the hazard zone to operate the system, a fixed isolation would complicate the access. In such cases, an ACTIVE PROTECTIVE MEASURE (Rauhamäki & Kuikka 2015) can be used to enable opening such as a door, a gate, or a hatch to be placed on the isolation and to use a control system to operate the system or the opening to retain safety. In such situations, one can consider the combination of the ISOLATE HAZARD and ACTIVE PROTECTIVE MEASURE approaches to enable the system under control to respond to the user actions in a safe manner or affect user possibilities mastered by a control system to a support safe operation of the system.

In this paper, we present patterns utilizing the ACTIVE PROTECTIVE MEASURES to promote safety. That is, the patterns describe protective measures that include a functional part. The functional part affects the operation of the system to retain the safety of people.

The target audience of the patterns and the pattern language (see Section 2.1) described on the paper contains people involved in safety system development such as system architects, safety engineers, hardware and software developers and designers. Primarily, the pattern language aims to serve people

Authors' address: J. Rauhamäki | Seppo Kuikka, P.O. Box 692, FI-33101 Tampere, Finland; emails: jari.rauhamaki@tut.fi seppo.kuikka@tut.fi

EuroPLoP '15, July 08 - 12, 2015, Kaufbeuren, Germany

ACM 978-1-4503-3847-9/15/07…$15.00
DOI: http://dx.doi.org/10.1145/2855321.2855365

with low experience and expertise on safety system development. Secondarily, the language can support more experienced people with decision making and provide a guideline for the design process.

2. BACKGROUND

2.1 Pattern language for safety system development

The patterns presented in the paper are considered as a part of larger pattern collection considering safety system development. A section of the language is illustrated on Figure 1 and it illustrates how the patterns relate to previous work. The patterns presented on this paper have solid bold outline. The root pattern on Figure 1 suggests applying a risk based approach on safety. That is, to select the risk mitigation methods according to the risk (magnitude and significance). From there onwards, the arrows suggest a potential path across the pattern language indication the potential application order of the patterns and strategies. Thumbnails for the patterns described and referred to the paper are given in Table 1.

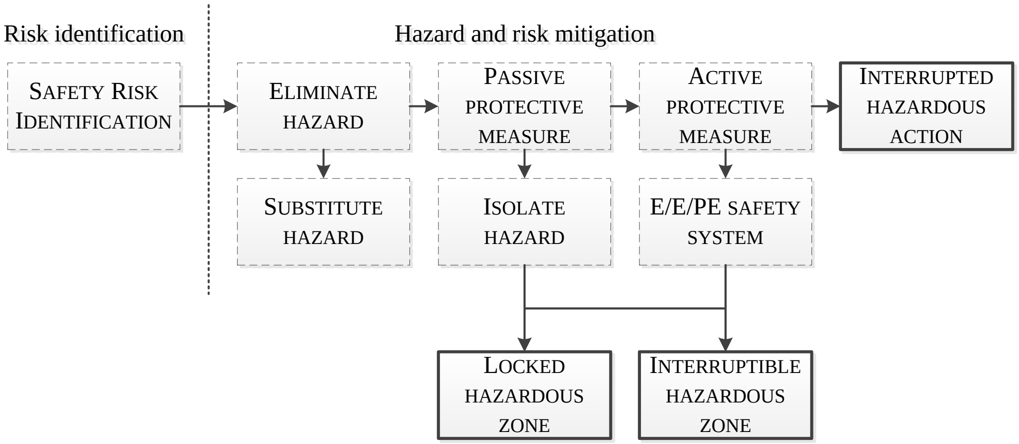

Risk identification
Hazard and risk mitigation
SAFETY RISK IDENTIFICATION
ELIMINATE HAZARD
PASSIVE PROTECTIVE MEASURE
ACTIVE PROTECTIVE MEASURE
INTERRUPTED HAZARDOUS ACTION
SUBSTITUTE HAZARD
ISOLATE HAZARD
E/E/PE SAFETY SYSTEM
LOCKED HAZARDOUS ZONE
INTERRUPTIBLE HAZARDOUS ZONE

Fig. 1. Section of pattern language for safety system development

Table 1: Patlets (aka. short descriptions) for the patterns

Pattern	Patlet
SAFETY RISK IDENTIFICATION	To make conscious decisions considering hazard and risk management, information on these aspects needs to be available. Therefore, use structured and/or systematic method(s) in order to identify the hazards and associated risks introduced by the system. (Rauhamäki & Kuikka 2014)
ELIMINATE HAZARD	You want to maximize the likelihood that a hazard introduced by a system cannot cause harm in any part of the system lifecycle. Therefore, eliminate the hazard completely by removing the component introducing the hazard from the system. (Rauhamäki & Kuikka 2014)
SUBSTITUTE HAZARD	A hazard needs to be eliminated from the system. Therefore, substitute the hazardous element with a non-hazardous or at least less hazardous element. (Rauhamäki & Kuikka 2014)
PASSIVE PROTECTIVE MEASURE	A hazard or a hazardous element remains in the system, but the related risk needs to be mitigated. Therefore, use a passive protective measure to mitigate the risk by non-functional design solutions, equipment or system design features that mitigate risk related to hazard. (Rauhamäki & Kuikka 2015)
ACTIVE PROTECTIVE MEASURE	A hazard or a hazardous element remains in the system and the related risk needs to be mitigated. Therefore, use an active protective measure to mitigate the risk by affecting the system operation through a defined functionality so that risk is reduced. (Rauhamäki & Kuikka 2015)
ISOLATE HAZARD	A hazardous element needs to remain in the system and with the current exposure profile, the related risk is intolerable. Therefore, isolate the hazard by physically isolating the hazardous element from the environment and people. (Rauhamäki & Kuikka 2014)
E/E/PE SAFETY SYSTEM	An active protective measure of relatively complex functionality needs to be implemented. Therefore, use an electric, electronic or programmable electronic (E/E/PE) safety system to implement the protective measure functionality. (Rauhamäki & Kuikka 2014)
INTERRUPTED HAZARDOUS ACTION	A user operated system element may enter hazardous operating range due to an operator initiated actions without the operator noticing this. Therefore, Interrupt the hazardous system operation before the hazardous operation range is reached and force the operator to acknowledge this.
INTERRUPTIBLE HAZARDOUS ZONE	The hazardous zone or element needs to be easily accessible but hazardous conditions within the zone may exist in defined situations. Therefore, Implement an interlocking guard over the hazardous element or zone.
LOCKED HAZARDOUS ZONE	A hazardous zone or element needs to be easily accessible but hazardous conditions within the zone may exist in defined situations. Therefore, Implement a locking guard to cover the hazardous element or zone.

2.2 Terminology

Table 2 provides definitions for some of the terms used on the paper. Especially, the patterns refer to various types of guards used as protective measures and these are defined according to (EN ISO 12100:2010).

Table 2: Definition for some of the terms used on the paper.

Term	Definition as given in (EN ISO 12100:2010)
Guard	"Physical barrier, designed as part of the machine to provide protection" and "Depending on its construction, a guard may be described as, for example, casing, shield, cover, screen, door, enclosing guard."
Fixed guard	"Guard affixed in such a manner (for example, by screws, nuts, welding) that it can only be opened or removed by the use of tools or by destruction of the affixing means"
Interlocking guard	"Guard associated with an interlocking device so that, together with the control system of the machine, the following functions are performed: – the hazardous machine functions "covered" by the guard cannot operate until the guard is closed, – *if the guard is opened while hazardous machine functions are operating, a stop command is given, and* – when the guard is closed, the hazardous machine functions "covered" by the guard can operate (the closure of the guard does not by itself start the hazardous machine functions)"
Interlocking guard with guard locking / locking guard	"Guard associated with an interlocking device and a guard locking device so that, together with the control system of the machine, the following functions are performed: – the hazardous machine functions "covered" by the guard cannot operate until the guard is closed and locked, – *the guard remains closed and locked until the risk due to the hazardous machine functions "covered" by the guard has disappeared, and* – when the guard is closed and locked, the hazardous machine functions "covered" by the guard can operate (the closure and locking of the guard do not by themselves start the hazardous machine functions)"

3. INTERRUPTED HAZARDOUS ACTION

3.1 Intent

The system interrupts actions leading to hazardous operating range and the user needs to take a deliberate action to take the system into a hazardous operating range such as releasing a water tap lock for the hot water side.

3.2 Context

A human operator operates a system, system part, functionality or parameter directly through a control interface. For example, the operator drives a mobile machine or a boom attached to it, operates the movement of a hydraulic press, or adjusts the number of revolutions per minute of a turning lathe. The operation is actuated through a control interface provided, such as a joystick, switch, button, keyboard, or slide. The operated entity introduces a hazard, such as a shear, impact, stability, or noise hazard. The hazard is present only in a defined operating range or states where the operator can control the system. The harm may occur for a person, the operated machine itself or surrounding machines, structures etc.

3.3 Problem

A user operated system element may enter hazardous operating range due to an operator initiated actions without the operator noticing this.

3.4 Forces

- The hazardous element cannot be eliminated from the system due to operational reasons. For instance, a shearing machine needs to be able to shear regardless the fact that it may also shear fingers and hot water needs to be available in household tap regardless the fact that hot water can cause a burn.
- In many cases, a human controller could be able to notice whether or not it is safe to proceed to the hazardous operating region. However, human senses are limited so that the safe operating region might be hard to detect reliably, such as detecting safe audio volume levels by ear. An operator may get bored, distracted, or make mistakes when the time to check the condition to proceed to the hazardous region should be made. For instance, when executing repetitive work one loses focus on the task and fails to check where the co-worker's fingers are located.
- The risk could be potentially mitigated by adjusting the overall operation of the system so that the hazardous conditions are avoided. However, this would potentially hinder the system capabilities, performance etc. For example, the maximum temperature of circulating hot water could be lowered, but this would also introduce a strict limit for the maximum temperature for the water one could take from the tap.
- The operated system could provide the operator with a non-interrupting notification of the hazard or hazardous situation. However, this does not require actions or conscious operation from the operator. The notification may go unnoticed due to many reasons such as boredom, distraction, or notification element malfunction.
- Using a control system to observe the hazardous element and related properties of the system, the hazardous action could be prevented by the means of the control system. However, in many situations it might be relatively hard to detect if the operation, that is, driving the considered element into hazardous operating range, could cause harm. For instance, detecting if a person is under a descending work platform in open environment or what is the maximum angular acceleration rating of a disc of an angular grinder is not easily observable by a control system.

