GSN Support of Mixed-Criticality Systems Certification

Carlos-F. Nicolas[1](✉), Fernando Eizaguirre[1], Asier Larrucea[1], Simon Barner[2],
Franck Chauvel[3], Goiuria Sagardui[4], and Jon Perez[1]

[1] IK4-Ikerlan, Mondragon, Spain
{cfnicolas,feizaguirre,alarrucea,jmperez}@ikerlan.es
[2] Fortiss, Munich, Germany
barner@fortiss.org
[3] SINTEF ICT, Oslo, Norway
franck.chauvel@sintef.no
[4] Mondragon Goi Eskola, Mondragon, Spain
gsagardui@mondragon.edu

Abstract. Safety-critical applications could benefit from the standardisation, cost reduction and cross-domain suitability of current heterogeneous computing platforms. They are of particular interest for Mixed-Criticality Product Lines (MCPL) where safety- and non-safety functions can be deployed on a single embedded device using suitable isolation artefacts and development processes. The development of MCPLs can be facilitated by providing a reference architecture, a model-based design, analysis tools and Modular Safety Cases (MSC) to support the safety claims.

In this paper, we present a method based on the MSCs to ease the certification of MCPLs. This approach consists of a semi-automated composition of layered argument fragments that trace the safety requirements argumentation to the supporting evidences. The core of the method presented in this paper is an argument database that is represented using the Goal Structuring Notation language (GSN). The defined method enables the concurrent generation of the arguments and the compilation of evidences, as well as the automated composition of safety cases for the variants of products. In addition, this paper exposes an industrial-grade case study consisting of a safety wind turbine system where the presented methodology is exemplified.

Keywords: Goal Structuring Notation (GSN) · Model-based development · Safety-critical systems · Product lines · Variability

1 Introduction

Modern *Heterogeneous Computing Platform* (HCPs) enable architectural simplification and standardisation across multiple application fields (e.g., automotive, railway, avionics) to implement embedded systems with a homogeneous

S. Tonetta et al. (Eds.): SAFECOMP 2017 Workshops, LNCS 10489, pp. 157–172, 2017.
DOI: 10.1007/978-3-319-66284-8_14

hardware (HW) and *software* (SW). The research on bringing determinism and fault isolation to HCP platforms enable safety-critical applications for heterogeneous processors, while also deploying non-safety-related applications. The cost reduction in multi-purpose HW components fosters a common platform development for multiple domains. However, HCPs lead to interferences in temporal and spatial domains due to their complexity, sophistication and high-performance resources. These interferences challenge the certification of modern HCPs and they are one of the main objectives of today's embedded system developers.

Certification represents the major cost driver in the project budget for developing safety-critical HCP systems. This process is a third-party attestation related to products, processes, systems or persons [14]. An attestation is the issue of a statement based on reviewer's decision that demonstrates the fulfilment of specified requirements or standards. In traditional certification, if a requirement of the systems changes, the whole system is re-assessed. This certification model increments the cost and the time-to-market. Modularity methodology enables dividing the system into independent modules which may be developed and certified with different criticality levels (e.g., Safety Integrity Level (SIL) 1 to 4 according to IEC 61508). This method enables improving the reusability and scalability of the overall system and allows the reduction of the complexity and the certification cost of mixed-criticality systems.

IEC 61508 is the safety-related standard for Electrical/Electronic/Programmable Electronic (E/E/PE) functional safety systems. This standard considers safety as an *emergent system property*, resulting from the inherent safety of its components, the system structure and the interactions between its parts, the operational context, and the development process. Safety standards rooted in IEC 61508 follow a stereotyped development work-flow (V-model development process) with interleaved analysis, refinement and review tasks. The IEC 61508 standard also recommends the use of models to assess the compliance with the established practices where the developer verifies the safety behaviour of the safety-related system.

The development process redundancy is also mandated by IEC 61508 for high integrity systems. The redundancy consists of the separation of concerns, staff roles and artefacts between the design and development and the *Verification and Validation* (V and V) activities. The process redundancy decreases the likelihood of systematic errors relying on diverse interpretations of the requirements. However, in practice, a file-based application environment does not support the concurrent and independent development, which is required to certify high-integrity *Mixed-Criticality Product Lines* (MCPLs) cost-effectively.

In the scope of the European project *Distributed REal-time Architecture for Mixed Criticality Systems* (DREAMS) [8] the safety certification of MCPLs according to the IEC 61508 standard is one of the objectives. This paper presents a shared certification artefact based on a *Database Management System* (DBMS) to overcome the limitation introduced in the previous paragraph. Furthermore, the presented solution provides support for different use-cases for collaborative safety-projects. Those collaborative projects can handle and share safety certificates, evidences and reference documents common to a MCPL.

The project can also collect the arguments and the documents concurrently and semi-automatically optimise the design and post-design of MCPLs.

The paper is organised as follows. Section 2 recalls some related works about product line development. Section 3 introduces the European DREAMS project, presents its architectural style and exposes a toolset for generating and checking safety argumentation models. Section 4 presents a Platform Based Design (PBD) work-flow to design safety product lines. Section 5 exemplifies the integration of the methodology proposed in a safety wind turbine product line system. Section 6 reflects the lessons learned. Finally Sect. 7 presents the conclusion and outlook.

2 Related Work

The key to provide modularity in safety certification are the safety cases. A safety case is a structured argument supported by a body of evidence that provides a compelling, comprehensive and valid case that a system is safe for a given application in a given operating environment. When these components are integrated into a mixed-criticality system, then, specific global safety arguments can also be assembled to show the validity of the safety claims.

The safety case approach is already accepted in safety applications domains like railway applications (according to the requirements of EN 50139 [7]) or air traffic management systems. EUROCONTROL publishes a safety case development manual [11] for aviation management applications, based on *Goal Structuring Notation* (GSN) notation [15]. Other safety standards also allow the use of safety cases, even if there is no specific guidance on the safety case structure or the overall structure for cross-reference.

