FP7 Project 2007- Grant agreement n°: 218575

Project Acronym: **INESS**

Project Title: **INtegrated European Signalling System**

Instrument: Large-scale integrating project
Thematic Priority: Transport

Document Title: **INESS_WS G_Deliverable 2.1_WS_Finalized_Report_Ver2009-09-08**

Due date of deliverable	2009-06-30
Actual submission date	2009-09-08

Deliverable ID:	D.G.2.1
Deliverable Title:	Definition of long term goals
WP related:	Safety Case Process Strategy Development
Responsible partner:	TUBS
Task/Deliverable leader Name:	Jörg R. Müller
Contributors:	TUBS, Funkwerk, DLR, DB, BBR, RFI, Ansaldo, Invensys

Start date of the project: 01-10-2008 Duration: 36 Months

Project coordinator: Paolo De Cicco
Project coordinator organisation: UIC

Revision: Dissemination Level[1]: CO

[1] PU: Public, PP: Restricted to other programme participants (including the Commission Services), RE: Restricted to a group specified by the consortium (including the Commission Services), CO: Confidential, only for members of the consortium (including the Commission Services).

Document Information

Document type: Report
Document Name: INESS_WS G_Deliverable 2.1_WS_Finalized_Report_Ver2009-09-08
Revision: 7
Revision Date: 2009-09-08
Author: Jörg. R. Müller
Dissemination level: CO

Approvals

	Name	Company	Date	Visa
WP leader	Jörg R. Müller	TUBS		
WS Leader	Jörg R. Müller	TUBS		
Project Manager				
Steering Board				

Document history

Revision	Date	Modification	Author
1	2009-06-22	Creation of document	J. Schröder
2	2009-07-02	Moved attachment to report D.G.2.3	J. Schröder
3	2009-07-30	Restructured, Rewritten	J. R. Müller
4	2009-07-31	GSN-introduction/notation moved To D.G.2.2/D.G.4.2	J. R. Müller
5	2009-08-24	Section Commercial Tools, Remote Access added	G. Buxhoeveden
6	2009-09-01	Estimation of Efforts	Jörg R. Müller
7	2009-09-07	Adjustment to PMC comments	G. Buxhoeveden

TABLE OF CONTENTS

Glossary

The following abbreviations are applied in this document:

C	Communication
CMIS	Content Management Interoperability Services
CS	Creation of Safety Case
DB	Data Base
DM	Document Management
DMS	Document Management System
GSN	Goal Structuring Notation
RFC	Request For Comments
SaCa	Safety Case
SM	Safety Management
W	Workflow
WebDAV	Webbased Distributed Authoring and Versioning
ROCS	Railway Operation Control

Section 1 – Executive Summary

1.1 The context of workstream G

The aim of workstream G is to reduce time and money for the Safety Case in industry, i.e. operators as well as suppliers, by avoiding unnecessary or redundant procedures. To achieve this aim one can identify four phases in workstream G (see figure 1).

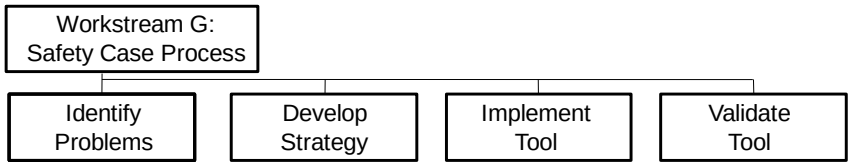


Figure 1: One can specify four phases to achieve the aim of Workstream G

According to the DoW for the second and third phase the following procedure was planned:

1. Define long term and short term goals (task 2.1 and task 2.2)
2. Estimate economical benefit (task 2.3)
3. Specify requirement description technique (task 3.2)
4. Specification of improved Safety Case Process (task 4.1)
5. Specification of system and software requirements (task 4.2)
6. Implementation of tools (task 5.1)

