RESEARCH REPORT

VTT-R-06629-13

Systems Engineering Artefact Model – SEAModel

Authors: Jarmo Alanen, Peetu Valkama

Confidentiality: Project confidential (will be public after the project)

Report's title Systems Engineering Artefact model – SEAModel	
Customer, contact person, address Tekes, Matti Säynätjoki Kyllikinportti 2, P.O. Box 69, FI-00101 Helsinki, FINLAND	Order reference Tekes: 40204/12
Project name Computational methods in mechanical engineering product development	Project number/Short name 78634/SIMPRO
Author(s) Jarmo Alanen, Peetu Valkama	Pages 42
Keywords Systems Engineering, Data model	Report identification code VTT-R-06629-13

Summary

A reference model for systems engineering main artefact types was created. The model is called SEAModel (Systems Artefacts Engineering Model). The model consists of five model packages, System, Requirement, Behaviour, Structure and Specialty Engineering packages.

System package consists of System-of-interest concept model, System context concept model and System element type concept model.

Requirement package consists of Requirement concept model.

Behaviour package consists of Behaviour view concept model and System function concept model.

Structure package consists of Structure view concept model and Network concept model (CANopen case).

Specialty engineering package consists of Risk Assessment concept model.

For each of these models a set of artefact types and their relations are presented as SysML block definition diagrams. Furthermore, each of the artefact types are described in more detailed in tabular form. Artefact type attributes are not listed.

A short discussion on using requirement templates is provided. Furthermore, discussion on applying SEAModel in case of virtual engineering is supplied.

Confidentiality	Project confidential (will be public after the project)

Tampere 23.10.2013

Written by	Reviewed by	Accepted by
Jarmo Alanen, Senior Scientist	Juha Kortelainen, Principal Scientist	Riikka Virkkunen, Technology Manager

VTT's contact address

VTT Technical Research Centre of Finland, P.O. Box 1300, FI-33101 Tampere, Finland

Distribution (customer and VTT)

Tekes/Matti Säynätjoki, 1 original
VTT/archive, 1 original

Preface

This report was created as a result of SIMPRO VTT subproject Task 3.1 (Fixing the data model for simulation based system development to report the Systems Engineering Artefact Model (SEAModel) that was designed for management of engineering data, especially traceability of such data.

SIMPRO (Computational methods in mechanical engineering product development) is funded by Tekes – the Finnish Funding Agency for Technology and Innovation.

We thank Ali Muhammad for support in analysing the simulation support of SEAModel.

Tampere 23.10.2013

Authors

Contents

1. Introduction

A project called TIKOSU was carried out by VTT and Aalto University during the years 2009–2011. In the project, a set of concept models for engineering artefacts of programmable control systems of machinery were created and demonstrated. The aim of the TIKOSU model was to provide a data model that can be implemented by database oriented tools, like Application Lifecycle Management (ALM) or Product Lifecycle Management (PLM) tools. TIKOSU was created to provide a systematic set of engineering artefacts and their traceability.

The TIKOSU model is currently used in various customer projects carried out by VTT. In the SIMPRO project[1], the scope of the work goes beyond programmable control systems. Hence there is a need to enhance the TIKOSU model to also support design and simulation of mechanical systems.

1.1 Terms and abbreviations

Term/abbreviation	Description
ALM	Application Lifecycle Management
CAN	Controller Area Network
CANopen	An application layer protocol for CAN. Specified by CiA (CAN in Automation)
HAZOP	Hazard and operability study
PECS	Programmable Electronic Control System
PLC	Programmable Logic Controller
PLM	Product Lifecycle Management
SDO	Service Data Object (A CANopen term)
SW	Software
SysML	Systems Modeling Language

2. Goal

The goal of the work was to enhance the TIKOSU model to accommodate mechanical systems. The original TIKOSU model was focused on programmable electronic control systems. The main motivation for the model was the need for systematic set of engineering artefacts and their traceability.

3. Revisiting the TIKOSU model

The TIKOSU model published in (Alanen, Vidberg et al. 2011) consists of the following parts:

- System and system context concept model
- Requirement concept model
- Risk assessment concept model
- Behaviour concept model
- Structure concept model
- System function concept model
- Document concept model
- Network concept model.

The aim of the models was to systematically define the artefacts created during the engineering processes. This means a systematic set of artefact types, systematic meta-data (i.e. attributes) of the artefact types, and systematic set of relations between the artefacts. Well defined relations between

[1] See details at www.hankegalleria.fi → search SIMPRO.

the artefacts are a necessity to provide traceability between the engineering artefacts. Furthermore, well defined work flows to control the status of the artefacts were defined for the risk assessment part of the model. The TIKOSU model was demonstrated by implementing it onto two platforms, a cross-relational database (MySQL) and a commercial Application Lifecycle Management (ALM) tool (Polarion ALM). Integrations to different system engineering tools were created, e.g. to Vertex ED (electrical CAD) and CoDeSys (PLC programming tool by 3S-Smart Software Solutions). Parts of the model have been applied in industrial customer cases.

4. SEAModel

During the SIMPRO project, the TIKOSU model was updated to cover mechanical systems in order to support traceability between the system requirements and virtual engineering artefacts, like CAD models and simulation models and results. The updated model is called Systems Engineering Artefacts Model (SEAModel). It defines the key artefacts of a system and the relations between the artefacts. Hence it is a data model, not a process model, neither a model of any particular system.

The main motivation for SEAModel is the need to provide traceability between different artefacts, especially from requirements to design and implementation, from implementation to test execution, and from test execution to verification and validation reporting. Another motivation is to provide a model that can easily be implemented onto a relational database, onto a Product Lifecycle Management (PLM) tool or onto an Application Lifecycle Management (ALM) tool or whichever tool that provides management of structured data.

SEAModel consists – at the time of writing – of five packages (see Figure 1).

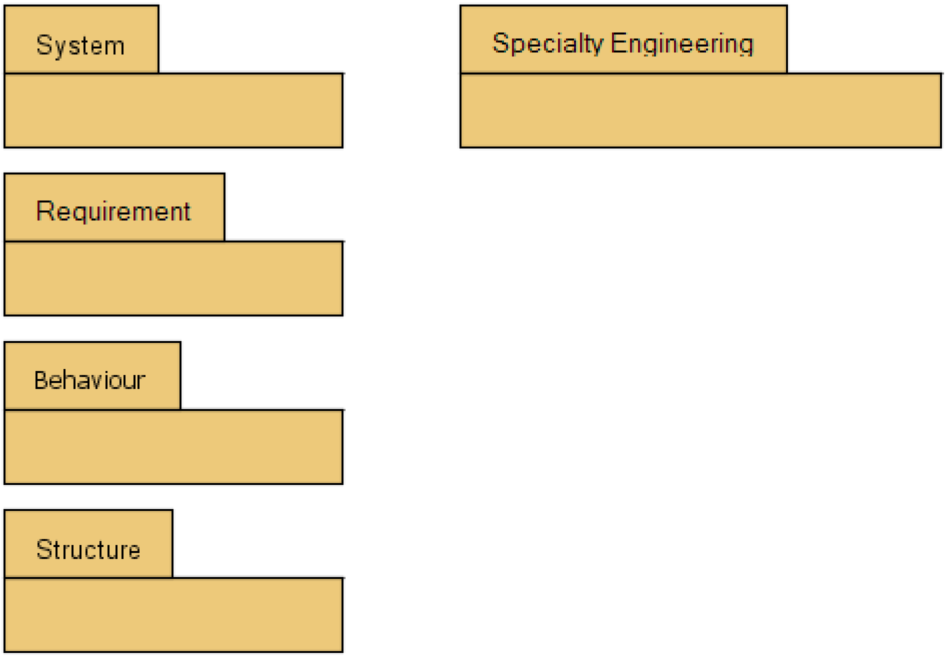

Figure 1. SEAModel packages.

The packages are described in the following sections. The notation for the subsequent models follows the SysML block definition diagram notation. A short guide for the notation is provided below (Figure 1).

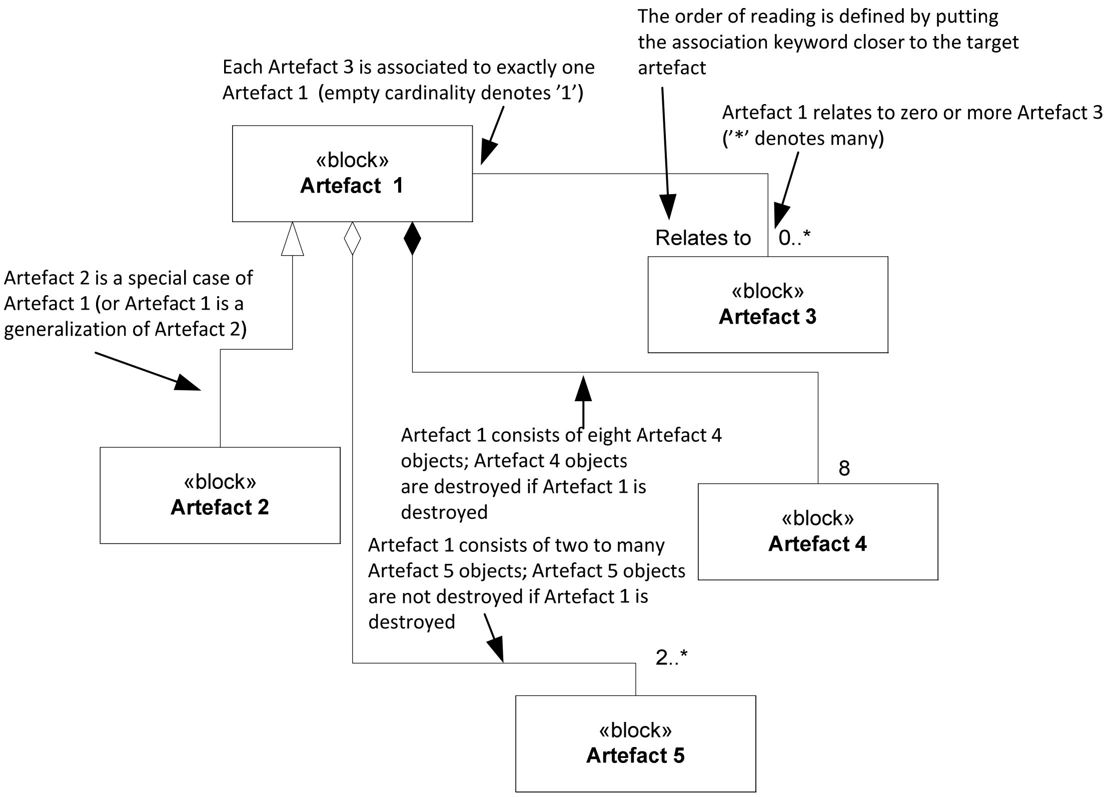
The order of reading is defined by putting the association keyword closer to the target artefact
Each Artefact 3 is associated to exactly one Artefact 1 (empty cardinality denotes ’1’)
Artefact 1 relates to zero or more Artefact 3 (’*’ denotes many)
«block»
Artefact 1
Relates to
0..*
«block»
Artefact 3
Artefact 2 is a special case of Artefact 1 (or Artefact 1 is a generalization of Artefact 2)
Artefact 1 consists of eight Artefact 4 objects; Artefact 4 objects are destroyed if Artefact 1 is destroyed
8
«block»
Artefact 2
«block»
Artefact 4
Artefact 1 consists of two to many Artefact 5 objects; Artefact 5 objects are not destroyed if Artefact 1 is destroyed
2..*
«block»
Artefact 5

Figure 2. SysML block definition diagram notation guide.

4.1 System package

System package provides the upper level model that defines the artefacts and their relations of the **system-of-interest** and its **context**. Also the artefacts and their relations to publish the **system type** to be applied in an application are modelled. The three models are described as block definition diagrams in the following sub-sections.

To better understand the sub-sequent models, it is necessary to understand the hierarchical structure of the systems. The system hierarchy model according to ISO/IEC/IEEE 15288 is illustrated in Figure 3. The system structure is modelled by two artefact types: System (system-of-interest being its special case) and System element, where a system element can be a sub-system or an atomic element, i.e. a component.

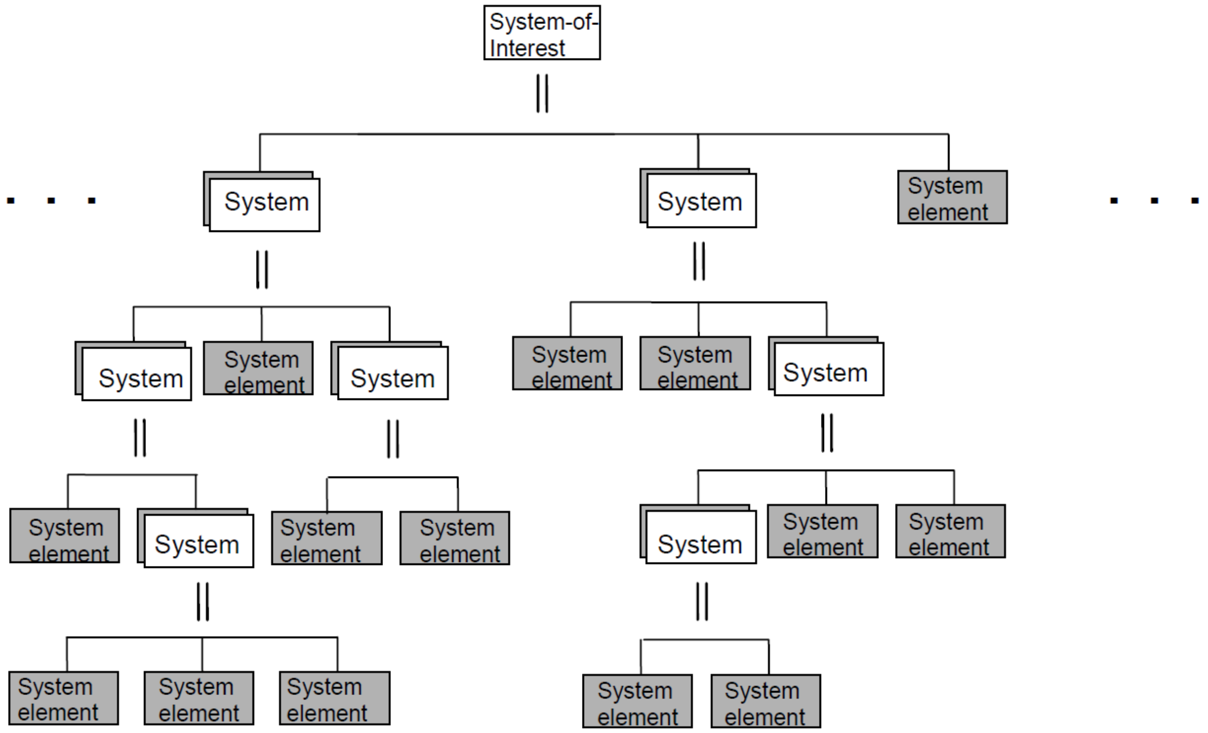

Figure 3. ISO/IEC/IEEE 15288:2008 system-of-interest structure model.

The model in Figure 3 is partially redrawn in Figure 4 to better illustrate the fact that a system only consists of system elements, and that a system element can be a sub-system or an atomic element, and that a sub-system does not have a special modelling element, but is a system (in fact a system-of-interest) from the point of view of the developer of the sub-system.

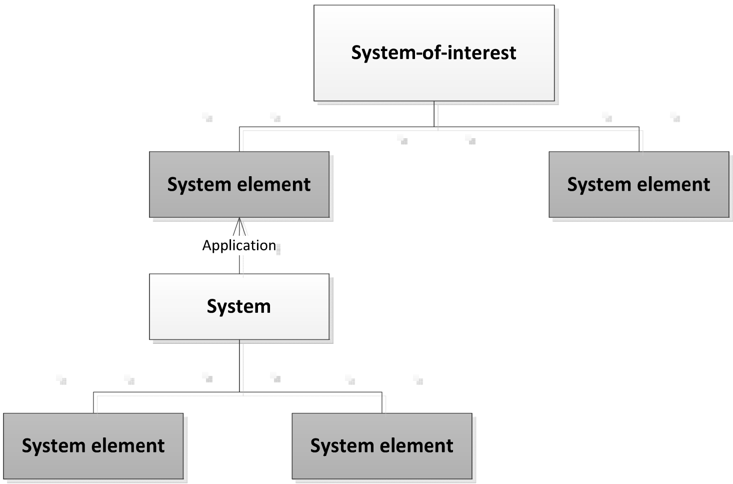

Figure 4. ISO/IEC/IEEE 15288:2008 partially redrawn.