GSN [15] is a safety case notation language proposed by Kelly to develop, document and maintain safety-cases. It was developed following the Toulmin approach [26]. This notation language uses the goal, strategy, solution, context, assumption and justification elements to express the safety-related requirements of a system (see Fig. 1). In addition, GSN language supports modularity [17,21],

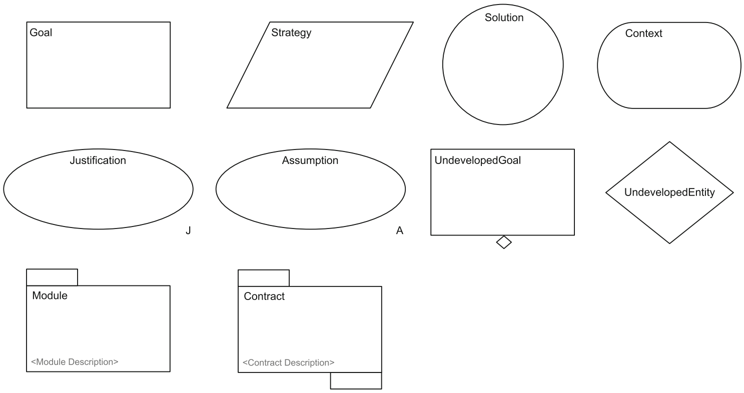

Fig. 1. GSN elements. (Source [16])

adding the module and contract elements. The module element is a package of arguments that abstract the view of the argument structure. The contract element is a package that represents the relationship between two or more modules, defining how a claim in one supports the argument in the other.

On the other hand, the OPENCOSS project [3] tackles the problem of certifying a product for multiple application domains for which different safety standards may apply. Such domain-specific safety standards differ in the definition of the safety property and its scope, as well as the compliant processes or the approval criteria from the competent certification body. OPENCOSS aimed at integration and interoperability of SW tools, including requirements management and test automation tools. To that end, it relies on DBMS repositories to build certification arguments. This project introduces:

- a common certification meta-model, the Common Certification Language (CCL) that can be transformed to the certification requirements mandated by the domain-specific safety standard
- argumentation patterns to arrange the product and process compliance arguments
- a customizable process to generate certification artefacts (e.g. documents).

The ongoing AMASS project [1] builds on OPENCOSS developments. AMASS proposes a reuse-oriented approach for architecture-driven assurance, multi-concern assurance and seamless interoperability between assurance and engineering activities. This project focuses on the loose coupling between SW design environments, retaking the Open Services for Lifecycle Collaboration (OSLC) integration.

3 European Project DREAMS

DREAMS [20] is a European project that aims at developing a cross-domain *real-time* (RT) architecture and design tools for complex networked multi-core mixed-criticality systems. This project delivers meta-models, virtualization technologies, model-driven development methods, tools, adaptation strategies and validation, verification and assessment methods for the seamless integration of mixed-criticality to establish security, safety and real-time performance as well as data, energy and system integrity. It also defines a cross-domain system architecture of a hierarchically distributed platform for mixed-criticality applications combining the logical and physical views.

Logically, the architecture style of DREAMS consists of heterogeneous application subsystems with different criticality levels (SIL 1 to 4 according to IEC 61508), timing (firm, soft, hard and non-real-time) and computation models such as Time-Triggered (TT) messages, data-flow and shared memory. Application subsystems can have contradicting requirements for the underlying HW platform such as different trade-offs between predictability, safety and performance in processor cores (i.e., Zynq-7000, Hercules), hypervisors (i.e., XtratuM, PikeOS), operating systems (i.e., Windows CE, Linux) and networks (i.e., on-chip network, off-chip network). They can be further split into SW components (e.g., diagnosis partitions and safety protection partitions which are responsible for executing a safety state in the case of a failure).

Figure 2 shows the architecture style defined in DREAMS project where blocks highlighted in grey represent *core platform services*, blocks with dotted boundary are the *optional platform services*, and the blocks with diagonal lines are the *application related platform services*. Partitioning establishes this system perspective, enabling the decomposition of the system into multiple application subsystems which can be independently certified to the respective level of criticality.

The HW architectural style proposed by the European project DREAMS ensures determinism and temporal independence to simplify the timing and resource analysis. Temporal predictability and low jitter also promote the quality of control of mixed-criticality systems. This architecture style is used in the following sections to define the methodology for developing mixed-criticality product lines.

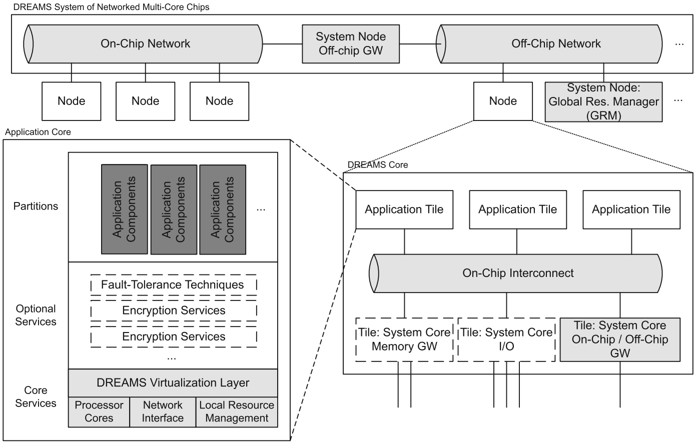

Fig. 2. The DREAMS architecture style (Source [20]).

In the scope of the DREAMS project [20] the safety certification of MCPLs according to the IEC 61508 standard is one of the objectives. The subject of this project are families of dependable mixed-criticality systems that embody variable sets of features (e.g., safety-related and non safety-related features). In DREAMS, we generate several argumentation models [18,19] which may be completed by evidences, analysis and tests results to provide a robust and verified system.