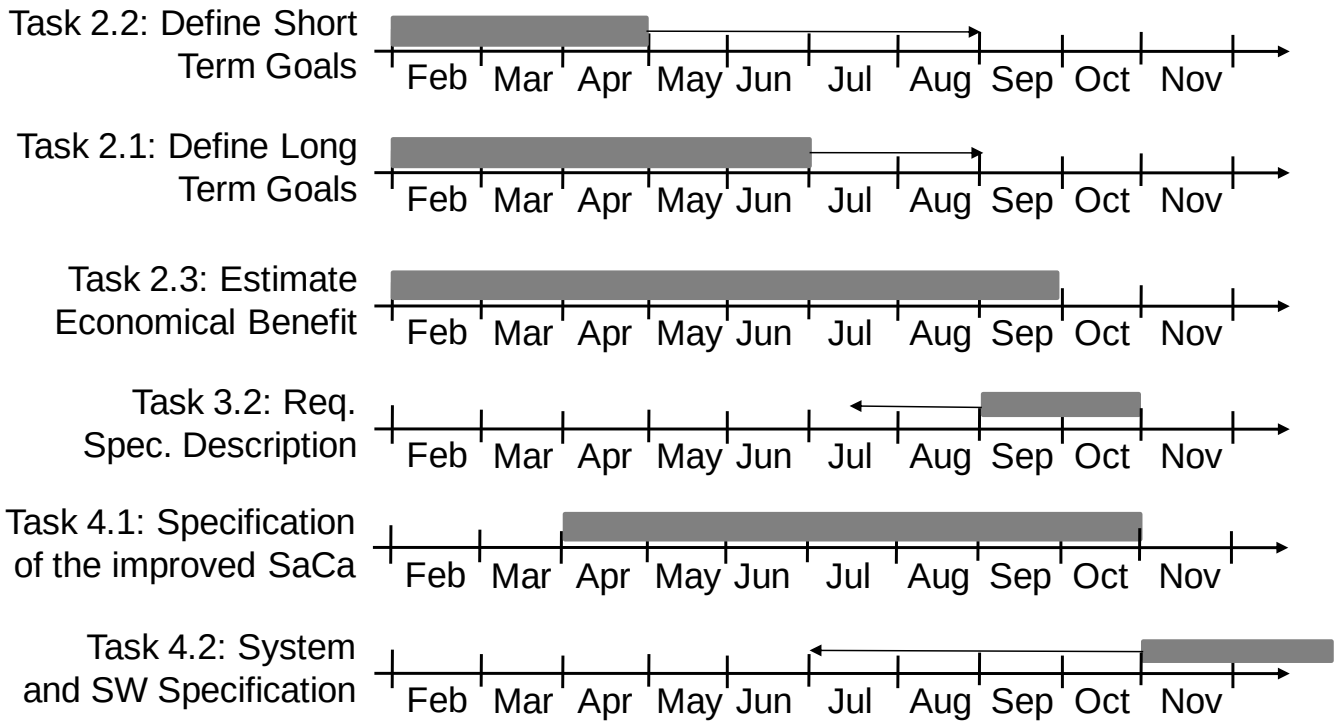


Figure 2: Rescheduling the Tasks in WS G

According to the interviews as well as to the findings of the second workshop, it became clear, that most problems to be solved are related to the realm of workflow and document management. Many of the desired functions have already been implemented in freely available open source applications. Therefore it was agreed, that resources shouldn't be wasted on implementing functions that are available elsewhere. The advantage of using open source software is that a lot of desired functions come "for free", thus offering "more benefit" for "less cost".

A more detailed analysis of open source software reveals a far broader perspective why open source software could be essential for businesses operating on long term projects. The case for open source software can be found in the annex.

The drawback for the project on the other hand is that a few tasks have to be rescheduled. The new project plan is somewhat less sequential but more interwoven, as some tasks now happen in parallel – see Figure 2.

1.2 The aim of task G.2.1

Not all desirable goals that will lead to an improvement of safety case processes and thus to saving of cost (economical benefit) can be achieved in INESS project because of limited time and resources. Therefore, in work package G.2.1 (long term goals) future goals for the time after the INESS project are specified.

This work package is closely linked with work package G.2.2 (short term goals), as every goal fits in only one of the both categories.

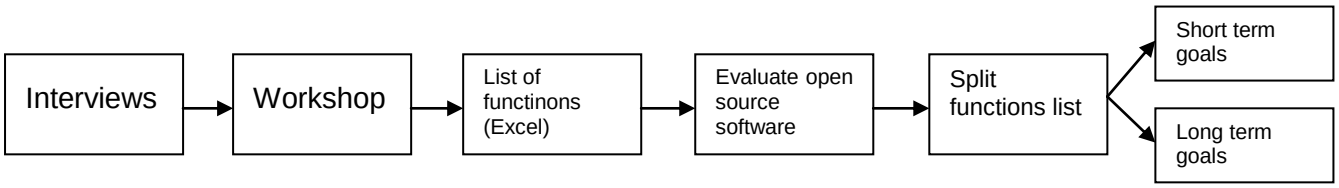
An essential part of long and short term goals are the functions to be implemented in the tool that has to be developed in the project, so results of both work packages will be lists of these functions.

It should be mentioned that the long term goals shall be realized on top of the short term goals. All the efforts undertaken for the software tool within the INESS project can stay as they are. Long term functions can simply be added to the tool, thus extending the functionality and adding more benefit.

Furthermore a long term vision is developed in this document. This will be done by taking the long term goals, short term goals and their identified functions to the next level by defining the outline of a future tool that supports the safety case writer to full extend.

Section 2 – Definition of Long Term Goals

From the practitioners interviews a list of functions was derived, which was split up in long term and short term goals. The short term goals can be found in Deliverable D.G.2.2. The process of identifying the goals can be depicted a follows.



This document will deal with the long term goals.

The identified long term goals fall into the following five categories derived from the function list. Explanations for each category are provided in the following sections of this document.

- Integration of the Goals Structuring Notation with the Documents Management System (see 2.1)
- Integration of the INESS tools with other available commercial tools (see 2.2)
- Adding remote access to the Document Management System (see 2.3)
- Providing additional country specific information for the safety case (see. 2.4)
- Integration of risk analysis tools and connection to requirements management tools (see 2.5)

2.1 On-line analysis of Goal Structures

The “Goal Structuring Notation” (GSN) is a technique to improve the structure, rigor, and clarity of safety arguments (see e. g. [1,2,3] and deliverables D.G.2.2 and D.G.4.2). The Goal Structuring Notation is a graphical argumentation notation that explicitly represents the individual elements of any safety argument (requirements, claims, evidence and context) and (perhaps more significantly) the relationships that exist between these elements (i.e. how individual requirements are supported by specific claims, how claims are supported by evidence and the assumed context that is defined for the argument). This technique is therefore predestined to support the safety case manager. A long term goal will be to integrate a goal structure into the DMS in a way, that the (sub-)goals of such a structure are automatically being evaluated according to modifications of the corresponding documents in the DB (figure 3).