In case of SEAModel, we enhance this model by distinguishing a development time system and a published system. For the purpose of published system, a new modelling element is introduced: System type. The System type modelling element stores the information necessary to apply the particular system type by an acquirer of the system to apply it for his specific purposes. It does not store all the development time information, that can partly be confidential, but the development time information is stored in the System (or it special case: System-of-interest) modelling element and its relating artefacts. In other words, the revised structure model provides three aspects of a system: development time information in the **System** model element, the published information in the **System element type** model element and the role of the published system as a sub-system or component in the **System element** model element. Figure 5 illustrates the revised system-of-interest structure model with distinguished model element for published systems.

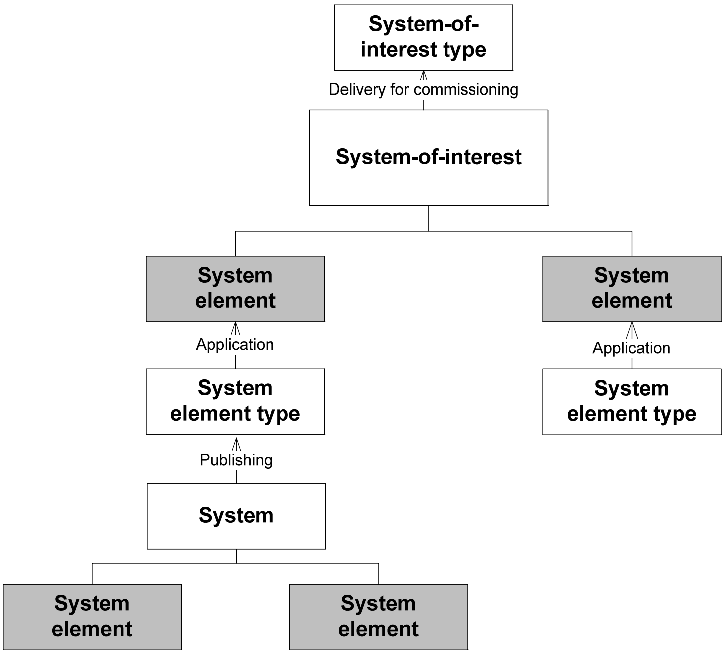

Figure 5. ISO/IEC/IEEE 15288:2008 customised by adding a new model element: System type.

There is yet another aspect of a system: system individual. When a published system is incorporated into the design it gets its application specific role in the System element model element, but when several instances of a system-of-interest and its constituent sub-elements are produced, each produced system or system element has got its instance specific data, like operating hours. Hence a fourth modelling element is introduced: ***Individual***. The supplier of the system and the acquirer of the system (and maybe some other stakeholders) may collect and store different kinds of information about the system or system element individuals. The completed system-of-interest structure model with the Individual model element is depicted in Figure 6.

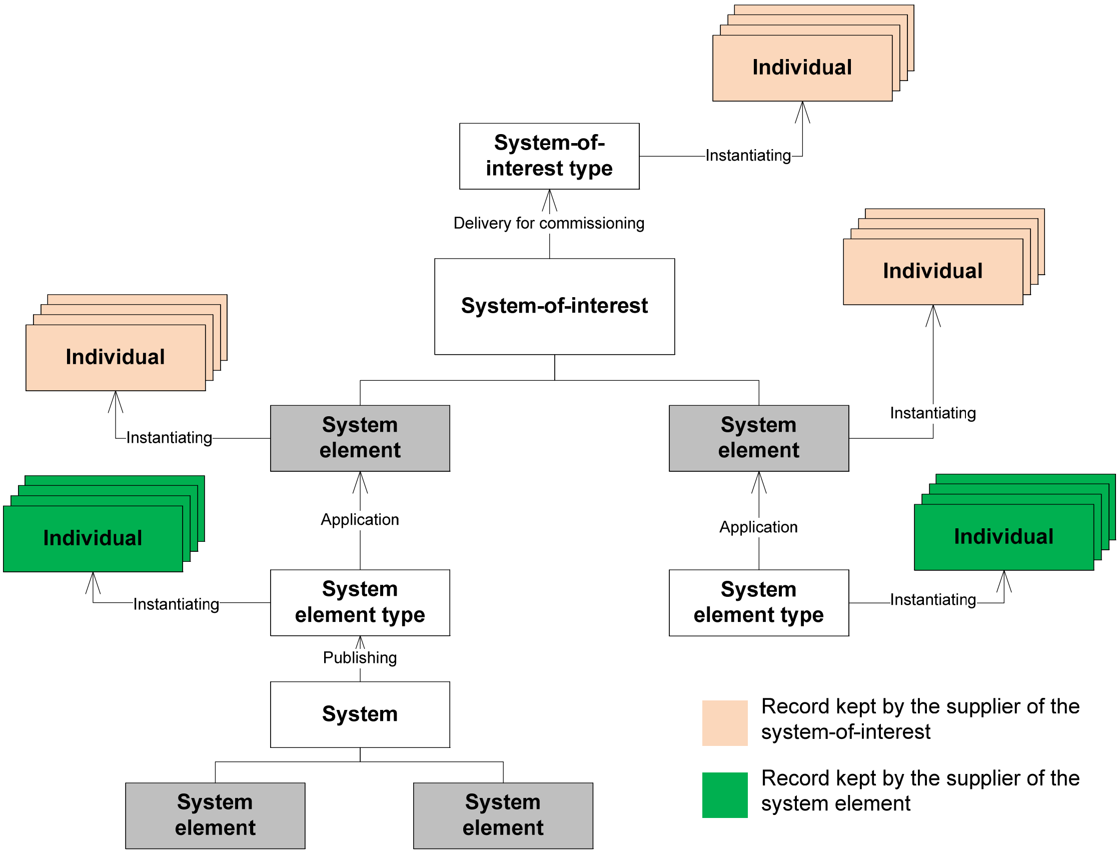

Figure 6. System-of-interest model with a model element for system element and system type individuals.

SEAModel is applied to each system, whether a system-of-interest or a subsystem developed in-house. For off-the-shelf sub-systems and components, SEAModel may or may not have been used; it does not matter for the systems engineer of the system-of-interest as long as he or she receives the necessary data from the sub-system or part manufacturer to be stored into the *System element type* artefact. In case SEAModel is not used by the system element developer, it is difficult to arrange seamless traceability of requirements from the main system to the system element, and traceability of verification artefacts from the system element to the main system. Such a full blown traceability in the hierarchy from top to bottom and back may be needed only in rare cases.

4.1.1 System-of-interest concept model

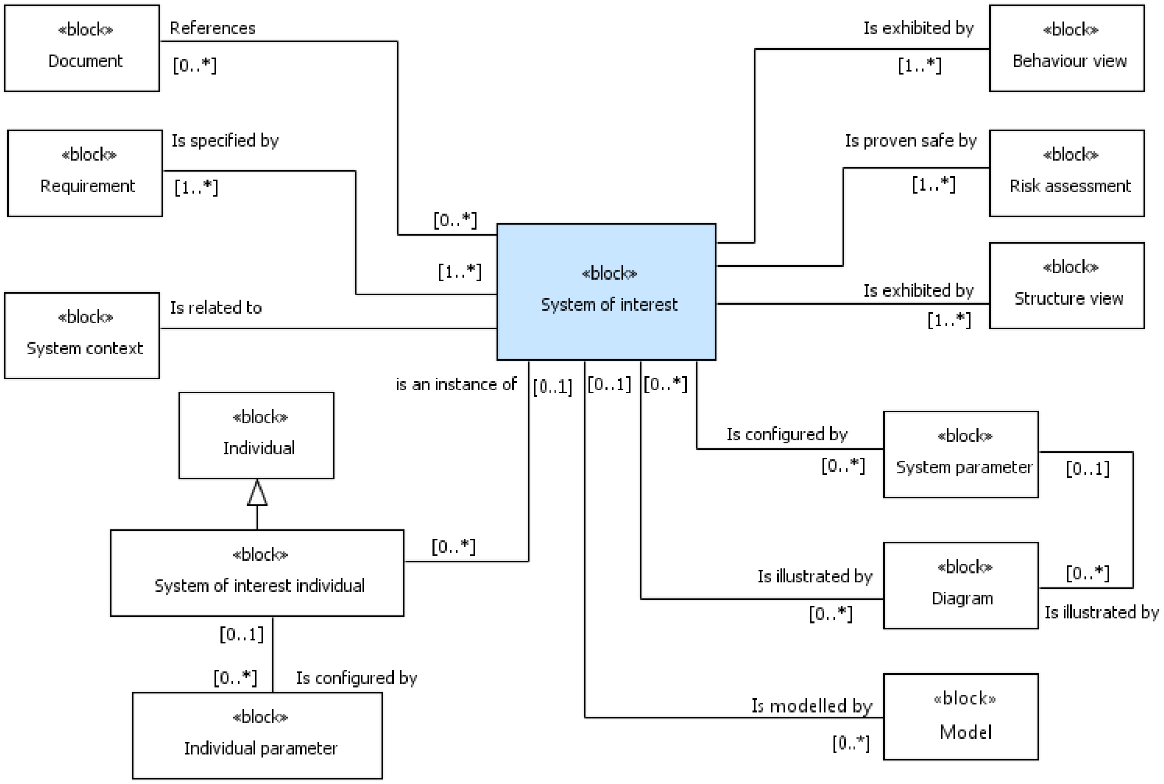

Figure 7. System-of-interest concept model diagram.

The System-of-interest model (Figure 7) is the main and top level model to define the artefacts relating to the system under development. The main parts of the system model (besides the System-of-interest block) are the System context, Requirement, Structure View, Behaviour View and Risk assessment blocks. A separate model for each of these is provided in later sections.

The artefact types (i.e. the ‘blocks’) in Figure 7 are described in Table 1.

Table 1. Description of the artefacts types shown in the System-of-interest concept model diagram (Figure 7).

Title of the artefact type	Description
Behaviour view	The Behaviour view artefact is used to provide different perspectives to system functional architecture. It can be used to provide views to categorised sets of functionality or to completely different functionalities of machines with dual or more usage purposes.
Diagram	The Diagram artefact can accommodate diagrams, pictures and photographs of any kind. The most important attribute is the link to the actual diagram file.
Document	The Document artefact defines any type of document, except diagrams and photos, which have a dedicated artefact type (Diagram). The most important attribute (or feature) is the link to the actual document file.
Individual	The Individual artefact records information about the supplied system individuals. Such information can include static (e.g. serial number) and dynamic information (e.g. operation hours).
Individual parameter	Besides the information provided by the Individual artefact and its attributes, a set of individual parameters (colour, existence of options etc.) can be assigned to a system individual.
Requirement	The Requirement artefact defines a requirement and its attributes, like type and source. A requirement can be a stakeholder requirement or a system requirement.

Title of the artefact type	Description
Risk assessment	The Risk assessment artefact works as an assignment to carry out a risk assessment task. It, however, also contains a short description of the results of the risk assessment (the actual results are reported in comprehensive analysis reports).
Structure view	Structure view provides a means to present different views to the system's physical architecture. The views can be partial views of the whole system structure or different viewpoints, like development time view and manufacturing time view
System context	The System context artefact provides description e.g. about the following issues requested by ISO 12100:2010: ergonomic principles, energy sources, space limits, life limit, service intervals, other time limits, housekeeping policy, material properties, other limits, external systems interaction and experience of use. The System context artefact also works as the main node to collect the system context related artefacts through associations.
System of interest	The System of interest artefact defines the system under development and during all the life cycle phases. It only includes a small number of attributes, mainly title and a short description of the system, i.e. the system identification.
System of interest individual	The System-of-interest individual artefact is a specialisation of the Individual artefact with no additional attributes.
System parameter	The System parameter artefact stores any kind of system parameters to configure e.g. a system function, system element or a port. Note that this parameter affects the system design; a separate parameter, Individual parameter, is used to configure system individuals.

4.1.2 System context concept model

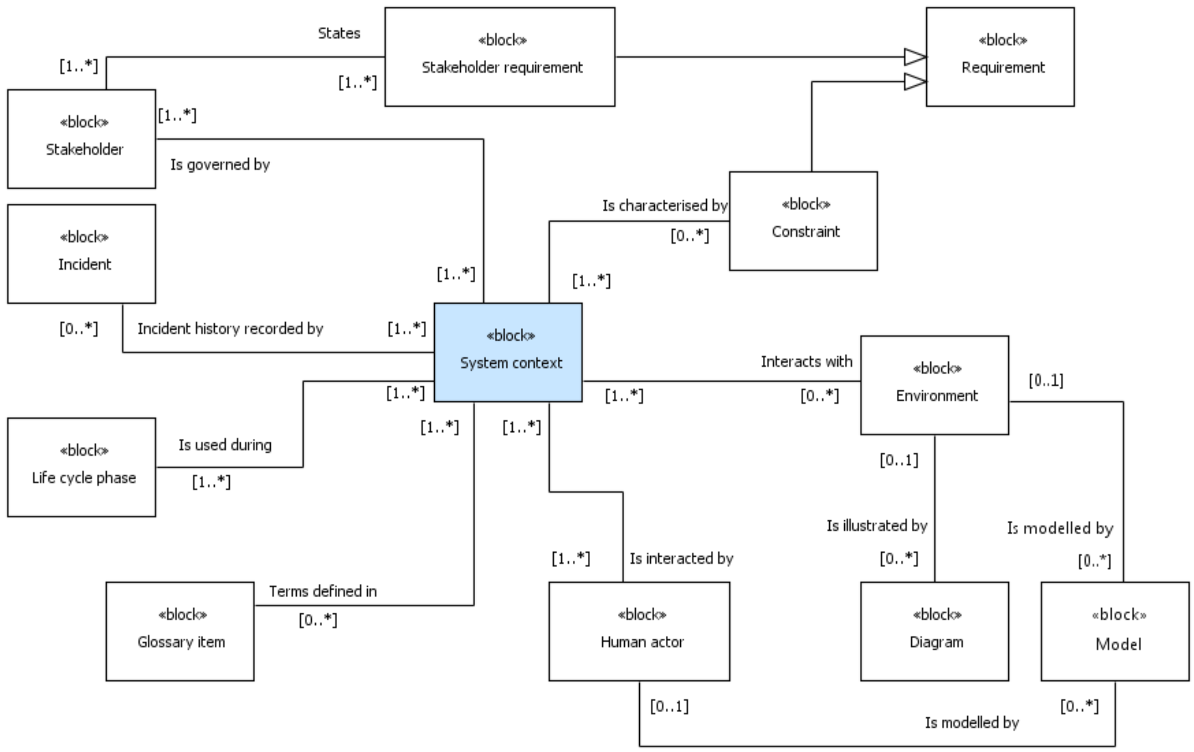

Figure 8. System context concept model diagram.

The system context model (Figure 8) models the context in which the system will be used. The context includes both concrete (like environment and human actors) and abstract issues (like incident history and glossary).

The artefact types (i.e. the ‘blocks’) in Figure 8 are described in Table 2.

Table 2. Description of the artefacts types shown in the System context concept model diagram (Figure 8).