As this European project aims at providing cost-effective tools and procedures for the certification, we tackle the compilation of the whole set of safety information required by each variant of a mixed-criticality system product line.

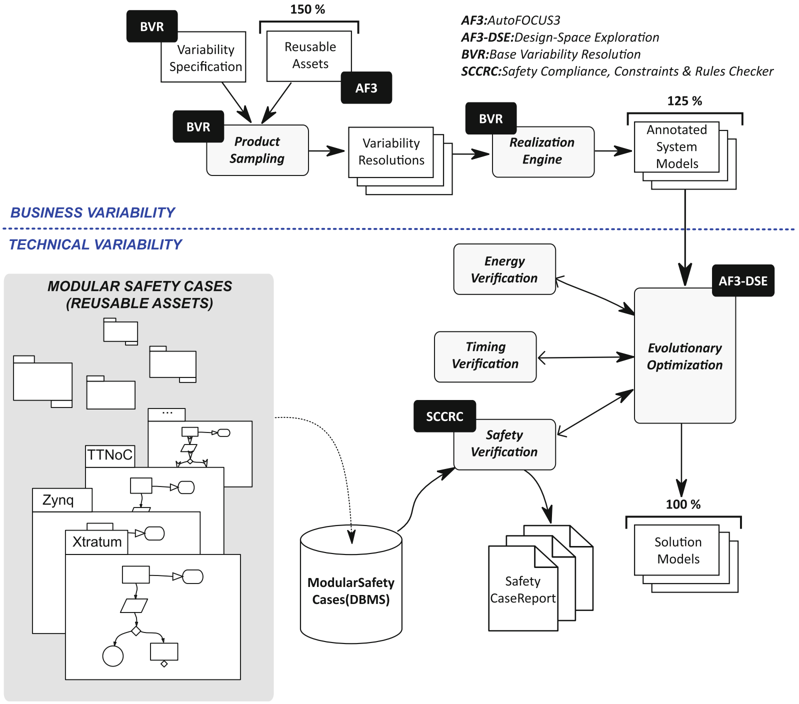

Fig. 3. Design Space Exploration (DSE).

Such compilation is costly and time-consuming, even when two variants show minor differences; their safety argumentation may share only some small fragments. To improve cost-effectiveness, we automate the construction of preliminary safety arguments, after a favourable safety evaluation of a candidate mixed-criticality system. To that end, several plug-ins for safety verification are developed based on the *Design Space Exploration* (DSE) extension of the open source model development tool *AutoFOCUS* 3 (AF3) [4,6]. Further plug-ins are also developed to give support for energy and timing verification, although they are out of the scope of this paper.

Safety related plug-ins shown in Fig. 3 may be used to capture the requirements of MCPL systems, define the variability models, sample and assess the properties of the variability models and build and refine the variant safety argumentation of those systems [9]. The plug-ins for safety developed in DREAMS projects are the following:

- *Safety Case Argumentation Generator:*
  This generator enables constructing safety case argumentations (i.e., *Modular Safety Cases* (MSCs)) by instantiating and composing a set of GSN diagram patterns. The GSN argumentation models that we generate represent the certificates as a *Solution* element, constrained by an *Obligation* (i.e. the

requirement for conformance demonstration). In the context of IEC 61508, a safety certificate usually has an accompanying mandatory report, emitted by the certification body.

The components that compose a reliable system shall be supported by reference workflows, which guide the developers to use the item safely. Therefore the project team must to justify how they manage the item, demonstrate the proper adoption of safety measures from design to integration and system validation, justify deviations from the recommended practice and execute verification activities.

In DREAMS project, these activities spread through different project phases of the development process. The proper handling of safety-compliant items is justified at the design phase.

- *Safety Case Checker:*

  Once we get a safety case argumentation, the DREAMS Safety Compliance Constraints and Rules Checker (SCCARC) [9] performs sanity checks following these rules:

  - *Rule 1:* Concrete argumentation shall not contain optional elements.
  - *Rule 2:* The goals shall be supported by strategies.
  - *Rule 3:* The strategies shall be supported by other goals or solutions.
  - *Rule 4:* At the final development stage, the related goals, strategies and solutions shall be developed and instantiated.

  As required in the IEC 61508 safety standard, the validation, verification and testing activities (V and V) shall be accomplished independently from the design. Therefore the V and V related information and evidences shall be provided by a separate team. The DSE tool generates a blueprint system based on custom evaluation results at its completion. After building the actual products, the properties predicted are verified, by carrying out further analysis and experiments.

- *Safety Case Documenter:*

  This post-processing feature generates a detailed description of the safety arguments and exposes the results from the safety case checker in a report. The safety argumentation is generated using the safety case argument generator. The safety case documenter traverses the argumentation model for the feasible product variants, writing a LaTeX transcript with a pre-defined safety-case template. The template contains 11 chapters, including an introduction, a system description, an overall safety argumentation, an analysis of every safety argument and its evidences, assumptions, issues, limitations detected and recommendations. In addition, this template is extended with an annexe that includes a user guide of the template.

  The automated generation of the preliminary safety case helps at keeping the overall documentation synchronised and eases the completion of the argumentation with new safety-relevant information collected at later development stages [15].

4 Safety Mixed-Criticality Product Line Development

Modularity methodology gives rise to the System of System (SoS) where independently useful systems are integrated into large systems with unique capabilities [23]. Concepts from SoS engineering can be helpful in Product Line Engineering (see Fig. 4). When dealing with SoS engineering, we can consider each system to be an instantiation of a product of a product line. The motivation to consider systems (in a SoS context) to be products of a product line can come from multiple causes. In many cases, a supplier of systems may have families of similar systems. Product line techniques promise considerable benefits in systematically handling such families of products. Therefore, product lines can be seen as a mechanism to develop components, sub-systems and systems in a SoS approach. From the perspective of the end user, it can be beneficial to handle groups of systems together rather than addressing them independently.

