Support for Modular Certification of Safety-Critical Embedded Systems in DECOS – the Generic Safety Case*

Erwin Schoitsch[1], Egbert Althammer[1], Gerald Sonneck[2], Henrik Eriksson[3], Jonny Vinter[3]

[1] (Austrian Research Centers GmbH – ARC)
{erwin.schoitsch, egbert.althammer}@arcs.ac.at
[2] (Tribun IT-Consulting, Austria)
gerald.sonneck@tribun.at
[3] (SP Technical Research Institute of Sweden)
{henrik.eriksson, jonny.vinter}@sp.se

Abstract. The integrated EU-project DECOS (Dependable Embedded Components and Systems) aims at developing an integrated architecture for embedded systems to reduce life-cycle costs and to increase dependability of embedded applications. To facilitate the certification process of DECOS-based applications a modular approach has been implemented which is based on the usage of generic safety cases. This means concretely that an application safety case merely contains the application-specific issues and re-uses the safety arguments of the generic safety cases of the DECOS platform. The safety cases are based on validation-plans (v-plans) comprising the steps to validate the safety requirements and contain the evidence which prove that these requirements are fulfilled. The Generic Test Bench which has been developed in the realm of DECOS supports the overall validation and verification process and particularly provides guideline to generate the generic safety cases.

1 Introduction

"Smart Systems" are based on intelligent embedded control systems, which are distributed within the application systems, and often hidden to the every-day life user. E.g., more and more functions in today's cars are realized by electronics and software, 80-90% of the new innovative features are realized by distributed embedded systems. Eventually, even highly safety critical mechanical and hydraulic control systems will be replaced by electronic components. Value of electronics in cars will increase beyond 40% of the total value. Even today, upper class cars contain up to 80 ECUs, several bus systems, and about 55% of all failures are caused by electronics, software, cables and connectors [3], [6].

The DECOS project [2] aims at making a significant contribution to the safety of dependable embedded systems by facilitating the systematic design and deployment of integrated systems [1]. DECOS (Dependable Embedded Components and Systems)

* Research supported in part by EU IST-FP6-511764 (DECOS).

is a European Integrated Project in FP6, Embedded Systems area, scheduled for the period 2004-2007. Coordinator is Austrian Research Centers GmbH (ARC). Work is performed by 19 partners: Two research centers (ARC, SP (Sweden)), six universities (Universities of Technology Vienna, Darmstadt, Budapest, Hamburg-Harburg, Universities of Kassel and Kiel), three technology and tool providers (TTTech Vienna, Esterel Toulouse, Infineon Munich), and eight demonstrator/application related industrial partners (Audi AEV, CR Fiat, Hella, Airbus, Thales, EADS, Liebherr Aerospace, Profactor).

In federated systems, each application subsystem is located on a dedicated processor. The federated approach provides natural separation of application functions, but causes increased weight, electric energy consumption and cost due to resource duplication and the large number of wires, buses and connectors. Integrated systems not only help to alleviate this problem, they also permit communication among application functions. A remarkable feature of the integrated DECOS architecture is that hardware nodes are capable of executing several tasks of application subsystems of different criticality. Throughout this paper, we will use the notion of a node instead of processor or component.

An integrated architecture provides a fixed number of nodes, each of which has certain properties (e.g., size of memory, computational power, I/O resources). All tasks have to be allocated such that given functional and dependability constraints are satisfied. This is discussed in detail in [4].

This paper focuses on the description of the certification process in DECOS which is implemented in a modular way and uses the concept of so-called generic safety cases and is supported by the Generic Test Bench which has been designed and implemented within subproject 4 (validation and certification). An overview of the other subprojects is found in [5].

2 Definition and Structure of a (Generic) Safety Case

One of the definitions of a safety case is as follows:
“A safety case is a document to convince a licensing authority that a given product or system is safe enough to be taken into use.”

