Using UML Profiles for Sector-Specific Tailoring of Safety Evidence Information

Rajwinder Kaur Panesar-Walawege[1,2], Mehrdad Sabetzadeh[1], and
Lionel Briand[1,2]

[1]Simula Research Laboratory, Lysaker, Norway
[2]University of Oslo, Oslo, Norway
{rpanesar,mehrdad,briand}@simula.no

Abstract. Safety-critical systems are often subject to certification as a
way to ensure that the safety risks associated with their use are suffi-
ciently mitigated. A key requirement of certification is the provision of
evidence that a system complies with the applicable standards. The way
this is typically organized is to have a generic standard that sets forth the
general evidence requirements across different industry sectors, and then
to have a derived standard that specializes the generic standard accord-
ing to the needs of a specific industry sector. To demonstrate standards
compliance, one therefore needs to precisely specify how the evidence
requirements of a sector-specific standard map onto those of the generic
parent standard. Unfortunately, little research has been done to date on
capturing the relationship between generic and sector-specific standards
and a large fraction of the issues arising during certification can be traced
to poorly-stated or implicit relationships between a generic standard and
its sector-specific interpretation. In this paper, we propose an approach
based on UML profiles to systematically capture how the evidence re-
quirements of a generic standard are specialized in a particular domain.
To demonstrate our approach, we apply it for tailoring IEC61508 – one of
the most established standards for functional safety – to the Petroleum
industry.

Keywords: Safety Certification, UML Profiles, Evidence Information
Models, IEC61508.

1 Introduction

Safety-critical systems are typically subject to safety certification, whose aim
is to ensure that the safety risks associated with the use of such systems are
sufficiently mitigated and that the systems are deemed safe by a certification
body. A key requirement in safety certification is the provision of evidence that
a system complies with one or more applicable safety standards. A common
practice in defining standards for certification is to have a generic standard
and then derive from it sector-specific standards for every industry sector that
the generic standard applies to. The idea behind such a tiered approach is to
unify the commonalities across different sectors into the generic standard, and
then *specialize* the generic standard according to contextual needs. The generic
standard is sometimes referred to as a *metastandard* [19].

A notable example in safety certification is the specialization of IEC61508 [10] – a generic standard that deals with the functional safety of electrical / electronic / programmable electronic safety-critical systems. In the process industry, this standard is adapted as IEC61511 [9], in railways as EN 50129 [8], in the petroleum industry as OLF070 [5], and in the automotive industry as the forthcoming ISO 26262 [4].

For specialization to be effective, it is important to be able to precisely specify how the evidence requirements stated in a generic standard map onto those stated in a derived standard. Unfortunately, there has been little work to date on systematizing the specification of the relationship between generic and sector-specific standards. This has led to a number of problems. In particular, Feldt et. al. [11] cite the lack of agreed-upon relationships between generic and derived standards as one of the main reasons behind certification delays, caused by ambiguities in the relationships and the need for subjective interpretations by the certification body and system supplier. Furthermore, Nordland [12] notes the lack of a well-formulated process for showing that a derived standard is consistent with a generic standard. This too is directly attributable to the lack of precise and explicitly-defined relationships between the standards.

In this paper, we propose a novel approach based on UML profiles [3] to capture the relationship between the evidence requirements of a generic standard and those of a sector-specific derivation. Briefly, our approach works by (1) building conceptual models for the evidence requirements of both the generic and sector-specific standards, (2) turning the conceptual model of the generic standard into a profile, and (3) using the profile for stereotyping the elements in the conceptual model of the sector-specific standard. Our approach offers two main advantages: First, it provides a systematic and explicit way to keep track of the relationships between a generic and a derived standard in terms of their evidence requirements. And second, it enables the definition of consistency constraints to ensure that evidence requirements are being specialized properly in the derived standard.

While the overall ideas behind our approach are general, we ground our discussions on a particular safety standard, IEC61508, and a particular derivation, OLF070 (used in the petroleum industry). On the one hand, this addresses a specific observed need in safety certification of maritime and energy systems; and on the other hand, it provides us with a concrete context for describing the different steps of our approach and how these steps fit together. The conceptual model characterizing the IEC61508 evidence requirements has been described in our earlier work [18]. The one for OLF070 has been developed as part this current work. Excerpts from both conceptual models will be used for exemplification throughout the paper.

The remainder of this paper is structured as follows: In Section 2, we review background information for the paper. In Section 3, we describe our UML profile for IEC61508 and in Section 4 we discuss how the profile can be used for specialization of safety evidence. Section 5 compares our work with related work. Section 6 concludes the paper with a summary and suggestions for future work.

2 Background

In this section, we provide a brief introduction to safety certification (based on IEC61508), how safety evidence requirements can be structured through conceptual modeling, and UML profiles.

2.1 IEC61508-Based Certification

Safety-critical systems in many domains, e.g., the avionics, railways, and maritime and energy, are subject to certification. One of the most prominent standards used for this purpose is IEC61508. The standard sets forth the requirements for the development of electrical, electronic or programmable electronic systems containing safety critical components. This standard is concerned with a particular aspect of overall system safety, called functional safety, aimed at ensuring that a system or piece of equipment functions correctly in response to its inputs [10]. The standard defines requirements for hardware development, software development, and the development process that needs to be followed. The standard applies to systems with different required safety margins. This is encoded in the standard in the form of Safety Integrity Levels (SILs). The levels range from SIL 1 to SIL 4 and indicate the level of risk reduction measures that need to be in place based on the failure rate of the implementation and the acceptability of the risks involved. A number of sector-specific standards specialize IEC61508. These include IEC61511 in the process industry [9], EN 50129 [8] for railways, OLF070 [5] for the petroleum industry, and the upcoming ISO26262 [4] for the automotive industry.