On the other hand, the DREAMS project implements modularity and provides a backup of safety argumentation models, consisting of a structured set of GSN MSCs. This set of GSN models encapsulate the MSCs in a composable mode and provide a guideline to carry out IEC 61508 compliant assessment. For instance, the MSC for an IEC 61508 compliant hypervisor and a Commercial Off-The-Shelf (COTS) multi-core device are defined in [18, 19].

On the basis of the safety-related arguments defined in the MSCs and the product line hierarchy introduced at the beginning of this section, we identify the following four levels of abstraction to represent a modular mixed-criticality product line development process.

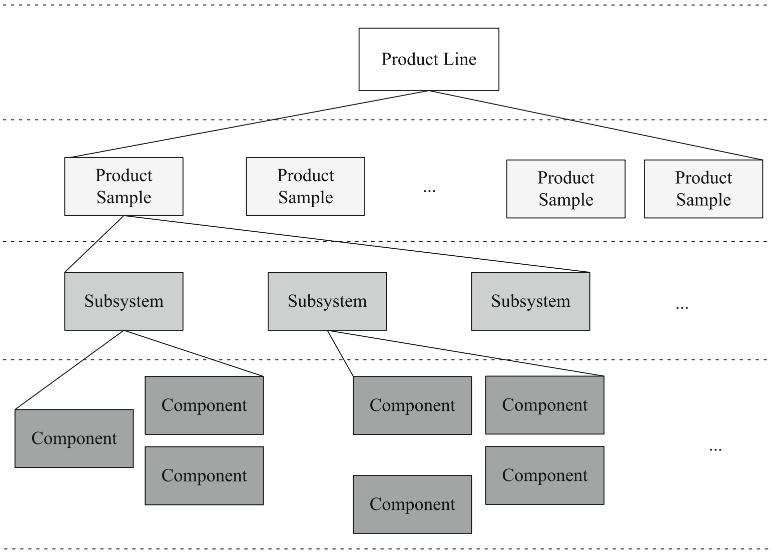

Fig. 4. Product line abstraction layers.

- the first layer represents the *safety-related arguments regarding a product line* that is based on the argument framework introduced in [13]
- the second layer defines the *safety arguments of a product sample* that may be composed of a set of components represented in the fourth layer of abstraction
- the third abstraction layer defines the *generic safety-related arguments that a safety component* shall fulfil to be considered a compliant item. This abstraction layer is the meet-in-the-middle point for developing a product line
- the last layer defines the *safety-related arguments for commercial and custom safety components* which could be used for developing a certain product sample, e.g.: a COTS multi-core device, a hypervisor, a mixed-criticality network, an operating system.

Variability is the quality, state or degree of a system to be changeable. For example, the product samples of a product line can vary depending on the safety standard (i.e., IEC 61508, ISO 26262, IEC 50126) and the level of criticality (i.e., SIL 1 to 4 according to the IEC 61508, Automotive Safety Integrity Level (ASIL) A to D according to ISO 26262).

- *Variation Points from Standards*
  DREAMS tackles the safety certification approach according to the IEC 61508 standard. IEC 61508 is the safety standard for E/E/PE functional safety systems and it is the basis for other domain-specific safety standards such as the ISO 26262 (automotive), EN 50126 (railway) or ISO 13849 (machinery). Most domain-specific safety standards require further characterization of the components, require a specific argumentation structure and provide mandatory safety-case guidelines. In general, a safety standard does not provide a fully objective evaluation guideline, and therefore, they require some subjective interpretation.
  On the other hand, there is a trend to harmonise the underlying requirements from multiple safety standards. However, no cross-domain development environment can cope completely with these differences [3].
  The work presented herein scopes the IEC 61508 safety standard, a similar approach may be used for other application domains ruled by different standards. The same may be applied to security or timing related standards.
- *Variation Points from Safety Requirements*
  Given a particular application domain (i.e., automotive, railway), safety standards set different requirements regarding the development process, the product design and the integration. In addition, the product manufacturer may target different safety levels (e.g., ASIL, SIL) for developing the product samples of a product line. In those cases, the safety requirements of those product samples may be mapped to several variation points that provide the right to choose between components with different criticality levels (e.g., SIL 1 to 4 according to the IEC 61508 safety standard). For instance, different measures and diagnostic techniques are recommended by the IEC 61508 safety standard depending on the required SIL.

The argument database may also host argument models for COTSs artefacts that may be used to implement parts of a product line safety argumentation.

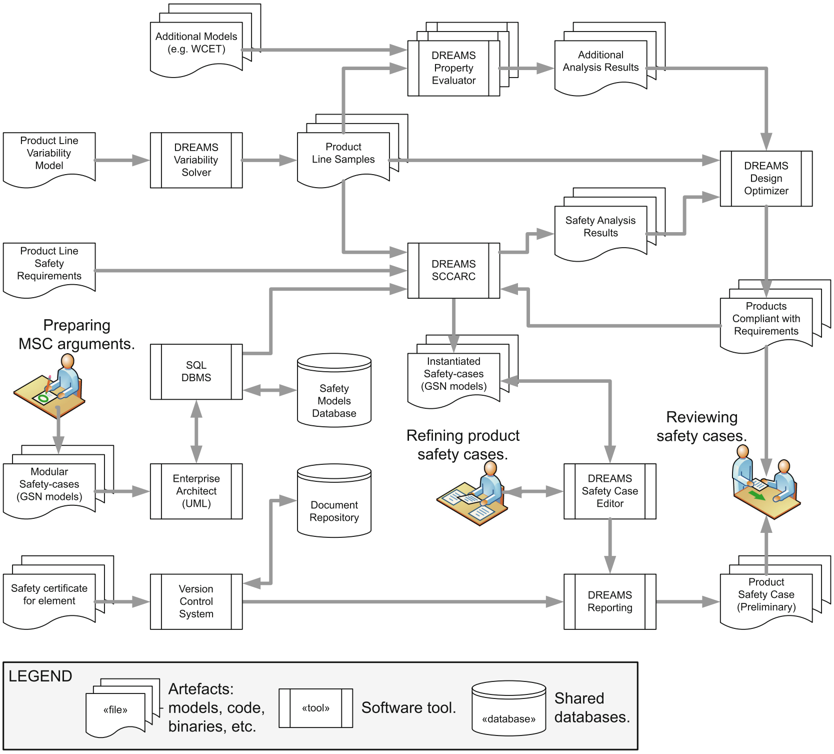
Additional Models (e.g. WCET)
DREAMS Property Evaluator
Additional Analysis Results
Product Line Variability Model
DREAMS Variability Solver
Product Line Samples
DREAMS Design Optimizer
Safety Analysis Results
Product Line Safety Requirements
DREAMS SCCARC
Products Compliant with Requirements
Preparing MSC arguments.
SQL DBMS
Safety Models Database
Instantiated Safety-cases (GSN models)
Reviewing safety cases.
Refining product safety cases.
Modular Safety-cases (GSN models)
Enterprise Architect (UML)
Document Repository
DREAMS Safety Case Editor
Safety certificate for element
Version Control System
DREAMS Reporting
Product Safety Case (Preliminary)
LEGEND
«file»
Artefacts: models, code, binaries, etc.
«tool»
Software tool.
«database»
Shared databases.