Title of the artefact type	Description
Constraint	Constraint is a special case of a requirement. In one sense, it is not a requirement, because it only states facts e.g. about the system’s environment (like the dimensions of the space where the system will be installed); on the other hand, its effect in the design is similar to that of requirements.
Diagram	The Diagram artefact can accommodate diagrams, pictures and photographs of any kind. The most important attribute is the link to the actual diagram file.
Environment	Description of the mechanical, climatic, chemical, ergonomic and external system environment, especially in regard to their effect in the system of interest. Domain knowledge can be described here, too.
Glossary item	Definitions, terms and abbreviations are presented in the Glossary item artefact.
Human actor	Any human actor that interacts with the system, whether an assembly man, operator, maintenance man, cleaner, etc., or a bystander who is situated in the hazard zone of the system.
Incident	A record of accidents and incidents history, including near misses, of the this type of systems or similar system
Life cycle phase	The life cycle model is recorded in the Life cycle phase artefact, e.g. Concept, Development, Production, Utilisation, Support and Retirement according to ISO/IEC TR 24748-1:2010.
Requirement	The Requirement artefact defines a requirement and its attributes, like type and source. A requirement can be a stakeholder requirement or a system requirement.
Stakeholder	The Stakeholder artefact lists the stakeholders that may state requirements for the system (i.e. that have interest in the system). Stakeholders can include e.g. system users, domain experts, principal, investors, board of directors, corporate management, authorities, laws, standards, customers, maintenance staff, training staff, system engineer, buyers of the system and marketing and sales.
Stakeholder requirement	A special case of a requirement: requirement set by a stakeholder
System context	The System context artefact provides description e.g. about the following issues requested by ISO 12100:2010: ergonomic principles, energy sources, space limits, life limit, service intervals, other time limits, housekeeping policy, material properties, other limits, external systems interaction and experience of use. The System context artefact also works as the main node to collect the system context related artefacts through associations.

4.1.3 System element type concept model

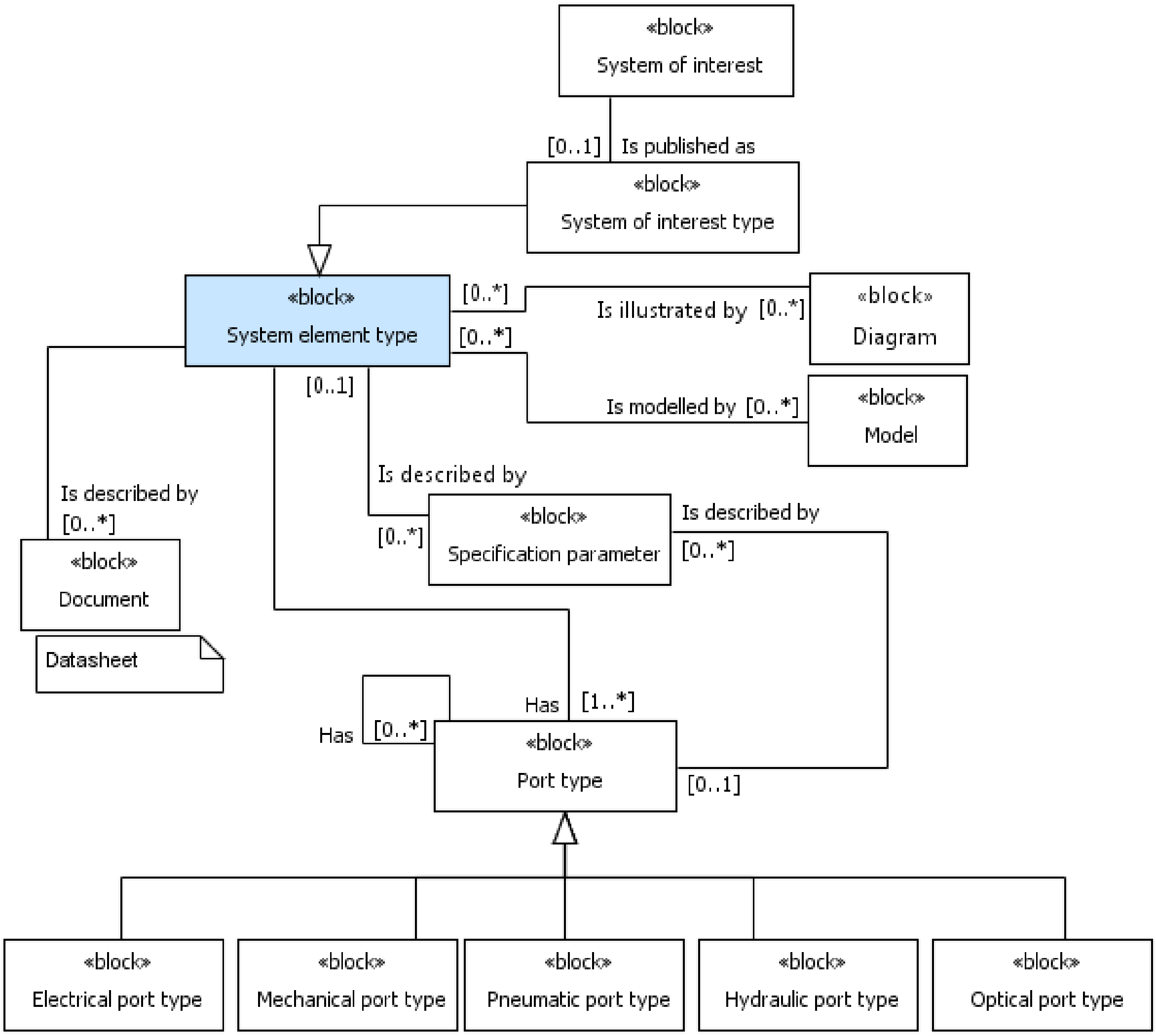

Figure 9. System element type concept model diagram.

The System element type concept model depicted in Figure 9 provides a model for system types, i.e. published systems to be applied by the system developers or acquirers of the system. The System type model works as a model for a library of sub-systems and components (i.e. parts) incorporated into the system-of-interest; both sub-systems and components are stored using the System element type artefact. But it also provides the platform for the developer of the system-of-interest to publish the system under development when it is ready (this is done by the System of interest type artefact that is a special case of System element type).

The set of System element types and their ports constitute a system element library that contains all the generic information about the system elements and their interfaces (ports). A system element is an instance of a system element type (see Figure 14), and a port is an instance of a port type (see Figure 14). Hence a system element inherits all the information from its system element type. Similarly, a port inherits all the information from its port type.

The idea is that the datasheet information of the system types is stored into a structured data repository. However, it is also possible to attach a conventional datasheet with a system type if necessary. The optimal workflow of course would be such that the component manufacturers provide the datasheets in XML files that can easily be incorporated into the artefact repository according to SEAModel. This would require broad acceptance of SEAModel by the machine manufacturers and their sub-contractors.

System type and its port types can be specified with several specification parameters. The specification parameters cannot be changed; they have been defined by the component vendor, e.g. a component (i.e. an atomic system element) can have as its specification parameter, weight, dimensions, allowable temperature range, etc. Such information is normally presented in an easily readable format in conventional datasheets, but the provision of such a structured way of storing specification parameters in the Spec parameters artefact facilitates showing of the specification parameters in the application specific documents or drawings generated from a SEAModel based data repository. The Spec parameter artefact has been motivated by MSRSYS specification (MSR Consortium 2002) and contains many of the attributes defined by it. The System element type artefact has been defined such that it can accommodate the *DeviceIdentity* element from the CANopen XML-based device profile according to CiA DSP 311.

The artefact types (i.e. the ‘blocks’) in Figure 9 are described in Table 3.

Table 3. Description of the artefacts types shown in the System type concept model diagram (Figure 9).

Title of the artefact type	Description
Document	The Document artefact defines any type of document, except diagrams and photos, which have a dedicated artefact type (Diagram). The most important attribute (or feature) is the link to the actual document file.
Electrical port type	The electrical case of the Port type artefact
Hydraulic port type	The hydraulic case of the Port type artefact
Mechanical port type	The mechanical case of the Port type artefact
Model	The Model artefact represents any kind of model, SysML model, virtual model, mock-up, etc.
Optical port type	The optical case of the Port type artefact
Pneumatic port type	The pneumatic case of the Port type artefact
Port type	The port type specifies (with its specification parameters) the interfaces of the system element type. A port type may consist of sub-ports. A typical case is a connector with several pins.
Specification parameter	Specifies a parameter of the system type; a ‘datasheet parameter’, like weight, maximum temperature, etc.
System element type	The system element type artefact contains the overall identification and description of the library component or sub-system, or of the published system-of-interest.
System of interest	The System of interest artefact defines the system under development and during all the life cycle phases. It only includes a small number of attributes, mainly title and a short description of the system, i.e. the system identification.
System of interest type	A special case of the System element type artefact; the overall identification and description of the published system of interest

4.2 Requirement package

The Requirement package consists of one concept model, the Requirement concept model, which is presented in Section 4.2.1. In Section 4.2.2, however, a more detailed discussion on using requirement templates is provided.

4.2.1 Requirement concept model

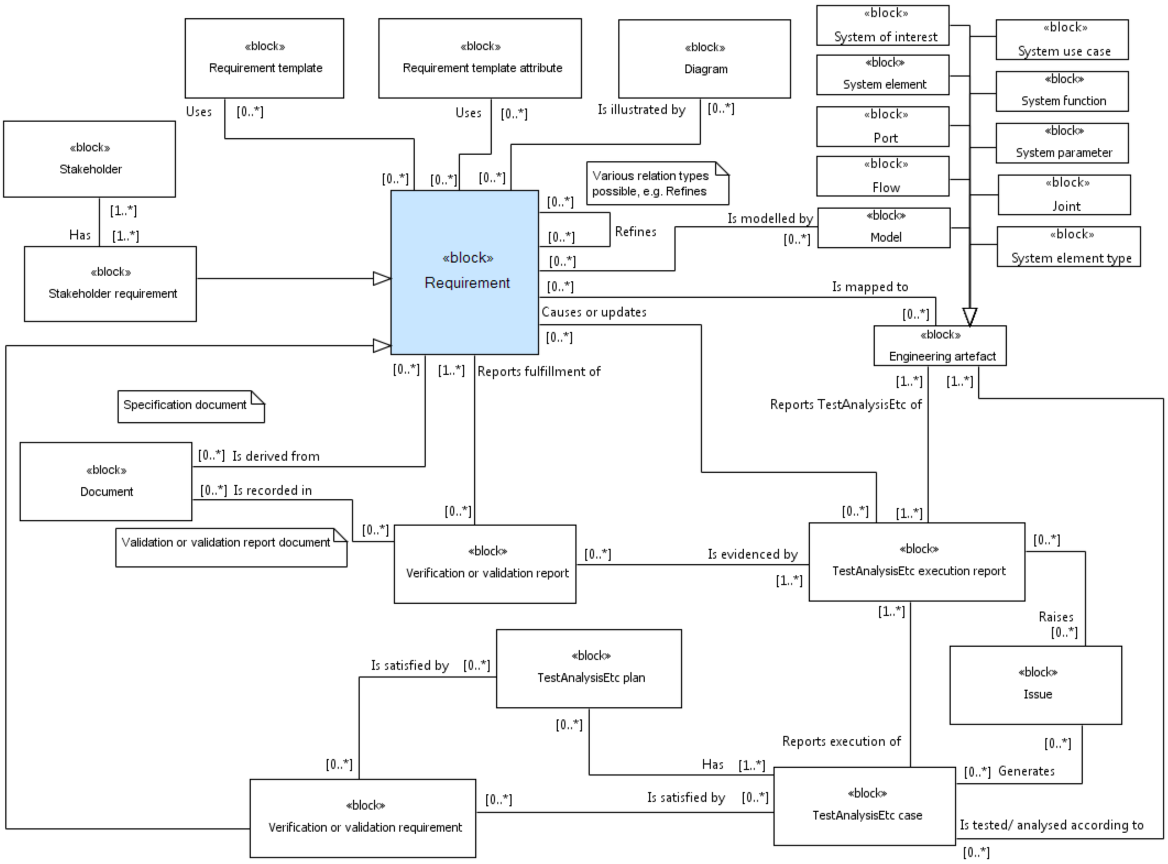
«block» Requirement template
«block» Requirement template attribute
«block» Diagram
«block» System of interest
«block» System element
«block» Port
«block» Flow
«block» Model
«block» System use case
«block» System function
«block» System parameter
«block» Joint
«block» System element type
«block» Stakeholder
«block» Stakeholder requirement
«block» Requirement
«block» Engineering artefact
Specification document
«block» Document
Validation or validation report document
«block» Verification or validation report
«block» TestAnalysisEtc execution report
«block» TestAnalysisEtc plan
«block» Issue
«block» Verification or validation requirement
«block» TestAnalysisEtc case
Various relation types possible, e.g. Refines
Uses
Is illustrated by
Has
Refines
Is modelled by
Is mapped to
Causes or updates
Reports fulfillment of
Reports TestAnalysisEtc of
Is derived from
Is recorded in
Is evidenced by
Raises
Is satisfied by
Reports execution of
Generates
Is tested/ analysed according to
[0..*]
[1..*]

Figure 10. Requirement concept model diagram.

The core model for traceability from requirements to design and implementation, from implementation to test execution, and from test execution to verification or validation reporting is presented in Figure 10. The central point for the requirement concept model is the Requirement artefact that stores both the stakeholder and system requirements. The Stakeholder requirement is a special case of a requirement: it relates to one or more Stakeholders and needs to be validated at the end of the project to prove that the system fulfils the expectations of the stakeholders. A system requirement is the ‘normal’ case of a requirement. The top level system requirements are derived from the stakeholder requirements. A requirement can have child requirements, that refines the parent requirement. In this case, the trace role is ‘refine’, but other roles can also be supported. Diagrams, documents and models can be linked to the requirements where necessary.

The realisation of the requirement is validated at the end of the development project. A special set of artefact types are provided for this purpose: Verification or validation report, TestAnalysisEtc execution report and TestAnalysisEtc case. Tests, analyses or any other methods are needed for evidence of fulfilling the requirements for the design. A TestAnalysisEtc execution report must be traced to the artefact under verification or validation to prompt for a test (or analysis etc.) re-execution if the artefact under test is updated.

Based on the results of the execution of a test or an analysis, an issue can be raised to call for corrective actions.

TestAnalysisEtc plan collects a set of TestAnalysisEtc cases to form a specific sequence of tests for a specific purpose, such as for Factory Acceptance Test (FAT).

The validation model is designed to work with the ISO 13849-2 (2003) validation procedure. For example, when validating the category of a safety function, observance of the basic safety principles must be inspected. The basic safety principles of electrical systems are listed in Appendix D of ISO 13849-2. Let us consider the following basic safety principle: *Proper selection, combination,* *arrangements, assembly and installation of components/system*. The particular issue is written into the TestAnalysisEtc case artefact, e.g. as: *Inspect that components/systems are properly selected, combined, arranged, assembled and installed*. In ISO 13849-2, there are several such tasks, not only the basic and well-tried safety principles that can be presented as test or analysis cases. The fault modes to be considered can also be presented as test or analysis cases. The result of the test or analysis is stored in the TestAnalysisEtc execution artefact, and the validation result is recorded in the Verification or validation report artefact. It may be necessary to execute more than one test or analysis cases for the evidence of successful result of the validation. Hence the Verification or validation report artefact has a one-to-many relation to the TestAnalysisEtc execution report artefact.

The concept model in Figure 10 provides a simple support for using requirement templates. A detailed discussion on requirement templates is provided in Section 4.2.2.

The artefact types (i.e. the ‘blocks’) in Figure 10 are described in Table 4.

Table 4. Description of the artefacts types shown in the Requirement concept model diagram (Figure 10).