2.2 Conceptual Modeling Of Compliance Evidence Information

In general, standards, irrespective of the domains they are targeted at, tend to be expressed as textual requirements. Since the requirements are expressed in natural language, they are subject to interpretation by the users of the standards. To make the interpretation explicit and develop a common understanding, we develop a conceptual model that formalizes the evidence requirements of a given standard. Such a model can be conveniently expressed in the UML class diagram notation [3].

For illustration, we show in Fig. 1 a small fragment of the conceptual model that we have built in our previous work on IEC61508 [18]. Concepts are represented as classes and concept attributes – as class attributes. Relationships are represented by associations. Generalization associations are used to derive more specific concepts from abstract ones. When an attribute assumes a value from a predefined set of possible values, we use enumerations. Finally, we use the package notation to make groupings of concepts and thus better manage the complexity.

The diagram in Fig. 1 presents the concepts for describing the development process, packaged as `Process Concepts`, and how these relate to concepts in the `Issue Concepts`, `Artifact Concepts` and `Requirements Concepts` packages. From

these other packages, we show only the concepts that related to those in `Process Concepts`. The central concept in the diagram of Fig. 1 is the notion of `Activity`, representing a unit of behavior with specific input and output. An activity can be further decomposed into sub-activities. A (life-cycle) phase is made up of a set of activities that are carried out during the lifetime of a system. Each activity utilizes certain techniques to arrive at its desired output, given its input. The selection of techniques is related to the safety integrity level that needs to be achieved. For example, if the activity in question concerns software verification, constructing formal proofs of correctness is usually unnecessary for low integrity levels, whereas, formal proofs are highly recommended for the highest integrity level. Each activity requires certain kind of competence by the agents performing it. The agent itself can be either an individual person or an organization. In either case, the agent is identified by the type of role it plays, for example the agent may be the supplier of a system or the operator. Agents can be made responsible for certain development artifacts. Further detail about the other packages shown can be found in [18].
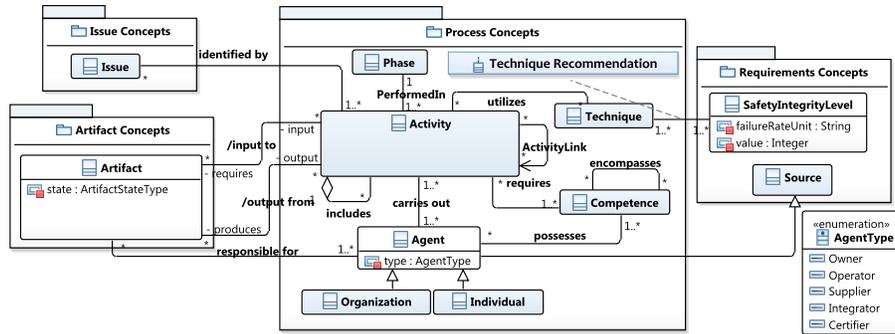

Fig. 1. IEC61508 Process Concepts and Their Links

2.3 UML Profiles

UML profiles [3] aim at providing a lightweight solution for tailoring the UML metamodel for a specific domain. The same mechanisms used by UML profiles for tailoring the UML metamodel can also be effectively used for tailoring standards compliance evidence according to domain-specific needs.

Briefly, UML profiles enable the expression of new terminology, notation and constraints by the introduction of context-specific stereotypes, attributes and constraints. Stereotypes are a means of extending a base metaclass. We extend the `Class`, `Property` and `Association` metaclasses, creating stereotypes for the concepts, their attributes and their relationships respectively. Moreover, constraints can be defined in a profile by using the Object Constraint Language (OCL) [13] to ensure that certain semantics are maintained in the new models to which the profile is applied. By using profiles the new models that employ the profile are still consistent with the UML metamodel.

As we describe in the subsequent sections, we use this mechanism to create a profile of the IEC61508 conceptual model (Section 3) and then use it to specialize the IEC61508 standard for the petroleum industry (Section 4).

3 UML Profile of the IEC61508 Standard

Our approach for specializing a generic standard is through the use of a UML profile. In Fig. 2, we show the methodology we propose for this purpose. The methodology consists of four main steps: (1) creating a conceptual model of the generic standard, we do this using a UML class diagram; (2) creating a UML profile based on the generic conceptual model; (3) creating a conceptual model of the sector-specific standard and applying the stereotypes from the UML profile of the generic standard; and (4) validating the OCL constraints of the profile over the sector-specific conceptual model to ensure that it is consistent with the generic standard. We apply this methodology for specializing the generic IEC61508 standard to the OLF070 standard for the petroleum industry.
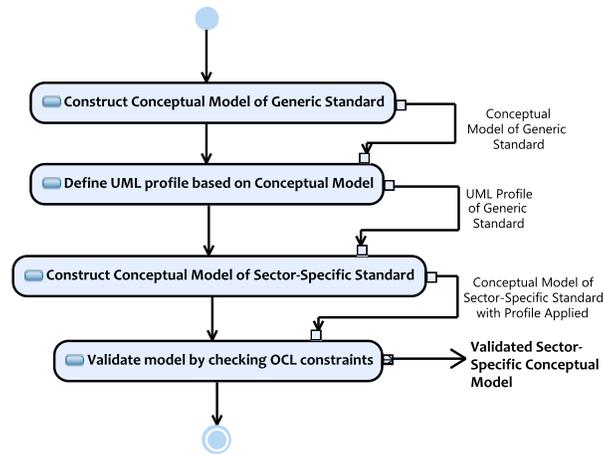

Fig. 2. The Methodology for Specialization of a Generic Standard.