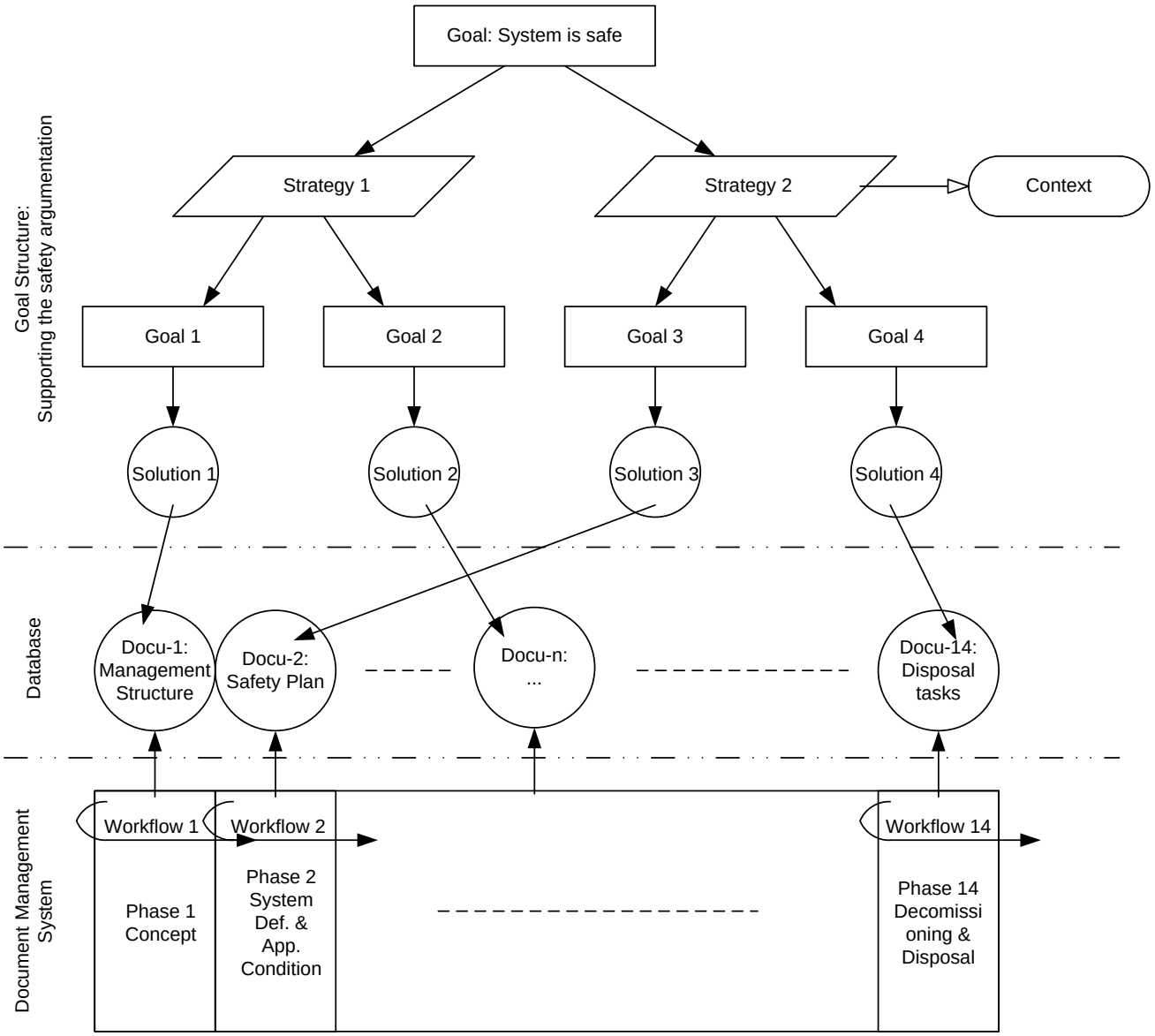


Figure 3: Integration of DMS and Goal Structure

2.2 Broadening the Interoperability with commercial Tools

The software modules developed in WS G will serve the participating parties of the workstream as useful tools to assist in the management of the safety case. As the tools will be validated by workstream participant Funkwerk, a complete company integration will happen according to Funkwerk's IT-environment. To realize a flexible integration of these tools in a business environment of any company interested in such tools, adjustments to the tools have to be made.

Before these adjustments can be done an assessment of needed interfaces to existing solutions has to be performed. It can be assumed that most companies already use software tools to support the safety case to some extend. Word processing packages and spread sheet applications can be regarded as a standard. Document Management Systems might be in use, but are somewhat less standardized. Tools for Requirements Management might also be in use.

To realize a thorough integration of the INESS tools, the existing solutions within a company will have to be analyzed. At least in theory to kinds of adjustments are possible:

- Adjustments to the INESS tools communication interfaces
- Adjustments to other open source tools (i.e. DMS)

So far it is planned to implement known and established communication interfaces in the INESS tools which require none or only minor adjustments to work together with other solutions that implement the same interfaces (WebDAV, CMIS). A sensible prognosis about the amount of work to extend the interfaces of the INESS tools can only be done, once the concrete IT-system is known in which the INESS tools should be integrated.

The following categories with need of interoperability could be identified:

- Integration with existing databases
- Integration with existing DMS
- Integration with requirements management tools
- Integration with risk management tools (see 2.5)

2.2.1 Integration with existing databases

Most companies already use a vast variety of software packages to perform their business. Solutions that are already at work in a company which might be helpful to be integrated with the INESS tools could be:

Knowledgebase: A database where "company knowledge" is stored. These systems come in a variety of flavours. It could be an online communication forum, it could be a list of frequently asked questions, it could even be a wiki where every employee contributes. These systems all depend on acceptance by the employees, but once they are established offer a good source of information.