Title of the artefact type	Description
Diagram	The Diagram artefact can accommodate diagrams, pictures and photographs of any kind. The most important attribute is the link to the actual diagram file.
Document	The Document artefact defines any type of document, except diagrams and photos, which have a dedicated artefact type (Diagram). The most important attribute (or feature) is the link to the actual document file.
Engineering artefact	An Engineering artefact can be any artefact that is a result of the engineering work during the life cycle of the system.
Flow	Specifies the flow between ports. A flow can be electrical (signal), hydraulic (fluid), mechanical (momentum), optical (light), etc.
Issue	Any kind of issue prompting actions by the development organisation, e.g. to change specifications or to do redesign.
Joint	The logical connection between ports. It does not specify the actual physical implementation of the connection. E.g. in case of electrical connection, it represents the joint galvanic point that connects two or more electrical pins, but it does not specify the wires and cables used to implement the galvanic connection.
Model	The Model artefact represents any kind of model, SysML model, virtual model, mock-up, etc.
Port	The interfaces are modelled by the Port artefacts. Port is a logical interface entity that is specified in detail by the flows it can carry.
Requirement	The Requirement artefact defines a requirement and its attributes, like type and source. A requirement can be a stakeholder requirement or a system requirement.
Requirement template	A requirement template that forms the body of a requirements sentence
Requirement template attribute	Requirement template attributes are the system specific terms or system elements that are placed to the placeholders of a requirement template to create a well-formed requirements sentence.
Stakeholder	The Stakeholder artefact lists the stakeholders that may state requirements for the system (i.e. that have interest in the system). Stakeholders can include e.g. system users, domain experts, principal, investors, board of directors, corporate management, authorities, laws, standards, customers, maintenance staff, training staff, system engineer, buyers of the system and marketing and sales.
Stakeholder requirement	A special case of a requirement: requirement set by a stakeholder
System element	A System element can be a sub-system or a component. System elements together constitute the system of interest. The typical distinguishing factor between a sub-system and a component is the existence of a part number: a subsystem does not normally have a part number.
System function	A system level function, e.g. boom movement

Title of the artefact type	Description
System of interest	The System of interest artefact defines the system under development and during all the life cycle phases. It only includes a small number of attributes, mainly title and a short description of the system, i.e. the system identification.
System parameter	The System parameter artefact stores any kind of system parameters to configure e.g. a system function, system element or a port. Note that this parameter affects the system design; a separate parameter, Individual parameter, is used to configure system individuals.
System use case	A system level use case. Use cases can be specified according to SysML, but the system use case descriptions can be supplemented by safety related information.
TestAnalysisEtc case	The specification for a single test, analysis, demonstration, review etc. case
TestAnalysisEtc execution report	Records the results of the test, analysis, demonstration, review etc. case executions.
TestAnalysisEtc plan	The TestAnalysisEtc plan artefact collects a set of test, analysis, demonstration, review etc. cases to form a specific sequence of tests for a specific purpose, such as for Factory Acceptance Test (FAT).
Verification or validation report	The result of verification or validation is recorded here. The main content is a simple pass/no-pass verdict.
Verification or validation requirement	A special case of the Requirement artefact. In many cases, the requirements specification or safety standards set requirements as to how the design shall be verified or validated.

4.2.2 Embracing requirement templates

Requirements templates (a.k.a. requirement boilerplates) can be used to set a fixed framework for requirements sentences. The requirement templates constrain the freedom for composing requirements sentences in order to provide well-formed, good quality requirements. E.g. in case of the following requirement template *(*Hull et al. 2010*)*,

If <operational condition> the <system> shall <function>,

the requirement instance could be:

"If the operator is still using the remote controller the mobile elevating platform shall disable changing of the operating mode."

Where:

<operational condition> = the operator is still using the remote controller

<system> = mobile elevating platform

<function> = shall disable changing of the operating mode.

Requirement templates are presented by several authors. Table 5 lists some examples of requirement templates from (Hull et al. 2010).

Table 5. Example requirement templates (Hull et al. 2010).

Performance/capability	The <system> shall be able to <function> <object> not less than <performance> times per <units>
Performance/capability	The <system> shall be able to <function> <object> of type <qualification> within <performance> <units>
Performance/capacity	The <system> shall be able to <function> not less than <quantity> <object>
Performance/timeliness	The <system> shall be able to <function> <object> within <performance> <units> from <event>
Performance/periodicity	The <system> shall be able to <function> not less than <quantity> <object> within <performance> <units>
Interoperability/capacity	The <system> shall be able to <function> <object> composed of not less than <performance> <units> with <external entity>
Sustainability/periodicity	The <system> shall be able to <function> <object> for <performance> <units> every <performance> <units>
Environmental/operability	The <system> shall be able to <function> <object> while <operational condition>

A similar approach has been published by the SOPHIST GmbH company, the template model of which is illustrated in Figure 11.

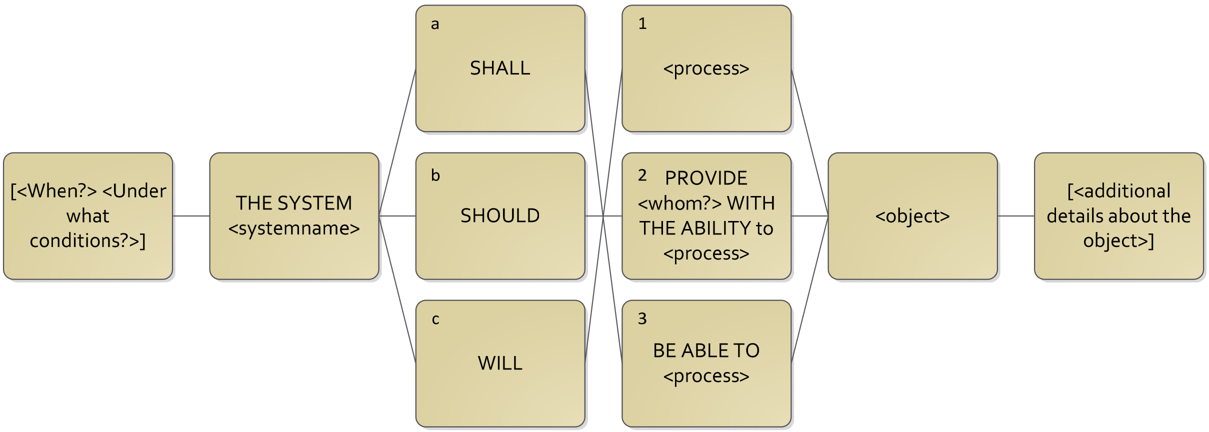
[<When?> <Under what conditions?>]
THE SYSTEM <systemname>
a
SHALL
b
SHOULD
c
WILL
1
<process>
2
PROVIDE <whom?> WITH THE ABILITY to <process>
3
BE ABLE TO <process>
<object>
[<additional details about the object>]

Figure 11. The SOPHIST GmbH model for requirement templates (Rupp et al. 2009).

The first selection (a, b or c) is based on the degree of obligation, whereas the second selection (1, 2 or 3) is based on the requirement type:

1. **Autonomous system action**. Example: “If the customer data entered is already present in the system, the system shall display the error message „customer already exists“.”
2. **User interaction**. Example: “If a customer is not yet registered on the library system, the library system shall provide the librarian with the ability to enter a customer‘s name and date of birth.”
3. **Interface requirement**. Example: “As long as the library system is in operation, the library system shall be able to receive software update data from a central administration computer via the local area network.”

All examples above are provided by Rupp et al. (2009).

Yet another model for requirements templates has been provided by (Mavin and Wilkinson 2010). It is called Easy Approach to Requirements Syntax (EARS). The EARS model consists of only five requirement templates as listed in Table 6.

Table 6. EARS templates (Mavin and Wilkinson 2010).

Type	Description	Template	Example
Ubiquitous	Ubiquitous requirements have no preconditions or trigger, but are always active	The <system name> shall <system response>	"The control system shall indicate the engine oil quantity to the aircraft."
Event-Driven	Event-driven requirements are initiated only when a triggering event is detected at the system boundary and are designated by the keyword When	WHEN <optional preconditions> <trigger> the <system name> shall <system response>	"When continuous ignition is commanded by the aircraft, the control system shall switch on continuous ignition."
Unwanted Behaviour	Unwanted behaviour requirements define the required system response to mitigate an unwanted event, or to prevent the system from entering an unwanted state. They are logically equivalent to event-driven requirements, but are designated by the If-Then keywords. This explicitly differentiates requirements that handle unwanted behaviour.	IF <optional preconditions> <trigger>, THEN the <system name> shall <system response>	"If the computed airspeed fault flag is set, then the control system shall use modelled airspeed."
State-Driven	State-driven requirements are active while the system is in a defined state and are designated by the keyword While.	WHILE <in a specific state> the <system name> shall <system response>	"While the aircraft is in-flight, the control system shall maintain engine fuel flow above XXlbs/sec."
Operational Feature	Operational feature requirements are applicable only in systems that include a particular feature and are designated by the keyword Where.	WHERE <feature is included> the <system name> shall <system response>	"Where a control system component acts as a firewall, the component shall be Fireproof."

SEAModel incorporates requirement templates by introducing two artefact types: Requirement template and Requirement template attribute (see Figure 10). If we consider the example in the beginning of this section (Section 4.2.2), the <system> attribute and the <function> attribute could well be in the set of SEAModel design artefacts, namely System element and System function respectively, but the <operational condition> attribute, "*the operator is still using the remote controller*", is a free form piece of text that has no placeholder in SEAModel. The <operational condition> consists, in fact, of three attributes, <actor> ("*operator*"), <action> ("*is using*") and <object> ("remote controller"). Of these, it is easy to map the <actor> attribute onto the Human actor artefact of SEAModel, <action> plus <object> can be mapped onto the Use case act artefact, but the additional word "*still*" is not included in the use case act description; the additional words and the 'the' articles are not included neither in the requirement template nor in the artefact descriptions.

Hence the only reasonable solution is to provide a single list of requirement template attributes instead of using the already defined artefact types of SEAModel. In SEAModel, the Requirement template attribute artefact supplies these attributes. However, this may make the requirements definition process distinct from the other processes in terminology. This makes the requirements template methodology less systematic. Hence advantages of requirement templates are questionable. Another problem is the set of existing prewritten requirements e.g. from safety standards; they do not follow the requirement templates scheme. As a consequence, the advantages of the quite disciplined way of recording simple requirements with the requirement template method may be minimal. It is, however, recommended to use requirement templates in the phase of capturing the stakeholder requirements, especially the customer requirements; the consistency of the customer requirements is important due to the fact that they are in many cases attached with the contract between the customer and the contractor, and have thus juridical importance.

4.3 Behaviour package

Behaviour package includes the models to define system behaviour artefacts and their relations from the actor perspective (Behaviour view Model; Section 4.3.1) and from the system perspective (System Function Model; Section 4.3.2).

4.3.1 Behaviour view concept model

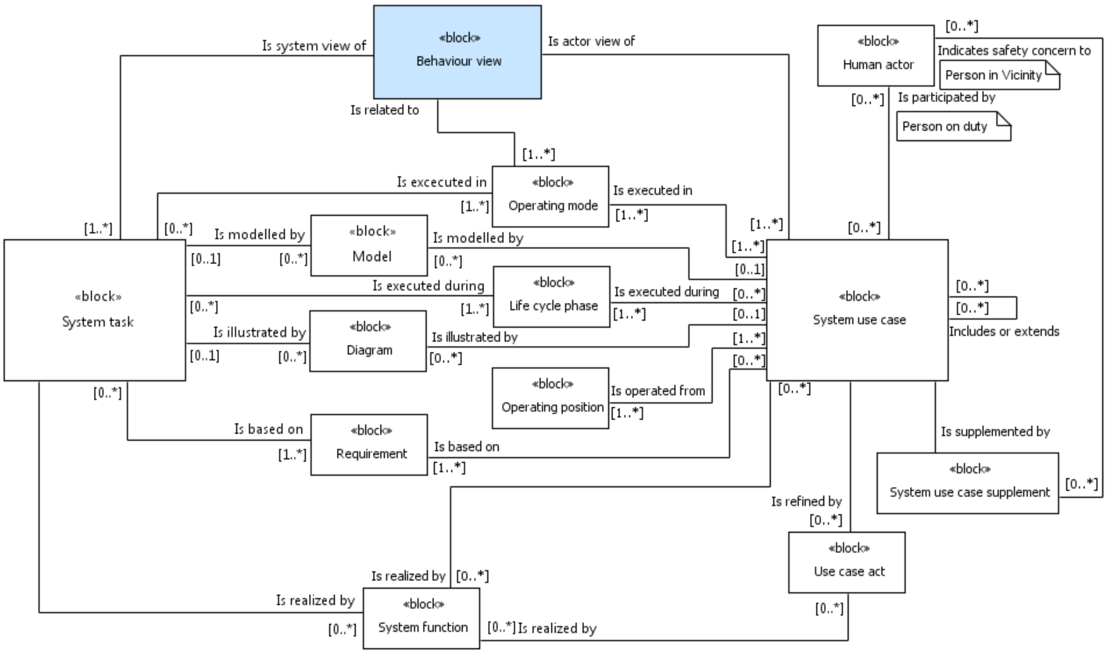

Figure 12. Behaviour view concept model diagram.

The Behaviour view concept model encompasses the description of the functionality of the system. Its core artefact is the System use case artefact. The Behaviour artefact only gives a basic description of the system functionality in verbal format as captured from the stakeholders, i.e., a description of the work to be performed by the machine (the 'intended use' of the machine as phrased by the risk assessment standard ISO 12100, but it also stores the description of the reasonably foreseeable misuse that must be considered according to ISO 12100).

It is possible to define several Behaviour view artefacts for a single system. This is useful in cases where the system or the machine has clearly separate ways of working or separate functional features.

The system behaviour is described in a more systematic way in system use cases and system tasks, the former being the actor view of the behaviours and the latter the system view.

System use case artefacts are created to describe the functional requirements stated by the stakeholders in a systematic way. This is why the system use cases are related to the Requirement artefact as depicted in Figure 12. Any system use case can be illustrated by any behaviour related model, like activity diagram, sequence diagram, state machine diagram or use case diagram, or any other diagram.

A system use case can include finer grained use cases or extend another system use case. The sequence of acts of a system use case is stored in the Use case act artefact. The reason for separating the Use case act artefact from the System use case artefact is that during the Use Case Safety Analysis (UCSA) we need to be able to link a single use case act to an identified hazard to provide traceability. It must be ensured that to extend traceability such that if e.g. the set of Human actor artefacts is changed not only the System use case artefacts related to the changed Human actor artefacts are marked suspect but also the related Use case act artefacts and the related hazards. This is reasonable if we think about a case in which a new actor is introduced into the system. It is then highly relevant to re-analyse, using the UCSA method, all the system use cases to which the new actor is linked; or, if one of the human actors is removed from the human actor list, one or more hazards identified by UCSA may become irrelevant or need to be updated.

One possible way to work with system use cases and use case acts is to write the system use cases in a platform independent way (i.e. with no reference to the underlying system elements) and the use case acts with platform dependent way. In this way, the use case acts are written later than system use cases, i.e. after the first release of the physical architecture of the system is available.

System use cases are realised by System functions. A use case can use one or more system functions and a system function can belong to several use cases. System use cases and use case acts provide the actor view of the system behaviour, whereas system functions provide the system view of the system behaviour.

The System task artefact is the system realisation view of the behaviour. It defines the flow of system functions. It is especially useful in cases where human actors are not involved, but it is also used to identify system functions that cannot be identified from system use cases or use case acts.

The artefact types (i.e. the ‘blocks’) in Figure 12 are described in Table 7.

Table 7. Description of the artefacts types shown in the Behaviour view concept model diagram (Figure 12).

