GSN Support of Mixed-Criticality Systems Certification

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

[1] IK4-Ikerlan, Mondragon, Spain
[2] Fortiss, München, Germany
[3] SINTEF ICT, Oslo, Norway
[4] Mondragon Goi Eskola, Mondragon, Spain
{cfnicolas,feizaguirre,alarrucea,jmperez}@ikerlan.es,
barner@fortiss.de, franck.chauvel@sintef.no,
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* (HCP) enable architectural simplifications and standardisation across multiple application fields to implement embedded systems with a homogeneous *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. In the same vein, 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 the attached complexity and high performance. These interferences challenge the certification of modern HCPs and are one of the main objectives of today's embedded system developers.

The final publication is available at Springer via: http://dx.doi.org/10.1007/978-3-319-66284-8_14

The certification process represents the major cost driver in the overall project budget for developing safety-critical systems. Certification is a third-party attestation related to products, processes, systems or persons [7]. An attestation is the issue of a statement, based on a decision the following review, which fulfilment of specified requirements or standard is demonstrated. In traditional certification, if a requirement of the systems changes, the whole system is re-assessed. Modularity enables dividing the system into independent modules which may be developed and certified with different criticality levels, thus improving the re-usability and scalability of the overall system and reducing the complexity and the certification cost.

The IEC 61508 safety standard considers safety as an *emergent system property*, resulting from the inherent safety of its components, the system structure and its interactions with its operational context and between its parts and the development process. Safety standards rooted in IEC 61508 follow a stereotyped development workflow (e.g., V-model development process) with interleaved analysis, refinement and review tasks. In the scope of the European project *Distributed REal-time Architecture for Mixed Criticality Systems* (DREAMS) [4] the safety certification of *Mixed-Criticality Product Line* (MCPL) according to the IEC 61508 standard is one of the objectives. The subject of DREAMS are families of dependable mixed-criticality systems that embody variable sets of features (e.g., safety-related features).

IEC 61508 recommend the use of models for analysis purposes, which assess the compliance with the established practice where the developer verifies the safety-related behaviour of the real system. This implies that the argumentation models obtained after a *Design Space Exploration* (DSE) are partial and shall be completed by evidences, complementary analyses and the results of the tests. For instance, *AutoFOCUS 3* (AF3) [1] is a file-oriented application that handles models as single blocks of information. This limits re-usability in AF3 to model libraries. IEC 61508 mandates the *development process redundancy* for high-integrity systems. This 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. Process redundancy decreases the likelihood of systematic errors by relying on diverse interpretations of the requirements. The safety rationale collates the generated information from both activity branches in a collection of arguments. In practice, a file-based application environment does not support the concurrent and independent development, which is required to certify high-integrity MCPLs in a cost-effective manner.

This paper presents a shared certification artefact based *Database Management System* (DBMS) to overcome the limitation introduced in the previous paragraph. Furthermore, the presented solution provides support for different use-cases required for collaborative safety-projects. Those collaborative projects can handle and share safety certificates, evidences and reference documents common to the DREAMS MCPL, concurrently collect the arguments and document and semi-automatically optimise the design and post-design of MCPLs. The paper is organized as follows. Section 2 introduces the *Goal Structuring Notation* (GSN) notation language, the Platform Based Design (PBD) methodology and the DREAMS architecture style. Section 3 presents the PBD workflow to design safety product lines. Section 4 exemplifies the integration of the method-

ology proposed in a safety wind turbine product line system. Section 5 presents the conclusion and outlook.

2 Background

2.1 GSN Argumentation Models

A safety case specifies the interpretation and implementation of safety requirements of a system, including the engineering decisions and the rationale that attests the safety assessment. This information can be presented in textual language, what suffices for simple safety cases. However, pure textual language descriptions often lead to unclear, unstructured and ambiguous safety cases for complex safety requirements. To overcome these problems, Toulmin [18] introduced a notation for representing the hierarchical structure of safety cases. Following the Toulmin approach, Kelly proposed a method to develop, document and maintain safety-cases using the GSN [8] language. This notation languages uses the *goal*, *strategy*, *solution*, *context*, *assumption* and *justification* elements to express the safety-related requirements of a system. In addition, GSN language supports modularity [13], [9], 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.