It can also be used in the developing organization and by the organization using the safety-related system. The purpose of a safety case is usually described by the following quotation:
“A safety case should communicate a clear, comprehensive and defensible argument that a system is acceptably safe to operate in a particular context.” [7]

The keywords here are argument and context. In a safety case there are claims and there is evidence. Claims are specific statements about the safety properties of the system. Evidence results from V&V activities or specific safety mechanisms such as e.g. a back-up mechanism. The *arguments* are the links between claims and evidence. This means that the arguments are really important; it is useless to have evidence which is not related to a claim via arguments.

The *context* is also important since the safety case and its evidence is usually only valid under certain restrictions and conditions.

The concept of safety cases is well-established in e.g. the railway domain and according to the safety standard for railway applications EN 50129, safety cases are divided into three classes:

1. Generic product safety case (independent of application)
2. Generic application safety case (for a class of applications)
3. Specific application safety case (for a specific application)

The former two are, as the name implies, generic and can consequently be re-used for a class of applications or for independent applications, respectively.

Regardless of category, the content of a safety case is basically the same, see Fig. 1 below.

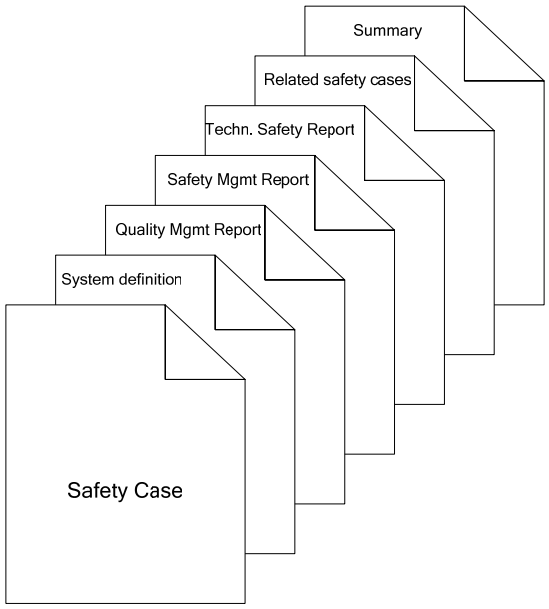

Fig. 1. Contents of a safety case

- The **system definition** shall precisely define or reference the system/subsystem/equipment to which the safety case refers, including version numbers and modification status of all requirements, design, and application documentation.
- The **quality management report** shall contain the evidence of quality management. Here aspects from e.g. ISO 9001 shall be covered, e.g. organizational structure, quality planning, as well as handling and storage.
- In the **safety management report,** the evidence of safety management shall be presented. Here documentary evidence to demonstrate compliance with all

elements of the safety management process throughout the lifecycle shall be provided.

- The **technical safety report** shall contain the evidence of functional and technical safety. The technical safety report shall explain the technical principles which assure the safety of the design, including (or giving references to) all supporting evidence (for example, design principles and calculations, test specifications and results, and safety analyses).
- In the **related safety cases** part, there shall be references to the safety cases of any subsystem or equipment on which the main safety case depend.
- The **conclusion** shall summarize the evidence presented in the previous parts of the safety case, and argue that the relevant system/subsystem/equipment is adequately safe, subject to compliance with the specified application conditions.

As long as the safety-related application conditions of a generic safety case are fulfilled by the more specific safety case or are carried forward to the more specific safety case, it is not necessary to repeat the generic approval process for each application.

3 Goal of the Generic Safety Case – Simplification of the Certification Process by Providing a Modular Approach

The goal of the generic safety case is the simplification of the certification process by providing a modular approach. One of the reasons that this makes sense is the fact that certification is a significant cost factor in the development of safety-critical systems. For instance, in the avionic domain between 60% and 70% of the cost of an avionics component is verification and certification cost [8]. Depending on the level of criticality, certification according to DO-178B increases the cost of developing software by 300–500% [9]. Consequently, there is a need for architectures that are designed for validation [10]. Design for validation occurs by devising a complete and accurate reliability model, by avoiding design faults, and by minimizing parameters that have to be measured. In the automotive area, with the upcoming standard ISO 26262, similar problems will arise, although criticality is restricted to SIL3 by definition. Modular certification will be a requirement and is most important for automotive industry because of the multi-tier supplier structure.