Existing projects database: Existing projects offer a good source of information to learn for current or future projects. Access to this kind of information would be helpful for the safety case.

Office management tools: Information about timelines or meetings is often stored in different places. Access to this kind of information would make the safety case more transparent. It should be kept in mind that acceptance of tools drops dramatically if it is necessary to input identical information in different tools.

Bug tracking software: Within the software development departments bug tracking packages are commonplace. The information aggregated in such systems can provide useful information about the current state of software modules and could be used by the safety manager to get more insight to software development.

2.2.2 Integration with existing DMS

Document Management Systems can be found in lager companies but are nowadays also available for small companies. An interface to a non documented legacy system will most probably not be possible. But once the DMS implements one or another open standard, chances are good that it is possible to adjust the INESS tools to work together with such an existing system. Currently two interfaces could be identified to be useful for integration purposes.

WebDAV: The Web Distributed Versioning and Authoring Standard is an extension to the http-protocol and it exists since the early 90s. It is standardized in RFC 4918 and is at least in parts implemented in most DMSs.

CMIS: The Content Management Interoperability Standard is a very new interface that is currently being developed by major providers of DMSs (Alfresco, Day Software, EMC, IBM, Microsoft, Open Text, Oracle and SAP). Software libraries that implement the preliminary standard are already freely available. CMIS itself is based on open protocol standards such as SOAP (Simple Object Access Protocol) and REST (Representational State Transfer).

2.2.3 Integration with requirements management tools

If requirements change at any stage of a project that has some safety implications it is quite likely that effects on the safety case occur. Therefore it would be helpful to integrate the management of the safety case with existing requirements management solutions.

In a simple first step scenario it would already be helpful to perform an import of safety relevant requirements into the tools developed within the INESS projects. If the data format of the requirements management software is well documented, such an import interface shouldn't be too complicated to implement.

In a more demanding scenario the INESS tools would have "live" access to the requirements database and could track changes as they happen. In this scenario the communication interface and structure of the database of the requirements software has to be known and well documented to implement interfaces within the INESS tools.

2.3 Remote Access to Safety Case Documents

Once a working DMS has been established in a company and it is being used for the management of the safety case a next step should be to integrate external parties (i.e. assessors, operators) into the whole process. If an assessor has the chance to accompany the whole development of the safety case, he can intervene at any time, if he notices a practice that does not conform to the norm or if he sees required documents missing.

It can be assumed that it is easier to be involved early in the whole process of developing the safety case, than to get a large amount of documentation close to the end of a project.

The goal of remote access to documents is closely related to the administration of the companies IT-system. Probably nothing has to be adjusted regarding the tools that were developed within the INESS project. But as these tools live in a software ecosystem yet unknown no reliable forecast can be performed to what extend the INESS tools might have to be adjusted to realize remote access.

2.4 Provision of Country specific Advices

Currently most companies have experience with the operators and markets they deliver their products to. To extend their business to other markets often only little knowledge is available in a company what procedures to follow and whom to ask to get acceptance of their products in a yet unknown environment.

A software supported guideline or help system that can be consulted to find out about deviations of standards in a country would help safety managers to take care of their documents at an early stage in the project.

2.5 Supporting the Risk Analysis and the Identification of Safety Requirements

The risk analysis and the derivation of the safety requirements have been identified as major tasks that cause many problems for the industry. Even if “all” requirements are well thought of and are written in an unambiguous way earlier or later a change request is necessary. This could be either by a customer (i.e. a special “feature” is needed) or could be technical reasons due to wrong assumptions (e.g. a subcontracting supplier can not deliver a certain component to a given specification). This implies that until a product is completely delivered and assessed it is almost certain that it will evolve and change during the concept and development phases. It is desirable to reduce this fact to a minimum. Nevertheless the changes that can not be avoided have to be traced carefully and have to be integrated into the planned system.

(General Dario Nosengo, Specific 47), Relation to D4.4: there “pragmatic”, INESS-specific and qualitative.

In the following sections a formal approach to modelling failure rates and dependability is described. The approach is a very generic one and can be applied to qualitative and quantitative data. A “pragmatic” implementation of this approach is described in Deliverable D.4.4. A description of the relationship to D.4.4 can be found in section 2.5.x.

2.5.1 Supporting the Identification of Risks

To support the identification of product risks, the use of creativity techniques (see e.g. [11]) has proven to be valuable. As this task is highly dependent on unquantifiable expert knowledge, the aim is to support the use of creativity techniques like

- Brainstorming (Generating many radical ideas)
- Mind Mapping (quickly identify and understand the structure of a subject)
- Morphological thinking (Using comparisons to express ideas and solve problems)
- Starbusting (Understanding new ideas by brainstorming questions)
- Affinity Diagrams (Organizing ideas into common themes)
- Crawford’s Slip Writing Method (Gathering ideas from many contributors)

After INESS-time it could be worth examining the tools that are available to deal with these techniques. An evaluation of these tools with respect to

- the specific needs when performing risk analyses within the railway area
- their ability to be integrated in the tool environment used in the railway domain, i.e. the tools’ interfaces.

After having identified appropriate tools, their integration with the commonly used DMS is to be considered.