Fig. 5. DREAMS workflow to support the certification of mixed-criticality product lines.

E.g., a commercial model-based design and coding environment, a safety Programmable Logic Control (PLC). Adjoined to a certified component we usually find a certificate stating the safety score of the component, a certificate report from the certification body, detailing the context for which the certificate was granted, as well as a fault analysis, the identified risks and the prevention measures, a safety manual and a reference workflow stating how the development process accommodates specific measures required by the component. For instance, a model-to-code transformer may require the developers to subscribe or regularly check an alert service from the application manufacturer to warn about detected defects in the tool that could bring errors to the implementation.

The DREAMS modelling toolset intends to support the certification of safety-critical embedded product lines. To attain this goal, in DREAMS we define a PBD workflow that covers several possible low-level refinements [24]. PBD supports the meet-in-the-middle process [12], where successive refinements of specifications meet with abstractions of potential implementations and identification of precisely defined layers. I.e., the platforms [25]. This workflow, that is shown in Fig. 5, consists of the following steps:

- Build the argumentation meta-models for the common components.
- Set the design objectives into the design space explorer [22].
- Run the optimizer.
- When a product line configuration meets the safety requirements, a safety argumentation model is generated by the safety-validator.
- The report generator translates the argumentation model for a given design solution into a set of documents with proper references to already available information (e.g. pre-built argumentation).

The proposed workflow shares a data base (DB) store of safety cases generated using GSN notation, as well as safety certificates and related documents for commercial-of-the-shelf elements. DBs ease tool integration into a collaborative framework, collecting the pre- and post-design information contributed by actors with different roles in the safety project. AF3 extensions compose pre-built safety cases, which are generated in DREAMS according to the compliant product configurations, then document the preliminary safety cases with cross-references to either available or due documents.

5 Validation – Wind Turbine System

This section exemplifies the application of the methodology introduced in the previous section for developing industrial-grade safety product line systems. This case study consists of a wind turbine controller that bases on the DREAMS architecture style defined in Fig. 2, which is designed and deployed using the DREAMS modelling toolset shown in Fig. 5.

The HW architecture for the *Wind Turbine Controller* (WTC) is composed of the supervision, control and protection units. The WTC operates some distributed *input/output* (I/O) nodes networked over an EtherCAT field-bus (see Fig. 6). The wind turbine control system is composed of the *Galileo* and the *DREAMS harmonized platform* (DHP) platforms, which are interconnected

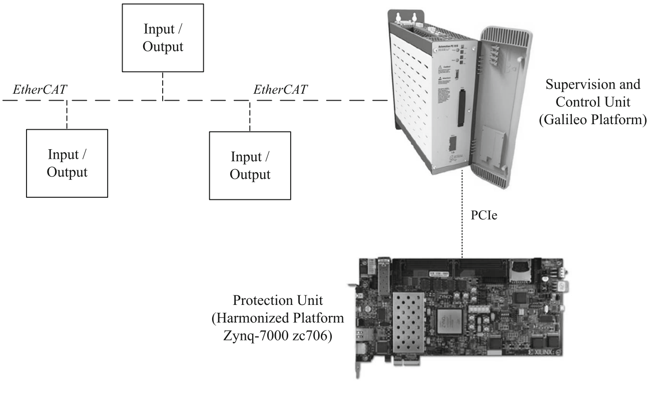
Input / Output
EtherCAT
EtherCAT
Input / Output
Input / Output
Supervision and Control Unit (Galileo Platform)
PCIe
Protection Unit (Harmonized Platform Zynq-7000 zc706)

Fig. 6. Wind turbine system HW architecture.

through a Peripheral Component Interconnect (PCI) Express (PCIe) bus. The RT platform named *Galileo* that supervises and controls the wind turbine system. This platform consists of an APC 910 industrial computer [5] with the customised operating system and SW. On the other hand, the DHP intends to implement the safety-related functions of the wind turbine system. The DHP integrates a Xilinx Zynq-7000 zc706 multi-core System on a Chip (SoC), integrating into a single silicon chip a dual-core ARM Cortex A9 and a Programmable Logic (PL).

This system architecture supports the execution of functionalities with different criticality levels (such as SIL 1 to 4 according to IEC 61508). To that end, XtratuM hypervisor [2] is used, splitting the CPUs of the Processing System (PS) and the soft-core(s) of the PL into partitions where the functionalities with different criticality are executed. The protection unit of the wind turbine system communicates with external sensors (e.g., wind speed sensor) and actuators (e.g., safety relay) through a safe field bus protocol composed of a non-safe field-bus EtherCAT and a Safety Communication Layer (SCL) integrated on top of a Network-on-Chip (NoC).