Another key element for controlling certification cost is architectural support for modular certification. Modular certification [11] is a certification strategy that promises a massive reduction in certification cost through modularization and reuse of certification arguments.

Modular certification is very important in e.g. the automotive domain where the sheer number of options and variants makes the design extremely complex and consequently also certification. For example, one automotive platform is used for several variants (sedan, hatchback, cab, etc) with different engine choices, e.g. petrol or diesel. Then the customer can select from a multitude of options ranging from electronically adjustable driver seats to navigation systems. All these options and

variants shall be possible without compromising the safety and thus keeping the certification valid.

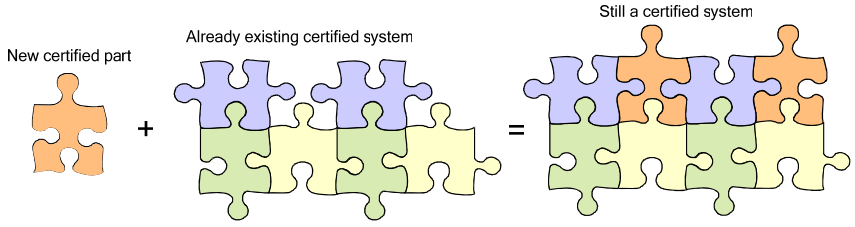

Fig. 2. Incremental certification

Modular certification requires error containment and encapsulation services from the integrated architecture. These services also permit incremental certification [12], i.e. the maintaining of a safety argument for the remaining part of the system when a DAS (distributed application subsystem) is modified or added. Incremental certification requires the ability to integrate new DASs without the need to re-certify the whole system, see Fig. 2. In particular, modifications to non safety-critical DASs do not involve a recertification of safety-critical DASs.

4 Generic Safety Cases in DECOS – The DECOS Certification Process

The DECOS architecture [1] offers modular certification by separating the certification of core services and architectural services from applications (enabling generic application safety cases (for class of applications) and for individual (specific) safety cases by supporting independent safety arguments for different DASs).

1. **Separating certification of architectural services from certification of applications**. The clear interfaces between the platform and applications, provided via the platform interface are a prerequisite for the separation of the certification of architectural services from the certification of applications. The certified architectural services of the integrated architecture establish a baseline safety argument for the certification of the overall system [12].
2. **Separating certification of different DASs**. The integrated architecture allows the independent certification of different DASs, instead of considering the system as an indivisible whole in the certification process. The safety argument for each DAS is provided to the integrator by the suppliers along with the compiled application code of the jobs in the corresponding DAS. In order to construct the safety argument for the overall system, the system integrator combines the safety arguments of the independently developed DASs and acquires additional evidence, such as results of a formal verification of the architectural services. The decomposition of the overall system into encapsulated DASs with different criticality levels reduces the overall certification efforts and allows focusing on the most critical parts. Furthermore, the separate certification of DASs is beneficial, if functionality is reused in different systems. In this case, the safety argument for the functionality needs to be constructed only once.

Therefore, as a modular safety case the safety case for a complete DECOS system will consist of the following parts:

- **Generic Safety Case for the DECOS core services** – to demonstrate the dependability of the DECOS core services (these are assumed to be proven by many years of research and even formally for different platforms such as TTP/C).
- **Generic Safety Case for DECOS nodes** – to demonstrate the dependability of the DECOS nodes (including especially DECOS high level services). Based on the requirements defined for the DECOS High level Services, a Generic Safety case was performed. For details see the next chapter.
- **Safety Case for DECOS applications** – to demonstrate the dependability of an application based on the DECOS architecture. Because of the separation of concerns by validating independent DASs (Distributed Application Systems) deployed on DECOS nodes following the architectural requirements, single applications (specific application safety case) as well as classes of applications (generic application safety case) can be validated and certified in an incremental manner.

In DECOS, modular certification is based on integrated architecture arguments (fault containment, encapsulation, separation in time and space), which are basis for the generic safety case.
The generic safety case proves by valid argument, that the DECOS architecture is fit for hosting SIL4 applications, i.e. the requirements are sufficient.