Using profiles for specialization offers the following key advantages:

– We can incorporate the specific terminology used by a generic standard and still allow the use of context-specific terminology. For example, in IEC61508, we have the general concept of `ProgrammableElectronicSystem` (`PES`). OLF070 instead refers to very specific types of `PES`s in the petroleum industry, e.g., Fire and Gas system (F&G), Process Shut-Down system (PSD), Emergency Shut-Down system (ESD). These sector-specific concepts can all be stereotyped as `ProgrammableElectronicSystem` to capture the correspondence. It is of course possible to directly extend the conceptual model of a generic standard for a specific domain by adding new elements to it. However, this makes it hard to keep track of which concepts are from the generic standard and which are from the sector-specific one. When a profile is used, all the stereotypes are known to be from the generic standard, hence a clear distinction is made between the terminologies.
– Stereotypes establish an explicit and rigorous mapping between the generic and sector-specific standards. This mapping can be used to ensure that, for a specific project, all the necessary evidence for demonstrating compliance has

been collected. Further, the existence of such an explicit mapping makes it possible to define pairwise consistency rules between the generic and derived standards (using UML's rich constraint language, OCL), and to provide guidance to the users about how to resolve any inconsistencies detected.

As shown in Fig. 2, the basis of our profile of IEC61508 is the conceptual model of the IEC61508 standard. The process of creating a conceptual model of the evidence requirements of a given standard involves a careful analysis of the text of the standard. It requires skills in modelling, systems development and knowledge of the process of certification beyond merely reading the standard. To some extent, this can be viewed as a process of qualitative data analysis, where the data is the text of the standard and it is being analysed to identify from it, all the salient concepts and their relationships. This retrieved information from the text is used create a common understanding of the standard and as a means of explicitly showing the relationships that exist between the salient concepts.

We exemplify the process of creating the conceptual model of IEC61508 by showing an excerpt of the standard, and the concepts and relationships that have been gleaned from the excerpt. Fig. 3 shows a section of the IEC61508 standard that is dedicated to requirements applicable to the software of a safety-related system. In Fig. 3, we can see the salient concepts and relationships identified in the text - these have been highlighted by enclosing the relevant text in a box and numbering the identified section. Box 1 shows that the concepts `Phase` and `Activity` are of importance during the software development lifecycle (in Fig. 3 we have used the concept names shown earlier in Fig. 1). Box 2 identifies some key relationships between phases and activities. An activity is performed during a phase and has specified inputs and outputs. Box 3 indicates that a generic life cycle is prescribed by the standard while not precluding deviations in terms of phases and activities. Box 4 presents the concepts: technique, safety integrity level and techniques recommendation - indicating that activities utilize certain techniques based on the safety integrity level. The same concepts and relationships may be found in several places in the standard. Once the text has been marked up in this manner, a glossary is created to ensure that consistent terms are used to refer to the same concepts and relationships. A part of this glossary, describing the most important concepts is shown in Table 1. The conceptual model is created from this set of concepts and their relationship and serves as the metamodel of the profile.

Fig. 4 shows a bird-eye's view of the different packages that make up the metamodel for our IEC61508 UML profile. The packages contain abstractions for modelling of the main concepts of IEC61508. We briefly explain each package. For more details, see [18]. The `System Concepts` package describes the breakdown of the system at a high level; the `Hazard Concepts` package contains the abstraction for describing the hazards and risks for the system; the `Requirements Concepts` package for the different types of requirements, including safety requirements; the `Process Concepts` package for describing the development process (details given in Section 2.2); the `Artifact Concepts` package for describing the different types of artifacts created as supporting evidence; the `Guidance` package for describing
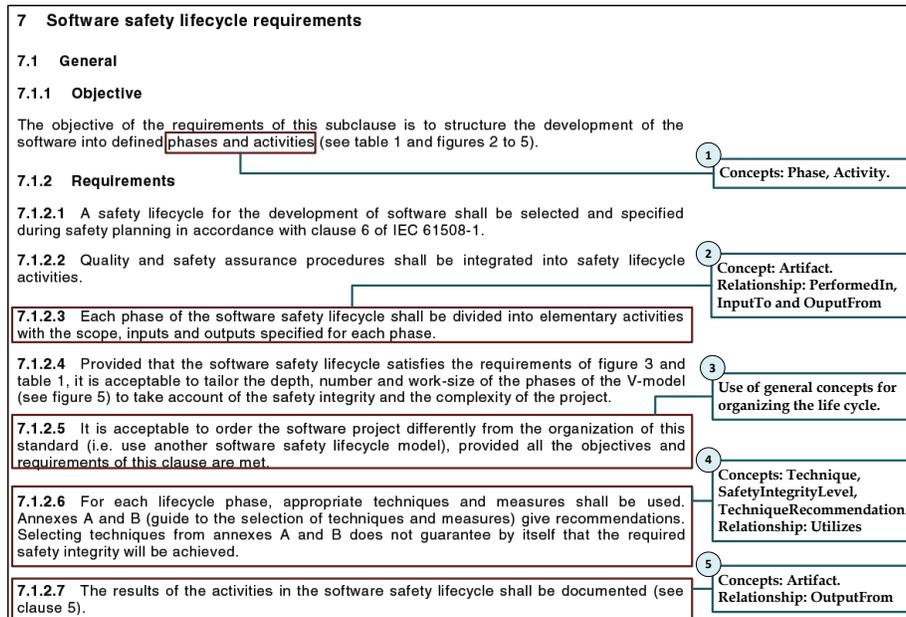
Fig. 3. An Excerpt of IEC61508 showing the textual source of some of the Process elements