The combination of the NoC and the SCL enables temporal and spatial independences, which depend if a shared memory is used or not to communicate the partitions. The NoC implemented in this case study is the STmicroelectronics' NoC (STNoC), which is complemented with the NoC SCL cross-domain pattern. The SCL guarantees a safe communication between the partitions.

Based on this HW architecture, the DREAMS toolset (safety argumentation generator, safety case checker and safety case documenter) and the variable product line development methodology presented in this paper, we present a wind turbine development process in this section.

Figure 7 presents the top abstraction layers (highlighted in grey) which are used for representing a wind turbine product line development. Those layers contain the safety argumentation of the modules that compose a product line. In addition, this figure exposes two contracts which select the optimum combination of components that should be used for composing a wind turbine (see Fig. 7). These contracts are managed by the AF3 DSE toolset, which would choose the optimum components depending on the safety arguments specified (e.g., integrity level, application environment) and the evidences that are provided by the components that compose the safety argumentation database (e.g., certification accreditation).

As defined in this paper, this representation hierarchy can be extended for include variability, thus enabling developing product lines of different application domains (e.g., railway, automotive). Each domain specific safety standard defines additional requirements and measures and diagnostic techniques that shall be met to accomplish safety certification. In addition, this representation hierarchy can be extended to develop product samples with different levels of criticality. Figure 8 presents a partial representation of the safety argumentation for COTS multi-core devices that support the variations from safety standards and safety requirements. In addition, the modular development methodology

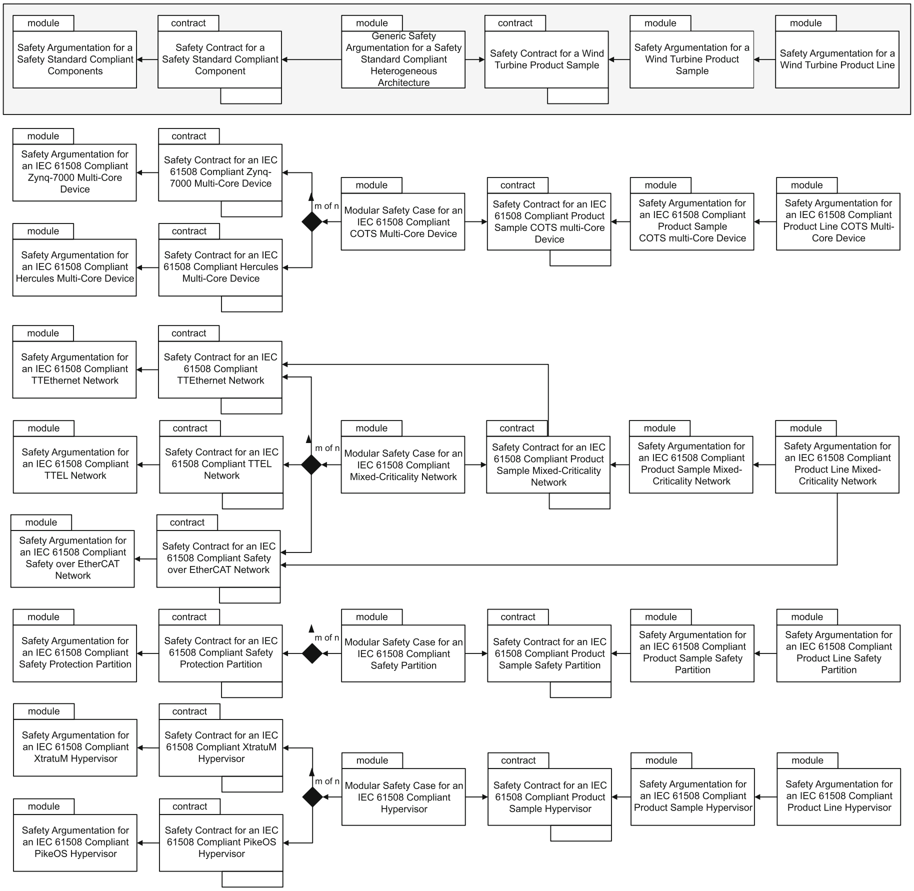
module
Safety Argumentation for a Safety Standard Compliant Components
contract
Safety Contract for a Safety Standard Compliant Component
module
Generic Safety Argumentation for a Safety Standard Compliant Heterogeneous Architecture
contract
Safety Contract for a Wind Turbine Product Sample
module
Safety Argumentation for a Wind Turbine Product Sample
module
Safety Argumentation for a Wind Turbine Product Line
module
Safety Argumentation for an IEC 61508 Compliant Zynq-7000 Multi-Core Device
contract
Safety Contract for an IEC 61508 Compliant Zynq-7000 Multi-Core Device
m of n
module
Modular Safety Case for an IEC 61508 Compliant COTS Multi-Core Device
contract
Safety Contract for an IEC 61508 Compliant Product Sample COTS multi-Core Device
module
Safety Argumentation for an IEC 61508 Compliant Product Sample COTS multi-Core Device
module
Safety Argumentation for an IEC 61508 Compliant Product Line COTS Multi-Core Device
module
Safety Argumentation for an IEC 61508 Compliant Hercules Multi-Core Device
contract
Safety Contract for an IEC 61508 Compliant Hercules Multi-Core Device
module
Safety Argumentation for an IEC 61508 Compliant TTEthernet Network
contract
Safety Contract for an IEC 61508 Compliant TTEthernet Network
module
Safety Argumentation for an IEC 61508 Compliant TTEL Network
contract
Safety Contract for an IEC 61508 Compliant TTEL Network
m of n
module
Modular Safety Case for an IEC 61508 Compliant Mixed-Criticality Network
contract
Safety Contract for an IEC 61508 Compliant Product Sample Mixed-Criticality Network
module
Safety Argumentation for an IEC 61508 Compliant Product Sample Mixed-Criticality Network
module
Safety Argumentation for an IEC 61508 Compliant Product Line Mixed-Criticality Network
module
Safety Argumentation for an IEC 61508 Compliant Safety over EtherCAT Network
contract
Safety Contract for an IEC 61508 Compliant Safety over EtherCAT Network
module
Safety Argumentation for an IEC 61508 Compliant Safety Protection Partition
contract
Safety Contract for an IEC 61508 Compliant Safety Protection Partition
m of n
module
Modular Safety Case for an IEC 61508 Compliant Safety Partition
contract
Safety Contract for an IEC 61508 Compliant Product Sample Safety Partition
module
Safety Argumentation for an IEC 61508 Compliant Product Sample Safety Partition
module
Safety Argumentation for an IEC 61508 Compliant Product Line Safety Partition
module
Safety Argumentation for an IEC 61508 Compliant XtratuM Hypervisor
contract
Safety Contract for an IEC 61508 Compliant XtratuM Hypervisor
m of n
module
Modular Safety Case for an IEC 61508 Compliant Hypervisor
contract
Safety Contract for an IEC 61508 Compliant Product Sample Hypervisor
module
Safety Argumentation for an IEC 61508 Compliant Product Sample Hypervisor
module
Safety Argumentation for an IEC 61508 Compliant Product Line Hypervisor
module
Safety Argumentation for an IEC 61508 Compliant PikeOS Hypervisor
contract
Safety Contract for an IEC 61508 Compliant PikeOS Hypervisor