The three DECOS processes (development and design, verification and validation, certification) are represented in the Certification Process Model (see Fig. 3).

The certification process runs in parallel with the other two processes and shall be started as early as possible to avoid making design decisions that have to be remade later on when the certification authority has complaints. Note that the architectural design and software development processes have been merged into one to improve the readability.

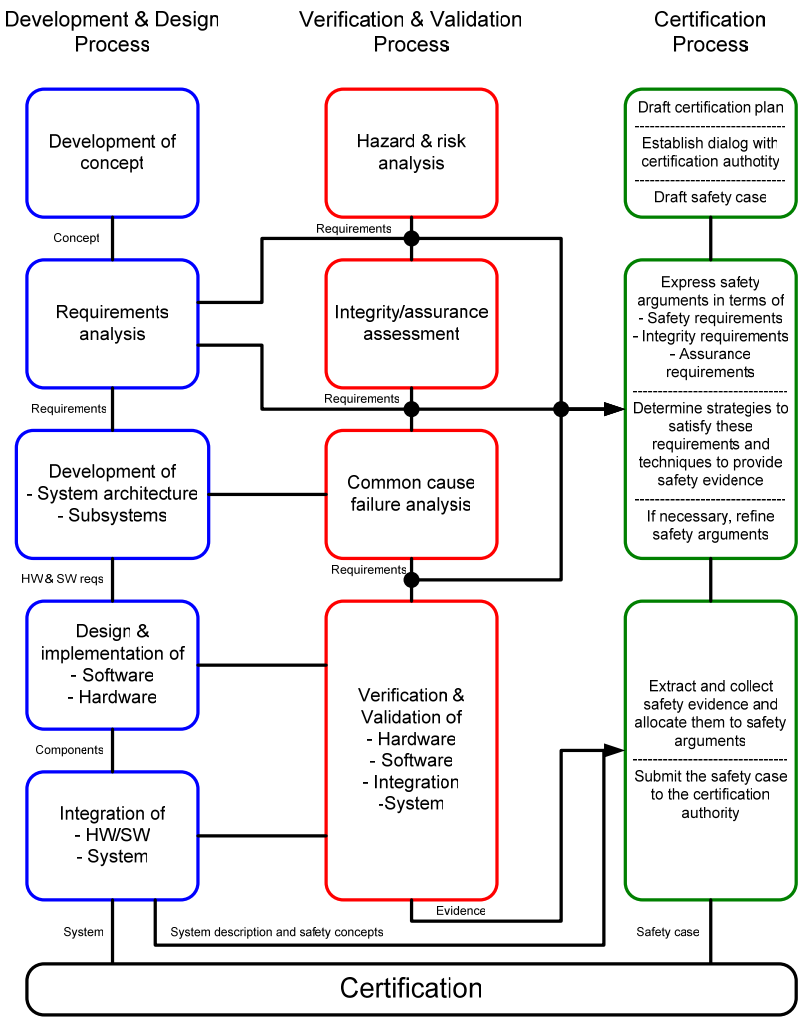

Fig. 3. The certification process and its relations to the other processes

5 The Generic Safety Case for DECOS Nodes

This chapter contains a brief summary of the following parts of the Generic Safety Case for DECOS nodes (according to Fig. 1): system definition (purpose and functionality of the DECOS node and system boundary), technical safety report (list

of the top-level functional and safety requirements, an example for a validation report for one of these requirements) and the conclusion of the Generic Safety Case.

System Definition

Purpose and Functionality

DECOS is an integrated architecture for distributed dependable embedded control systems in a wide variety of industrial sectors such as aerospace, automotive, railway, or industrial control. It provides a communication system which can be used simultaneously by several subsystems in a way which ensures that subsystems do not disturb with each other. The goal is to save communication hardware in highly distributed and complex applications.