2.5.2 Model-based Safety Analysis – The PROFUND approach

Model-based approaches offer an alternative solution to traditional risk analysis methods. Model-based approaches are based on the use of formal descriptions that have recently proved useful as well as practical numerous times, for instance, software development or system specification. Moreover, there is more and more research in the field of safety analysis trying to formally use these methods (see e.g. [10]). These approaches namely cover application fields that also include railway systems and all have their origin in computer science. In this respect they made an attempt to validate a formal system specification using formal specified safety requirements.

In order to define and validate the safety requirements for the system's functions and the system components according to the CENELEC RAMS norms, it is necessary to describe the relation of the occurrence of undesirable operational events and the functional and component-specific failure potential, respectively (dependability). Figure 4 depicts the influential aspects to be considered in a safety analysis, which are:

- The process with its regular and hazardous behaviour, including possible failures,
- The system's functionality with its regular and failure behavior, as well as
- The system implementation including regular and failure behavior of the implementation components.

In other words, the necessary modeling must be done for the function of stochastic-deterministic behavior of the process to be controlled, the deterministic behavior of correct system functionality and implementation, respectively, as well as the stochastic behavior of function and implementation dependability. The integrated observation of process (**PRO**cess), functionality (**FUN**Ctionality), and dependability (**D**ependability) forms the basis for the PROFUND approach (see e.g. [4])

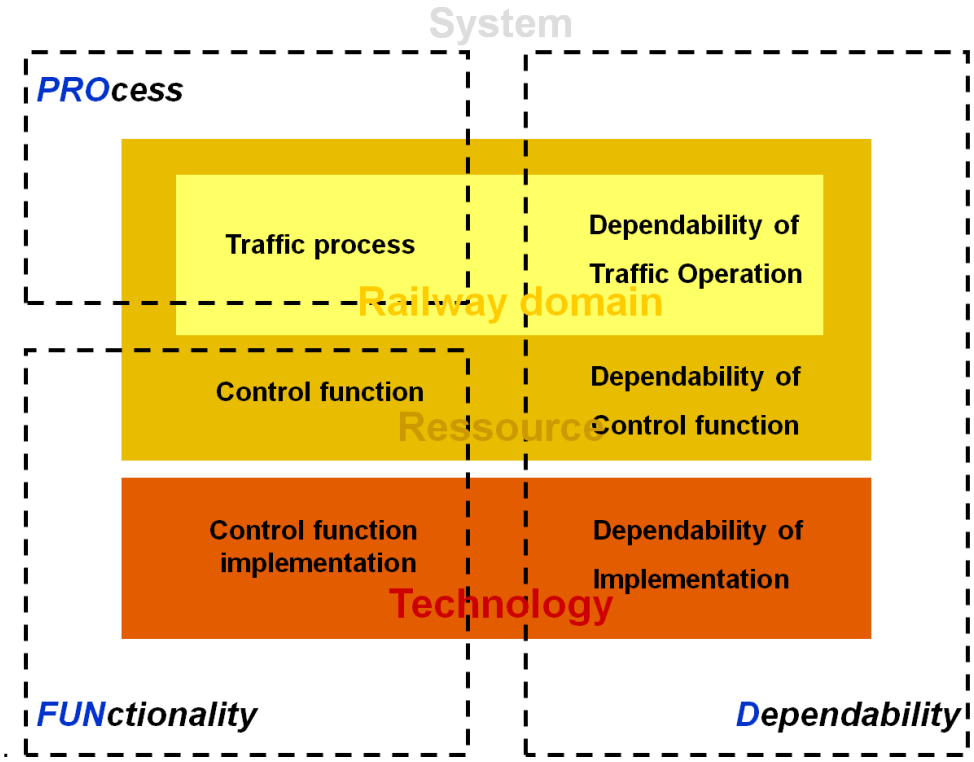


Figure 4: Process, Function and Implementation according to PROFUND

The PROFUND approach serves to continuously support all of the safety analysis' tasks with a formal description. It bases on a model which can be gradually set up during the safety analysis. It also can be used to perform a quantitative evaluation at any given point of time. Figure 5 graphically indicates this procedure.