Fig. 7. Wind turbine product line representation.

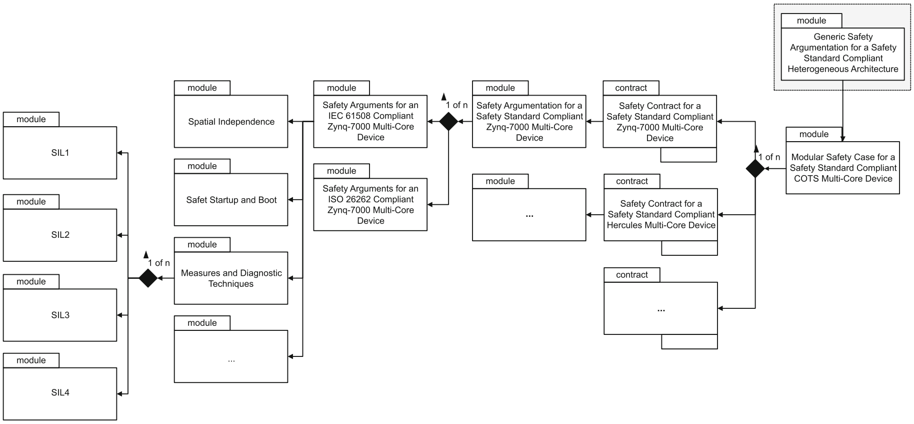
module
Generic Safety Argumentation for a Safety Standard Compliant Heterogeneous Architecture
module
SIL1
module
SIL2
module
SIL3
module
SIL4
1 of n
module
Spatial Independence
module
Safet Startup and Boot
module
Measures and Diagnostic Techniques
module
...
module
Safety Arguments for an IEC 61508 Compliant Zynq-7000 Multi-Core Device
module
Safety Arguments for an ISO 26262 Compliant Zynq-7000 Multi-Core Device
1 of n
module
Safety Argumentation for a Safety Standard Compliant Zynq-7000 Multi-Core Device
module
...
contract
Safety Contract for a Safety Standard Compliant Zynq-7000 Multi-Core Device
contract
Safety Contract for a Safety Standard Compliant Hercules Multi-Core Device
contract
...
1 of n
module
Modular Safety Case for a Safety Standard Compliant COTS Multi-Core Device

Fig. 8. Standard variable product line argumentation – Overall.

enables developing product lines with variable requirements as it enables reusing the safety argumentation blocks of the commercial and custom components. Figure 7 presents a safety requirement variable wind turbine product line where, as explained in the previous paragraph, can vary depending on the client needs.

6 Lessons Learned

The main challenge is handling an incomplete argumentation structure while splitting and cross-referencing the information according to a customised documentation structure. The documentation pattern shall be defined in the Functional Safety Management (FSM) procedure [10]. Mapping the argument fragments to the FSM documents and automatically cross-reference to existing documents -e.g. risks and faults analysis- or documents to be provided at a later development stage. E.g., a compilation of test results and their analysis.

To assemble the library of cross-referenced document artefacts, we would require a shared file system or a configuration management server, where digital representations of the evidences would be stored when available. For other tools, a relational database management system also provides a common storage point.

While these tools suffice for low-complexity products, to develop complex safety systems a better scalability would be required. This could be accomplished by switching to a different application interface (e.g. Open Services for Lifecycle Collaboration (OSLC)) supporting a loosely coupled application framework.

7 Conclusion and Outlook

DREAMS platform-based design supports re-using pre-certified components to deploy mixed-criticality systems. These HW and SW elements also enable a partially-automated design-space exploration, while easing the generation of the design rationale documentation as is required by the certification process. To this end, DREAMS provides a collection of safety arguments as a foundation to argue the satisfaction of the overall safety requirements. The safety-case approach supports modularity, yet for developing product lines where a per-product safety analysis and the justification of compliance are required to certification. Justification includes the linking analyses of the components, the freedom from interferences between the components and the prevention and tolerance of systematic errors in the development process.

A database of modular certification arguments provides a convenient information arrangement to support the modular composition of safety arguments. Our work shows how this can be even partially automated using the GSN to model the re-usable safety arguments. As an example, we developed the safety arguments for a generic IEC 61508 compliant wind-turbine product line which consist of a DREAMS wind turbine product sample composed of a set of commercial components. Furthermore, we identify several variation points that may extend the modular argumentation database. Those variation points include the variability of safety-related standards (i.e. DO 178C, ISO 26262), and the integrity level of the components (i.e., SIL 1 to 4 according to IEC 61508).