the other standards and recommended practices that will be used to develop the system, the `Issue Concepts` package for describing the defects or enhancements that may have given rise to changes; the `Configuration Management Concepts` package for describing the unique versions for all the components that make up the system, the `Justification Concepts` package to capture the assumptions and rationale behind the various decisions that are made during development; and the `Domain-Specific Concepts` package for capturing the enumerations for concept attributes in other packages (e.g., requirement type, system operating mode). The elements of the conceptual model are mapped almost directly into the profile. The concepts become stereotypes that extend the metaclass `Class`, the relationships become stereotypes that extend the metaclass `Association` and the attributes of these two extend the metaclass `Property`.

Table 1. Description of Main Concepts from the IEC61508 Metamodel

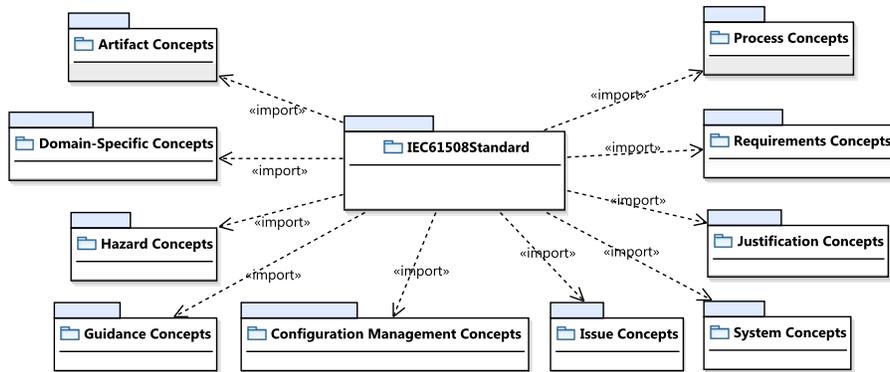| Stereotype	Description
Activity	A unit of behaviour in a process.
Agent	A person or organization that has the capability and responsibility for carrying out an activity.
Artifact	One of the many kinds of tangible by-products produced during the development of a system.
Assumption	A premise that is not under the control of the system of interest, and is accepted as true without a thorough examination. Assumptions can, among other things, be related to the environment of the system, the users, and external regulations.
Block	Entity of hardware or software, or both, capable of accomplishing a specified purpose.
Change	A modification made to the PES, Block or Artifact.
Competence	The ability to perform a specific task, action or function successfully.
Continued on next page ...	

Fig. 4. Packages of the IEC61508 Metamodel

Continued from previous page ...	
Stereotype	**Description**
ControlledItem	A PES, Block or Artifact for which meaningful increments of change are documented and recorded.
Defect	An error, failure, or fault in a system that produces an incorrect or unexpected result, or causes it to behave in unintended ways.
Description	A planned or actual function, design, performance or activity (e.g., function description).
DesignatedState	The state of the EUC related to safety, the EUC is either in a safe state or an unsafe state.
Diagram	Specification of a function by means of a diagram (symbols and lines).
Enhancement	Provision of improved, advanced, or sophisticated features.
Error	Discrepancy between a computed, observed or measured value or condition and the true, specified or theoretically correct value or condition.
Event	A single occurence in a series of occurences that cause a hazard to occur.
Failure	Termination of the ability of a functional unit to perform a required function.
Fault	Abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function.
GeneralStandard	A standard that provides generic recommendations on a specific subject to a number of related domains.
HardwareBlock	Any entity of hardware – this may be mechanical, electrical or electronic that is used in the composition of the system.
HazardousElement	The basic hazardous resource creating the impetus for the hazard, such as a hazardous energy source such as explosives being used in the system.
Hazard	Any real or potential condition that can cause injury, illness, or death to personnel damage to or loss of a system, equipment or property or damage to the environment.
Individual	Refers to a person.
InitiatingMechanism	The trigger or initiator event(s) causing the hazard to occur. The IM causes actualization or transformation of the hazard from a dormant state to an active mishap state.
Instruction	Specifies in detail the instructions as to when and how to perform certain jobs (for example operator instruction).
Interface	An abstraction that a block provides of itself to the outside. This separates the methods of external communication from internal operation.
Issue	A unit of work to accomplish an improvement in a system.
List	Information in a list form (e.g., code list, signal list).
Log	Information on events in a chronological log form.
Mistake	Human action or inaction that can produce an unintended result.
NonProgrammable-HardwareBlock	Electro-mechanical devices (electrical) solid-state non-programmable electronic devices (electronic).
Continued on next page ...	