The GSN notation language is adopted in work presented in this paper for representing a product line development process. The main reason for that is its acceptance in the safety domain, available guidelines and tool support. On the other hand, the *Unified Modeling Language* (UML) modelling application Enterprise Architect (EA), from Sparx Systems [17], is used to model the safety arguments for the *Modular Safety Case*s (MSCs)[5]. EA supports user's extensions named Model Driven Generation (MDG) Technologies, to extend EA's modelling capabilities to specific domains and notations. Kuono [10] released a basic MDG GSN extension for EA, which we extended by adding the modular GSN extensions, as well as other GSN stereotypes found in safety cases. EA supports concurrent engineering work-flow by storing the model in an external DBMS, supporting concurrent access by multiple users. The modular GSN MSCs for DREAMS are stored in a SQL DBMS.

2.2 Platform-based Design for Mixed-Criticality Product Lines

The PBD is an abstraction that covers several possible low-level refinements [15]. PBD supports the meet-in-the-middle process [5], where successive refinements of specifications meet with abstractions of potential implementations and identification of precisely defined layers –i.e, the *platforms* [16]. The *meet-in-the-middle* approach aims

[5] The DREAMS tool-set is a derivative of the AF3 environment, for which there is a GSN extension to model arguments. However, AF3 relies on information file-storage, and the AF3' GSN extension underwent adaptations to integrate with other DREAMS analysis tools. EA DBMS is implemented to support collaborative teamwork, enabling a concurrent development of the DREAMS baseline of MSCs. These tools also enable developing application-specific compliance and V and V-arguments.

at preventing the convergence to non-feasible solutions sometimes found with a *top-down* approach, as well as preserving abstractions to tackle the complexity arising in a *bottom-up* approach.

A platform may be composed of a set of elements together within their constraints and rules. It can be thought as a library of elements which can interconnect through communication components. Each element is characterised regarding its functionality and expected behaviour. The meet-in-the-middle methodology applies a top-down design (application design) for a high level of abstraction and implements a bottom-up design for a low level of abstraction (platform design). Both designs converge where the platform is ready to host an application, which is ready to be hosted on a platform. From a HW perspective, the bottom-up approach is supported (low to high abstraction level). This enables the adaptation of the HW both at the design time and the run-time –using dynamic and partial reconfiguration.

In the context of IEC 61508-2 the following two requirements may be implemented to show the absence of systematic faults. The first requirement implies to meet the requirements of *Route* 2_S *or Route* 3_S. *Route* 2_S *"Proven-in-use approach"* establishes the compliance with the requirements of proven in use components. A component shall only be regarded as proven-in-use when it has a clearly restricted and specified functionality and when the absence of systematic faults is demonstrated (see Subsection 7.4.10 of IEC 61508-2). *Route* 3_S *"Pre-existing SW"* establishes the compliance with the requirements of IEC 61508-3, including the requirements for pre-existing and re-used SW. On the other hand, the second requirement implies to *provide a safety manual* that includes a precise and complete description of the pre-existent components, enabling the assessment of the integrity of a specific safety function that depends wholly or partly on the pre-existing SW components (see Annexe D of IEC 61508-2 and IEC 61508-3).

2.3 DREAMS Architecture Style

DREAMS is an European project that aims at developing a cross-domain *real-time* (RT) architecture and design tools for complex networked systems where application subsystems of different criticality executing on networked multi-core chips are supported. DREAMS 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. This research project also defines a cross-domain system architecture of a hierarchical distributed platform for mixed-criticality applications combining the logical and physical views (see Figure 3).