3.5 Solution

Interrupt the hazardous system operation before the hazardous operating region is reached and force the operator to acknowledge this. That is, the operator (or any other instance) should not be able to drive or control the system into the hazardous operating region accidentally and unnoticed. In practice, implement an extra functionality for the operator to drive the system into a potentially dangerous operation region. The functionality interrupts the potentially hazardous actions, movements, events, processes, etc. initiated or controlled by the operator, until the operator has, in some way, confirmed that she is conscious about the operation and wants to continue the hazardous action.

Figure 2 illustrates state behaviour of an interrupted hazardous action principle from the interrupting function point of view. Initially, the system is typically in the safe operation region. In this region, user can freely control and drive the system. When the interrupt functionality observers the system is driven on the edge of hazardous operation region, the functionality interrupts the system operation and takes the Waiting for user acknowledgement state. Typically, in such case, the proceeding towards the hazardous operation region is stopped, regardless the state of the control devices. To proceed to the hazardous operation region, the user needs to confirm the operation. Often times this is achieved by re-affecting the controls. For example, a joystick or a footswitch needs to be released to the neutral position and end re-engaged to continue the hazardous operation.

Returning from the hazardous operation region can be typically done without an interruption. However, the decision is case specific and one needs to consider the hazard and risk assessment to make the final conclusion regarding the need for interruption when returning to the safe operation region.

In the best case, the interrupt forces the operator to check if it is safe to proceed the action. This is especially important if the system is operated by multiple people at the same time, but only one person actually provides the operating commands or for the system. For instance, some shearing machines are operated by two people. Both people lift and handle the material to be sheared but only one is operating the machine. If this cannot be achieved, it is still better to catch the operator's attention and force her make the decision consciously.

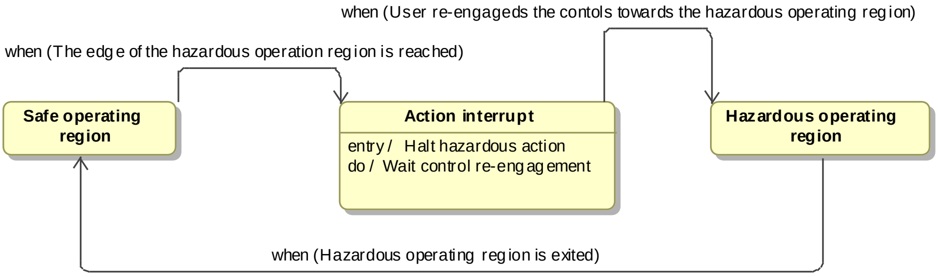

when (User re-engageds the contols towards the hazardous operating region)
when (The edge of the hazardous operation region is reached)
Safe operating region
Action interrupt
entry / Halt hazardous action
do / Wait control re-engagement
Hazardous operating region
when (Hazardous operating region is exited)

Fig. 2. State machine behaviour of interruptible hazardous action without interrupt on return from the hazardous operating region

3.6 Consequences

+ Operator is aware that the system has been driven to a hazardous range as the system requires the operator to perform a defined action before letting the hazardous action to continue.

+ It is harder to drive the system into the hazardous region accidentally, e.g. by an unconscious operator or an unintentional effect on system controls. The effectiveness of this depends on the implementation of the interrupt and the way it is acknowledged. For instance, if a joystick type control needs to be released in the centre position before the action can be continued, it is relatively unlikely that a fainted person or a fallen object could drive the system past the interrupt.

+ The possibility of a co-worker or an outsider to avoid the harms increases. It takes some time to perform the required action when an interrupt occurs. Depending on the case this may take some tenths of a second to a couple of seconds. Such time, even relatively short, provides the co-operator (and others) time to react on the hazardous situation before the harm occurs.

+ The interrupting functionality serves also as a form of a dead man's switch. The hazardous action cannot be achieved by jamming a control into a certain position and waiting the system to enter the hazardous operating region.

+ The operator may control the system freely within the non-hazardous area, range, and states.

+ The information regarding the acknowledgement of entering a hazardous zone is available and can be logged and used to e.g. monitor system usage and user characteristics.

– The interrupting function might get annoying if it interrupts workflow continuously (and in some cases it will), which might cause the function being bypassed. The potential for bypassing should be assessed in terms of risk. If the risk for bypassing is found intolerable through its likeliness or consequences, countermeasures need to be applied to prevent or complicate the successful implementation of a bypass[1].

– If the interrupting function is encountered continuously during the system operation, the effect of raising awareness of the operator may degrade and bypassing the interrupt becomes part of the work routine.

– The interruption functionality may add stress to the system. For example, in a case where the interrupt introduces a fast stop of moving parts of fluids, the pressure impacts in pneumatic and hydraulic systems and mechanical joints add stress on the corresponding parts.

– Not all the hazards or access to the hazardous zone are easily detectable. This may cause either unsuitability of the solution or expensive sensors to detect the situation of interest.

[1] The actual design and implementation of the countermeasures for bypassing are out of scope of the paper.

3.7 Known use

A bending machine operated by one or two people employs hazardous action interruption functionality considering the upper jaw control (see Figure 3). When the jaw is lowered above approximately one inch above the lower jaw, the upper jaw is stopped by the control or safety system. In this case, the stop of the upper jaw movement is the interruption of the hazardous action. To proceed to the hazardous operating region (jaws closed), the operator needs to release the jaw control switch and re-engage it. In this case the function (most likely) protects the operators' fingers, as the fingers need to be located in some cases very near the jaws of the machine and there is no guard or protective measure to mitigate the risk.

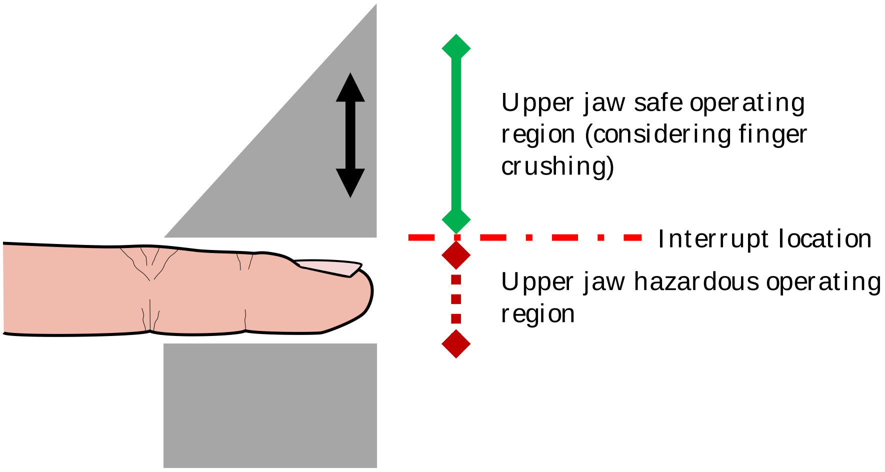

Upper jaw safe operating region (considering finger crushing)
Interrupt location
Upper jaw hazardous operating region

Fig. 3. Illustration of the upper jaw operating regions of in a bending machine application. (© Jari Rauhamäki)

An articulated jack passenger hoist stops at a defined height when it is lowered. The purpose of this is to provide the people below the platform the possibility to escape. Again, the operator needs to release the control switch to neutral and re-engage it to lower the platform into the final low state.

A boom type passage hoist stops as it approaches the range limits. After the stop the boom may still extend further, but typically its movement speed is lowered.

Samsung Galaxy S5 smartphone (and potentially other similar devices) notifies the user when sound volume is raised above a threshold level. The purpose is to protect and warn the user from high volume levels. User needs to release the volume increase button and press it again to raise the volume above the threshold level. The threshold is predefined and is possibly valid only for the original packed earphones.

In an operating system environment, a user with normal privileges is asked to elevate the privileges by either confirming a prompt or providing a password for elevated privileges account. The former case resembles the described solution whereas the latter alternative resembles a more enhanced/enforced version of the solution, as special information is required to enable the potentially hazardous operation. In the elevated privilege mode (potentially) hazardous operations, such as installing and uninstalling software or changing system configuration, can be typically done. For instance, Windows 7 (run as administrator), Linux, and Mac OS X (sudo) operating systems resemble the approach to elevating the privileges by requiring user action.

3.8 Related patterns

The pattern implements an ACTIVE PROTECTIVE MEASURE, that affects the system operation to prevent harm from occurring. In this case, the functionality is to stop the system before entering the hazardous operation region.

4. INTERRUPTIBLE HAZARDOUS ZONE

4.1 Intent

A system is taken into a safe state by means of control whenever user is able to access a hazard (source). For example, the microwave radiator is shut down whenever the door of the microwave oven is opened.

4.2 Context

There is a hazardous zone introduced a by the system under consideration, which requires frequent access by operators. This may be due to, for example, maintenance, operational or other reasons, such as lubrication, replacement worn parts, adjustment, changing tool bit or a workpiece or blockage removal need to be executed periodically within the hazardous zone.

The hazardous conditions within the hazardous zone can be removed relatively fast by means of controlling the system in relation to the time required to reach or get exposed to the hazard after accessing the hazardous area. That is, the control system is able to drive the system in such a state that the hazardous conditions are removed relatively fast. For instance, moving parts of a machine are stopped in order to prevent crushing, shearing, or impact hazards from occurring.

4.3 Problem

A hazardous zone or element need to be easily accessible but hazardous conditions within the zone may exist when the zone is accessed.

4.4 Forces

- The hazardous zone needs to be relatively easily accessible. Thus, a fixed guard is not an option as it instead of promoting, hinders the accessibility. Usage of a fixed guard to isolate location, which needs to be frequently accessed, leads more likely to complete the removal or modification of the guard, the machine, or the system to enable the access. This again conflicts with the original purpose of the guard.
- The system could control the access to the hazardous zone by locking it out whenever the hazardous conditions exist. However, people tend to prefer the situation that they are in charge of operation and free to interrupt the system when it suits them best.
- The more freely a user or an operator can access the hazardous zone (so that hazardous conditions are removed beforehand), potentially the more efficiently the user can carry out her tasks. Depending on the case, a system adapting to users work routine and phase can save a lot of time. The value of this time may overcome the additional investment into a machine build to adapt the user operation.

4.5 Solution

Implement an interlocking functionality to protect users from the hazardous element or zone. The interlocking functionality drives the system (or part of it) in a safe state, so that the hazardous conditions are eliminated within the zone. In practice, the interlocking functionality observes for access to the hazardous zone and controls the system in order to remove the hazardous conditions before harm occurs.

In many cases, the interlocking functionality is implemented through an interlocking guard. The guard is connected with a safety function to take the system or hazardous element under the guard into a safe state whenever the guard is opened or removed to access the hazardous zone (see Figure 4). This approach utilizes the ISOLATE HAZARD approach complemented by an opening guard to prevent access to the hazard when the guard is closed. When the guard is closed, it should prevent people, body parts and other non-wanted object for accessing the hazard zone. Whenever the guard is opened, the hazardous conditions are eliminated within the hazardous zone. One needs to ensure and/or take into account in guard design that the system needs certain amount of time to be taken into the safe state before one is able to access the hazard after opening the guard.