Stereotype	Description
Continued from previous page ...	
OperatingMode	The different modes that a system can be operating in, e.g. normal, maintenance, test, emergency.
Organization	A social arrangement which pursues collective goals, which controls its own performance, and which has a boundary separating it from its environment.
Phase	A set of activities with determined inputs and output that are carried out at a specific time during the life of a system.
Plan	Explanation of when, how and by whom specific activities shall be performed (e.g., maintenance plan).
Programmable-ElectronicSystem	System for control, protection or monitoring based on one or more programmable electronic devices, including all elements of the system such as power supplies, sensors and other input devices, data highways and other communication paths, and actuators and other output devices.
Programmable-HardwareBlock	Any physical entity based on computer technology which may be comprised of hardware, software, and of input and/or output units.
Rationale	The fundamental reason or reasons serving to account for something.
RecommendedPractice	Sound practices and guidance for the achievement of a particular objective.
Report	The results of activities such as investigations, assessments, tests etc. (e.g., test report).
Request	A description of requested actions that have to be approved and further specified (e.g., maintenance request).
Requirement	A necessary attribute in a system; a statement that identifies a capability, characteristic, or quality factor of a system in order for it to have value and utility to a user.
ResidualRisk	Risk remaining after protective measures have been taken.
Risk	Combination of the probability of occurrence of harm and the severity of that harm.
SafeState	The state of the EUC when safety is achieved.
SafetyIntegrity-Level	The probability of a safety-related system satisfactorily performing the required safety functions under all the stated conditions within a stated period of time.
SafetyRequirement	A prescriptive statement that ensures that the system carries out its functions in an acceptably safe manner.
SectorSpecific-Standard	A standard that provides recommendations for a specific industrial sector (e.g., the energy sector).
SoftwareBlock	Any entity of software that may be used for controlling the system – this may be embedded or application software or even different levels of software such as module, component, subsystem, system.
SoftwareLevel	The different levels into which a software system can be decomposed, e.g. System, subsystem, component and module.
Source	An abstract concept that can represent a person, organization or standard that can be a source of requirements to a system.
Specification	Description of a required function, performance or activity (e.g., requirements specification).
Standard	An established norm or requirement, typically provided as a formal document that establishes uniform engineering or technical criteria, methods, processes and practices.
Technique-Recommendation	A particular technique recommended based on the safety integrity level of the requirements that have been allocated to the block in question.
Technique	A procedure used to accomplish a specific activity or task.
UnsafeState	The state of the EUC when safety is compromised.
UserRole	An aspect of the interaction between a PES and the human elements.

Our IEC61508 profile consists of:
- 57 stereotypes that extend the metaclass Class, used to characterize the evidence elements
- 53 stereotypes that extend the metaclass Association, used to characterize the traceability links amongst the various evidence elements.
- 6 stereotypes extend the metaclass Property, used on the role names of the corresponding associations.

Besides these stereotypes, stereotypes extending the Class and Association metaclasses have OCL constraints to ensure they are used consistently. We will discuss these constraints and provide examples later in this section.
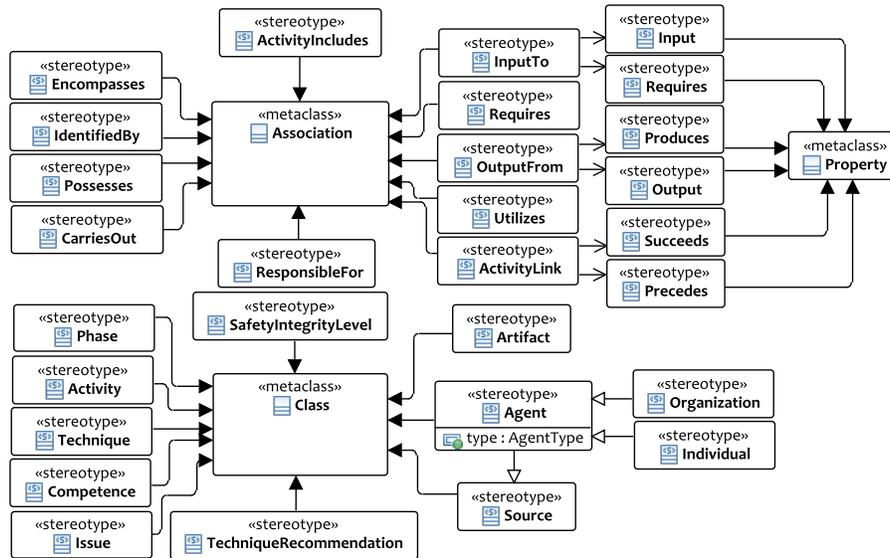
Fig. 5. IEC61508 Profile Fragment for the System Development Process

Since the profile is quite large and cannot be fully explained in this paper, as an example, in Fig. 5, we show the stereotypes created to manage the development process. These are the stereotypes derived from the partial conceptual model shown in Fig. 1. The IEC61508 standard does not mandate a specific development life-cycle such as the waterfall or iterative lifecycle; it does however state that a number of specific activities should be carried out. We have the stereotype `Activity` to model this. An `Activity` can itself include other sub activities and this is modelled by the association stereotype `ActivityIncludes`. Certain activities may precede or succeed others and this is modelled via the association stereotype `ActivityLink` along with its properties `Precedes` and `Succeeds`.

In safety-critical systems, it is very important to ensure that all work is carried out by personnel with the required knowledge and skills. IEC61508 mandates that this information be part of the compliance evidence. Hence, for each activity, we model both the required competence and that of the agent performing the activity via the stereotypes `Agent` and `Competence` along with `CarriesOut`, `Requires` and `Possesses`. An activity may have certain artifacts that are needed in order to carry it out and it will produce certain artifacts upon its completion. These concepts are modelled using the stereotypes `Artifact`, `InputTo`, `OutputFrom`, `Requires`, `Produces`, `Input` and `Output`. Finally, each activity will use certain techniques to create its output. These techniques are chosen based on the level of safety required and hence we have the stereotypes `Technique` and `TechniqueRecommendation`.