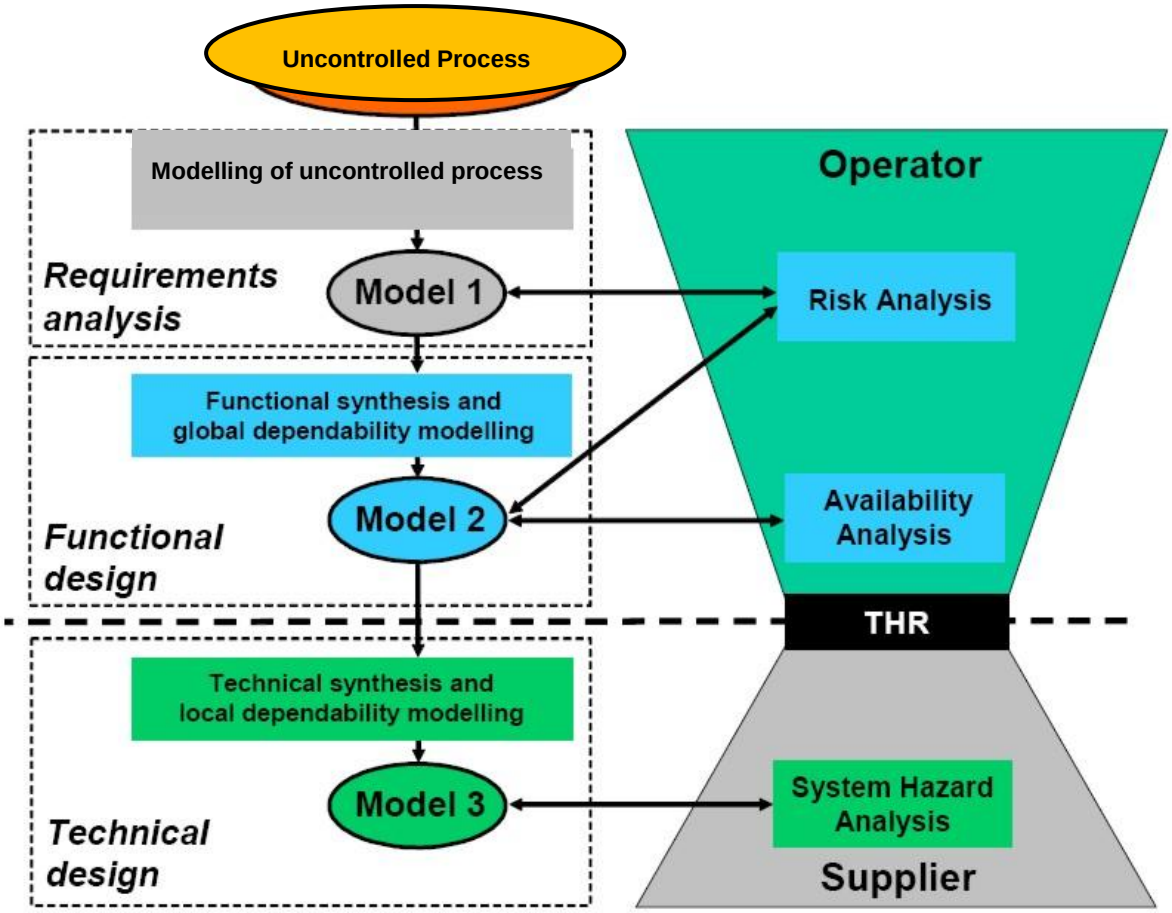


Figure 5: PROFUND: Risk-Analysis – Methodical Approach

The first step is to identify hazards and their possible consequences and to model the behaviour / process itself (Model 1). The analysis of this model in itself can be used to evaluate the operational risk. However, since the controller-functionality was not included, the results only reveal information about the hypothetical operational risk of uncontrolled behaviour / process. Nevertheless, if a suitable criterion for tolerable risk is applied, they can be used to quantify the requirements for risk reduction necessary. Creating a model of the behaviour / process makes it possible to formally determine a foundation for comparison of influences of different functional or technical specifications of the control-function.

The next step is to describe the functionality of the controller important for the safety analysis (Model 2). Due to a clear distinction of the behaviour / process to be controlled and controller functionality, the description can also be extended by adding the potential functional dependability using the functional threats identified in the Risk analysis. Connecting such a model and the model of the uncontrolled process, allows the interpretation of the functional safety goals.

The technical threats identified in the risk analysis form the basis for describing the intended technical implementation and its dependability (Model 3). A model of that kind can be used to determine the safety goals for system components by expressing them in tolerable failure rates (risk analysis result). However, this procedure requires the consideration of the functional safety goals, or a direct connection to Model 2 (uncontrolled behaviour / process and system functionality). In case these goals are technically impossible to achieve, an (iterative) extension of the basic technical specification must be

performed. In order to do so, further measures (e.g. monitoring, redundancy, etc.) to attain an increase of dependability are necessary. As soon as the functional safety goals are achieved, the resulting technical specification including the safety goals for system components can be used to implement the control device.

2.5.3 The Modelling Language

As required by the CENELEC Standards, a suitable formal modelling language has to be applied as a support of the requirements analysis, and of the functional and technical design. The constructed PROFUND model should allow the combined description of the logical and functional system behaviour as well as the temporal and dependable device behaviour and its qualitative as well as quantitative analysis.

The main aim of the qualitative analysis is to provide formal support for the examination of the reachable state space of the modelled process. It should be used for identifying the hazardous situations based on undesired operational events. Therefore on the one hand a global view is necessary. On the other hand, there are partly autonomous processes of operational and technical objects, which have to be modelled locally in a concurrent way.

Using the quantitative analysis (stationary and transient analysis), the probability of undesired states or hazardous events related to any time of the system-lifecycle has to be evaluated. Inputs can be the statistical properties and parameters of the operational process as well as diverse stochastic characteristics of the failure behaviour of the technical components in use.

To fulfil the mentioned requirements different formal methods can be used. Several approaches to the formal functional modelling of ROCS (Railway Operation Control) can be found e.g. in ([5] and [6]). In order to enable even the quantitative analysis, a subclass of Stochastic Petri Nets could be chosen as the modelling language. In general Petri Nets are used for the description of discrete event systems in a causal and temporal view. The structure of a Petri Net is visualized as a bipartial graph. The two disjunctive types of nodes connected by arcs are places (circles) and transitions (rectangles). Places can be interpreted as local system states or conditions for events in system behaviour and transitions represent events themselves. The meeting of conditions or reaching of states is represented by a marking within the place – see Figure 6. The set of all marked places represents the global state of the modeled system. An event may occur, i.e. the transition is allowed to fire, if all preconditions (input places) are marked. The occurrence of a transition is related to the change of a global system state. In this way a Petri net model represents not only the static structure but also the dynamic behaviour of the modeled system.