Further, add a mechanism to detect if the guard is closed or not, that is, if the guard prevents the access to the hazard or not. Connect the information on the guard state to control system or safety function. Whenever the guard is removed or opened, the safety function ensures that the hazardous conditions are removed or minimized. In practice, moving parts are stopped, and/or radiation, noise, etc. are removed or lowered to a tolerable level. In addition the system needs to retain this state until the guard is closed in place and valid start command is given (to prevent unexpected start-up).

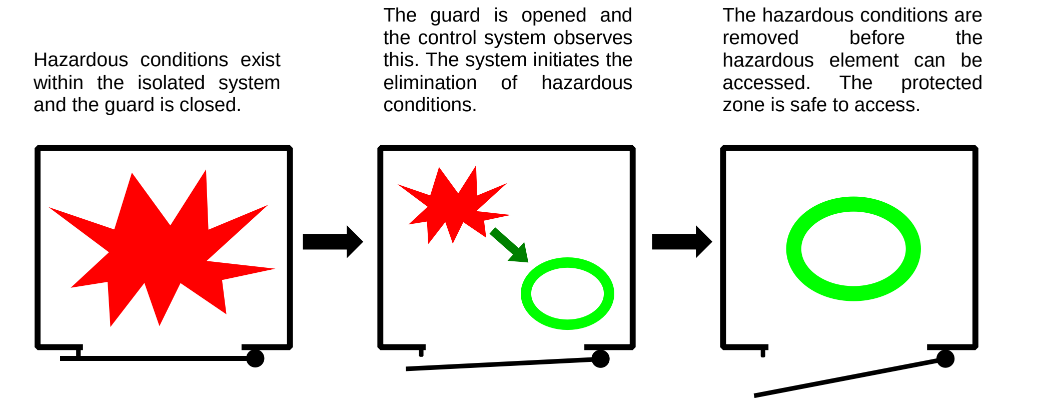

Hazardous conditions exist within the isolated system and the guard is closed.
The guard is opened and the control system observes this. The system initiates the elimination of hazardous conditions.
The hazardous conditions are removed before the hazardous element can be accessed. The protected zone is safe to access.

Fig. 4. Illustration of the operation of an interlocking guard operation (with ISOLATE HAZARD approach). Reproduced from (Kivistö-Rahnasto 2015)

In general, as mentioned previously, it is not necessary to isolate the hazardous zone mechanically. This, however, requires that there is no need or reason for the mechanical isolation of the hazardous zone or system part. In this case monitoring the hazardous zone should suffice. Monitoring can be established using non-contacting sensors such as light curtains. If potential for projecting shrapnel, load, sparkles, etc. exist, due to the hazard, mechanical isolation is required.

4.6 Consequences

+ The hazardous zone can be accessed free through path or access way dedicated to this. Whenever the hazardous zone is accessed the system eliminates the hazardous conditions within the hazardous zone before they can cause harm.

+ The system adapts to the operations and workflow of the user/operator. The machine obeys and follows the user and diminish the hazardous conditions automatically without requiring user actions other than accessing the hazardous zone (using a valid access or path way).

+ As the system adapts to user actions and protects her from harm, the user can potentially operate more optimally and save time and other resources.

+ No need for a fixed guard that might degrade quality attributes, such as the performance, capacity, or usability of the system.

– The system introducing the hazard needs to be designed to be interruptible and to remove the hazardous conditions before they can be reached. This may induce requirements on components, mechanics, maximum speeds etc.

– Typically the approach requires a greater distance between the isolating boundary and the hazard compared with the LOCKED HAZARDOUS ZONE approach. The distance needs to make sure the hazard cannot be accessed before hazardous conditions are eliminated. Thus, the shorter the distance, the shorter the time in which the hazardous conditions need to be eliminated and vice versa.

– The system needs to tolerate the stress related to removing the hazardous conditions, such as stopping moving parts or shutting down radiating components.

– If the stress on the system during the hazardous condition removal process needs to be decreased, the distance from the hazard to the isolating cover or monitoring line needs to be increased. For instance, if one second stopping equals one meter distance between the isolating guard and hazard, two seconds stopping time would require, for example, two and (a half meter) distance.

– An active protective measure needs to be developed. Minimally this requires a sensor to observe if the hazardous zone is, is to be, or can be accessed, logic to implement needed control activities to eliminate the hazardous conditions and actuating element to eliminate the hazardous conditions when needed. These fulfil the *observability*, *model* and *action* conditions as defined by (Leveson 2012) to enable the control of the system. Depending on available components and required functionality this may be burdensome and expensive. In some cases, a simple access hull operated relay switch might suffice, but in other cases software based functionality could be required.

- More complex logic as potentially multiple sensors needs to be observed.
- The users may develop a complete dependency on the protective functionality and lose their own judgement about the situation.

4.7 Example

Consider a rotating power transmission shaft, which needs frequent maintenance. Thus, the shaft should be easily accessible. The shaft is not completely located inside the machine body, where it would not cause harm, but it would also be hard to access frequently. The shaft is guarded with an opening mechanical guard. Whenever the guard is opened, the shaft is stopped before it can be reached.

4.8 Known use

The solution model is found in domestic microwave ovens and dish washers. In the former case, the microwave radiation is halted whenever the door of the microwave oven is opened to prevent user exposure to microwave radiation (see Figure 5). In the latter case the operation of the dish washer is stopped to, primarily, protect user from exposure to hot water, and secondly, to prevent building structures (floors, wall, ceiling, etc.) and inferiors from exposure to water.

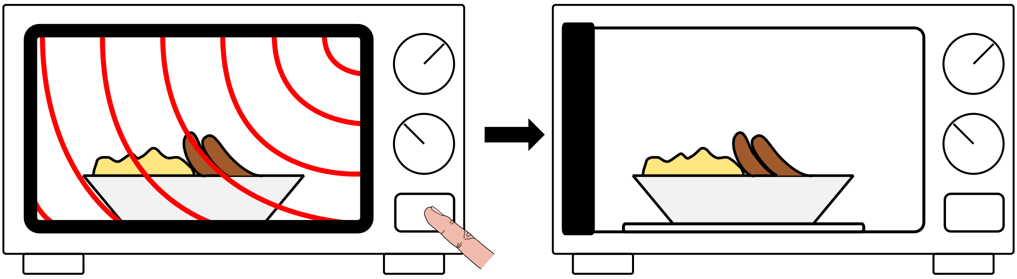


Fig. 5. A microwave oven can be opened freely by the user even the oven is on. The radiation is removed whenever user opens the door. (© Jari Rauhamäki)

4.9 Related patterns

An ACTIVE PROTECTIVE MEASURE (Rauhamäki & Kuikka 2014) such as E/E/PE SAFETY SYSTEM (Rauhamäki & Kuikka 2014) can be used to implement the mechanism to drive the system into a safe state whenever the guard is opened.

The LOCKED HAZARDOUS ZONE pattern illustrates a solution to the same problem. In this case, the system decides when the hazardous zone can be accessed and allows the access only when hazardous conditions are not present.

5. LOCKED HAZARDOUS ZONE

5.1 Intent

A hazardous zone is locked for access until the control system determines it is safe to access the area. For example, a washing machine door is locked until the end of the program.

5.2 Context

Hazardous zone or element has been mechanically isolated (ISOLATE HAZARD (Rauhamäki & Kuikka 2014)) from people, environment and other systems to mitigate the related risk due to hazardous conditions within the zone. The isolation is achieved using, for example, fences, enclosures, etc. Maintenance, operational or other tasks such as changing a tool bit or a workpiece, adjustment, blockage removal, lubrication or replacing worn parts need to be executed periodically within the zone.

5.3 Problem

A hazardous zone or element need to be accessible but hazardous conditions within the zone may exist in defined situations.

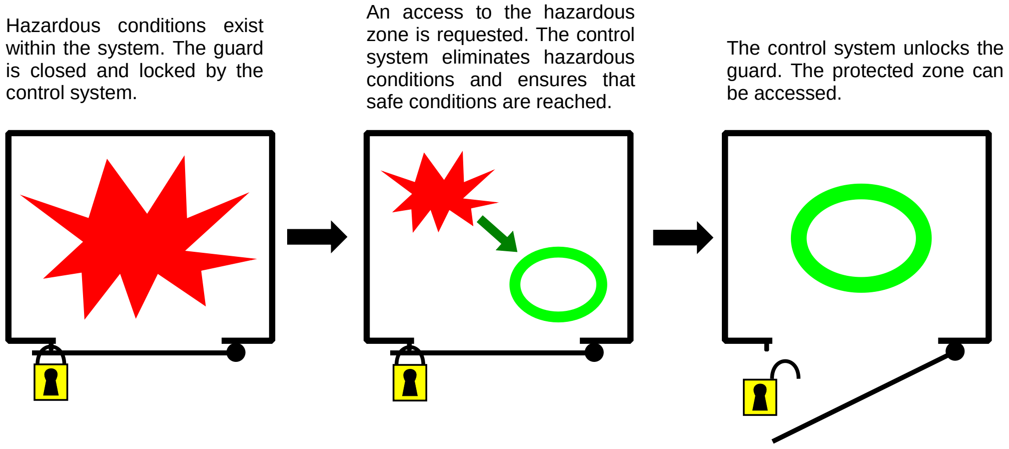
5.4 Forces

- The hazardous conditions cannot be removed fast enough by means of controlling the system introducing the conditions. That is, the control system is not able to drive the system in such a state that the hazardous conditions are removed in a sufficiently short period of time after the guard or the isolation is opened. For instance, moving parts of the machine cannot be stopped to prevent crushing, shearing, and impact hazards from occurring. This can be due to, for example, high inertia combined with low deceleration capacity, which effectively prevent stopping moving a part in a sufficiently short period of time. Another example is extreme temperature, which might be hard to compensate in a short period of time into a safe level.
- As the isolated zone needs to be accessible, a fixed guard is not an option as it instead of promoting, hinders the accessibility. Usage of a fixed guard to isolate location, which needs to be frequently accessed, leads likely to the complete removal or modification of the guard, the machine, or the system to allow the access. This again conflicts with the original purpose of the guard.
- The system operation optimization outweighs the user freedom to operate and access the hazardous zone. The primary control of access to the hazardous zone can be given to a machine and it is sufficient to let user request the access.
- The spatial requirements for the system are considerable so that the resulting system should have smaller dimensions instead of larger ones. Thus the distance between the hazardous element or zone and the isolating barrier should be as small as possible to reduce the area or the volume of the hazardous zone.