As stated earlier, there are OCL constraints for the class and association stereotypes. These constraint enforce the structural consistency of the evidence information in the sector-specific derivations. Specifically, for any association stereotyped with X, we must check that the endpoints of the association are

stereotyped correctly according to the endpoints of X in the profile metamodel. For example, consider the `CarriesOut` stereotype. We need a constraint to ensure that any association with this stereotype connects two elements stereotyped `Agent` and `Activity`, respectively. This constraint is shown in Table 2. A similar constraint is shown for `OutputFrom`, to ensure that any association having this stereotype has endpoints that are stereotyped `Artifact` and `Activity`.

For stereotypes extending the `Class` metaclass, we need to verify that any stereotyped element respects the multiplicity constraints of the profile metamodel. We show an example in Table 2: we have constraints to ensure that an element with the `Activity` stereotype is linked to at least one element with the `Artifact` stereotype and at least one element with the `Agent` stereotype.

The profile only needs to be created once per standard, and then can be reused for specializing the generic standard to any number of domains. Once the profile is created, the stereotypes of the profile are applied to the conceptual model of the domain-specific standard, also expressed as a UML class diagram. For the derived standard there are three things to bear in mind to ensure its consistency with the generic standard: (1) which concepts will be used directly from the generic standard (possibly with different terminology), (2) which concepts are specific to the domain and thus new, and (3) which concepts, from the generic standard, have been deliberately left out as they may not be applicable to the domain, in which case this omission is clearly noted and explained. The conceptual model of the derived standard is created in a manner similar to the generic standard, except that, the profile stereotypes are applied and the OCL constraints are checked to enforce the semantics of the specialization and guide the user in creating a structurally sound information model for a derived standard.

Table 2. OCL Constraints on Stereotypes

Stereotype	Constraint
CarriesOut	```self.base_Association.memberEnd->` ` select(p:Property	not (p.class.getAppliedStereotype(` ` 'IEC61508Profile::Activity').oclIsUndefined()))->size()=1` `and` `self.base_Association.memberEnd->` ` select(p:Property	not (p.class.getAppliedStereotype(` ` 'IEC61508Profile::Agent').oclIsUndefined()))->size()=1```
OutputFrom	```self.base_Association.memberEnd->` ` select(p:Property| not (p.class.getAppliedStereotype(` ` 'IEC61508Profile::Activity').oclIsUndefined()))->size()=1` `and` `self.base_Association.memberEnd->` ` select(p:Property| not (p.class.getAppliedStereotype(` ` 'IEC61508Profile::Artifact').oclIsUndefined()))->size()=1```
Activity	```1: self.base_Class.ownedAttribute->collect(` ` c:Property | c.association)->select(` ` a:Association	not a.getAppliedStereotype(` ` 'IEC61508Profile::OutputFrom').oclIsUndefined())->size()>0` `2: self.base_Class.ownedAttribute->collect(` ` c:Property|c.association)->select(` ` a:Association	not a.getAppliedStereotype(` ` 'IEC61508Profile::CarriesOut').oclIsUndefined())->size()>0```

4 Specializing IEC61508 for the Petroleum Industry

OLF070 is a derivation of IEC61508, elaborating the safety concerns that are specific to control systems in the petroleum industry. We discuss at a high level how OLF070 refines IEC61508. Recall the packages shown in Fig. 4: the `Artifact Concepts`, the `Configuration Management Concepts`, the `Issue Concepts`, the `Guidance`, and the `Justification Concepts` are the same in OLF070 as in IEC61508. The `Hazard Concepts` are the same, apart from the fact that in OLF070, the most common hazards have been defined in the standard already. The change in the `System Concepts` is that in addition to specifying the breakdown of the system, a particular component can be specified as either being part of a local safety function (e.g., process shutdown) or a global safety function (e.g., emergency shutdown). The `Requirements Concepts` specify that the SIL level of most common components can be obtained from a table provided in the standard unless there is a deviation in the component from what is described in the standard, in which case the SIL level is calculated using the procedures specified by IEC61508. The `Process Concepts` and the `Domain-Specific Concepts` are different in that there are specific processes and specific terminology used in the petroleum industry for developing the systems. In this section, we illustrate the specialization process by showing how the profile described in the previous section can be used for tailoring the evidence required by the OLF070 standard [5].

To preserve the continuity of our examples from the previous section, we focus on the development process aspects of OLF070, and more precisely on one of the phases envisaged in the standard, called the Pre-Execution Phase. This phase is concerned with developing a Plan for Development and Operation (PDO) of an oilfield. The PDO contains the details of all the systems that need to be created to make the oilfield functional. The phase ends with the creation of the PDO document that is then sent to the authorities to get permission for the project and used to select the main engineering contractor. In this phase, a number of activities are carried out: (1) all the equipment to be installed at the field and all the safety instruments systems (SIS) are defined; (2) hazards are identified; (3) a risk analysis is performed to gauge the extent of the risks that need to be mitigated; (4) safety functions (such as fire detection, gas detection, process shut-down) and the safety integrity levels are specified based on the results of the risk analysis.