The architecture style of DREAMS consists of an heterogeneous application subsystems with different criticality levels (e.g., Safety Integrity Level (SIL)1 to 4 according to IEC 61508), timing (e.g., firm, soft, hard, non-RT) and computation models such as Time-Triggered (TT) messages, data-flow and shared memory. The application subsystems can have contradicting requirements for the underlying platform such as different trade-offs between predictability, assessment and performance in processor cores (i.e., Zynq-7000 processor), hypervisors (i.e., XtratuM hypervisor), operating systems (i.e., Windows CE) and networks (i.e., on-chip and off-chip networks). This platform

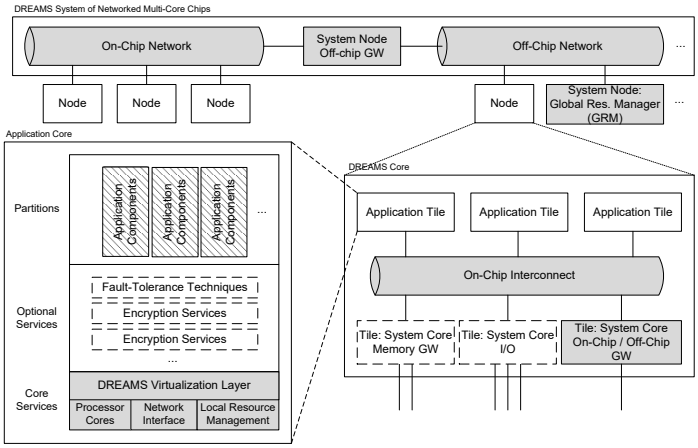

Fig. 1. The DREAMS architecture style. Blocks in grey background represent *core platform services*, blocks with doted boundary are the *optional platform services* and the blocks with diagonal lines are the *application related platform services*.

architecture provides global and local resource management units for executing multiple application subsystems, their component and different execution environments and resources such as memories and *input/output*s (I/Os).

This architecture style is used in the following sections to define the methodology for developing mixed-criticality product lines.

3 Safety Mixed-Criticality Product Line Development

Modularity gives rise to the System of System (SoS) where independently useful systems are integrated into large systems with unique capabilities [14]. Concepts from SoS engineering can be helpful in Product Line Engineering. 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. First, in many cases, a supplier of systems may have families of similar systems. Product line techniques promise considerable benefits in handling such families of products in a systematic fashion. Therefore, product lines can be seen as a mechanism to develop components, sub-systems and systems in a SoS approach. Second, from the perspective of an end user of systems, it can be beneficial to handle groups of systems together rather than addressing them independently.

Safety case notation languages may be used for representing the safety-related arguments of those granularity levels. For instance, the GSN notation language may be used to that end. A safety case is a documented body if evidences that provides a convincing and valid argument that a system is adequately safe for a given application in a specific environment (such as automotive, railway, lift). Regarding modularity, MSCs are safety cases that limit the impact of changes to specific modules of the system, enable reusability and reduce the complexity of the system (simplification strategy).

The DREAMS project implements modularity and provides the safety argumentation backup, consisting of a structured set of GSN MSCs. This set of GSN models encapsulate the MSCs in a composable format 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 [11, 12].

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 (see Figure 3).

1. The first layer represents the *safety-related arguments regarding a product line* that is based on the argument framework introduced in [6].
2. 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.
3. 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.
4. 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.

3.1 Variation Points from Safety Standards

DREAMS tackles the safety certification approach according to the IEC 61508 standard. IEC 61508 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 or provide mandatory safety-case guidelines. The safety standards do not provide a fully objective evaluation guidelines, and therefore, they require some subjective interpretation.

There is a trend to harmonise the underlying requirements from multiple safety standards, but currently, no cross-domain development environment can cope completely with these differences [3]. Although the work presented herein scopes IEC 61508, a similar approach may be used for other application domains ruled by different standards.

3.2 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., Automotive Safety Integrity Level (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., SIL1 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.