5.5 Solution

Implement a locking guard to cover the hazardous element or zone. A locking guard is a guard that is locked by the control system until hazardous conditions are removed from the hazardous zone so that the hazard cannot cause harm. The control system releases the guard lock when the hazardous conditions are removed and the hazardous zone can be accessed (Fig. 6).

The hazardous zone should already be isolated using an enclosure, a barrier, a fence, etc. To enable access to the hazardous zone, design an opening part such as a door, gate, or hatch in the isolation. When



Hazardous conditions exist within the system. The guard is closed and locked by the control system.

An access to the hazardous zone is requested. The control system eliminates hazardous conditions and ensures that safe conditions are reached.

The control system unlocks the guard. The protected zone can be accessed.

Fig. 6. Illustration of the operation of a locking guard operation. Reproduced from (Kivistö-Rahnasto 2015)

closed, the isolation should prevent people, body parts and other non-wanted objects for accessing the hazardous zone. Note, if implemented by a guard, the guard should be primarily left attached to the machine when opened (Machinery directive).

As the access to the hazard zone is prevented when hazardous conditions exist, one does not need to consider the time to drive the system into a safe state in terms of the considered hazardous conditions (still, user requirements may introduce time constraints). Due to this, there is no need to place the isolating cover far away from the hazard zone including the access locations and paths to the hazardous zone (unless other restrictions inflict this).

Equip the opening part with a locking mechanism, which is operated by the control system. That is, not the user or operator of the system should be able to release the lock (excluding emergency situations). The operation is as follows. User requests access to the hazardous zone. The control system drives the system into such state that the hazard(s) cannot cause harm, that is hazardous conditions are removed. The control system needs to be able to determine that the hazardous conditions have been removed through a measurement. In practice, hazardous condition removal might include, among others stopping moving parts, or shutting down radiation or noise sources. In some cases, the control system may not be able to directly remove the hazardous conditions by means of controlling the system. For instance, there may be no way for the control system to actively cool hot surfaces. In such cases, the control system can only wait for the surfaces to cool down to safe temperature. After the hazardous conditions are removed, the control system opens the guard lock and it is released to open.

The control system needs to be aware that the guard is closed, so in addition to locking mechanism, it needs a way to detect that the guard is closed before allowing the system to start again. In some cases, the locking mechanism may include a suitable sensor to identify the guard state. The start-up should be primarily actuated by user to prevent unexpected start-ups.

5.6 Consequences

+ The system part, for instance, hazardous movement, radiation, etc. does not have to be designed for (relatively) fast stopping. This may promote the usage of less expensive components or enable potentially less stress on the system due to the fast removal of the hazardous conditions.

+ The control system can ensure the removal of hazardous conditions before allowing access to the hazardous zone. This adds (potential) additional layer to protect the user from accessing the hazardous zone before the removal of hazardous conditions.

– The control system has to ensure the removal of hazardous conditions before allowing access to the hazardous zone. This indicates the following:

– Increased amount of sensors. The control system needs sensors to ensure that hazardous conditions are removed and that the guard is closed before allowing hazardous conditions to reappear (typically to restart the system by user).

– To be able to ensure the hazardous zone is not accessed before it is safe, a locking mechanism needs to be added to the isolating guard and it needs to be controllable by the control system. This increases cost, complexity and maintenance effort compared with a non-locking approach.

– More complex logic as potentially multiple sensors needs to be observed.

5.7 Known use

Although not machines in terms of the Machinery Directive, the solution model is found in domestic washing machines. The washing machine door is locked in the closed position, throughout the washing sequence/program. When the washing drum is stopped, energy to its driver is cut off, and water removed from the machine, the machine control system releases the door lock (see Figure 7). Only after these procedures the machine lets the user open the door.

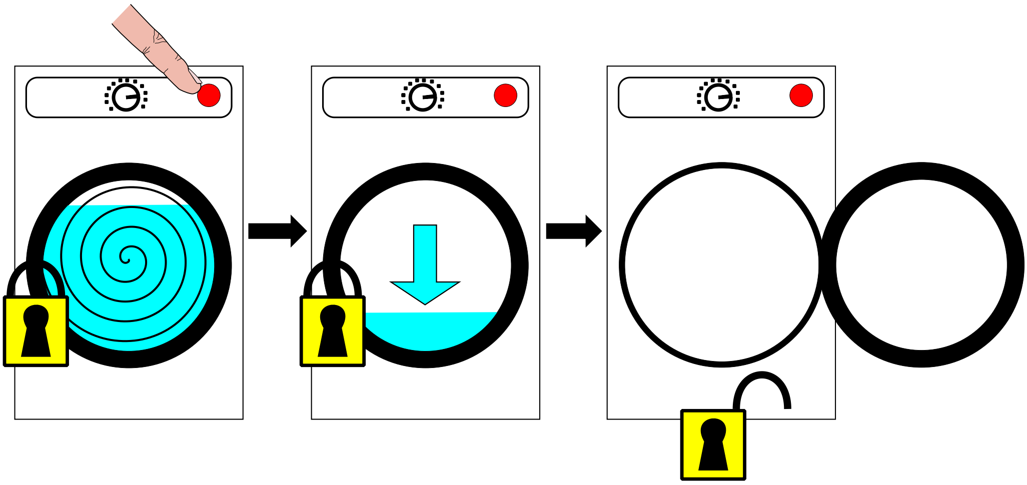


Fig. 7. Illustration of washing machine access during a program. The washing machine first needs to remove the water from the drum and prevent the motor from driving the drum before the feed door can be opened. (© Jari Rauhamäki)

Another device group in which the pattern is applied in some CD-, DVD-, and Blu-ray -drives using a tray for the disc loading. The drive does not open until the disc has stopped rotating inside the drive, that is, the disc comes to rest. When user requests eject, the disc is first stopped and only after that the tray opens. In this case the functionality primarily protects the disc itself, which could be scratched or otherwise damaged if lowered on the tray while spinning.

5.8 Related patterns

An ACTIVE PROTECTIVE MEASURE (Rauhamäki & Kuikka 2014) such as E/E/PE SAFETY SYSTEM (Rauhamäki & Kuikka 2014) can be used to implement the mechanism to drive the system into a safe state whenever the hazardous zone should be accessed.

The INTERRUPTIBLE HAZARDOUS ZONE pattern illustrates a solution to a similar problem. In this case, the user is free to access the hazardous zone and the system needs to conform to this. In practice, the system needs to detect if the hazardous zone is accessed and act accordingly to remove the hazardous conditions.

6. ACKNOWLEDGEMENTS

The author would like to greatly thank the shepherd Ralf Laue and the workshop group of EuroPLoP 2015 for all the suggestion, comments, and support to improve the paper.

REFERENCES

EN ISO 12100:2010 2010. Safety of machinery - General principles for design - Risk assessment and risk reduction (ISO 12100:2010).

Kivistö-Rahanasto, J. 2015. Vaaratilanteiden torjunta, suojukset ja turvalaitteet. Lecture material in Finnish.

Leveson, N. 2011. Engineering a Safer World : Systems Thinking Applied to Safety. In Engineering Systems series (eds.) Moses, J. (Chair), de Neufville, R., Heitor, M., Morgan, G., Paté-Cornell, E., Rouse, W. Massachusetts Institute of Technology. ISBN 978-0-262-01662-9.

J. Rauhamäki, J. and S. Kuikka, S. 2014. Strategies for Hazard Management ProcessPatterns for Hazard Mitigation Process. Presented in workshop of the EuroPLoP 2014 Conference, 9.-13.7.2014, Irsee, Germany.

Rauhamäki, J. J. and S. Kuikka, S. 2015. Strategies for Hazard Management Process II. Presented in workshop of the VikingPLoP 2015 Conference, 14.-17.5.2015, Ribaritsa, Bulgaria.

Publication VI

Rauhamäki, J., Kuikka, S. (2015). Strategies for Hazard Management Process II. In *Proceedings of the 20th European Conference on Pattern Languages of Programs.* [3] (ACM International Conference Proceeding Series). New York: ACM.

DOI: 10.1145/2855321.2855325

Strategies for Hazard Management Process II

JARI RAUHAMÄKI, Tampere University of Technology
SEPPO KUIKKA, Tampere University of Technology

Regarding hazard and risk management, one of the early decisions to be taken is to select the strategy for mitigating risk related to hazard. The most effective way is to eliminate the hazard from the system completely. This is, however, not always possible due cost, performance, usability, or other reasons. In such case, other measures need to be considered to mitigate the risks. In this paper, we present two of these strategies, namely active and passive protective measures, in a design pattern format. In many cases, a passive protective measure should be considered initially and preferred over an active protective measure whenever meaningful. Still, both strategies have their applications, and it is finally the designers' decision whether either is good fit for the considered risk.

Categories and Subject Descriptors:

Categories and Subject Descriptors: **K.4.1 COMPUTERS AND SOCIETY]**: Public Policy Issues—*Human safety*; **K.6.1 [MANAGEMENT OF COMPUTING AND INFORMATION SYSTEMS]**: Project and People Management—*Systems development*

General Terms: Human Factors

Additional Key Words and Phrases: Risk mitigation, hazard management

ACM Reference Format:

Rauhamäki, J. and Kuikka, S. Strategies for Hazard Management Process II. In proceedings of 20th European Conference on Pattern Languages of Programs, EuroPLoP 2015. 7 pages.

1. INTRODUCTION

When a hazard with intolerable associated risk is identified form a system, the risk needs to be mitigated. For this purpose hazard and risk mitigation methods need to be applied. In this paper, two abstract hazard and risk mitigation methods, namely active and passive protective measures, are introduced. This paper extends the work introduced in (Rauhamäki and Kuikka 2014).

Active and passive protective measures may have different meaning in different domains. For instance, in automotive industry an active safety measure is considered as functionality, etc. that tries to prevent harm from occurring. Such system is for example an automatic breaking system. A passive safety measure is considered any measure that is reduces the severity of the consequences after or during the accident. Such measures include for instance airbags and safety belts.

In context of this paper, however, we consider safety measures from the point of view of machinery and process systems. In these domains, a *passive safety measure* achieves risk reduction by means requiring no information on the system and/or environment state or a need to affect system operation by controlling it. Typical examples of such measures are guards (fixed, adjustable, or movable) and mechanical limits. In contrast, an *active safety measure* is considered achieving risk reduction by means of being aware of the system and/or environment state (e.g. in terms of one or more process variables) and affecting the state or operation of the system/environment/hazard source in an appropriate way.