In Fig. 6, we present a small excerpt of the OLF070 conceptual model and show the concepts we have just described as the different activities that take place during the Pre-Execution Phase. The stereotypes from our IEC61508 profile have already been applied. The phase is documented in the artifact called `PlanForDevelopmentAndOperation`. This is in compliance with IEC61508, whereby each phase should have a plan documenting it. For some of the activities, we show the relevant inputs and outputs and the agents that need to perform them. We use the stereotypes from our IEC61508 profile to show how this OLF070 model excerpt relates to IEC61508. Some of the stereotype we have already explained in Section 3. The four new ones here are `DocumentedIn` for the result of a phase, `BasedOn` to show whether an artifact is based on a standard, `Standard` to indicate
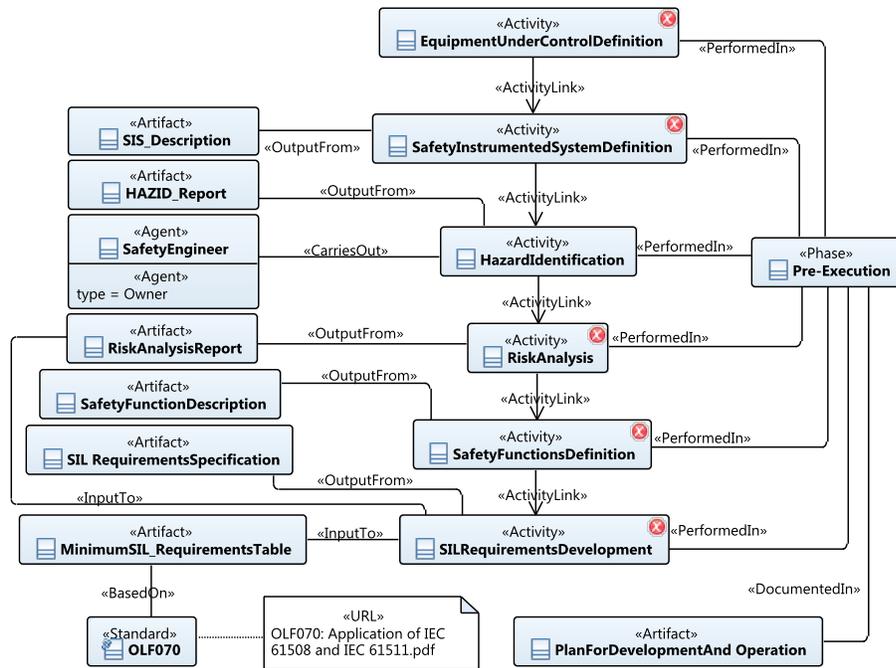
Fig. 6. An example phase from OLF070

a type of material used to create an artifact and `PerformedIn` for indicating which phase an activity is performed in. Note that stereotypes can have attributes, e.g the attribute `type` for the stereotype `Agent`, shown in Fig. 6, has the value `Owner` to indicate that the Safety Engineer is employed or commissioned by the owner of the system to be developed. For linking to artifacts and facilitating navigation to them, we can include URLs and file references in the conceptual model. An example is shown in the figure, where we link the OLF070 element to the actual document for the standard.

As discussed in Section 3, we use OCL constraints for enforcing consistent use of the profile. Once the stereotypes have been applied to the modelled elements, we can validate the model using an OCL checker, e.g. the Rational Software Architect OCL tool [1] that we use here. In Fig. 6, we can see that five of the elements have a red cross in their upper right-hand corner. These element have failed the OCL validation. The errors generated are shown in Fig. 7. The first five errors concern the constraint that an activity should have an agent performing it. The model elements `EquipmentUnderControlDefinition`, `SafetyInstrumentedSystemDefinition`, `RiskAnalysis`, `SafetyFunctionsDefinition`, and `SILRequirementsDevelopment` do not have a corresponding agent element. For `EquipmentUnderControlDefinition`, a further constraint has been violated: there is no output specified from that activity, indicated by the last error in the snap-

shot of Fig. 7. Thus, in addition to providing a means to explicitly show the relationships between the generic and sector-specific standard, the profile enables users to check whether the requirements of the generic standard are maintained in the sector-specific one.

Fig. 7. Error Report showing violated OCL Constraints

5   Related Work

Using UML profiles to adapt UML to a specific context is very common. The Object Management Group have so far standardized three profiles: the UML Profile for Modeling and Analysis of Real-time and Embedded Systems (MARTE) [16], the UML Profile for Modeling QoS and Fault Tolerance Characteristics and Mechanisms (QFTP) [15], and the UML Profile for Schedulability, Performance and Time (SPT) [14]. All three include safety-relevant concepts. However, in contrast to our work, none of these were designed for characterizing the evidence required for compliance to safety standards.

Zoughbi et. al. [20] propose a UML profile for the RTCA DO-178B standard[2] used in commercial and military aerospace software. This profile enables software engineers to directly add certification information to software models. The concepts modeled are targeted at addressing a major requirement of RTCA DO-178B having to do with traceability between requirements and design and eventually code. This information together with evidence of other quality assurance activities would form the basis of full compliance to the standard. The approach we propose in this paper differs from [20] in the following ways: Firstly, we focus on a different and broader standard; secondly, our profile includes a wide range of concepts related to the management of the development process in safety-critical systems, whereas [20] deals primarily with requirements and design; and thirdly and most importantly, we use profiles as a basis for sector-specific specialization – specialization is not tackled in [20].