Therefore, the main emphasis of the DECOS architecture lies on reliable communication between jobs of Distributed Application Subsystems (DASs). The distributed allocation of jobs (and DASs) can have various reasons. An obvious one is dependability. For the same reason DECOS provides strong encapsulation between jobs (i.e., spatial and temporal error containment, all component resources are allocated statically and interaction between jobs is restricted to ports), fault tolerance and diagnostic services. Figure 2-4 provides a schematic overview of a typical DECOS application.

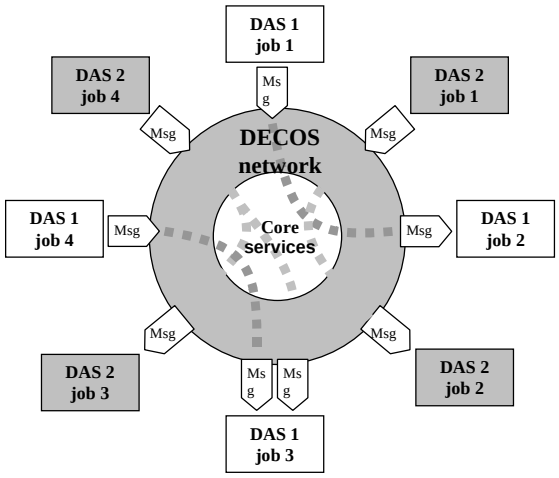

Fig. 4. Jobs of DASs communicating via a DECOS network

Fig. 4 illustrates a possible communication scenario among jobs of different DASs. In general DASs are self contained, but communication between different DASs is supported via so called gateways.

In order to be usable in the intended wide variety of industrial domains and specific applications, it is of crucial importance that DECOS does not deal with application data content and semantics. Its scope is limited to data transmission and job execution and related performance and dependability aspects.

On a functional level, DECOS assumes the existence of a set of so-called *core services*, on top of which it provides a set of so-called *high-level services*. The *Core Services* provided by DECOS (and contained in the inner circle in Fig. 4) are

- Predictable, Deterministic and Timely Transport of Messages
- Fault-Tolerant Clock Synchronization
- Strong Fault Isolation
- Consistent Diagnosis of Failing Nodes

Based on these Core Services, DECOS provides the following *High-Level Services*:

- Encapsulation Service
- Virtual Network Service
- Hidden Gateway Service
- Fault Tolerance Service
- Diagnostic Service

An important aspect of DECOS is that the DECOS communication network is exclusively controlled by itself. Hence, no external units like sensors can directly feed their data into the network. Instead, all sensors (and actuators) are to be controlled by DAS jobs, which therefore build the bridge between DECOS and the "outside world".

System Boundary

A DECOS application consists of hardware *nodes* that run DAS (software) *jobs* which communicate over the DECOS network.

Each DECOS node consists of an application computer and a dedicated communication interface (where both elements may be realized on the same piece of hardware). A DAS subsumes an arbitrary number of communicating and semantically correlated jobs, which may be distributed over several application computers. A major aim of DECOS is to allow the execution of several (dependent or independent) jobs on each application computer (in order to keep the hardware costs low). This implies the need for a strong encapsulation (with respect to fault isolation) of jobs.

Technical Safety Report

Overview of the Design

The standard EN 50129 states: "This section shall provide an overview description of the design, including a summary of the technical safety principles". Since the overview of the design has been provided already in the chapter "System definition" of this report, this chapter focuses on the technical safety principles. It includes technical safety principles of the Core Services and possible Applications also, because the DECOS nodes provide the link between these.

Technical safety principles of the Core
These are the core services which have been already listed in the section “Systems Definition”.