1.1 Pattern language

The patterns relate to a larger collection of patterns considering safety system development. They position in the beginning of the collection forming the initial steps in safety system development and illustrate fundamental choices selecting the methods for risk reduction to achieve acceptable system safety. Figure 1 illustrates the patterns introduced in this paper (bold outline) alongside their closely related patterns (dashed outline). Table 1 provides short descriptions (patlets) of the patterns.

Authors' addresses: Jari Rauhamäki | Seppo Kuikka, Korkeakoulunkatu 3, FI-33720, Tampere, Finland; emails: jari.rauhamaki@tut.fi | seppo.kuikka@tut.fi.

EuroPLoP '15, July 08 - 12, 2015, Kaufbeuren, Germany
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3847-9/15/07…$15.00
DOI: http://dx.doi.org/10.1145/2855321.2855325

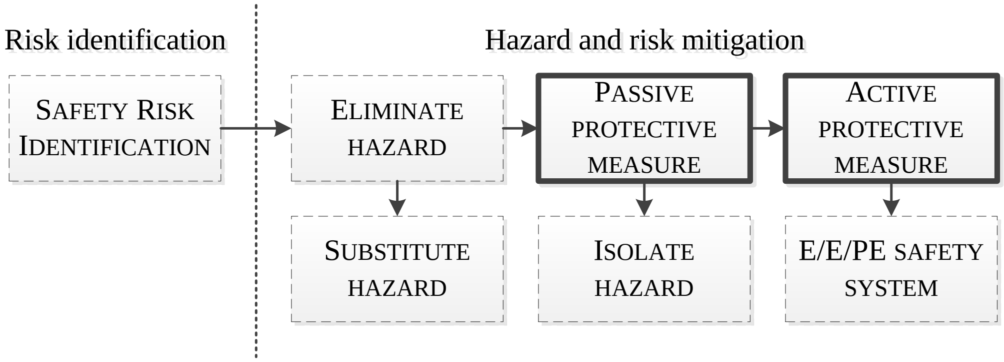

Risk identification
Hazard and risk mitigation
SAFETY RISK IDENTIFICATION
ELIMINATE HAZARD
PASSIVE PROTECTIVE MEASURE
ACTIVE PROTECTIVE MEASURE
SUBSTITUTE HAZARD
ISOLATE HAZARD
E/E/PE SAFETY SYSTEM

Fig 1. The patterns and their closely related patterns modified from (Rauhamäki&Kuikka 2014)

The hazard and risk mitigation methods introduced in this paper are likely known by experienced safety engineers and professionals. However, people with less experience in hazard and risk mitigation may potentially benefit from the illustration and thus form the most potential audience of the pattern described in this paper.

Table 1 Short description of the presented and related patterns

Pattern	Patlet
Safety risk Identification	To make conscious decisions considering hazard and risk management, information on these aspects needs to be available. Therefore, use structured and/or systematic method(s) to identify the hazards and associated risks introduced by the system. (Rauhamäki&Kuikka 2014)
Eliminate hazard	You want to maximize the likelihood that a hazard introduced by a system cannot cause harm in any part of the system lifecycle. Therefore, eliminate the hazard completely by removing the component introducing the hazard from the system. (Rauhamäki&Kuikka 2014)
Substitute hazard	A hazard is wanted to be eliminated from the system. Therefore, substitute the hazardous element with a non-hazardous or at least less hazardous element. (Rauhamäki&Kuikka 2014)
Passive protective measure	The system introduces a hazardous element of intolerable risk, which cannot be eliminated from it. Therefore, implement a passive protective measure to mitigate the risk.
Active protective measure	Passive protective measures have not provided an adequate approach to mitigate the risk. Therefore, implement an active protective measure to mitigate the risk.
E/E/PE safety system	Active protective measure of relatively complex functionality needs to be implemented. Therefore, use an electric, electronic or programmable electronic (E/E/PE) safety system to implement the protective measure functionality. (Rauhamäki&Kuikka 2014)

2 PASSIVE PROTECTIVE MEASURE

2.1 Context

The system under consideration introduces a hazard, which is according to risk assessment intolerable. The ELIMINATE HAZARD approach has been tried, but no way to eliminate the hazard has been identified due to, for instance, technical, system quality attributes degradation, operability, or cost restrictions.

2.2 Problem

The system introduces a hazardous element of intolerable risk, which cannot be eliminated from it.

2.3 Forces

- Introduction of a functional protective measure provides the system with the capability to react on, e.g., user and environment actions and events in terms of retaining the safety of the user. This opens possibilities to implement more advanced and complex protective measures, which could, e.g., improve the system usability, accessibility, performance, etc. However, such approach to mitigate the risk requires awareness, logic, and ability to affect the system. This increases the system complexity, cost, etc. Not all risks are worth the added complexity, cost, and design burden required to implement a functional protective measure.
- An added component, part or subsystem promotes functionality, adaptability, etc. of the risk mitigation method. However, every added component, part, or subsystem also contributes to the overall safety system fault characteristics including, among others, Mean Time to Fail (MTTF).
- Users could be protected against certain hazards applying administrative measures or personal protective equipment. Such non-engineering measures are, more or less, under user, customer, or operator consideration and control. Therefore, such measures cannot be directly influenced or enforced by the designer or manufacturer of the system. However, an engineering approach to mitigate risk should be considered prior to a non-engineering measure such as administrative methods or personal protection equipment. The engineering measures are applied by the manufacturer/designer of the system, and consequently under their consideration and control.

2.4 Solution

Implement a passive protective measure to mitigate the risk. A passive protective measure is a non-functional design solution, equipment, or system design feature that mitigates the risk related to a hazard. A passive protective measure does not remove or eliminate the hazard, but reduces the risk related to the hazard.

The main founding principle of a passive protective measure is the restriction. In practice, the likelihood of realization and/or the severity of the consequences of a risk are reduced. For example, the likelihood of risk realization can be decreased by restricting user access to the hazardous zone or restricting the freedom of operation. The severity of consequences can be restricted by, for example, capturing hazardous material in an appropriate container in case the material leaks out from the main container.That is, a passive protective measure is based on (reliable) structural and/or physical design or structure to reduce the risk by introducing a physical restriction (or property).

A passive protective measure does not (need to) sense or observe the state of the system, environment, or process variable, actively respond to such state changes or introduce functionality to the system to reduce risk. (Center for Chemical Process Safety 2012). There is no need for logic to be implemented in form of electronics, software, etc. Instead, the restrictions promoting safety and safe operation are built in the structure and physics of the designed solution. This promotes the reliability of the approach.

A passive protective measure is typically fixed in terms of self-adjustment. The measure may be adjustable or configurable by a human operator in certain cases such as cutting chip stopper in a lathe. However, the measure is and remains unaware of itself and its state.

2.5 Consequences

+ The risk is mitigated by the restrictive protective measure. In the best case, a passive protective measure may *eliminate the risk* if the measure prevents either the likelihood or the consequences of the risk realization. Still, the extent of mitigation depends on the case.

+ There is no logic involved in a passive protective measure that may malfunction or affect the system state. Because of this, there is no sensor, logic, or actuator that could fail.

+ Potentially a lower operating cost compared with ACTIVE PROTECTIVE MEASURE due to potentially lower maintenance cost (Center for Chemical Process Safety 2012).

+ The protective measure is designed and implemented by the designer and/or manufacturer of the system under consideration.

− *The hazard remains in the system*. A passive protective measure does not remove the hazard or hazardous element from the system.

− A passive protective measure needs to be scaled to and fixed for the worst case situation regardless of the parameters of the system or the environment. For instance, the thickness of fireproofing material needs to be scaled for the worst case (the highest) temperature and exposure time although the expected values for the temperature and exposure time would be lower. This may hinder other quality attributes of the system such as increase structure weight or cost.

− A passive protective measure is not aware of itself, the hazard, or its environment. Thus it is unable to carry out self-diagnostic operations or notice if it has been defeated or removed. The latter case can be implemented by providing the system the ability to detect the defeat or removal of the measure.

− A passive protective measure requires maintenance and inspections. The protective measure may wear, break, or get removed or bypassed etc., which results in the measure not mitigating the risk as intended (provides no or reduced mitigation).

2.6 Example

A passive protective measure can appear in various forms including, but not restricted to, for example:

- Fireproofing: A steel beam is coated with fireproofing material to increase the structure durability under fire situation.
- Fixed non-interlocking guard: A guard is positioned between an operator and a cutting blade.
- Fence: A fence restricts access to robot working area.
- Roll cage: a race car is equipped with a sturdy roll cage to retain the body shape in case of the car rolls on its roof.
- Fixed lines: Free running hose can be accidentally located in a hazardous position e.g. on a passage, whereas for fixed lines need to be initially located away of such positions.
- Leak dike: A tank containing hazardous material is located above a dike capable of containing the contents of the tank.
- Incompatible connectors: To prevent misconnection of multiple lines, each line is equipped with a distinct connector that is only fits into the correct supply connector. In more general, this approach is also known as poka-yoke, which purpose is to prevent defective conditions. (Shingo and Dillon 1989).

The connective aspect in all the aforementioned cases is that the hazard is not removed, but it is mitigated by applying a passive protective measure that does not sense the state of the system or its environment or respond to the system.

2.7 Known use

Connectors used in medical appliances are moving towards incompatibility between appliances intended for different purposes. The EN ISO 80369-1 states: “small-bore connectors of each application category specified in this International Standard shall be non-interconnectable with any of the small-bore connectors of every other application category for risks to be acceptable, unless otherwise indicated”. (EN ISO 80169-1:2010 2010). That is, for example, a connector for the neuraxial category use shall be non-interconnectable with the enteral gastric category.

Passive roll-over protection systems have been used widely in tractors and race cars. In practice, such systems are mechanically sturdy structures that restrict the vehicle crushing the passengers by preserving a space between ground and the vehicle body.

A hydraulic guillotine shear has a finger guard attached in front of the blade and the moving work piece holders (Baykal Machine Tools 2015). The guard has two objectives. Firstly, to reduce the likelihood to locate any body part under the guillotine blade when the machine is operated, that is, prevent a body part from accessing the blade and sheet holders’ operation zone. Secondly, to reduce the severity of the consequences of the harm, if one manages to locate a body part in the operating zone of the blade or a sheet holder. One might be able to squeeze a finger under the guard, but it is relatively hard to squeeze a hand or an arm under the guard.

2.8 Related patterns

The ACTIVE PROTECTIVE MEASURE strategy describes a solution to a similar problem with a different approach. An ACTIVE PROTECTIVE MEASURE is functional and thus observes and affects system operation to mitigate the risk. The ISOLATE HAZARD is a way to implement a PASSIVE PROTECTIVE MEASURE.