The Software Assurance Evidence Metamodel (SAEM) [17] is a proposal from the OMG, concerned with managing assurance evidence information. A main distinction between our work and SAEM is that we aim at characterizing the evidence that needs to be collected for certification based on a standard. Instead, SAEM is standard-independent and mainly directed towards linking the evidence to claims and the evaluation of the claims in light of the evidence. An abstract specification of evidence such as the one given by SAEM will therefore need to be complemented with an evidence conceptual model for a specific standard, e.g.,

our IEC61508 conceptual model. Indeed, just as we use profiles for specializing IEC61508 for a specific sector, one can use profiles to incorporate SAEM into the conceptual model of a given standard and create a metamodel that captures both the evidence requirements for compliance, and also the evaluation of whether the evidence is sufficient to substantiate the claims.

Chung et. al. [7] study the problem of compliance of a user-defined workflow with the activities envisaged in IEC61508. Their approach is to check (process) compliance by comparing user-defined activities in an organization against models of the activities in the standard. Our work is close to [7] in its goal to model compliance information; however, we go beyond the process aspects of IEC61508 and provide an evidence information model for the entire IEC61508, which can in turn be specialized to sector-specific needs through the use of profiles.

6 Conclusion and Future Work

In this paper we presented a methodology for ensuring that a generic standard can be specialized in a systematic manner for a particular domain. We do this by capturing the generic standard as a conceptual model using a UML class diagram and use this as a basis for creating a UML profile. The profile is then applied to the conceptual model of a sector-specific standard and used as an explicit means of keeping track of the relationships between the two. We exemplify our methodology by showing excerpts of the IEC61508 conceptual model that we have created, the UML profile based on this model and how we apply this profile to a conceptual model of the OLF070 standard which is a sector-specific derivation of IEC61508 for the petroleum industry.

Our approach offers two key benefits: (1) It incorporates the specific concepts used by a generic standard into the sector-specific standard whilst making a clear distinction between the two; and (2) It explicitly captures the mapping between two standards and defines consistency rules between them, which can be automatically verified and used for providing guidance to the users about how to resolve any inconsistencies.

Having established a means to capture the evidence required for a specific standard, we are now working on a means to create instantiations of these conceptual models such that we can create repositories of evidence for safety certification. Subsequently, we plan to carry out case studies to assess the cost-effectiveness of our methodology in the context of certification. Another prime concern is the ability to certify a system to multiple and often overlapping standards. For example, in the petroleum industry, it is quite common to certify a system to both OLF070 and to one of the NORSOK standards such as the NORSOK I-002 for Safety Automation Systems [6]. In future work, we plan to extend our methodology so that we can express how a repository of evidence information addresses each standard in a collection of inter-related standards. Finally, to aid the certification process from the perspective of a certification body, we would like to extend our work to the evaluation of evidence as proposed by the SAEM. This would lay the groundwork for a complete certification infrastructure based on industry standards.

References

1. IBM Rational Software Architect. `http://www.ibm.com/developerworks/rational/products/rsa/`.
2. DO-178B: Software considerations in airborne systems and equipment certification, 1982.
3. UML 2.0 Superstructure Specification, August 2005.
4. Road vehicles – functional safety. ISO draft standard, 2009.
5. The Norwegian Oil Industry Association. Application of IEC61508 and IEC61511 in the Norwegian Petroleum Industry, 2004.
6. Norwegian Technology Centre. Safety and automation system (SAS), 2001.
7. P. Chung, L. Cheung, and C. Machin. Compliance flow - managing the compliance of dynamic and complex processes. *Knowledge-Based Systems*, 21(4):332–354, 2008.
8. International Electrotechnical Commission. Railway Applications Safety-related electronic railway control and protection systems., 1999.
9. International Electrotechnical Commission. Functional safety - safety instrumented systems for the process industry sector(IEC 61511)., 2003.
10. International Electrotechnical Commission. Functional safety of electrical / electronic / programmable electronic safety-related systems (IEC 61508), 2005.
11. R. Feldt, R. Torkar, E. Ahmad, and B. Raza. Challenges with software verification and validation activities in the space industry. In *ICST'10*, pages 225–234, 2010.
12. O. Nordland. A critical look at the cenelec railway application standards. `http://home.c2i.net/odd_nordland/~SINTEF/tekster/A_critical_look_at_rail_standards.htm`, 2003.
13. Object Management Group (OMG). OMG object constraint language. http://www.omg.org/spec/OCL/2.0/, 2006.
14. Object Management Group (OMG). UML profile for schedulability, performance and time. http://www.omg.org/spec/SPTP/, 2006.
15. Object Management Group (OMG). UML profile for modeling quality of service and fault tolerance characteristics and mechanisms specification. http://www.omg.org/spec/QFTP/1.1/, 2008.
16. Object Management Group (OMG). UML profile for modeling and analysis of real-time and embedded systems (marte). http://www.omg.org/spec/MARTE/1.0/, 2009.
17. Object Management Group (OMG). Software Assurance Evidence Metamodel (SAEM). http://www.omg.org/spec/SAEM/, 2010.
18. R.K. Panesar-Walawege, M. Sabetzadeh, L. Briand, and T. Coq. Characterizing the chain of evidence for software safety cases: A conceptual model based on the IEC 61508 standard. In *ICST'10*, pages 335–344, 2010.
19. M. Uzumeri. Iso 9000 and other metastandards: Principles for management practice? *Academy of Management Executive*, 11, 1997.
20. G. Zoughbi, L. Briand, and Y. Labiche. Modeling safety and airworthiness (RTCA DO-178B) information: conceptual model and uml profile. *Software and Systems Modeling*, pages 1–31, 2010.