Figure 6 shows the graphical representation of the elements in a stochastic Petri Net. Such models allow qualitative and also quantitative analyses for proving performance and safety properties of the system described by the net (see e.g. (G)). The transient analysis is especially relevant to the dependability analysis, because it allows an analytical calculation of occurrence probability of certain system states in relation to the system life time. This kind of Petri net class as well as this analysis technique is supported by software tools like “TimeNET 3.0” ([8]) or the „Pi-Tool“ ([9]) which was used to model analysis carried out to obtain the results presented in the paper.

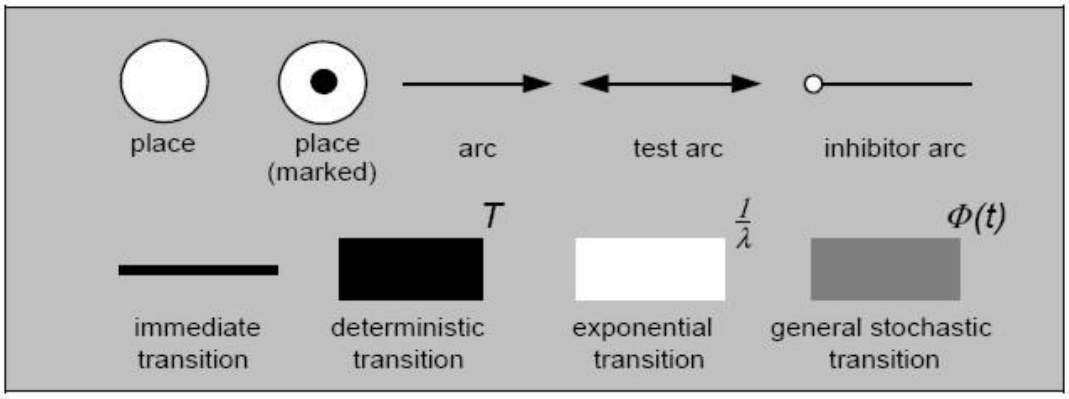


Figure 6: Graphical Elements of stochastic Petri Nets

2.5.4 Specific implementation of the approach in D.4.4

In section 4.1 of D.4.4 a hazard analysis is performed by identifying accidents and hazards, performing a causal analysis and linking those to safety requirements. The following links can be applied from the generic approach presented here to the specific implementation in D.4.4:

- Relations: D4.4 hazards and accidents can be identified in the process model when no function-model is taken into account.
- The safety requirements will always be fulfilled by the functional model, when no dependability model is taken into account.
- In addition to the approach in D.4.4 the quantitative measures can be found by taking into account the dependability model.

2.6 Migration plan

Currently it is planned to have the tools that are to be developed within the INESS project in an independently working state. I.e. the tools do not rely on any existing software within a specific company. The whole toolset should work autonomously. Nevertheless a single tool, like the GSN-Tool shall be able to work on its own, even without a document management system. To get the full potential out of the tool, communication with existing solutions would be helpful.

In addition to the long term goals described in the preceding sections the functions list that was discussed on the second WS G workshop provides information about users wishes for software support of the safety case. Most of these functions could be assigned to one of the already stated goals. But nevertheless these functions, as they were described by the workshop attendees provide valuable information to software developers, what shall be implemented in the future.

The tools being developed in the scope of short term goals shall be developed in a modular, on open standards based way to minimize the need of customization. The tools shall work in their “short term goal” state and shall be extendable with the functions mentioned in the former sections.

A rough estimation of the efforts to implement the functionalities described in sections 2.1 to 2.5 is listed in the following table: Please note: The actual effort depends on the experiences made within INESS time.

Section: 2.1

Name of functionality: On-line analysis of Goal Structures

Estimated Effort: 6 PM

Remark: A “deep” integration of the GSN tool with a background logic to evaluate the current working status has to be implemented.

Section: 2.2

Name of functionality: Broadening the interoperability with commercial Tools

Estimated Effort: 3 PM for each interface / commercial tool

Remark: It is probable that the interoperability with some existing tools will be quite easy and the interoperability with others will be quite time consuming.

Section: 2.3

Name of functionality: Remote access to Safety Case Documents

Estimated Effort: 2 PM

Remark: general administration of network(s).

Section: 2.4

Name of functionality: Provision of Country specific advises

Estimated Effort: 1 PM

Remark: pure administrative work

Section: 2.5

Name of functionality: Supporting the risk analysis and Identification of Safety Requirements

Estimated Effort: 30 PM

Remark:

- To support the identification of risks corresponding tools have to be evaluated among others with respect to a possible integration (3 PM).
- The presented PROFUND-concept is clear and is being standardized (IEC 62551, CD2). This method has to be explained in a user friendly way supported by easy examples (3 PM).
- There exist open-source Petri-net modelling tools. However, these have to be evaluated and at least adapted to the specific application area. If there is no appropriate modelling tool available, a new tool has to be implemented (18 PM).
- A tool to guide the practitioners and integrate the chosen / developed PN-tool as well as the creativity tool has to be developed and integrated (6 PM).

2.7 Resulting Long Term Goals

Up to now, the following list of long term goals can be identified:

1. Highly automated resolving of goal structures
2. Broadening the interoperability with commercial tools on the basis of a set of interfaces
3. Remote Access to Safety Case Documents
4. The provision of country-specific advices
5. Integrated concept to support the risk and safety analysis

Apart from these long term goals, which were derived from the function list, an overall vision of a future tool shall be helpful to get an idea of how such a tool might enhance the safety case process.

The following ideas may be considered for future development:

Completeness check: This function should check within a GSN tree, if all CENELEC required documents are present in the safety argument. The resulting documents should be less error prone.