3 ACTIVE PROTECTIVE MEASURE

3.1 Context

The system under consideration introduces a hazard, which is according to risk assessment intolerable. To mitigate the risk, a protective measure is being designed as the hazard could not be eliminated (see ELIMINATE HAZARD). Application of a PASSIVE PROTECTIVE MEASURE has not lead to satisfactory solution in terms of risk reduction or quality attribute degradation. That is, risk is not mitigated enough, a passive solution would not achieve required functionality, or other quality attributes such as usability, maintainability, or performance of the system have degraded intolerably with attempts to use one or several PASSIVE PROTECTIVE MEASURES.

3.2 Problem

Passive protective measures have not provided an adequate approach to mitigate the risk.

3.3 Forces

- A PASSIVE PROTECTIVE MEASURE is typically a simple approach to mitigate a risk. However, its lacks the ability to introduce any functionality in terms of retaining safety. This may hinder the usability, operability and their related performance of the system under consideration. For example, isolating barriers and fences automatically restrict the access to the hazard or hazardous zone (which is their purpose), but in some cases such restriction could hinder the system usability. Thus, an approach to allow increased freedom in system operation and functionality is sought after.
- In some cases it is sufficient to mitigate the likelihood and consequences of a risk realization in a restrictive manner. Making the protective measure unaware of itself, the hazard, and the environment promotes the simplicity of the measure. However, in such case the measure needs to be scaled for the worst case situation. Still, depending on the case, the state of the system does not always require the full restriction. In identified cases, a protective measure may use a relaxed restriction or operation compared with the worst case situation. Nevertheless, the protective measure must be prepared for the worst case, but it doesn't need to be fixed, if the measure adjusts its operation according to the state of the system.
- Users could be protected against certain hazards by applying administrative measures or personal protective equipment. Such non-engineering measures are more or less under the user, customer, or operator consideration and control and cannot be directly influenced or enforced by the designer or manufacturer of the system. However, an engineering approach to mitigate risk should be considered prior to a non-engineering measure such as administrative methods or personal protection equipment. The engineering measures are applied by the manufacturer/designer of the system, and therefore are under their consideration and control.

3.4 Solution

Implement an active protective measure to mitigate the risk. Instead of eliminating the hazard or using passive protective measures, implement a protective functionality that recognizes the hazardous situation and alters the system functionality to retain the safety of people. In case a hazardous situation occurs or is developing, an active protective alters or controls the system operation so that the hazardous situation is halted or its realization is prevented (or its consequences are reduced).

An example of the former case is a residual current device. The purpose of such device is to break an electric circuit if the input and output currents measured by the device deviate from each other more than a specified limit. Such situation occurs, for instance, when part of current flows through a person to ground. Another example of such case is a stop function that halts a moving machine if a person (or an object) passes through a light-curtain.

An active protective measure is (and needs to be) able to observe or monitor the state of the system under control and affect the operation of the system through control to mitigate the exposure to or the consequences of realization of a hazard. The measure is, however, limited by the abilities it is given. An active protective measure can only:

1) Observe and quantify phenomena for which information is provided to the measure either through sensors or data input from other systems.
2) Reason within the logic implemented in it.
3) Affect the system by the means provided to the measure either through the direct control of the actuators or otherwise affecting the system state, for example, by setting the control set point of the system or affecting the system structure.

That being said, an active protective measure is not omnipotent, but its abilities can be extended beyond the capabilities of a PASSIVE PROTECTIVE MEASURE. An active protective measure may be, e.g., able to run and communicate self-diagnostic data, adjust its operation according to environment, such as allow higher speed or larger area of operation. In addition, an active protective measure may be able to detect if it is potentially defeated somehow.

3.5 Consequences

+ The risk is mitigated by the protective measure. In the best case, an active protective measure may *eliminate the risk* if the measure prevents either likelihood or the consequences of risk realization. Still, the extent of mitigation depends on the case.
+ The protective measure can observe the state of the system and its environment and act accordingly. This provides the measure the ability to adjust its operation according to the state of the system and its environment. Thus, the measure does not have to be continuously scaled for the worst case situation. For instance, an elevating work platform can have larger load when the cage is sufficiently near the body, and the overweight protection can take this into account if it operates as an active measure.
+ The protective measure does not have to restrict system operation continuously and all situations. Instead, it can allow different operation in different operation points. For instance, a rate of flow in a pipeline can be allowed freely under certain conditions and restricted under other conditions.
+ The protective measure is designed and implemented by the designer and/or manufacturer of the system under consideration. This provides the designer with the capability and control over the protective measure and it is not so much under the influence of the user or user organization.
+ If an active protective measure is aware of itself, the system or the environment to some extent, it may have the ability to identify failures or attempts to defeat the measure.
− The hazard remains in the system. An active protective measure does not remove the original hazard or hazardous element from the system.
− An active protective measure has a potentially higher operating cost compared with PASSIVE PROTECTIVE MEASURE due to a potentially higher maintenance cost (Center for Chemical Process Safety 2012). For instance, the logic, sensors, and actuators need to be potentially tested and changed periodically to be retained operational.
− There is some sort of measurement, logic (or functionality), and actuation associated with the protective measure. Designing and implementing such logic is (typically) more complex than designing and implementing a passive protective measure. Each of these elements needs to be functional in order to achieve the protective functionality.

3.6 Example

An active protective measure can appear in various forms including, but not restricted to, for example:

- Seat belt: A mechanism observes the speed/acceleration of the belt going through the mechanism. In case the belt speed/acceleration exceeds the specified limit, the belt feed mechanism locks down and prevents further movement of the belt.
- Robot working area protection: A working area of a robot is isolated (ISOLATE HAZARD) with a fence. The fence door activates a safety function that stops or slows down the robot when the door is opened (which would allow a person to enter the hazardous zone).
- Process variable control: The temperature of a process vessel is monitored using a temperature sensor. If the temperature crosses a specified limit, an active protective measure cuts off the heat source of the vessel, applies cooling, or other counter measure to prevent further temperature increase.

3.7 Known use

Electricity driven hydraulic guillotine shear has a fixed finger guard attached in front of blade and plate holders (Baykal Machine Tools 2015). Part of the guard can be opened by the machine operator. Whenever the guard is lifted, the machine motion, including the blade, sheet holders and rear supports are halted. The machine also employs a light curtain monitored area behind the machine to detect a person or object entering the hazardous area and to trigger similar halt function as described previously.

In machinery and process system perspective, seat belts implement an active protective measure. A seatbelt mechanism observes the acceleration of the belt and locks down in case acceleration beyond a specified limit occurs. This prevents the belt from loosening in case of an impact (which causes the high acceleration of the belt).

Rupture disks and fuses are extremely simple active protective measures. They observe a process variable, pressure and current respectively and alter system operation opening line and breaking a circuit respectively. Rupture disks cannot typically recover once ruptured and needs to be replaced. If similar functionality and recovery ability is required, pressure relief valves could be considered. However, the applications of the two options are typically distinct.

3.8 Related patterns

An E/E/PE Safety system is a form of Active protective measure. In an E/E/PE Safety system the logic implementing the protective functionality is implemented using an electric, electronic, programmable electronic approach. This provides potential for relatively complex logic, especially when programmable electronics are considered.

4 ACKNOWLEDMENTS

The authors would like to thank the shepherd of the paper Timo Vepsäläinen and workshop participants for their valuable and insightful comments, suggestions and effort to help the authors to improve the paper.

5 REFERENCES

Baykal Machine Tools 2015. Hgl: User-friendly control unit and software. Baykal Machinery. Site: http://www.baykal.com.tr/en/products/shears/hgl/explore [accessed 14.1.2015]. Photo available: http://web.archive.org/web/20150114055044/http://www.baykal.com.tr/en/products/shears/hgl/fotograflar/productgallery_hgl-kullanici-dostu-kontrol-paneli-ve-yazilim_20140607171946.jpg [accessed 14.1.2015].

EN ISO 80169-1:2010 2010, Small bore connectors for liquids and gases in healthcare applications - Part 1: General requirements (ISO 80369-1:2010).

Long, B. 2013. Mercedes-Benz SL: R129-series 1989 to 2001. Veloce Publishing Ltd. ISBN: 978-1-845844-48-6. p. 208.

Rauhamäki, J. and Kuikka, S. 2014. Strategies for Hazard Management Process, In proceedings of EuroPLoP '14, July 09 - 13, 2014, Irsee, Germany. ACM 978-1-4503-3416-7/14/07. http://dx.doi.org/10.1145/2721956.2721966.

Rauhamäki, J. and Kuikka, S. 2014. Strategies for Hazard Management Process. Presented in workshop of the EuroPLoP 2014 Conference, 9.-13.7.2014, Irsee, Germany.

Shingo, S and Dillon, A. P. 1989. A Study of the Toyota Production System From an Industrial Engineering Viewpoint. Translated by Dillon A. P., Productivity Press. Revised Edition. ISBN 0-915299-17-8 p. 257.

Publication VII

Rauhamäki, J., Kuikka, S. (2014). Patterns for Sharing Safety System Operation Responsibilities between Humans and Machines. In *Proceedings of the 8th Nordic Conference on Pattern Languages of Programs VikingPLoP 2014, Sagadi Manor, Estonia, 10.4.-13.4.2014.* (pp. 68-74). [7] (ACM International Conference Proceeding Series). New York: ACM.

DOI: 10.1145/2676680.2676687

Patterns for Sharing Safety System Operation Responsibilities between Humans and Machines

JARI RAUHAMÄKI, Tampere University of Technology
SEPPO KUIKKA, Tampere University of Technology

Although the automatic operations of machines, processes, and systems have increased, human operators are still typically required to operate or monitor the system or operate in proximity of the considered process or system. As people operate the systems and in their proximity, they have a role in the overall safety system operation. Automated safety systems, which primarily ensure a safe operation of the system under control, have properties that humans are not capable of, but they are not perfect either. In this paper, two patterns for sharing the responsibilities between automated safety systems and human operators are presented. The strengths of automated safety systems and human operators are combined so that the weaknesses of the other can be overcome.

Categories and Subject Descriptors: **[Hardware]:** Robustness - *Safety critical systems*

General Terms: Human Factors

Additional Key Words and Phrases: E/E/PE, Redundancy, User roles

ACM Reference Format:

Rauhamäki, J. and Kuikka, S. 2014 Patterns for Sharing Safety System Operation Responsibilities between Humans and Machines. VikingPLoP 2014 (Vihula, Estonia, 2014). ACM ICPS series

1. INTRODUCTION

Processes, machines and other systems rarely operate completely autonomously though control of these kinds of systems is typically automated. Instead, human operators are typically required to operate, monitor, and maintain the systems. As people operate the systems and work in their proximity, they have a role in keeping the system in safe operating region. In this paper, a safety system refers to an automated system of which purpose is to decrease risk related to a hazard. An example of such safety system is a system for stopping a cutting machine if human body part is potentially detected in the blade work area. Typically, in modern systems, these kinds of safety systems are implemented using an E/E/PE (Electric/Electronic/Programmable Electronic) approach.