Technical safety principles of the DECOS Node
- Encapsulation service (to guarantee that activities do not disturb each other) is realized by using hardware protection mechanisms (such as memory region protection) provided by the Memory-Management Unit of the Application Computer, and by the scheduler which makes sure that a job does not use more computation resources than it was assigned to.
- Temporal partitioning (to prevent temporal interference of different subsystems): strict separation of the execution of the basic connector unit’s copy operation from the behavior at ports.
- Spatial partitioning (to prevent spatial interference of different subsystems): the basic connector unit transfers messages autonomously from the memory elements to the outer ports.
- Push and pull: The control flow at an inner port is always directed towards the connector unit, i.e. message transmissions are performed as an information *push* and message receptions proceed according to the information *pull* principle. Thus, the application is never interrupted by incoming messages and can disseminate outgoing messages without relying on a control signal from the connector unit.
- Temporal firewall: Each state message port of the Safety-Critical Connector Unit provides a *temporal firewall* interface. This firewall prevents that a job (which might access its port at any time) sends data outside of its schedule, and allows for a consistent diagnosis of failing nodes.
- Fault-tolerant layer: The FT layer performs bit-wise comparison of redundant messages transmitted in two channels and in case of a match it provides one message to the host. In case of a mismatch the host is informed about an invalid reception of a message. The FT layer supports the packing and unpacking of messages that cannot be transmitted in a single frame. This is provided by the *Message Group Service*. One message group consists of several messages received in several frames. Upon a reception of one invalid message (frame) from a message group the status of the message group shall be set to invalid. The *Message Group Service* assigns three message statuses for the message groups: *Valid*, *Null*, *Invalid*. The *Null* status is set in case there was no valid frame received that carries the content of the message The validation mechanisms shall inject faults in frames containing a messages of a message group, in order to validate the correct operation of the *Message Group Service* in presence of faults.

Possible technical safety principles of Applications
A number of important safety techniques, such as redundancy, will be provided by the applications.

Fulfillment of Functional and Safety Requirements
For each functional and safety requirements the fulfillment is shown. This is done using tables which use the following structure:

- Identifier: unique identifier of the requirement
- Requirement text: contains a short description of the requirement
- Reference: reference to a document which contains the evidence, e.g. the validation report (see section 6).

An example for such a table is shown in Table 1.

Table 1. List of requirements

Identifier	Requirement text	Reference
SC_GT_01	A service providing global time information shall be available	ValReport1
SC_VTS_01	Validity time span must be considered	ValReport2
SC_TRA_01	The Transport Service shall support periodic data transfer	ValReport3
SC_COM_01	Message transmissions have to occur on a sparse time base in order to facilitate the definition of global consistent system state	ValReport4
NSC_COM_01	The Transport Service shall support unicast communication	ValReport5
NSC_COM_02	The Transport Service shall support multicast communication	ValReport6
NSC_COM_03	The Transport Service shall support broadcast communication	ValReport7
DOM_CAN_01	High bus loads, caused by malfunctioning of another node, shall not block the CAN controller or application processor	ValReport8
DOM_CAN_02	Network-wide data consistency shall be ensured within the fault hypothesis of the DECOS architecture (this is known as atomicity requirement).	ValReport9
TT_ P_01	Message transmissions shall occur on a sparse time base in order to facilitate the definition of global consistent system state	ValReport10
…	…	

Summary

Note: According to the discussion in the chapter on purpose and scope of this Safety Case the following is valid only under the assumption, that all quoted analyses and reports have been done and all quoted requirements are satisfied.

The DECOS nodes provide the communication between the application-dependent jobs and the Core Services (see Fig. 4) and are thereby instrumental for the communication between the jobs. Additionally the DECOS nodes provide mechanisms for fault tolerance.

The arguments presented in the previous parts of the Safety Case shows that a DECOS node is adequately safe to be part of a safety-relevant system (subject to compliance with the specified application conditions).

This is guaranteed by the following principles and services:

- Fault-tolerant clock synchronization;
- Predictable, deterministic and timely transport of messages;
- Strong fault isolation (fault encapsulation);
- Fault tolerance service.

The software of the DECOS node is SIL 4.

6 Support for the Generation of Generic Safety Cases by the Generic Test Bench of DECOS

The Generic Test Bench [13] which supports validation and verification in DECOS supports the generation of safety cases by providing generic v-plans for safety standards, a documentation support and built-in user guidance in terms of a help file.