"Where am I" function: Clicking on a document, process or any clickable element should reveal to which CENELEC phase it belongs and to which other tasks it is linked and which persons are involved. A highly automated guidance system could help the safety case manager to ease the way through producing all necessary documentation.

Integration with software development tools: The GSN Editor is based on an open "plug-in" standard. The tool could be integrated with other source code editors or modelling tools or version management systems. Plug-ins for requirements management are also available. Linking those with the GSN would ease traceability of requirements to the safety case.

Linking requirements to source code and documentation: It would be helpful to trace requirements from every point in the development back to their source. Linking requirements to models of a model based architecture approach of software development would most certainly enhance transparency and quality of a software product.

Section 3 – Conclusions

The long term goals mentioned in this document may be a pool of ideas yet to be analysed in the future, once the short term goals are implemented. To kinds of long term goals could be identified.

Things like help systems or online guidelines (e.g. for country specific advices) require a large amount of regular “office work” to be done, before they could be integrated in a software tool. The integration itself (from a software developer’s point of view) shouldn’t be very complicated.

Things like the analysis of the goal structure and the implications for the safety case, once a document has changed need a thorough modelling approach. Development of such functions will take considerably more time, compared to the “office work” of the other goals. The inclusion of risk analysis in the whole process can be regarded equally time consuming.

Section 4 – Bibliography

[1] T. P. Kelly: Arguing Safety – A Systematic Approach to Safety Case Management, DPhil Thesis YCST99-05, Department of Computer Science, University of York, UK, 1998.

[2] Tim Kelly and Rob Weaver: The Goal Structuring Notation – A Safety Argument Notation, Department of Computer Science and Department of Management Studies, University of York, UK.

[3] European Organisation for the Safety of Air Navigaion: Safety Case Development Manual, European Air Traffic Management, Eurocontrol, 2006.

[4] R. Slovak, J. May, E. Schnieder: Profund Modelling for Holistic Risk and availability Analysis by means of stochastic Petri nets applied to a Level Crossing Control System. In: G. Tarnai, E. Schnieder, Eds.: Formal Methods for Railway Operation and Control Systems, p. 221-232, Budapest, 2003. Symposium FORMS 2003, L'Harmattan.

[5] M. Meyer zu Hörste, E. Schnieder: Formal Modelling and Simulation of Train Control Systems using Petri Nets: Proc. Of World Congress on Formal Methods in the Development of Computing Systems. LNCS, Springer-Verlag, p. 1867,1999.

[6] J. Zahradník, K. Rástoný and A. Janota: UML - based Specification of a Railway Interlocking and Signalling System, Proc. Of Workshop on Software Specification of Safety relevant Transportation Control Tasks (E. Schnieder, Ed.), VDI-Report, 131-142, 2003.

[7] R. German and S. Ross: Performance Analysis of Communication Systems – Modeling with Non-Markovian Stochastic Petri Nets. John Wiley and Sons, 2000.

[8] TimeNET: URL:http://www.tu-ilmenau.de/fakia/8086.html (July 2009).

[9] *π-Tool:* Tool for modeling and analysis with stochastical Petri nets developed at the Institute for Traffic Safety and Automation Engineering of the Technical University of Braunschweig, 2009.

[10] K. M. Hansen: Linking Safety Analysis to Safety Requirements. Doktorarbeit, University of Denmark, 1996.

[11] URL: http://www.mindtools.com/pages/main/ThemedIndex.htm, (July 2009).

[12] URL: http://gforge.enseeiht.fr/docman/view.php/52/2924/Slides_2h.pdf (September 2009).

ANNEX

A Case for Open Source Software

The problem of software development can be best explained with an example (taken from [12]).

The program for the AIRBUS A300 began in 1972 and ended in 2007 (i.e. 35 years of “tracing requirements” or adjustments to “customer wishes”). But it doesn’t end there. AIRBUS offers support for the A300 until 2050. So the need for software to accompany a product is 78 years in this case. Similar time spans can be expected for products in the rail domain. The question is how to provide a software-based management system for the source code and for the documentation?

There are basically two ways to deal with this problem. Either you find a proper set of tools from a commercial provider, which satisfies your needs you get the following pros and cons:

Pro:

- Support for money

Con:

- No influence on product development
- Actual “working” of a tool is regarded as “intellectual property” or “company asset”, no one can control, if the tool really works as expected
- Vendor of the tool could be bought by another company, uncertain future of the product, lack of continuity
- Tool could be scrapped altogether, if the vendor does not see commercial success in a small market
- No possibility to migrate old data if the tool is not developed any further
- Licensing costs, licensing management

Or you opt for an open source alternative which offers different pros and cons:

Pro:

- No licensing cost
- Source Code is “open”, i.e. can be examined by everyone and checked if it really does, what is expected
- Source Code can be adjusted to future requirements (there will be lots of them in 50+ years…)
- Data handled by open source tools can be easily migrated as it is transparent how things are read and saved
- Needed functionalities can be programmed. No one has to be “persuaded” to add a feature to a tool
- Expected boost for SME to provide programming and services

Con:

- As no one “owns” the code one could argue no one is “responsible”. Which translates into another Pro not yet mentioned. Open Source is a community based effort. Knowledge is shared and there is no single source dependency. It can be regarded as a free market of ideas, where ultimately the best (or most popular) idea survives, but with the advantage that the “other” ideas just do not disappear (as with an unsuccessful software product) but can be seen and analysed. Data handled by such unsuccessful approaches can still be understood and transformed to a more future-proof system. Or if it is commercially viable a business supports the inferior system and adjusts the source code to their needs.