At least two roles for humans in context of safety system operation can be identified. Primarily, humans are the objects that are protected from the hazards introduced by the system. In this role, humans are somewhat erratic and one should expect that humans will put themselves in danger at some point deliberately or subconsciously. Thus, there needs to be a safety system to mitigate the risk. Secondly, humans operate and monitor the system. Because of this, they are a part of the system functionality and control. Thus, the operators have a role in the protection of others as the operators are in control of the system.

Authors' address: Jari Rauhamäki | Seppo Kuikka, Korkeakoulunkatu 3, FI-33720, Tampere, Finland; emails: jari.rauhamaki@tut.fi | seppo.kuikka@tut.fi; ACM 978-1-4503-2660-5/14/04$15.00 http://dx.doi.org/10.1145/2676680.2676687

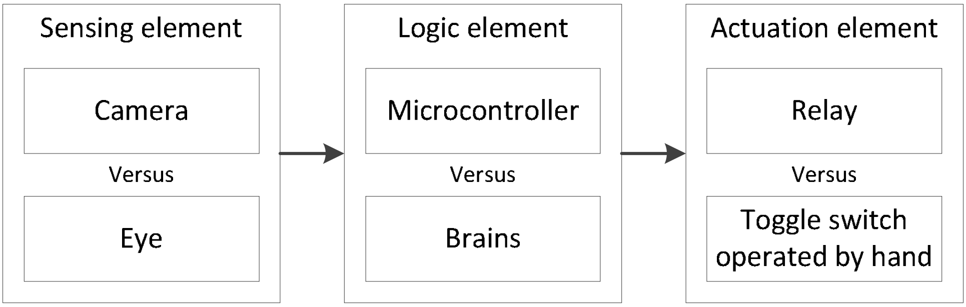


Figure 1: Human and machine elements in safety function context

As humans and automated system both can influence safety, how should one share the responsibilities between them? Humans and automated systems have their strengths and weaknesses considering their operation as a part of a safety system (Figure 1). Generalizing this line of thought to some extent, the strengths of one are the weaknesses of the other. For instance, humans are able to sense the environment and the system broadly whereas E/E/PE systems typically have more limited capabilities in this aspect. However, in some cases, especially inside a machine, device or system, humans cannot operate. Thus, observing such locations need to be left for machines. On the other hand, E/E/PE systems are tireless and have deterministic reactions, whereas human properties do not similarly prosper on this aspect. In addition they can be placed to observe locations humans cannot tolerate. This setup gives both actors a different role in context of safety system operation.

In this paper two patterns considering the role of human in context of safety system operation are presented. The patterns are targeted for designers with no or restricted experience in safety system design. The patterns discussed and referred to in this paper are shortly introduced on Table 1.

Table 1: Short descriptions of the patterns mentioned in this paper

Pattern	Patlet
AUTOMATED SAFETY SYSTEM	Humans are slow and unreliable decision makers and cannot deterministically react on random events. Therefore, Avoid humans as a part of the safety system functionality.
MANUAL SAFETY ABILITY	E/E/PE safety systems have limited possibilities to observe and interact with their environment and system. Therefore, provide the human operators with the ability to manually drive the system into a safe state.

2. AUTOMATED SAFETY SYSTEM

Context

A system under development introduces a hazard related to the operation of the machine. The system is operated or monitored by human who is able to obtain required amount of information to justifiably activate a safety function. To mitigate the hazard there is a need for constant awareness and deterministic actions to achieve the objective of hazard mitigation. However, humans are unreliable decision makers and observers and cannot deterministically react on random events to execute a desired response.

Problem

The safety function needs to be implemented with a system that is capable of deterministic reaction. That is, the actuation of the safety function must happen deterministically in a specific time window.

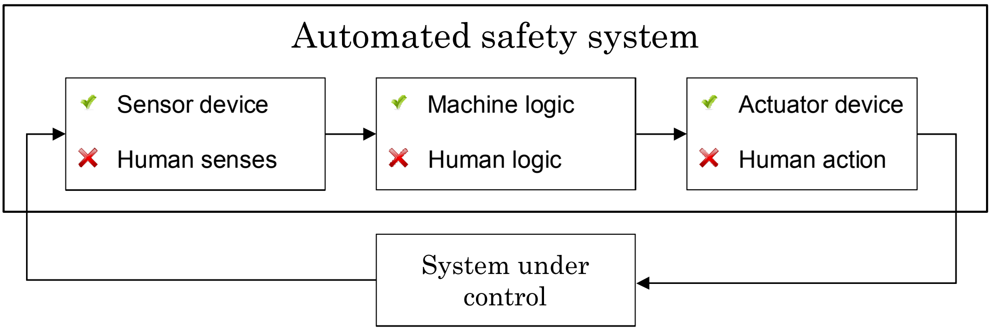
Forces

- Humans have relatively long reaction time compared with electronic systems. Human reaction time is about 140-160 milliseconds for auditory stimulus and 180-200 milliseconds for visual stimulus (Kosinski 2013) whereas automated system can react in microsecond time scale (consider for example microcontroller). In many cases human reaction time to actuate a safety function would be sufficient, but there are exceptions. For instance, shutting down a radiation source in case of radiation shield removal may require faster reaction time than human is capable of.

- Humans get bored executing repetitive (e.g. process value monitoring) task, are fallible and distractible, which may lead to ignoring or missing a critical event. As a consequence human reaction is not deterministic. For instance, human may be distracted by chatting with another person, they could sneeze on a critical moment or in a quick decision making situation they may misinterpret the state of the system and make wrong decision. Thus, typically human can react in one second, but in some cases the time may be several seconds, which in some cases is already insufficient reaction time. Typically, this is a more severe problem than the difference between absolute reaction time of human and automated system.
- Humans may not be able to resolve a meaningful safety function outcome under a stress of strict time limits. That is, in addition the reaction itself is not deterministic. For instance, if there are two seconds to react on an event and select correct reaction, human needs to interpret the input, resolve the situation and act correctly under a strict time limit. This may increase likelihood of selecting wrong action or missing the time limit.
- For humans it is hard to quantify an absolute value for magnitude that is, for example, used to determine whether a safety function should be activated or not. It is easier for humans to determine if a pressure or temperature is higher or lower than a reference, but very hard to determine what the absolute pressure or temperature value is.
- For humans it is hard to sense magnitudes or the state of events inside the system due to the observed location is unreachable (the system is enclosed as it is typically) or the magnitude to be determined is dangerous (high pressure, voltage or temperature, toxic substances or no air to breath, for example).

Solution

Implement the safety function applying an automated system approach and avoid humans as a primary part of this functionality. Human involvement should not be required (or at least required human involvement should be minimized) for a successful outcome of the objective of a safety system. Human factors are involved in most process industry incidents (Broadripp, 2012).



Let an automated safety system implement all the necessary elements of safety function implementation. This can be achieved, for instance, by electric, electronic or programmable electronic (E/E/PE) system. Other possibilities are, for example, hydraulic, pneumatic, and mechanical systems. However, often the latter are, or can be, controlled with an E/E/PE system. In context of automated safety systems, observing the system state, the processing according to safety function (logic), and actuation should be machine controlled to reduce the required human involvement. The target is to provide the automated safety system all the information (from sensing elements), ability to interpret this information (by safety functions), and means to control the system (through the actuators) to meet the safety function goals.

Though human operators should not primarily be part of safety system functionality as a basis, human operators should be provided with information about hazardous conditions and means to react accordingly.

There are situations, such as safety system failures or bypasses, where a human operator may be able to prevent or mitigate the consequences of a hazard realization (see) if they are just aware of the situation and able to affect the system accordingly. However, primarily the safety system should be as independent of human operators as possible and integrating humans as a functional part of a safety system should be considered as a supplementary approach.

Consequences

+ Automated safety system takes care of deterministic (including sufficiently fast) response in order to implement the safety function(s).
+ Actions of human operators required in operation of the safety system are minimized. That is, operators are liberated to observe the general view instead of focusing on specific safety function tasks.
+ Hardware and software are used to do what they do best, that is, routinely and deterministically monitor the system and actuate safety function if required.
− A dedicated safety system needs to be designed, build and maintained though an operator is nevertheless required to operate or monitor the system. This increases the cost of the system in terms of development, instrumentation, construction, use, and maintenance.
− It is difficult, expensive and still potentially inadequate for cover all possible events and event chains leading to harm in hazard and risk identification. Even though this could be achieved, covering all possible hazards with an automated system could prove impracticable.

Example

Consider a process vessel (a container, tank or similar equipment designed to hold a substance - gas, liquid, or solid material) that has potential to overheat, which would cause a serious hazard. To prevent the overheating hazard an alarm system could be implemented so that the operator is alarmed if the temperature rises over a specified limit in which case the operator would actuate the safety function (e.g. disconnect heat source from the vessel or open cooling water valves from control room). In such case, a human is involved in operation of a safety system. The operator to be alarmed could be e.g. distracted to notice the alarm information in which case the system could overheat to a dangerous level. To circumvent the human aspect, an independent E/E/PE safety system can be constructed to monitor the temperature of the vessel and to actuate the safety function if the measure surpasses the specified temperature limit.

Known use

A steel cutting machine is equipped with an E/E/PE safety system that stops the cutting operation whenever a person enters the backside of the machine. The stopping functionality is completely autonomous from a human operator.

Another application of this principle is taken into use in automotive industry. Automatic braking systems are used to prevent or mitigate the consequences of collisions have been implemented in recent models. The system initially provides a warning to the driver, but in case the driver does not respond to the warning, the system automatically decelerates the vehicle to prevent or mitigate collision consequences. (Grover et. al. 2008).

Related patterns

The MINIMIZE HUMAN INTERVENTION pattern (Hanmer, 2007) illustrates a similar approach in fault tolerance context suggesting implementing the error processing and recovery without human involvement to speed up the process. The MANUAL SAFETY ABILITY pattern illustrates a way to supplement an automated safety system with human abilities. E/E/PE SAFETY SYSTEM (Rauhamäki and Kuikka, 2014) pattern describes a potential approach to implement an automated safety system using electric, electronic, or programmable electronic approach.