Title of the artefact type	Description
Behaviour view	The Behaviour view artefact is used to provide different perspectives to system functional architecture. It can be used to provide views to categorised sets of functionality or to completely different functionalities of machines with dual or more usage purposes.
Diagram	The Diagram artefact can accommodate diagrams, pictures and photographs of any kind. The most important attribute is the link to the actual diagram file.
Human actor	Any human actor that interacts with the system, whether an assembly man, operator, maintenance man, cleaner, etc., or a bystander who is situated in the hazard zone of the system.
Life cycle phase	The life cycle model is recorded in the Life cycle phase artefact, e.g. Concept, Development, Production, Utilisation, Support and Retirement according to ISO/IEC TR 24748-1:2010.
Operating mode	Different types of control or operating modes can include e.g. the following modes: automatic, manual; remote, local; diagnostics; ‘limp home’.
Operating position	The positions at which the system is controlled and operated
Requirement	The Requirement artefact defines a requirement and its attributes, like type and source. A requirement can be a stakeholder requirement or a system requirement.
System function	A system level function, e.g. boom movement
System process	A special case of a system use case. Describes the sequence of system use cases in an explicit way (contrary to the implicit way in which the sequence of system use cases can be concluded from the post conditions of one use case and preconditions of another use case).
System task	The System task artefact is the system realisation view of the behaviour. It defines the flow of system functions. It is especially useful in cases where human actors are not involved, but it is also used to identify system functions that cannot be identified from system use cases or use case acts.
System use case	A system level use case. Use cases can be specified according to SysML, but the system use case descriptions can be supplemented by safety related information.
System use case supplement	A safety related supplement that adds attributes like ‘actor qualification’, ‘list of operator instructions’, ‘expected misuse’, ‘activation frequency’, ‘preliminary accident scenario’ to the system use case.
Use case act	The sequence of acts of a system use case is stored in the Use case act artefacts. A use case act is an atomic piece of a work task. System functions are often identified from the use case acts.

4.3.2 System function concept model

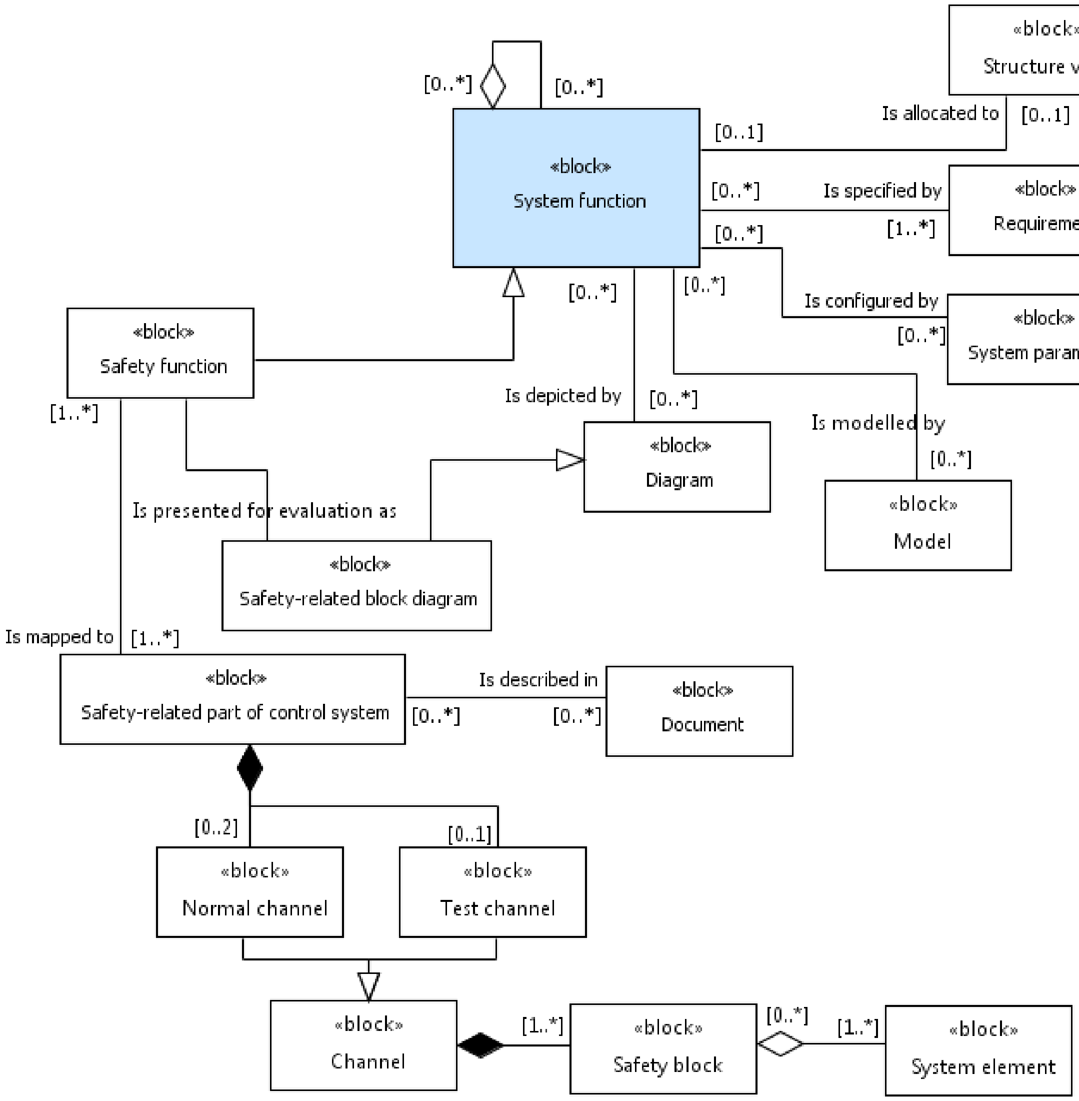

Figure 13. System function concept model diagram.

The System function concept model (Figure 13) completes the behaviour model of the system.

The system use cases are realized by System functions. A system function is specified exhaustively in this model such that the software engineer can implement the software for the system function based on the particular system function artefact. The attributes of the System function artefacts are selected such that the requirements for a safety function specification according to IEC 62061:2005 are fulfilled. A Safety function[2] is a special case of a system function and is often a 'sub-function'[3] or, to be more

[2] The concept of a safety function can be somewhat obscure to a machine engineer, and it may be difficult to identify and specify a safety function. For example, a boom movement is a normal operational function. When limiting its speed to a safe level, the speed-limiting facility can be called a safety function, but it may be difficult to point it out and show where it is because it may simply be a line of application software and a parameter embedded in the application software. Let us think of another safety function called 'prevention of unexpected movement': the boom movement is stopped when the joystick is released to its central position but, for safety reasons, a dead-man's switch and a hydraulic enabling valve are added. Now the requirement for the safety function 'prevention of unexpected movement' could be, e.g., PL d, which is achieved by a two-channel approach (i.e., with Category 3 according to ISO 13849-1). What are the two channels? The first one is the normal centre position stop and the second one is the dead-man's switch – enable valve – channel. Now this leads to the fact that half of the safety function is allocated onto the normal channel and the rest to the additional safety channel. In both of the examples it is difficult to separate the safety function from the operational function and hence the electrical control system that executes the normal system functions easily becomes a safety-related electrical control system as a whole. In some industry sectors the safety functions are separated from normal functions.

[3] Safety function is not a sub-function in the sense that the system function does not necessarily call it, but the safety function exists along with the system function to provide the necessary functional safety measures.

precise, a ‘parasitic function’ of a system function. Hence the model allows a system function to consist of one or more sub-functions. There can of course be sub-functions that are not safety functions.

Furthermore, the model contains almost all the information needed to make a safety analysis for a system function with analysis methods like FMEA, HAZOP or FTA. The missing part in this model in regard to safety analysis is the communications part that is described in a model of its own (see Section 4.4.2).

A system function is allocated to a Structure view artefact of its own. Such a system function specific structure is a partial view of the actual system structure to illustrate the part of the system structure that takes part in executing the system function.

A system function is specified by a set of Requirement artefacts, configured by System parameters and can be modelled by models and diagrams of any kind.

The model also includes the artefact types needed to carry out the Performance Level (PL) evaluation according to ISO 13849-1:2006. (Neither the IEC 62061 nor the IEC 61508 safety integrity levels (SIL) are currently supported by the model, although the System function specification is done according to IEC 62061[4].) Safety function must be represented for the PL evaluation in a manner that cannot be fulfilled by the Structure view concept model presented in Section 4.4.1. Hence a special set of artefacts is attached to Safety function. A safety-related block diagram needs to be drawn to define the logical structure of the safety function to illustrate, which safety blocks (i.e., unities of system elements) are logically connected in series and which in parallel in the fault tolerance sense. ISO 13849-1:2006 (in its Appendix B) gives guidance on creating such diagrams. Such a diagram is in theory drawn for each safety-related part of a control system (SRP/CS)[5], but the model requires a combined block diagram to be connected to the safety function instead of SRP/CS due to the fact that it is more common to present a combined diagram that has its context in the safety function, not in individual SPR/CS:s. The model also allows linking of Document artefacts to an SRP/CS artefact. Such a document can be a safety manual or a technical manual of an off-the-shelf safety device, such as a safety PLC.

A safety-related part of a control system[6] (SRP/CS) can be a one channel system or a two channel system. Such channels are denoted Normal channels in the model. Test channels may also be defined (in Category 2 solutions according to ISO 13849-1:2006). The Normal channels and Test channels are special cases of the Channel artefact. Each channel consists of blocks, and blocks consist of system elements.

The artefact types (i.e. the ‘blocks’) in Figure 13 are described in Table 8.

Table 8. Description of the artefacts types shown in the Behaviour view concept model diagram (Figure 13).

Title of the artefact type	Description
Channel	A logical entity that encompasses the signal processing flow from input to output through logic
Diagram	The Diagram artefact can accommodate diagrams, pictures and photographs of any kind. The most important attribute is the link to the actual diagram file.
Document	The Document artefact defines any type of document, except diagrams and photos, which have a dedicated artefact type (Diagram). The most important attribute (or feature) is the link to the actual document file.
Model	The Model artefact represents any kind of model, SysML model, virtual model, mock-up, etc.

[4] The reason for adopting the IEC 62061 function specification format while otherwise following ISO 13849-1 is that ISO 13849-1 does not provide such a systematic function specification template as IEC 62061. The specification template is not presented in this report.

[5] Very often a safety function is considered to consist of three SRP/CSs: input, logic and output. PL is evaluated for each of them and the combined PL is calculated according to the rules of ISO 13849-1.

[6] Note that the SISTEMA tool by BGIA calls these subsystems. As ISO 13849-1 does not use the term ‘subsystem’, we simply call them SRP/CSs and, in fact, SISTEMA treats them as SRP/CSs according to ISO 13849-1 even though it uses diverse terminology.

Title of the artefact type	Description
Normal channel	A normal functional channel from input (like sensors) to output (like hydraulic actuators) through logic (like programmable logic controller)
Requirement	The Requirement artefact defines a requirement and its attributes, like type and source. A requirement can be a stakeholder requirement or a system requirement.
Safety block	A set of system elements that constitute a specific portion of the overall safety related part of the control system
Safety function	A special case of a system function specified to provide a functional safety measure, like safety related stop, prevention of unexpected start-up and hold-to-run function
Safety related block diagram	A special case of diagram that illustrates the logical structure of the safety related part of the control system according to Annex B of ISO 13849-1:2008
Safety-related part of control system	“Part of a control system that responds to safety-related input signals and generates safety-related output signals” (ISO 13849-1:2008)
Structure view	Structure view provides a means to present different views to the system’s physical architecture. The views can be partial views of the whole system structure or different viewpoints, like development time view and manufacturing time view
System element	A System element can be a sub-system or a component. System elements together constitute the system of interest. The typical distinguishing factor between a sub-system and a component is the existence of a part number: a subsystem does not normally have a part number.
System function	A system level function, e.g. boom movement
System parameter	The System parameter artefact stores any kind of system parameters to configure e.g. a system function, system element or a port. Note that this parameter affects the system design; a separate parameter, Individual parameter, is used to configure system individuals.
Test channel	A special case of a channel that performs monitoring of the normal channel.

4.4 Structure package

The Structure package provides models for the physical architecture of the system. The Structure view model allows several views to the system's physical architecture artefacts and their relations. The Structure view model is presented in Section 4.4.1. The Structure package also includes a draft version of the Network model to define artefacts and their relations of communications networks. Currently, only CANopen network is covered. The Network model is presented in Section 4.4.2.

4.4.1 Structure view concept model

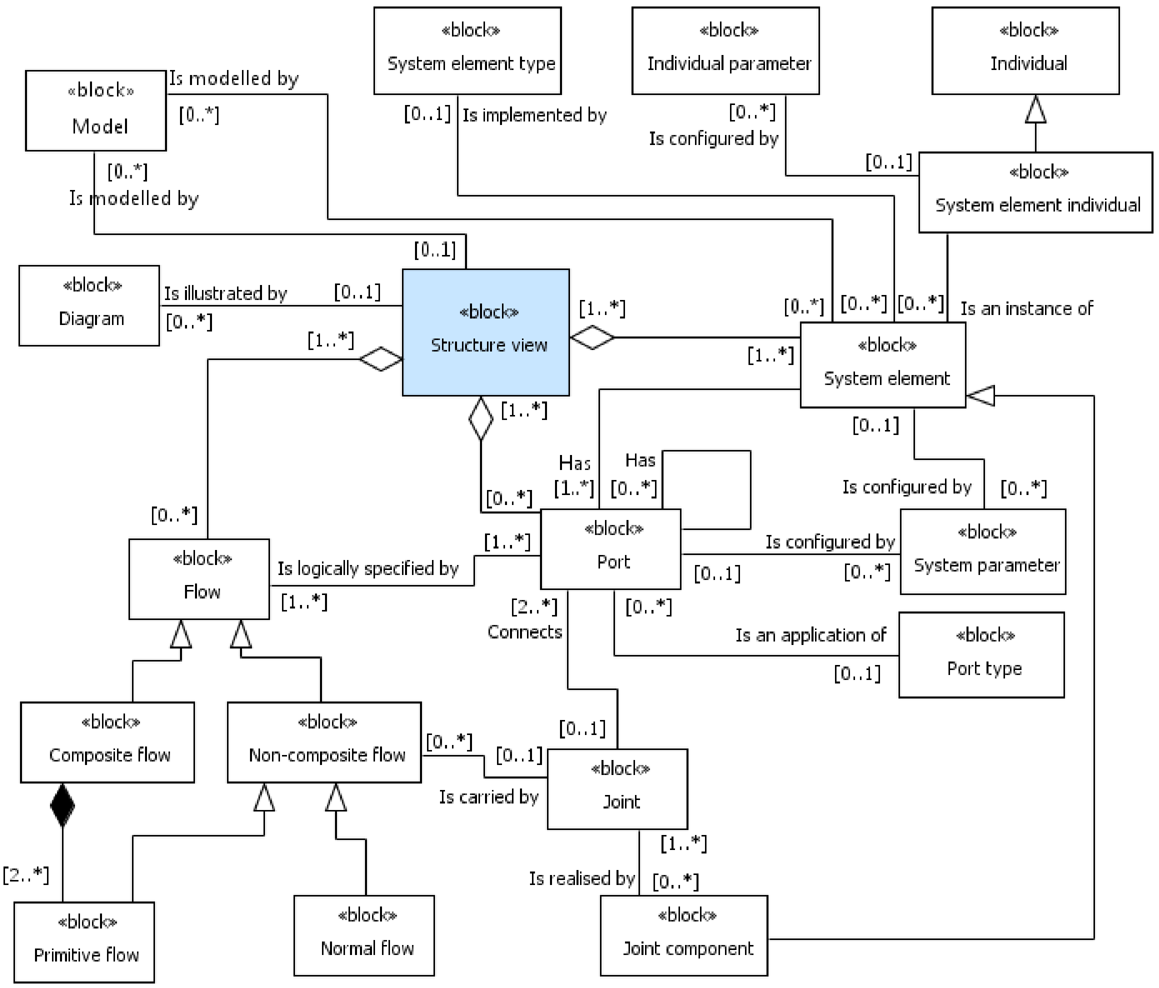

Figure 14. Structure view concept model diagram.

The Structure view concept model defines the physical structure artefacts and their relations. The main artefact is the System view artefact that provides different views to the structure of the system.

The model of the system structure is based on the AP233 system structure concept model presented in Figure 15.