3.3 Linking Argumentation for COTSs Components

The argument database may also host argument models for COTSs artefacts that would be used to implement parts of the system –e.g. a commercial model-based design and coding environment, a safety PLC. Adjoined to a certified component we usually find:

1. a certificate stating the safety score of the component
2. a certificate report from the certification body, detailing the context for which the certificate was granted, as well as the fault analysis, the identified risks and the prevention measures
3. the safety manual for the item, and
4. a reference workflow stating how shall the development process accommodate to specific measures required by the item[6]

3.4 DREAMS Model-based Workflow

The DREAMS modelling toolset intends to support the certification of safety-critical embedded product lines. To attain this goal, as represented in Figure 2, the DREAMS workflow consists of the following steps:

1. Build the argumentation meta-models for the common components.
2. Set the design objectives into the design space explorer.
3. Run the optimizer.
4. When a product line configuration meets the safety requirements, a safety argumentation model is generated by the safety-validator.
5. 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 argumentations).

4 Validation – Wind Turbine System

This section exemplifies the application of the proposed methodology for developing an industrial-grade safety product line systems. Our case study consists of a wind turbine controller that bases on the DREAMS architecture style defined in Figure 1 and is designed and deployed using the DREAMS modelling toolset shown in Figure 2.

The HW architecture for the *wind turbine controller* (WTC) is composed of the supervision, control and protection units. The WTC operates some distributed I/O nodes networked over an EtherCAT field-bus. The wind turbine control system is composed

[6] 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.

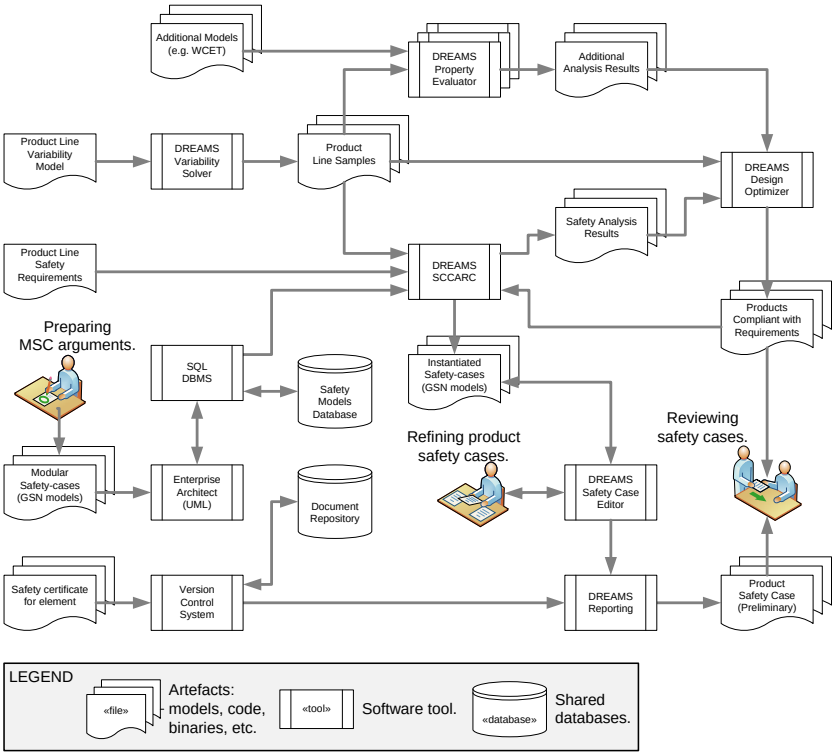

Fig. 2. DREAMS workflow to support the certification of mixed-criticality product lines. Shared DB store the GSN arguments for MSCs, as well as safety certificates and related documents for commercial-of-the-shelf elements. DB eases tool integration into a collaborative framework, collecting the pre- and post-design information contributed by actors with different roles in the safety project. AF3 extensions compose pre-built MSCs according to the compliant product configurations, then document the preliminary safety cases with cross-references to either available or due documents.

of the *Galileo* and the DREAMS harmonized computing platforms, which are interconnected through a Peripheral Component Interconnect (PCI) Express (PCIe) bus. The RT-platform *Galileo* supervises and controls the wind turbine system. This platform consists on an APC 910 industrial computer with customised operating system and SW. On the other hand, the *DREAMS harmonized platform* (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 SIL1 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.