Generic V-Plans Based on Safety Standards (e.g. IEC 61508)

In DECOS generic v-plans based on safety standards have been developed which shall support the development of v-plans for safety-critical systems. The generic v-plans have been designed in such a way that they contain the extracted V&V activities from a safety standard (e.g. IEC 61508) for each possible safety integrity level (SIL). Starting from these generic v-plans the Generic Test Bench supports the generation of a concrete v-plan for a specific SIL (e.g. SIL 4) which only contain those V&V activities which are relevant for this SIL. Together with the functional and safety requirements and the necessary evidence they are the basis for the generic safety cases.

Fig. 5 shows a generic v-plan for IEC 61508-3 and the generated v-plan for SIL 4.

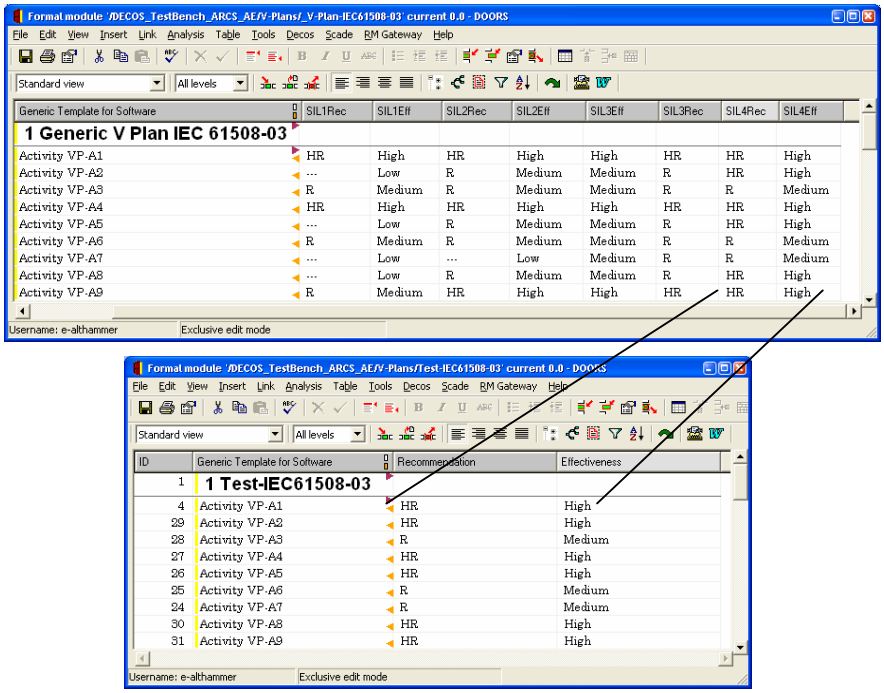

Fig. 5. Generic v-plan for IEC 61508-3 and generated v-plan for SIL 4

Documentation Support

When a v-plan has been completed (i.e. there is a valid evidence for each V&V activity) a validation report can be generated which corresponds to a table which contains the following information for each V&V activity: identifier, type of V&V activity and the name of the V&V activity report. The validation reports are referenced by the safety case (see e.g. Table 1).

Built-In User Guidance ("Help")

A specific feature of the integrated test bench is user support by user guidance menus and underlying documents which provide basic information for validation and certification support. The following screen shot in Fig. 6 shows the welcome page of the Test Bench built-in user guidance. Clicking e.g. on the hyperlink "Safety Case" the help for the development of a safety case will be shown.