3. MANUAL SAFETY ABILITY

Context

As a result of hazard and risk identification process, a system is identified to be able to cause damage or harm to humans, environment, the system itself or other systems. The system is operated and/or monitored by humans so that they can act to retain the system in a safe operating region. An E/E/PE (Electrical/Electronic/Programmable Electronic) system is deployed to implement a safety function to mitigate the identified risks.

Problem

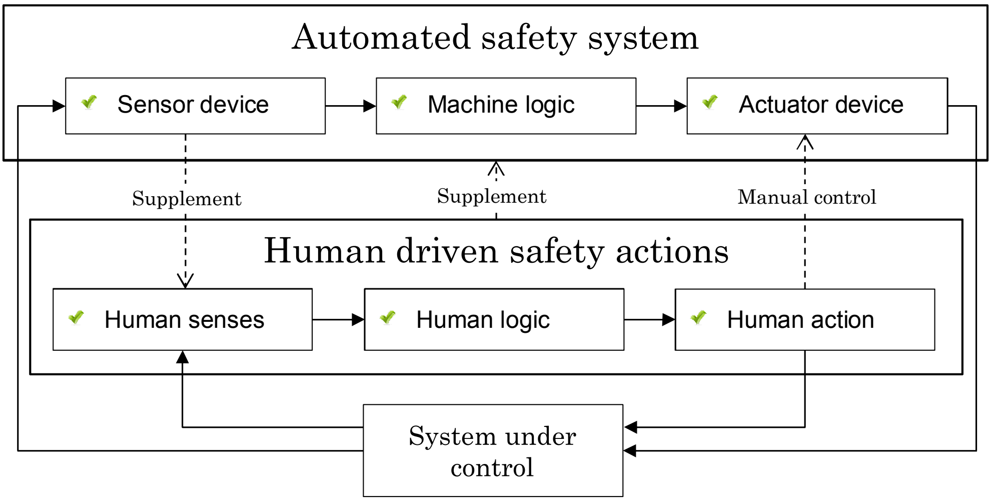
For an E/E/PE system it is hard to react correctly or in any way in situations which it was not designed for. E/E/PE safety systems have limited possibilities to do reasoning as well as observe and interact with their environment and the system under control due to limited sensors, logic and actuators provided for the safety system. Thus, the safety system may have limited capabilities to mitigate hazards related to unidentified and unexpected hazards or in case of system or component failure of the E/E/PE safety system.

Forces

- Hazard and risk analysis for the system under control may have omitted some of the potential hazards and associated risks. For a large process, machine, or a system it is relatively hard to identify all possible events and chain of events that may lead to a hazard. An automated safety system (typically) cannot react correctly on unforeseen hazards.
- In case of a failure of an automated safety system there is a need to retain the system in a safe operating region or transform the system into a safe state in another way.
- Safety systems have limited capabilities to sense the state of the system under control and its environment as the safety system utilizes a limited amount of sensors. Providing an E/E/PE safety system with broad capability to sense its environment and the hazardous system parts would be expensive and increase complexity in terms of safety system devices and design.
- Safety systems have limited capabilities to do logical reasoning as they can only do reasoning programmed or otherwise implemented into it. Providing an E/E/PE safety system with ability to do human like reasoning is problematic in general and if applied in safety system context it would be hard, expensive, and complex.
- Safety systems have limited capabilities to affect the system as they utilize a limited amount of actuators. Providing an E/E/PE safety system with broad capability to affect the system would be, expensive and increase complexity of the (safety) system.
- The human operators of the system potentially have a much better general view on the environment and thus they can observe environment in ways that safety systems typically cannot. For instance, it is relatively easy for human to detect a blowpipe flame (if one is visible), but for E/E/PE system this might require special hardware.
- A human operator reasoning is not similarly restricted as the predefined reasoning of the logics of E/E/PE safety systems is.
- Humans tend to feel more comfortable if they know they have some mechanism to have control over an automated system.

Solution

Provide the human operators with the ability to manually drive the system into a safe state (see SAFE STATE (Eloranta et. al 2014) or a safe operating region. This can be achieved for example by providing an emergency stop button which stops any movement of the corresponding machine or system. Any kind of automated safety measure may fail as a cause of bypassing, removal of the function, hardware or software errors, and unexpected conditions and so on. Human operators are capable of observing machine operation, its environment and others around the machine, do reasoning (beyond E/E/PE system logics) to identify hazardous situations and act to prevent a hazard from realizing (though relatively slowly compared with E/E/PE system). Thus, human operators have ability to operate beyond the safety system and, if needed, take the system into a safe state.



In practice, automated safety systems are hard to build to take account all possible hazards, mainly because it is difficult to identify all the possible hazards beforehand. Instead, the safety systems target the hazards that can be identified (and which risk needs to be mitigated). Design of practically any system includes some assumptions such as: a wall will hold any foreseeable pressure level, piping will not leak, or devices withstand the forces given in their specification and operate as expected. However, various reasons may lead to situations in which such assumptions do not apply and a hazardous situation is created. For an E/E/PE safety system it is very difficult to observe such an event for which it was not designed for because it has no suitable sensor element, it cannot reason what to do in the situation if no logic is provided for the task, or affect the system accordingly if there are no suitable actuators for the purpose. A human operator still may be eligible to perceive the situation, do logical reasoning and react accordingly.

Manual safety actuation can be implemented in various ways. The actual implementation is not as important as to give the human the ability to bring the system into a safe state. One typical approach is described in the known use section. Regardless of the implementation, the manual safety actuation should not be the primary safety measure, but only to be used as last resort. That is, a human operator can drive the system into a safe state if the primary safety system fails to do this.

Consequences

+ There is a possibility to apply the safe state or drive the system in a safe operating region by human operators when an (E/E/PE) safety system fails to react or has no possibility react on a hazardous situation. Thus, also hazard emerging from unexpected or unidentified events or chains of events in context of E/E/PE safety system can be handled (at least to some extent).

+ Human strengths are efficiently used to improve safety, i.e., to prevent or reduce the consequences of unpredictable hazards which an automated safety system is incapable to detect, reason, or handle. That is, human operators can supplement the E/E/PE safety system operation forming a redundant hazard mitigation resource especially considering abnormal situations.

– Additional hardware and possibly logic is required to implement the safety system. To take the system into safe state human needs a suitable interface such as an emergency stop button or a manually operable valve. These interfaces need to be located at least at the potential place of a hazard, but also at other places where they are seen appropriate.

- Humans provide no deterministic behaviour from the safety point of view. That is, though a human operator is able to actuate the safety function, it could happen in wrong time, wrong way or unnecessarily. In the latter case, the system availability is hindered as the system is taken towards a safe operation region when it is not actually necessary. For instance, an operator shuts down a machine inadvertently.

Example

Consider a mobile machinery application that operates using a hydraulic system. Due to wear of the piping or external reasons a breakage may appear in the piping in which case the hydraulic oil would start out of the hydraulic system. This kind of hazard could be mitigated with an E/E/PE safety system, but in the hazard and risk identification phase the risk related to this event was considered negligible and no E/E/PE system was deployed to mitigate the hazard. Thus, there is no automated safety system to react on such situation. Nevertheless, a human operator may identify the hazard and shut down the hydraulics to prevent further leakage of oil.

Known use

A known use of manual safety actuation is the emergency stop button, which takes the system into the safe state. Emergency stop buttons are used to take the system into a safe state when a hazardous situation is detected by a human. Emergency stop buttons are mandatory in machinery applications according to the Machine Directive (Machinery Directive, 2010).

Mobile elevating working platforms are equipped with manual safety actuation to drive the platform down on the ground level in emergency situations. Such situations are, for instance, incapable (e.g. fainted) platform operator or a movement halt of the platform due to the overload or insufficient ground contact of a support leg of the platform. The manual actuation elements reside on ground level (on the platform of the machine) so that they can be operated from by others.

Related patterns

The MAXIMIZE HUMAN PARTICIPATION pattern (Hanmer, 2007) illustrates a similar approach in fault tolerance context suggesting providing system experts possibility to participate in the system operation. The DE-ENERGIZED OVERRIDE pattern (Rauhamäki & Kuikka, 2013) describes how a safety system can override the control system to obtain a safe state. The approach can be applied for the implementation of manual safety ability. The SAFE STATE (Eloranta et. al 2014) pattern describes the concept of a safe state in which the machine or system introduces minimal risk for itself, its environment and people around it. The pattern does not take a stand on how the safe state is obtained and retained. The DE-ENERGIZED SAFE STATE (Rauhamäki & Kuikka, 2013) pattern suggests designing a system to take a SAFE STATE when it is de-energized. That is, if the system (or part of it) is not energized, it returns into a defined safe state.

4. ACKNOWLEDGEMENT

The authors would like to greatly thank the workshop group members in the VikingPLoP 2014 event and especially our shepherd Manuel Menghin for their valuable feedback and input to improve the patterns and the paper.

5. REFERENCES

Broadribb, M. P. 2012, "It's people, stupid!: Human factors in incident investigation", Process Safety Progress, vol. 31, no. 2, pp. 152-158.

Eloranta, V.-P., Koskinen, J., Leppänen M. and M., Reijonen, V. 2014. Designing Distributed Control Systems: A Pattern Language Approach. Wiley Software Patterns Series (1st ed.). John Wiley & Sons, Ltd, Chichester. ISBN: 9781118694152.

European Parliament and of the Council. 2006. Directive 2006/42/EC of the European Parliament and of the Council, vol. L 157/24.

Grover, C., Knight, I., Okoro, F., Simmons, I., Couper, G., Massie, P. and Smith, B. 2008. Automated Emergency Breaking Systems: Technical requirements, costs and benefits. Published project report. April 2008. 117 p. Available http://ec.europa.eu/enterprise/sectors/automotive/files/projects/report_aebs_en.pdf. Retrieved 26.2.2014.

Hanmer R. S. 2007. Patterns for fault tolerant software. John Wiley & Sons Ltd., Chichester. ISBN: 978-0-470-31979-6.

Kosinski, R. J. 2013. A literature review on reaction time, Clemson University. Available: http://biae.clemson.edu/bpc/bp/lab/110/reaction.htm Retrieved 11.2.2014.

Piampiano, J. M. & Rizzo, S. M. 2012. "Safe or Safe Enough? Measuring Risks & Its Variable Objectively", Professional Safety, vol. 57, no. 1, pp. 36-43.

Rauhamäki, J., Kuikka, S. 2013. Patterns for control system safety. In proceedings of EuroPLoP 2013.

Rauhamäki, J., Kuikka, S. 2014. Patterns for Hazard Mitigation Process. In proceedings of EuroPLoP 2014.

Publication VIII

Rauhamäki, J. (In press). Patterns for Functional Safety System Development. LNCS Transactions on Pattern Languages of Programming.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-3964-0
ISSN 1459-2045