4.1 Argumentation Database

The DREAMS approach relies on the reusable generic MSCs, cross-domain patterns and assets to ease the development of safety applications based on heterogeneous embedded platforms. MSCs constitute the basic blocks to build a product line. Some safety cases would provide a certificate as an evidence of compliance with safety standards, usually in the context of additional evidence for proper usage, integration with the rest of the system (e.g., the safety manuals) and additional verification requirements. In the GSN argumentation model we 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. When the component is integrated into a system, also a reference work-flow guides the developers to use the item in an acceptable safe way. Therefore the project team has 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 verification. In DREAMS these activities spread through different project phases. The proper handling of safety-compliant items shall be justified at the design phase.

The argumentation database presented in this paper provides a safety-argumentation database based on the abstraction layers introduced in Section 3 and the IEC 61508 safety standard. Figure 3 presents a partial representation of an IEC 61508 compliant product line system based on a safety COTS multi-core device. For instance, the product sample shown in the figure may be composed of a Zynq multi-core device or a Hercules device. On the other hand, as shown in Figure 4, this argumentation database can be extended to other safety-related standards (e.g., ISO 26262, DO-178B) and may support components with different criticality levels (e.g., SIL 1 to 4, ASIL A to D).

4.2 Argumentation Check

The GSN modular extensions support modelling the variability in an abstract argumentation graph [8]. For the certification process, we have to instantiate this argumentation, resolving all the options for the variants. Once we get a particular product sample, the DREAMS Safety Compliance Constraints and Rules Checker (SCCARC) component performs sanity checks on the argumentation following these rules:

- Concrete argumentation shall not contain optional elements.
- All the goals shall be supported by strategies.
- All the strategies shall be supported by other goals or solutions.

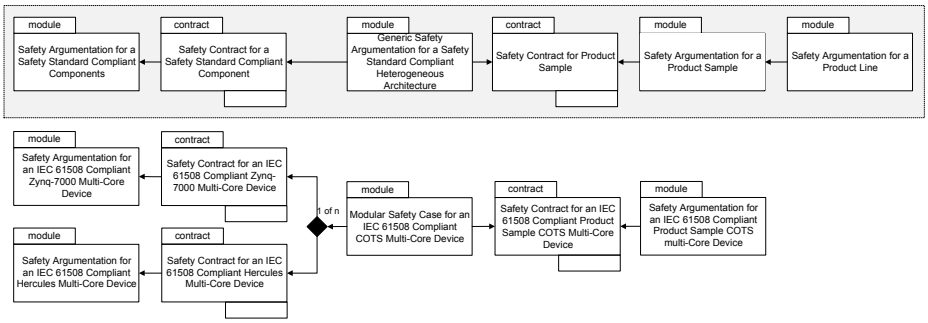

Fig. 3. Partial representation of a product line argumentation database.

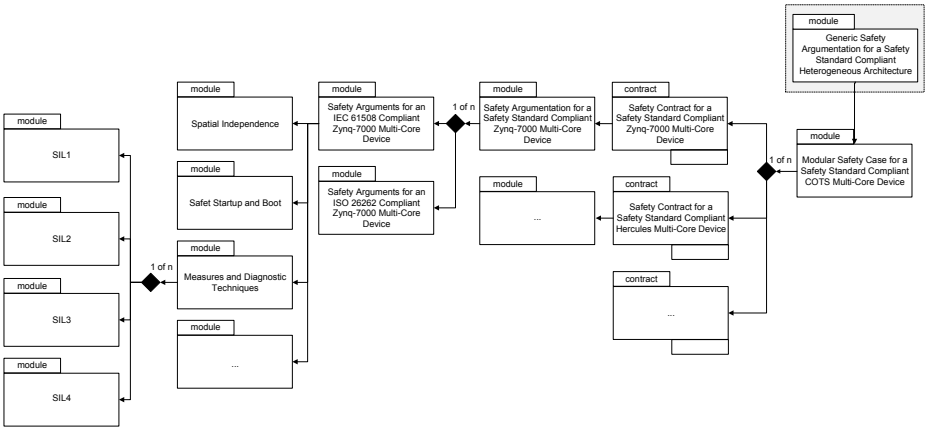

Fig. 4. An overview to the safety argumentation for COTS multi-core devices.