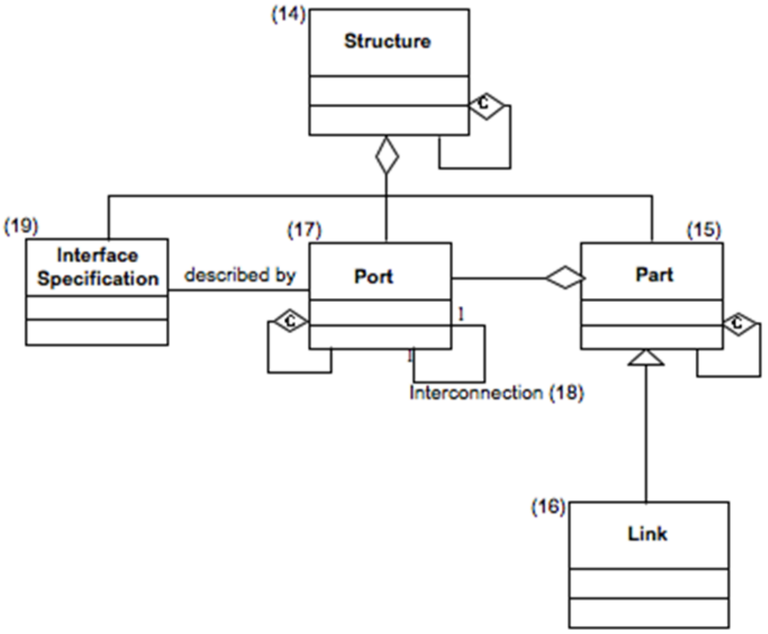

Figure 15. System structure concept model of AP233 (ISO/DIS 10303-233:2009).

The main idea in SEAModel is that Structure view is defined by its System elements (Parts in the AP233 model above) and their interfaces. The interfaces are modelled by Ports specified by Flows that flow through the ports. Port has can have one or more sub-port, e.g. a connector port can have several electrical pin ports. Ports can be connected together with Joint. A joint is a logical entity that may manifest itself in the real world as Joint components (specialisations of System element), e.g. as a mechanical rod or as a cable with cable splices etc. that are defined during the electrical CAD work.

In principle, the association from Structure view to Port seems unnecessary, because a structure view consists of all the Port artefacts under the System element artefacts anyway. In the model, however, a port can be directly associated with a structure view if necessary, and even such that the Port is not linked to any Part. It should be noted that the structure model of AP233 in Figure 15 has the same model in this respect: a structure consists of parts and ports, and a part consists of ports. The AP233 standard goes even further: a structure also consists of interface specifications, which in SEAModel are called Flows. SEAModel also provides such an association, and this association is in fact useful in the case of system functions model (see Section 4.3.2); a system function can be associated with a structure of its own such that the system elements, ports and flows that are used and needed by the particular system function are pointed out. Without the direct relation from Structure view to Port and Flow a system function structure would be impossible to present, because a system element inherits all its ports and a port inherits all its sub-ports, not only the ports and sub-ports that are relevant to the particular system function. Such a system function specific structure view is a partial structure of the whole system structure and is required e.g. by the safety analyst. It is also helpful for the maintenance persons to see, which system elements, ports and flows need to be faultless for the function to work correctly.

A structure view and a system element can be illustrated in one or more Diagrams, or it can be modelled by several models, like a block definition diagram and internal block diagram according to SysML. The functionality and characteristics of a system element and a port can be configured by one or more System parameters.

There are basically three types of Flows: Normal flow, Primitive flow and a Composite flow. The concept of a composite flow is needed in cases in which the actual flow is composed of two or more flows, which we call primitive flows. Such an example is a quadrature encoder sensor in which the position signal flow is composed of two primitive flows, Channel A pulses and Channel B pulses.

Flow is mapped to Joint for the purpose of risk analysis: During signal-based HAZOP, the cause of a deviation can be pointed out in the model, e.g. it can be shown that a possible cause of a deviation ‘no signal’ is a break in the joint between two ports. If, however, a more detailed estimation about the probability of the connection break is needed for the safety analysis, the Joint element artefact is consulted. In the case that the joint element has not yet been designed, the analysis may provide requirements for the structure and quality of the joint components. It is of course suggested that the risk analyses of system functions are carried out before implementation of the joint elements.

A system element is an application (an instance) of a system element type, and a port is an application (an instance) of a port type. For electrical ports it is quite normal to provide several functionalities for a single port type, like the I/O types analogue input and digital input, and these can be configured according to the application needs. Hence an electrical port cannot simply inherit the I/O type of the port type (because it is configurable). To denote the actual I/O type used in the application, special attributes, like *actual_io_type* and *actual_direction*, are needed in the Port artefact.

Table 9. Description of the artefacts types shown in the Behaviour view concept model diagram (Figure 13).

Title of the artefact type	Description
Composite flow	A special case of a flow where the flow consists of several flows called primitive flows. An example is a quadrature encoder the output of which consists of the two pulse train channels, Channel A and Channel B, with 90 degrees phase shift to each other; the sign of the phase shift report the direction of the rotation whereas the pulses report the rate of travel; hence the actual flow (the composite flow), rotation travel with its sign (direction), is calculated from the two channels.
Diagram	The Diagram artefact can accommodate diagrams, pictures and photographs of any kind. The most important attribute is the link to the actual diagram file.
Document	The Document artefact defines any type of document, except diagrams and photos, which have a dedicated artefact type (Diagram). The most important attribute (or feature) is the link to the actual document file.
Flow	Specifies the flow between ports. A flow can be electrical (signal), hydraulic (fluid), mechanical (momentum), optical (light), etc.
Individual	The Individual artefact records information about the supplied system individuals. Such information can include static (e.g. serial number) and dynamic information (e.g. operation hours).
Individual parameter	Besides the information provided by the Individual artefact and its attributes, a set of individual parameters (colour, existence of options etc.) can be assigned to a system individual.
Joint	The logical connection between ports. It does not specify the actual physical implementation of the connection. E.g. in case of electrical connection, it represents the joint galvanic point that connects two or more electrical pins, but it does not specify the wires and cables used to implement the galvanic connection.
Joint component	The physical implementation of the joint
Non-composite flow	A flow that does not constitute of several primitive flows
Normal flow	An atomic flow that does not need any other flows to constitute a flow. It Is thus a special case of Non-composite flow.
Port	The interfaces are modelled by the Port artefacts. Port is a logical interface entity that is specified in detail by the flows it can carry.
Port type	The port type specifies (with its specification parameters) the interfaces of the system element type. A port type may consist of sub-ports. A typical case is a connector with several pins.
Primitive flow	A flow that does not carry the actual flow itself but needs other primitive flows to constitute a flow (a composite flow). A primitive flow is atomic and is thus a special case of Non-composite flow.
Requirement	The Requirement artefact defines a requirement and its attributes, like type and source. A requirement can be a stakeholder requirement or a system requirement.

Title of the artefact type	Description
Structure view	Structure view provides a means to present different views to the system's physical architecture. The views can be partial views of the whole system structure or different viewpoints, like development time view and manufacturing time view
System element	A System element can be a sub-system or a component. System elements together constitute the system of interest. The typical distinguishing factor between a sub-system and a component is the existence of a part number: a subsystem does not normally have a part number.
System element individual	The System element individual artefact is a specialisation of the Individual artefact with no additional attributes.
System element type	The system element type artefact contains the overall identification and description of the library component or sub-system, or of the published system-of-interest.
System parameter	The System parameter artefact stores any kind of system parameters to configure e.g. a system function, system element or a port. Note that this parameter affects the system design; a separate parameter, Individual parameter, is used to configure system individuals.

4.4.2 Network concept model (CANopen case)

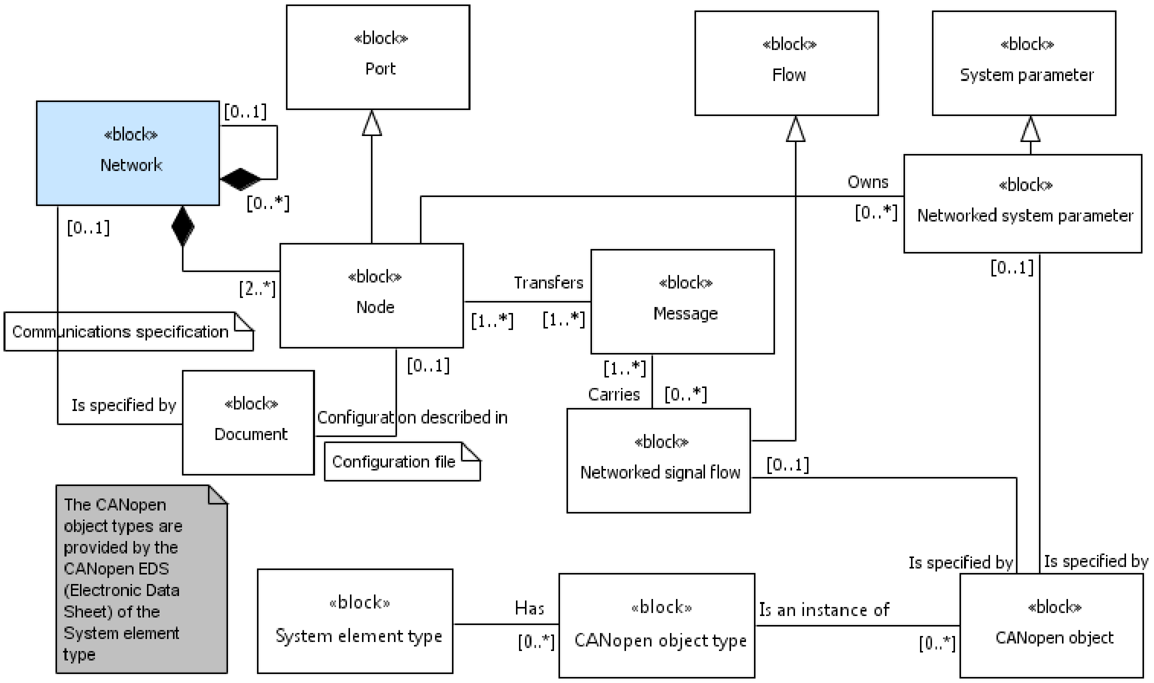

Figure 16. Network concept model diagram (draft).

The Network concept model defines artefacts and their relations of communications networks. The model in Figure 16 only supports CANopen networks. However, the main artefacts, Network, Node, Message, Networked signal flow and Network system parameter are supposed to be applicable to other communication protocols as well.

Network consists of Nodes and sub-networks. A node is a special case of Port (see description of Ports in Section 4.4.1). A network is specified by a protocol specification and by a message specification[7] structured as Message artefacts. Messages are owned by Nodes. A message can carry one or more Networked signal flows. A Networked signal is a special case of Flow (see description of Flows in Section 4.4.1). A networked signal flow is mapped to CANopen object (in the case of

[7] The protocol specification and message specification together constitute the communications specification.

CANopen networks). A CANopen object is an instance of CANopen object type owned by System element type.

Furthermore, some of the system parameters reside on remote nodes. Hence to make them accessible, Networked system parameters are defined as a special case of System parameters. In the case of CANopen, access by such parameters is through the SDO-service. The necessary parameters to do this can be found in CANopen object (actual value) and in CANopen object type (other parameters).

Note that this model is not complete and has neither been tested nor demonstrated. The model above was created to ensure that the Structure view concept model can be linked with the Network concept model.

Table 10. Description of the artefacts types shown in the Network concept model diagram (Figure 16).

Title of the artefact type	Description
CANopen object	CANopen object is an information item (an instance of a CANopen object type) carried by a communications message.
CANopen object type	CANopen object type according to CiA DS301 and DSP 311
Document	The Document artefact defines any type of document, except diagrams and photos, which have a dedicated artefact type (Diagram). The most important attribute (or feature) is the link to the actual document file.
Flow	Specifies the flow between ports. A flow can be electrical (signal), hydraulic (fluid), mechanical (momentum), optical (light), etc. Flows specify the ports (i.e. they inform, what type of flows the ports are able to transmit or receive)
Message	Specifies a communications message.
Network	A communication network segment with dedicated physical layer and data link layer parameters.
Networked signal flow	A signal flow that goes through at least one network segment.
Networked system parameter	A system parameter that can be accessed through a communication network
Node	Node is a port to a communications network.
Port	The interfaces are modelled by the Port artefacts. Port is a logical interface entity that is specified in detail by the flows it can carry.
System element type	The system element type artefact contains the overall identification and description of the library component or sub-system, or of the published system-of-interest.
System parameter	The System parameter artefact stores any kind of system parameters to configure e.g. a system function, system element or a port. Note that this parameter affects the system design; a separate parameter, Individual parameter, is used to configure system individuals.

4.5 Specialty engineering package

The Specialty engineering package contains the models for speciality engineering artefacts and their relations. Speciality engineering includes issues like risk assessment (human safety), security, usability, sustainability engineering, human factors engineering (ergonomics), dependability and logistics. Currently, concept model for risk assessment has been designed (see Section 4.5.1).

4.5.1 Risk Assessment concept model

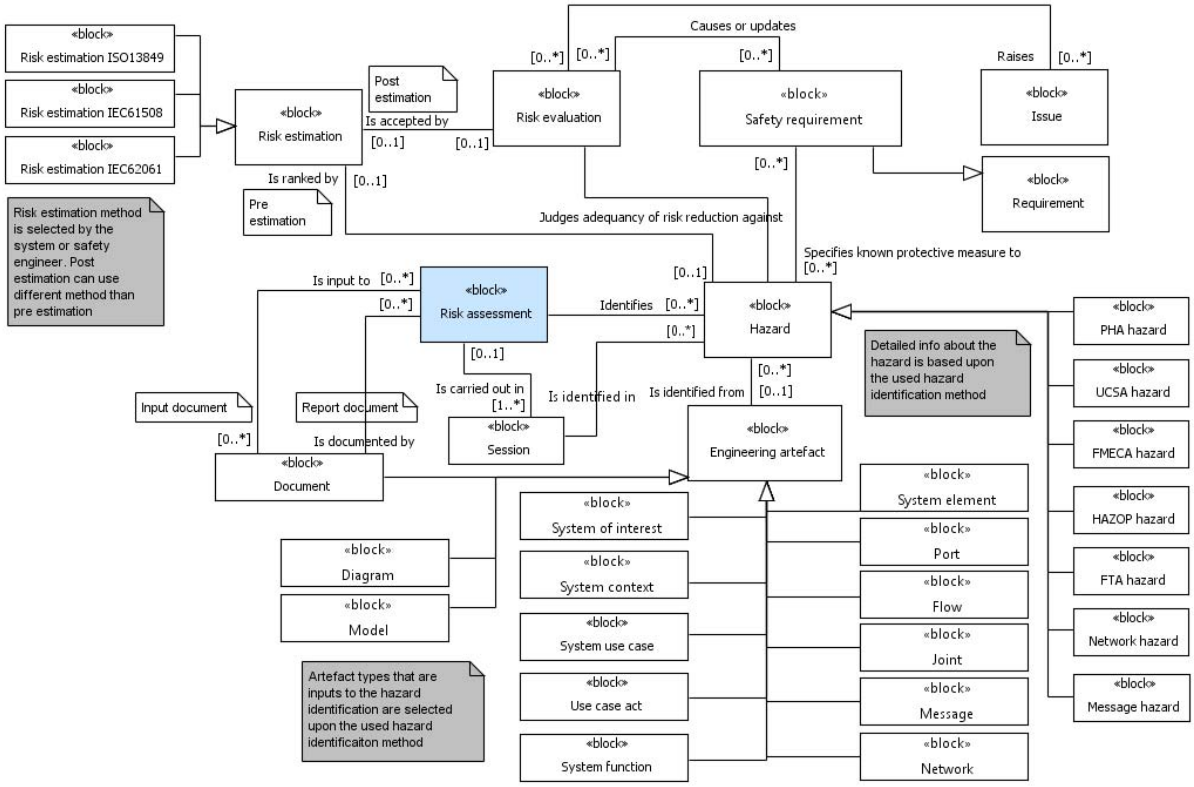

Figure 17. Risk assessment concept model diagram.

The risk assessment concept model in Figure 17 is based on the ISO 12100:2010 risk assessment model depicted in Figure 18.