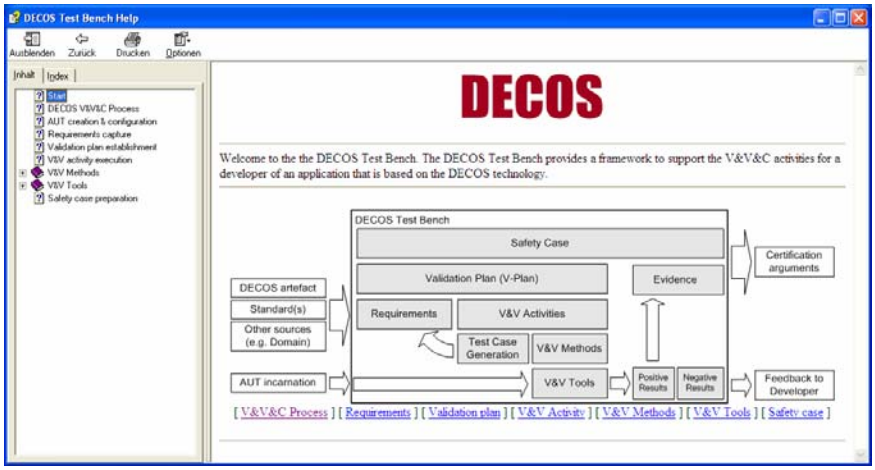

Fig. 6. Welcome page of Test Bench built-in user guidance

7 Conclusions and Future Work

One central goal of DECOS is to support the modular certifiability of integrated dependable systems (such as the DECOS-based systems) according to the safety standards, e.g. IEC 61508. This process is called the DECOS Certification Process. This means that the safety case for the application contains only the application specific evidence and reuses the results generic safety cases for the basic DECOS artifacts such as the DECOS nodes and the core services. The Generic Test Bench supports this process by providing generic v-plans for safety standards, documentation support in order to generate the validation report from the completed v-plans and built-in user guidance in terms of a help file.
The future work will comprise:

- Extending the set of generic v-plans for further safety standards for various domains (e.g. for the DO 178 B for aerospace)
- Establishment of further v-plans and evidence for the generic safety cases
- Finishing the built-in user guidance in order to cover all aspects of the validation and certification process. In the last phase of the project users will test for usability, completeness and understandability.

8 References

[1] H. Kopetz, R. Obermaisser, P. Peti and N. Suri, “From a Federated to an Integrated Architecture for Dependable Embedded Real-Time Systems”, TU Vienna University of Technology, Austria, and Darmstadt University of Technology, Germany, 2004

[2] DECOS: Dependable Embedded Components and Systems, Integrated Project within the EU Framework Programme 6, http://www.decos.at
[3] Association of German Car Manufacturers (VDA). HAWK2015 – Challenges for the automotive supply chain. Henrich Druck + Medien GmbH, Frankfurt am Main, 2003 (in German).
[4] G. Weißenbacher, W. Herzner, E. Althammer, “Allocation of Dependable Software Modules under Consideration of Replicas“, Proceedings of the ERCIM/DECOS Workshop on Dependable Software-Intensive Embedded Systems at Euromicro 2005, Porto, Portugal. Proceedings published by ERCIM, ISBN 2-912335-18-8, p. 51-58
[5] E. Schoitsch, “The Integrated Project DECOS, From a Federated to an Integrated Architecture for Dependable Safety-Critical Embedded Systems – an Overview”, Proceedings of the ERCIM/DECOS Workshop on Dependable Software-Intensive Embedded Systems at Euromicro 2005, Porto, Portugal, Proceedings published by ERCIM, ISBN 2-912335-18-8, p. 9-14
[6] E. Schoitsch, "Design for Safety AND Security of Complex Embedded Systems: A Unified Approach"; invited presentation des NATO Advanced Research Workshops, TU Gdansk, Springer, “Cyberspace Security and Defense: Research Issues”, ISBN-10 1-4020-3380-X, p. 161-174
[7] T. Kelly, “A Systematic Approach to Safety Case Management”, Proceedings of the SAE Conference, 2003
[8] NASA Formal methods site, http://www.nasa.gov
[9] R. Atkinson, “COTS Tools Reduce the Cost of Embedded Software Certification”, COTS Journal, 2002
[10] S.C. Johnson R.W. and Butler, “Design for Validation”, IEEE Aerospace and Electronic Systems Magazine, 1992
[11] J. Rushby, “Modular Certification”, SRI Technical Report, 2001
[12] M. Nicholson et al., “Generating and Maintaining a Safety Argument for Integrated Modular Systems”, In Proceedings of the 5th Australian Workshop on Safety Critical Systems and Software, 2000
[13] E. Schoitsch et al., “Validation and Certification of Safety-Critical Embedded Systems – The DECOS Test Bench”, Proceedings of the SAFECOMP 2006, Gdansk, Poland, Proceedings published by Springer, LNCS 4166, p. 372 – 385