– At the final development stage, all the related goals, strategies and solutions shall be developed and instantiated.

In the DREAMS process, the DSE precedes the argumentation evaluation. As required by safety standards the validation, verification and testing activities (V and V) shall be accomplished independently from the design. Therefore V and V information shall be provided by a separate team. At the completion of the DSE stage, we get a blueprint of the system based on the results of several evaluations. As the conclusions based on models and data only hold if these can also be verified in the real world. At DSE these are considered like uninstantiated solutions, that should be credited with additional evidences. After building the actual products, we shall verify that the properties predicted hold, by carrying out further analysis and experiments.

4.3 Safety case documentation

SCCARC has a post-processing feature to generate a detailed description of the safety arguments in document form. To this end, SCCARC traverses the argumentation model

for the feasible product variants, writing a LaTeX transcript with a pre-defined safety-case template. The automated generation of the preliminary safety-case helps at keeping the overall documentation synchronized, and eases the completion of the argument with new safety-relevant information collected at later development stages, as stated in [8].

5 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., SIL1 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. AutoFocus 3, `http://af3.fortiss.org/`
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. DREAMS: Dreams - distributed real-time architecture for mixed-criticality systems (2013), http://www.uni-siegen.de/dreams/home/
5. Fan Jiang, Y.Y., Kuo, J., Ma, S.P.: An embedded software modeling and process by using aspect-oritented approach. Soft. Eng. and Applications, J. of 4(2), 16 (Apr 2011), DOI:10.4236/jsea.2011.42012
6. Hutchesson, S., McDermid, J.: Trusted Product Lines. Inf. Softw. Technol. 55(3), 525–540 (2013), DOI:10.1016/j.infsof.2012.06.005
7. ISO/IEC: ISO/IEC 17000 Conformity assessment – Vocabulary and general principles (June 2004)
8. Kelly, T.: Arguing safety - A systematic approach to managing safety cases. PhD thesis (1998), https://www-users.cs.york.ac.uk/tpk/tpkthesis.pdf
9. Kelly, T.: Modular certification: Acknowledgements to the industrial avionic working group (IAWG) (2007)
10. Kouno, Takeshi: MDG Technology for GSN (Goal Structure Notation) (2017), http://community.sparxsystems.com/community-resources/
11. Larrucea, A., Perez, J., Agirre, I., Brocal, V., Obermaisser, R.: A modular safety case for an IEC 61508 compliant generic hypervisor (Aug 2015), DOI:10.1109/DSD.2015.27
12. Larrucea, A., Perez, J., Obermaisser, R.: A Modular Safety Case for an IEC 61508-compliant COTS multi-core device. In: DASC 2015 Conf. Proc. (Oct 2015), DOI:10.1109/DSD.2016.66
13. 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: Proc. of SBESC'14. IEEE (2014), DOI:10.1109/SBESC.2014.20
14. 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
15. Sangiovanni-Vincentelli, A., Martin, G.: Platform-based design and software design methodology for embedded systems. IEEE Design and Test of Computers 18(6), 10 (2001), DOI:10.1109/54.970421
16. 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'04. p. 5. ACM (2004), DOI:10.1145/996566.996684
17. Sparx Systems: Enterprise Architect UML modeling tool, http://www.sparxsystems.com/products/index.html#corp
18. Toulmin, S.E.: The Use of Argument. No. 241, Cambridge University Press (1958)