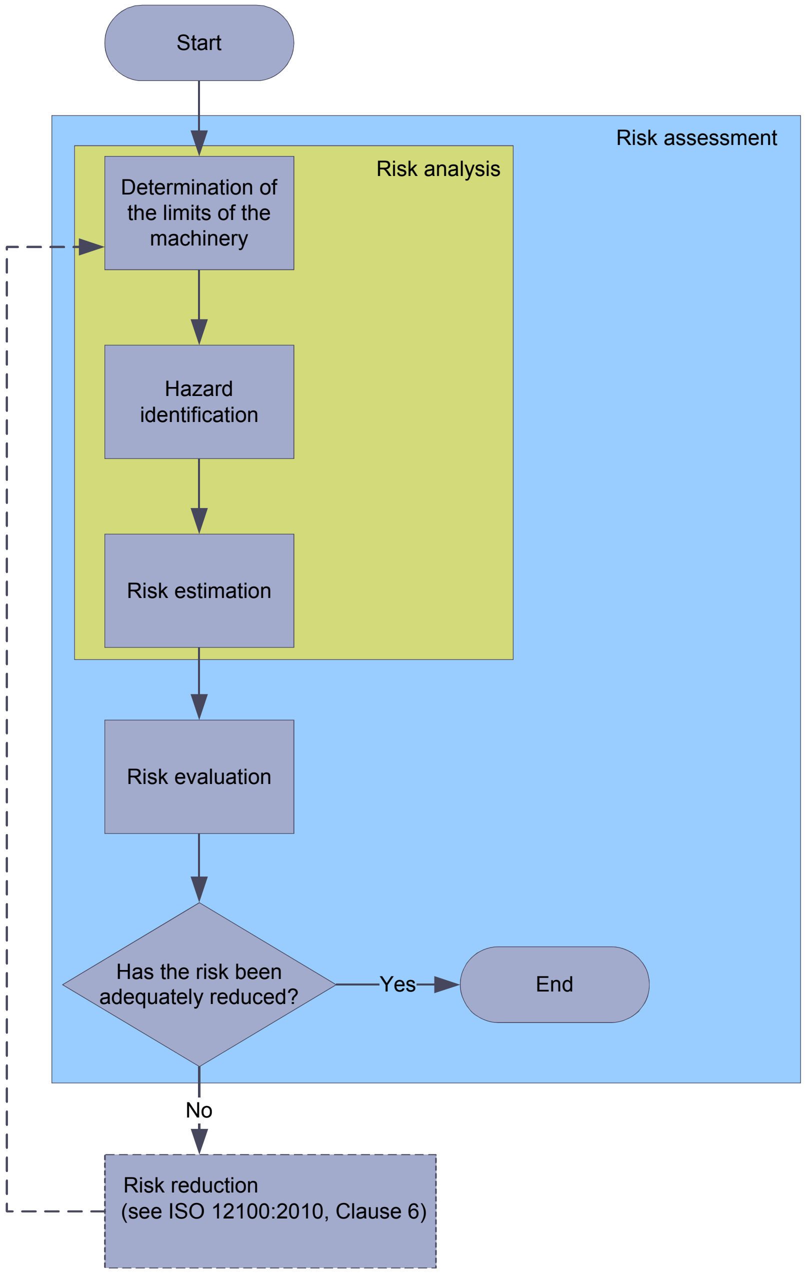
Start
Risk assessment
Risk analysis
Determination of the limits of the machinery
Hazard identification
Risk estimation
Risk evaluation
Has the risk been adequately reduced?
Yes
End
No
Risk reduction
(see ISO 12100:2010, Clause 6)

Figure 18. Risk assessment process model according to ISO 12100:2010 (Risk analysis = combination of the specification of the limits of the machine, hazard identification and risk estimation; Risk estimation = defining likely severity of harm and probability of its occurrence; Risk evaluation = judgment, on the basis of risk analysis, of whether the risk reduction objectives have been achieved; definitions from ISO 12100).

The generic information concerning the risk assessment is stored in the Risk assessment artefact. It specifies e.g. the type of risk analysis used for the particular assessment; currently the following analysis types are supported: Preliminary Hazard Analysis (PHA), Use Case Safety Analysis (UCSA), Failure Mode Effects and Criticality Analysis (FMECA), Fault Tree Analysis (FTA), Hazard and operability study (HAZOP), message safety analysis and network safety analysis.

A risk assessment is performed in several analysis sessions the minutes of which are recorded in the Session artefact. During the PHA-, UCSA- etc. sessions, Hazards are identified based on the analysis type specific methods. The source information for the analyses is typically found among the following set of design artefacts:

- System-of-interest (typically for PHA)
- System context (typically for PHA)
- System use case (typically for PHA and UCSA [the supplemented use case])
- Use case act (for UCSA only)

- System function (typically for FMECA, FTA and HAZOP)
- Flow (typically for HAZOP)
- System element (typically for FMEA)
- Port (typically for FMECA, but only for rare cases if any)
- Joint ([or in practice Joint component, i.e. a system element] typically for FMEA)
- Message (for message safety analysis only)
- Network (for network safety analysis only)
- any other artefact, like Document, Diagram and Model.

The analysis will result in different types of Hazards. In the model, they are categorised according to the analysis type that revealed the hazard. Hence there are seven special cases of the Hazard artefact:

- PHA Hazard
- UCSA Hazard
- FMEA Hazard
- HAZOP Hazard
- FTA Hazard
- Message Hazard
- Network Hazard.

After a hazard has been identified, its risk will be estimated and recorded in the Risk estimation artefact. The model enables several alternatives for the risk estimation method; currently the risk estimation methods of IEC 61508, IEC 62061 and ISO 13849-1 are supported. The risk estimation method is determined by an attribute in the Risk assessment artefact; this attribute is set by the systems engineer or the safety engineer.

Corrective actions will be recommended if the existing protective measures are not sufficient to reduce the risk. The existing protective measures must be evidenced in the form of existing safety requirements and linked to the particular hazard; e.g., during analysis sessions a person may point out that there is an overload limiting device in the system and thus claims that the risk of the identified hazard is negligible. The analyst must not simply write down the claimed protective measure, but he must browse the Requirement artefacts (a database or similar storage in practice) and pick up the requirement for the overload limiter and link the requirement to the hazard. This ensures that if a change is made to the specifications, e.g. the requirement for the overload limiter is removed, the particular hazard automatically becomes suspect, and an update to the particular analysis of the hazard is promptly requested.

Recommendations for corrective actions will be handed over to a team of evaluators who will judge upon the adequacy of the suggested risk reduction measures and decide upon the final implementation of the protective measures against the identified hazard. The judgement is recorded in the Risk evaluation artefact, but the actual result of the risk evaluation is one or more new safety requirements (if needed). The resulting safety requirements are not necessarily a direct copy of the corrective action recommendations by the risk analysis team, but may be modifications of the corrective action recommendations. Hence the Risk evaluation artefact includes rationale on the modifications or direct acceptance of the corrective action recommendations. The resulting safety requirements are linked to the risk evaluation to provide a trace to the hazard causing the safety requirement. In the end, the particular safety requirements are validated according to the requirements model in Section 4.2.1.

However, there are cases in which risk evaluation may lead to a change in the original specifications instead of creating new safety requirements; e.g., the risk analysis team may recommend equipping the machine with a collision avoidance system, but the risk evaluation team may find it too expensive to implement and creates an issue to change the original specifications, e.g. to strip off features that are difficult to implement cost-efficiently with an acceptably low safety risk.

The evaluator team together with the safety engineer can redo the risk estimation to ensure that the acceptable risk level has been reached with the stated new safety requirements.

The communications analysis is performed in two parts: a Message safety analysis and a Network safety analysis. The former is performed according to the model of (IEC 61784-3:2007) and the latter according to the network validation questions by the Swedish Pålbus-project (Hedberg and Wang 2001) with VTT modifications.

Besides the well-structured input artefacts, one or more Documents may be provided for the analysis team as input to the analysis. Such documents include e.g. the relevant safety standards.

The results of the risk assessment are recorded in a Document artefact, e.g. in a collective Risk Assessment Report.

The artefact types (i.e. the 'blocks') in Figure 17 are described in Table 11.

Table 11. Description of the artefacts types shown in the Risk assessment concept model diagram (Figure 17).

Title of the artefact type	Description
Diagram	The Diagram artefact can accommodate diagrams, pictures and photographs of any kind. The most important attribute is the link to the actual diagram file.
Document	The Document artefact defines any type of document, except diagrams and photos, which have a dedicated artefact type (Diagram). The most important attribute (or feature) is the link to the actual document file.
Engineering artefact	An Engineering artefact can be any artefact that is a result of the engineering work during the life cycle of the system.
Flow	Specifies the flow between ports. A flow can be electrical (signal), hydraulic (fluid), mechanical (momentum), optical (light), etc. Flows specify the ports (i.e. they inform, what type of flows the ports are able to transmit or receive)
FMECA hazard	Hazard identified by the Failure Modes, Effects and Criticality Analysis method
FTA hazard	Hazard identified by the Fault Tree Analysis method
Hazard	The Hazard artefact records the description of the hazards identified during the analysis sessions
HAZOP hazard	Hazard identified by the Hazard and operability study method
Issue	Any kind of issue prompting actions by the development organisation, e.g. to change specifications or to do redesign.
Joint	The logical connection between ports. It does not specify the actual physical implementation of the connection. E.g. in case of electrical connection, it represents the joint galvanic point that connects two or more electrical pins, but it does not specify the wires and cables used to implement the galvanic connection.
Message	Specifies a communications message.
Message hazard	Hazard identified by message safety analysis
Model	The Model artefact represents any kind of model, SysML model, virtual model, mock-up, etc.
Network	A communication network segment with dedicated physical layer and data link layer parameters.
Network hazard	Hazard identified by network safety analysis
PHA hazard	Hazard identified by the Preliminary Hazard Analysis method
Port	The interfaces are modelled by the Port artefacts. Port is a logical interface entity that is specified in detail by the flows it can carry.
Requirement	The Requirement artefact defines a requirement and its attributes, like type and source. A requirement can be a stakeholder requirement or a system requirement.
Risk assessment	The Risk assessment artefact works as an assignment to carry out a risk assessment task. It, however, also contains a short description of the results of the risk assessment (the actual results are reported in comprehensive analysis reports).

Title of the artefact type	Description
Risk estimation	The Risk estimation artefact stores the risk estimation parameters and the resulting risk level
Risk estimation IEC 61508	Risk estimation parameters and risk level according to IEC 61508-5:2010
Risk estimation IEC 62061	Risk estimation parameters and risk level according to IEC 62061:2006
Risk estimation ISO 13849	Risk estimation parameters and risk level according to ISO 13849-1:2006
Risk evaluation	The Risk evaluation artefact stores the judgment, on the basis of the risk analysis, of whether the risk reduction objectives have been achieved
Safety requirement	Safety requirements are a special case of normal requirements and are simply distinguished from normal requirements by a True/False flag or by a Requirement type attribute.
Session	The session meeting minutes are recorded in the Session artefact.
System context	The System context artefact provides description e.g. about the following issues requested by ISO 12100:2010: ergonomic principles, energy sources, space limits, life limit, service intervals, other time limits, housekeeping policy, material properties, other limits, external systems interaction and experience of use. The System context artefact also works as the main node to collect the system context related artefacts through associations.
System element	A System element can be a sub-system or a component. System elements together constitute the system of interest. The typical distinguishing factor between a sub-system and a component is the existence of a part number: a subsystem does not normally have a part number.
System function	A system level function, e.g. boom movement
System of interest	The System of interest artefact defines the system under development and during all the life cycle phases. It only includes a small number of attributes, mainly title and a short description of the system, i.e. the system identification.
System use case	A system level use case. Use cases can be specified according to SysML, but the system use case descriptions can be supplemented by safety related information.
UCSA hazard	Hazard identified by the Use Case Safety Analysis method
Use case act	The sequence of acts of a system use case is stored in the Use case act artefacts. A use case act is an atomic piece of a work task. System functions are often identified from the use case acts.

5. Modelling and simulation artefacts in SEAModel

In this chapter, a short discussion is provided on management of modelling and simulation artefacts in SEAModel. More comprehensive work on this subject will be carried out in the later phases of the SIMPRO project (in Task 3.2 of the VTT subproject).

The modelling artefacts are implemented in SEAModel by a single artefact type Model. The Model artefact can be linked to several key artefacts, like System-of-interest, System context, Requirement, System use case, System task, System function, System element and System element type.

In general, the simulation artefacts constitute of the following:

- **simulation requirements** (the requirements which state that simulation has to be carried out; more detailed requirements as to how simulation has to be carried out; the requirements can include rationale for the simulations)

- **simulation plans** (the plans for the simulation process, what is the purpose (the rationale) of the simulations, what are the items under simulation, simulation strategy, list of simulation cases, simulation schedule, simulation personnel)
- **simulation case specifications** (the rationale for the simulation case, the exact specifications of the simulation steps, list of tools to perform the simulation case, environment of the simulation, expected results)
- **simulation model** (an executable, purpose driven, view of the item [like model] under verification; note that simulation model is not needed if the actual model supports simulation of the characteristics under verification)
- **simulation model parameters** (the parameters that relate to the item under verification, and that are relevant for the simulation, are called as simulation model parameters; in principle the parameters of the item under analysis are changed, and thereupon the simulation model parameters are updated accordingly, but in some case where the actual model does not support simulation, like a physical prototype, the simulation model parameters are changed during the simulation, and the parameters of the model are updated according to the simulation results)
- **simulation execution parameters** (the parameters relating to the execution of the simulation, like the number of simulation runs)
- **simulation results** (the simulation results can include virtual measurement data, animation videos, pictures or sound files)
- **simulation results interpreter** (an instruction document or a program that provides means for interpreting the simulation results, like a program that plots a graph of the virtual measurement data).

Besides the artefacts above, the common artefacts like requirements and design artefacts are relevant during the simulation process. The design artefacts are the ones that are verified or validated by the simulation cases, and the requirements are the ones against which the justification of correct design, based on the simulation results, is done.

The following table outlines, how the simulation artefacts above can be mapped onto SEAModel. The relevant model in SEAModel is the requirement concept model presented in Section 4.2.1.

Table 12. Mapping of simulation related artefacts onto SEAModel.

Simulation artefact	Corresponding artefact in SEAModel
Simulation requirement	Verification or validation requirement
Simulation plan	TestAnalysisEtc plan
Simulation case specification	TestAnalysisEtc case
Simulation model	Model in most cases
Simulation model parameters	(System parameter)
Simulation execution parameters	No corresponding artefact. TestAnalysisEtc case can be used to define the parameters
Simulation results	TestAnalysisEtc execution report
Simulation results interpreter	No corresponding artefact

Table 12 points out that three new artefact types in SEAModel are needed to comprehensively support traceability of the simulation artefacts:

- **Verification model parameters** (if System parameter cannot be used directly; encompasses simulation model parameters)
- **Verification execution parameters** (encompasses simulation execution parameters)
- **Verification results interpreter** (encompasses simulation results interpreter).

Adding of the artefact types above into SEAModel is studied in future tasks of the SIMPRO project.

6. Tools to support SEAModel

Practical implementations of SEAModel require a software tool with the following features:

- Structured artefact repository where relations between the artefacts can be created and maintained
- Artefacts traceability with impact analysis
- Version control of artefacts and of a set of artefacts (with “baselines” feature)
- Modification control
- Automatic document generation
- Document management (or seamless integration to an existing document management system)
- Integration possibility with systems engineering tools like requirements management tools, CAD-tools, SW programming tools etc.
- Concurrent engineering capabilities
- Collaboration features with wiki-pages, task lists, discussion boards, announcements, etc.
- Metrics of various systems engineering issues, e.g. how many of the system requirements are covered by the design artefacts.

Besides the above features, the following issues need to be considered when selecting the tool:

- Cost
- Responsiveness (especially important when playing with the traceability features)
- Usability, user experience.