Future developments of the argumentation support would include additional attributes to represent the credibility of a given argument. Those attributes will enable capturing the subjective evaluation of the argumentation as done by a certification body. It is noteworthy that gathering this information is a challenging task. However, based on previous safety assessments and experiences with a certification body, a GSN model can represent a valuable asset to detect in advance the weakest link in the argumentation chain before actually facing the certification process.

Acknowledgement. This work was funded by the European Union's 7th Framework Programme under grant agreement No. 610640. Any opinions, findings and conclusions expressed in this article are those of the authors and do not necessarily reflect the views of funding agencies.

References

1. AMASS (Architecture-driven, Multi-concern and Seamless Assurance and Certification of Cyber-Physical Systems). http://www.amass-ecsel.eu/
2. XtratuM Hypervisor. http://www.fentiss.com/en/products/xtratum.html
3. OPENCOSS Open Platform for EvolutioNary Certification of Safety-critical Systems (2016). http://www.opencoss-project.eu/
4. Aravantinos, V., Voss, S., Teufl, S., Hölzl, F., Schätz, B.: AutoFOCUS 3: tooling concepts for seamless, model-based development of embedded systems. In: Proceedings of the 8th International Workshop Model-Based Architecting of Cyber-Physical and Embedded Systems (ACES-MB), pp. 19–26 (2015)
5. B&R Automation: Automation PC 910 (2015). http://www.br-automation.com/en/products/industrial-pcs/automation-pc-910/
6. Barner, S., Diewald, A., Eizaguirre, F., Vasilevskiy, A., Chauvel, F.: Building product-lines of mixed-criticality systems. In: Proceedings of the Forum on Specification and Design Languages (FDL 2016). IEEE, Bremen, September 2016
7. CENELEC: PD CLC/TR 50506–2: 2009 Railway applications. Communication, signalling and processing systems. Application guide for EN 50129. Part 2: Safety assurance, CENELEC (2009)
8. DREAMS: DREAMS - Distributed real-time architecture for mixed-criticality systems (2013). http://www.uni-siegen.de/dreams/home/
9. DREAMS: DREAMS 5.5.3 - Distributed real-time architecture for mixed-criticality systems - Methods for certifying mixed-criticality (2016)
10. DREAMS: DREAMS 5.6.1 - Distributed real-time architecture for mixed-criticality systems - Functional Safety Management (2017)
11. EUROCONTROL: Safety Case Development Manual version 2.2, 6 November 2006. http://www.eurocontrol.int/sites/default/files/article/content/documents/nm/link2000/safety-case-development-manual-v2.2-ri-13nov06.pdf
12. Fan Jiang, Y.Y., Kuo, J., Ma, S.P.: An embedded software modeling and process by using aspect-oriented approach. J. Softw. Eng. Appl. **4**(2), 16 (2011). doi:10.4236/jsea.2011.42012

13. Hutchesson, S., McDermid, J.: Trusted product lines. Inf. Softw. Technol. **55**(3), 525–540 (2013). doi:10.1016/j.infsof.2012.06.005
14. ISO/IEC: ISO/IEC 17000 Conformity assessment - Vocabulary and general principles, June 2004
15. Kelly, T.: Arguing safety - a systematic approach to managing safety cases. Ph.D. thesis (1998). https://www-users.cs.york.ac.uk/tpk/tpkthesis.pdf
16. Kelly, T.: Concepts and principles of compositional safety case construction, May 2001
17. Kelly, T.: Modular certification: acknowledgements to the industrial avionic working group (IAWG) (2007)
18. Larrucea, A., Perez, J., Agirre, I., Brocal, V., Obermaisser, R.: A modular safety case for an IEC 61508 compliant generic hypervisor, August 2015. doi:10.1109/DSD.2015.27
19. Larrucea, A., Perez, J., Obermaisser, R.: A modular safety case for an IEC 61508-compliant COTS multi-core device. In: Proceedings of the DASC 2015 Conference, October 2015. doi:10.1109/DSD.2016.66
20. Obermaisser, R., Weber, D.: Architectures for mixed-criticality systems based on networked multi-core chips. In: Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA). pp. 1–10, September 2014
21. de Oliveira, A.L., Braga, R.T.V., Masiero, P.C., Papadopoulos, Y., Habli, I.: A model-based approach to support the automatic safety analysis of multiple product line products. In: Proceedings of the SBESC 2014. IEEE (2014). doi:10.1109/SBESC.2014.20
22. Perez, J., Gonzalez, D., Trujillo, S., Trapman, T.: A safety concept for an IEC-61508 compliant fail-safe wind power mixed-criticality system based on multicore and partitioning. In: de la Puente, J.A., Vardanega, T. (eds.) Ada-Europe 2015. LNCS, vol. 9111, pp. 3–17. Springer, Cham (2015). doi:10.1007/978-3-319-19584-1_1
23. Prochnow, D., Hilton, L., Zabek, A., Willoughby, M., Harrison, C.: Systems of systems and product line best practices from the DoD modeling and simulation industry, Septemeber 2014. http://www.acq.osd.mil/se/webinars/2014_09_09-SoSECIE-Prochnow-brief.pdf
24. Sangiovanni-Vincentelli, A., Martin, G.: Platform-based design and software design methodology for embedded systems. IEEE Des. Test Comput. **18**(6), 10 (2001). doi:10.1109/54.970421
25. Sangiovanni-Vincentelli, A., Carloni, L., Bernardinis, F.D., Sgroi, M.: Benefits and challenges for platform-based design. In: Proceedings of the 41st Annual Conference on Design Automation - DAC 2004, p. 5. ACM (2004). doi:10.1145/996566.996684
26. Toulmin, S.E.: The Use of Argument, No. 241. Cambridge University Press, Cambridge (1958)