Typical tools to support at least the core set of the features above are among PLM (Product Lifecycle Management) tools and ALM (Application Lifecycle Management) tools. The main difference between these tool is that PLM tools are CAD-oriented, whereas ALM tools are software development oriented. Examples of PLM tools are ARAS PLM, Eurostep Share-A-Space, Dassault Enovia (CATIA V6), PTC Windchill and Siemens Teamcenter. Examples of ALM tools are IBM RELM[8] (Rational Engineering Lifecycle Management) with other Rational tools, Polarion ALM and Microsoft TFS (Team Foundation Server).

Some of the needed features can be achieved by a requirement management software or with database oriented content management systems (CMS) like MS SharePoint. In the context of the SIMPRO project, a trial will be made to integrate SharePoint with the THTH’s Simantics to provide a platform for tracing e.g. simulation artefacts according to SEAModel. SharePoint provides a well-known user interface, collaboration features and document management features, whereas Simantics is planned to provide the artefact repository with trace links. Furthermore, Simantics provides the platform for integration of CAD- and simulation tools with the artefact repository. Based on the Simantics trials with SharePoint, decision of the final SEAModel implementation platform for SEAModel demonstration will be made in Task 3.3 of the SIMPRO VTT sub-project.

[8] RELM is actually a Systems Engineering lifecycle management tool.

7. Conclusions

SEAModel provides a sound basis for tracing design and implementation artefacts to requirements, and from test and simulation cases and executions to the design and implementation artefacts. Furthermore, risk assessment related artefacts can be managed in a structured manner by using the model.

SEAModel is implementable onto a database oriented tool, like a PLM tool or a requirements management tool. To facilitate traceability of requirements and verification results between the main contractor and sub-contractor, all participating organisations should have access to the artefact repository that has been structured according to SEAModel.

The next step is to test and demonstrate SEAModel on selected platform, especially in the case of virtual engineering that applies simulation in verification and validation phase.

8. Summary

A reference model for systems engineering main artefact types was created. The model is called SEAModel (Systems Artefacts Engineering Model). The model consists of five model packages, System, Requirement, Behaviour, Structure and Specialty Engineering packages.

System package consists of System-of-interest concept model, System context concept model and System element type concept model.

Requirement package consists of Requirement concept model.

Behaviour package consists of Behaviour view concept model and System function concept model.

Structure package consists of Structure view concept model and Network concept model (CANopen case).

Specialty engineering package consists of Risk Assessment concept model.

For each of these models a set of artefact types and their relations are presented as SysML block definition diagrams. Furthermore, each of the artefact types are described in more detailed in tabular form. Artefact type attributes are not listed.

A short discussion on using requirement templates is provided. Furthermore, discussion on applying SEAModel in case of virtual engineering is supplied.

References

Alanen, J., Vidberg, I., Nikula, H., Papakonstantinou, N., Pirttioja, T. and Sierla, S. 2011. Engineering data model for machine automation systems. *VTT Tiedotteita – Valtion Teknillinen Tutkimuskeskus,* (2583), pp. 1–137.

Hull, E. Jackson, K. and Dick, J. 2010. Requirements engineering. DE: Springer.

Hedberg, H. and Wang, Y. 2001. PALBUS Work Package 10.10. 2001. Methods for Verification and Validation of Distributed Control Systems. Borås: SP Swedish National Testing and Research Institute. 66 p.

IEC 61784-3:2007. Industrial Communications – Fieldbus Profile – Part 3: Profiles for functional safety communications in industrial networks. 2007-12 ed. Geneva: International Electrotechnical Commission. 47 s.

IEC 62061:2005. Safety of machinery – Functional safety of safety-related electrical, electronic and programmable electronic control systems. 2005-01-20 ed. Geneva: International Electrotechnical Commission. 205 s.

ISO 12100:2010. Safety of machinery – General principles for design – Risk assessment and risk reduction. 2010-10-20 ed. Geneva: International Organisation for Standardization. 77 s.

ISO 13849-1:2006. Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design. 2006-10-30 ed. Geneva: International Organisation for Standardization. 85 s.

ISO/DIS 10303-233:2009. Industrial automation systems and integration – Product data representation and exchange – Part 233: Application protocol: Systems engineering. 2009-10-12 ed. Geneva: International Organisation for Standardization.

Mavin, A. and Wilkinson, P. 2010. Big Ears (The Return of "Easy Approach to Requirements Engineering"). Requirements Engineering Conference (RE), 2010. 18th IEEE International 2010, pp. 277–282.

MSR Consortium 2002. MSRSYS – ECU control system description [Homepage of Association for Standardisation of Automation and Measuring Systems (ASAM e.V.)], [Online]. Available: http://www.asam.net/images/stories/Standards/AE/AAS/asam_ae_aas_msrsys_v1.2.0.zip [2010, 01/29] .

Rupp, C. and a group of SOPHIST employees. 2011. Requirements Engineering und Management. 5th edition. SOPHIST GmbH.

Appendix 1. Artefact types

Behaviour view *1. The Behaviour view artefact is used to provide different perspectives to system functional architecture. It can be used to provide views to categorised sets of functionality or to completely different functionalities of machines with dual or more usage purposes.*

CANopen object *2. CANopen object is an information item (an instance of a CANopen object type) carried by a communications message.*

CANopen object type *3. CANopen object type according to CiA DS301 and DSP 311.*

Channel *4. A logical entity that encompasses the signal processing flow from input to output through logic.*

Composite flow *5. A special case of a flow where the flow consists of several flows called primitive flows. An example is a quadrature encoder the output of which consists of the two pulse train channels, Channel A and Channel B, with 90 degrees phase shift to each other; the sign of the phase shift report the direction of the rotation whereas the pulses report the rate of travel; hence the actual flow (the composite flow), rotation travel with its sign (direction), is calculated from the two channels.*

Constraint *6. Constraint is a special case of a requirement. In one sense, it is not a requirement, because it only states facts e.g. about the system's environment (like the dimensions of the space where the system will be installed); on the other hand, its effect in the design is similar to that of requirements.*

Diagram *7. The Diagram artefact can accommodate diagrams, pictures and photographs of any kind. The most important attribute is the link to the actual diagram file.*

Document *8. The Document artefact defines any type of document, except diagrams and photos, which have a dedicated artefact type (Diagram). The most important attribute (or feature) is the link to the actual document file.*

Electrical port type *9.The electrical case of the Port type artefact.*

Engineering artefact *10. An Engineering artefact can be any artefact that is a result of the engineering work during the life cycle of the system.*

Environment *11. Description of the mechanical, climatic, chemical, ergonomic and external system environment, especially in regard to their effect in the system of interest. Domain knowledge can be described here, too.*

Flow *12. Specifies the flow between ports. A flow can be electrical (signal), hydraulic (fluid), mechanical (momentum), optical (light), etc. Flows specify the ports (i.e. they inform, what type of flows the ports are able to transmit or receive).*

FMECA hazard *13. Hazard identified by the Failure Modes, Effects and Criticality Analysis method.*

FTA hazard *14. Hazard identified by the Fault Tree Analysis method.*

Glossary item *15. Definitions, terms and abbreviations are presented in the Glossary item artefact.*

Hazard *16. The Hazard artefact records the description of the hazards identified during the analysis sessions, the existing protective measures and the corrective actions recommendations.*

HAZOP hazard *17. Hazard identified by the Hazard and operability study method.*

Human actor *18. Any human actor that interacts with the system, whether an assembly man, operator, maintenance man, cleaner, etc., or a bystander who is situated in the hazard zone of the system.*

Hydraulic port type *19. The hydraulic case of the Port type artefact.*

Incident *20. A record of accidents and incidents history, including near misses, of the this type of systems or similar system types.*

Individual *21. The Individual artefact records information about the supplied system individuals. Such information can include static (e.g. serial number) and dynamic information (e.g. operation hours).*

Individual parameter *22. Besides the information provided by the Individual artefact and its attributes, a set of individual parameters (colour, existence of options etc.) can be assigned to a system individual.*

Issue *23. Any kind of issue prompting actions by the development organisation, e.g. to change specifications or to do redesign.*

Joint *24. The logical connection between ports. It does not specify the actual physical implementation of the connection. E.g. in case of electrical connection, it represents the joint galvanic point that connects two or more electrical pins, but it does not specify the wires and cables used to implement the galvanic connection.*

Joint component *25. The physical implementation of the joint.*

Life cycle phase *26. The life cycle model is recorded in the Life cycle phase artefact, e.g. Concept, Development, Production, Utilisation, Support and Retirement according to ISO/IEC TR 24748-1:2010.*

Mechanical port type *27. The mechanical case of the Port type artefact.*

Message *28. Specifies a communications message.*

Message hazard *29. Hazard identified by message safety analysis.*

Model *30. The Model artefact represents any kind of model, SysML model, virtual model, mock-up, etc.*

Network *31. A communication network segment with dedicated physical layer and data link layer parameters.*

Network hazard *32. Hazard identified by network safety analysis.*

Networked signal flow *33. A signal flow that goes through at least one network segment.*

Networked system parameter *34. A system parameter that can be accessed through a communication network.*

Node *35. Node is a port to a communications network.*

Non-composite flow *36. A flow that does not constitute of several primitive flows.*

Normal channel *37. A normal functional channel from input (like sensors) to output (like hydraulic actuators) through logic (like programmable logic controller).*

Normal flow *38. An atomic flow that does not need any other flows to constitute a flow. It Is thus a special case of Non-composite flow.*

Operating mode *39. Different types of control or operating modes can include e.g. the following modes: automatic, manual; remote, local; diagnostics; 'limp home'.*

Operating positions *40. The positions at which the system is controlled and operated.*

Optical port type *41. The optical case of the Port type artefact.*

PHA hazard *42. Hazard identified by the Preliminary Hazard Analysis method.*

Pneumatic port type *43. The pneumatic case of the Port type artefact.*

Port *44. The interfaces are modelled by the Port artefacts. Port is a logical interface entity that is specified in detail by the flows it can carry.*

Port type *45. The port type specifies (with its specification parameters) the interfaces of the system element type. A port type may consist of sub-ports. A typical case is a connector with several pins.*

Primitive flow *46. A flow that does not carry the actual flow itself but needs other primitive flows to constitute a flow (a composite flow). A primitive flow is atomic and is thus a special case of Non-composite flow.*

Requirement *47. The Requirement artefact defines a requirement and its attributes, like type and source. A requirement can be a stakeholder requirement or a system requirement.*

Requirement template *48. A requirement template that forms the body of a requirements sentence.*

Requirement template attribute *49. Requirement template attributes are the system specific terms or system elements that are placed to the placeholders of a requirement template to create a well-formed requirements sentence.*

Risk assessment *50. The Risk assessment artefact works as an assignment to carry out a risk assessment task. It, however, also contains a short description of the results of the risk assessment (the actual results are reported in comprehensive analysis reports).*

Risk assessment *51. The Risk assessment artefact works as an assignment to carry out a risk assessment task. It, however, also contains a short description of the results of the risk assessment (the actual results are reported in comprehensive analysis reports).*

Risk estimation *52. The Risk estimation artefact stores the risk estimation parameters and the resulting risk level.*

Risk estimation IEC 61508 *53. Risk estimation parameters and risk level according to IEC 61508-5:2010.*

Risk estimation IEC 62061 *54. Risk estimation parameters and risk level according to IEC 62061:2006.*

Risk estimation ISO 13849 *55. Risk estimation parameters and risk level according to ISO 13849-1:2006.*

Risk evaluation *56. The Risk evaluation artefact stores the judgment, on the basis of the risk analysis, of whether the risk reduction objectives have been achieved.*

Safety block *57. A set of system elements that constitute a specific portion of the overall safety related part of the control system.*

Safety function *58. A special case of a system function specified to provide a functional safety measure, like safety related stop, prevention of unexpected start-up and hold-to-run function.*

Safety related block diagram *59. A special case of diagram that illustrates the logical structure of the safety related part of the control system according to Annex B of ISO 13849-1:2008.*

Safety requirement *60. Safety requirements are a special case of normal requirements and are simply distinguished from normal requirements by a True/False flag or by a Requirement type attribute.*

Safety-related part of control system *61. "Part of a control system that responds to safety-related input signals and generates safety-related output signals" (ISO 13849-1:2008). It can be electrical, electronic, programmable electronic, mechanical, pneumatic, hydraulic, etc.*

Session *62. The session meeting minutes are recorded in the Session artefact.*

Specification parameter *63. Specifies a parameter of the system type; a 'datasheet parameter', like weight, maximum temperature, etc.*

Stakeholder *64. The Stakeholder artefact lists the stakeholders that may state requirements for the system (i.e. that have interest in the system). Stakeholders can include e.g. system users, domain experts, principal, investors, board of directors, corporate management, authorities, laws, standards, customers, maintenance staff, training staff, system engineer, buyers of the system and marketing and sales.*

Stakeholder requirement *65. A special case of a requirement: requirement set by a stakeholder.*

Structure view *66. Structure view provides a means to present different views to the system's physical architecture. The views can be partial views of the whole system structure or different viewpoints, like development time view and manufacturing time view.*

System context *67. The System context artefact provides description e.g. about the following issues requested by ISO 12100:2010: ergonomic principles, energy sources, space limits, life limit, service intervals, other time limits, housekeeping policy, material properties, other limits, external systems interaction and experience of use. The System context artefact also works as the main node to collect the system context related artefacts through associations.*

System element *68. A System element can be a sub-system or a component. System elements together constitute the system of interest. The typical distinguishing factor between a sub-system and a component is the existence of a part number: a subsystem does not normally have a part number.*

System element individual *69. The System element individual artefact is a specialisation of the Individual artefact with no additional attributes.*

System element type *70. The system element type artefact contains the overall identification and description of the library component or sub-system, or of the published system-of-interest.*

System function *71. A system level function, e.g. boom movement.*

System of interest *72. The System of interest artefact defines the system under development and during all the life cycle phases. It only includes a small number of attributes, mainly title and a short description of the system, i.e. the system identification.*

System of interest individual *73. The System-of-interest individual artefact is a specialisation of the Individual artefact with no additional attributes.*

System of interest type *74. A special case of the System element type artefact; the overall identification and description of the published system of interest.*

System parameter *75. The System parameter artefact stores any kind of system parameters to configure e.g. a system function, system element or a port. Note that this parameter affects the system design; a separate parameter, Individual parameter, is used to configure system individuals.*

System process *76. A special case of a system use case. Describes the sequence of system use cases in an explicit way (contrary to the implicit way in which the sequence of system use cases can be concluded from the post conditions of one use case and preconditions of another use case).*

System task *77. The System task artefact is the system realisation view of the behaviour. It defines the flow of system functions. It is especially useful in cases where human actors are not involved, but it is also used to identify system functions that cannot be identified from system use cases or use case acts.*

System use case *78. A system level use case. Use cases can be specified according to SysML, but the system use case descriptions can be supplemented by safety related information.*

System use case *79. A system level use case. Use cases can be specified according to SysML, but the system use case descriptions can be supplemented by safety related information.*

System use case supplement *80. A safety related supplement that adds attributes like ‘actor qualification’, ‘list of operator instructions’, ‘expected misuse’, ‘activation frequency’, ‘preliminary accident scenario’ to the system use case.*

Test channel *81. A special case of a channel that performs monitoring of the normal channel.*

TestAnalysisEtc case *82. The specification for a single test, analysis, demonstration, review etc. case.*

TestAnalysisEtc execution report *83. Records the results of the test, analysis, demonstration, review etc. case executions.*

TestAnalysisEtc plan *84. The TestAnalysisEtc plan artefact collects a set of test, analysis, demonstration, review etc. cases to form a specific sequence of tests for a specific purpose, such as for Factory Acceptance Test (FAT).*

UCSA hazard *85. Hazard identified by the Use Case Safety Analysis method.*

Use case act *86. The sequence of acts of a system use case is stored in the Use case act artefacts. A use case act is an atomic piece of a work task. System functions are often identified from the use case acts.*

Verification or validation report *87. The result of verification or validation is recorded here. The main content is a simple pass/no-pass verdict.*

Verification or validation requirement *88. A special case of the Requirement artefact. In many cases, the requirements specification or safety standards set requirements as to how the design shall be verified or validated.*