Safety-Related Application Conditions – A Balance between Safety Relevance and Handicaps for Applications

Friedemann Bitsch[1], Ulrich Feucht[2], and Huw Gough[2]

[1] Informatik Consulting Systems AG, Sonnenbergstr. 13, D-70184 Stuttgart, Germany
friedemann.bitsch@ics-ag.de
[2] Thales Rail Signalling Solutions GmbH, Lorenzstraße 10, D-70435 Stuttgart, Germany
{ulrich.feucht,huw-michael.gough}@thalesgroup.com

Abstract. Railway standards prescribe the use of Safety-related Application Conditions (SACs). SACs are demands to be observed when using a safety related system or a sub-system. The use of SACs can, however, easily be associated with difficulties. SACs of sub-systems can imply high efforts regarding their fulfillment at system level. Furthermore, SACs at sub-system level may become very obstructive for the user of the sub-system, if the safe application on system level has strong restrictions. Additionally, a large number of SACs may be very difficult to manage. In this way, SACs may obstruct the introduction of a system or a sub-system into the field. Particular hazards could arise from SACs, if they are formulated ambiguously, so that the originally intended safety-related measures are not taken at all. This paper presents the objectives and benefits of SACs and depicts difficulties and challenges associated with the use of SACs. The paper not only explains what should be the SAC content but also the quality criteria, the conditions for SAC creation and SAC fulfillment are described. The SAC management process introduced at Thales Rail Signalling Solutions GmbH is outlined. On the one hand, this process shall support the quality of SACs and on the other hand reduce the effort for SAC creation, fulfillment and evidence.

Keywords: Safety-related Application Conditions, SAC quality, conditions for defining SACs, process for defining and complying with SACs.

1 Introduction

Safety cases for safety-related railway control systems must be created for safety-related items[1]. A majority of the argumentation in the safety case is directed towards the internal attributes of the item. Moreover, also hazards are identified which cannot be covered by the internal attributes of the item itself, but rather through the adherence to certain requirements during the usage of the item in the intended superior

[1] The term "item" is used in this paper as an umbrella term for a system, a subsystem, a product or a component. A system can include several subsystems which can include several products. A product can be constructed from several components.

B. Buth, G. Rabe, T. Seyfarth (Eds.): SAFECOMP 2009, LNCS 5775, pp. 32–45, 2009.

context. These requirements are Safety-related Application Conditions (SACs). They must be documented in the item safety case and handed over to the responsibility of the user of the item. The superior context means either the application by an end user or the application in the development on a superior level (compare with the levels described in section 2 and Fig. 1). SACs document the conditions which must be followed during the usage of the item in a superior context due to safety-related reasons, so that hazards are avoided. The adherence to conditions remains the responsibility of the user. However, it is safety critical if SACs are not fulfilled by the user e.g. because of communication problems about the content of the SAC or because the user does not perceive why the instruction of the SAC is necessary for safety.

An example of non-compliance with a safety-related regulation is explained in the judgment [2] for the Transrapid accident in Emsland, Germany in 2006. According to [2] the regulation of the manufacturing company was not fulfilled which defined that the electronic route gate has to be set obligatory in case of shunting operations. [2] explains that this was not implemented in the operating rules.

It is often possible to decide whether a SAC which has been formulated can be solved by avoiding the SAC altogether if measures are designed within the boundaries of the item itself, otherwise the decision is made to make development improvements on superior system level. Such SACs which could have been avoided, can implicate high efforts at fulfillment on superior system level. Avoidable SACs also may be unneeded and unreasonable demands for appliers when the required safe application of a system or product is very extensive, highly restrictive, if the SACs are difficult to interpret or if the amount of SACs is unmanageably large. In this way SACs may obstruct the introduction of a product in the market. SACs without real safety character complicate and handicap the application of the item unnecessarily.

Therefore approaches are necessary which support the creation of SACs with clear and precise description of their content and clearness about their safety relevance, the decision in which cases SACs are necessary and in which other cases SACs should be avoided and the compliancy with SACs without high efforts.

In section 2 the benefit of SACs is pointed out and a definition for SACs is given. Requirements of safety standards and related works for the SAC topic are explained in section 3. On that basis challenges and risks with SACs are handled in section 4 and needs for creating, complying with and demonstrating SACs are derived. In the sections 5 and 6 criteria for SAC creation and quality are introduced. Processes for defining and handling SACs are presented in section 7. Important issues for SAC quality and efficient handling with SACs are summarized in section 8.

2 Meaning and Purpose of SACs

2.1 Benefits of SACs

Before it is defined what SACs exactly are the question shall be pursued for what SACs are useful and necessary. SACs involve several benefits in the Product Life Cycle. SACs assure safe operation of products by prescribing demands, which ensure the safe deployment of a system. SACs are important to give users clear safety-relevant instructions. Consequently, SACs are necessary for safety. They are prescribed compellingly

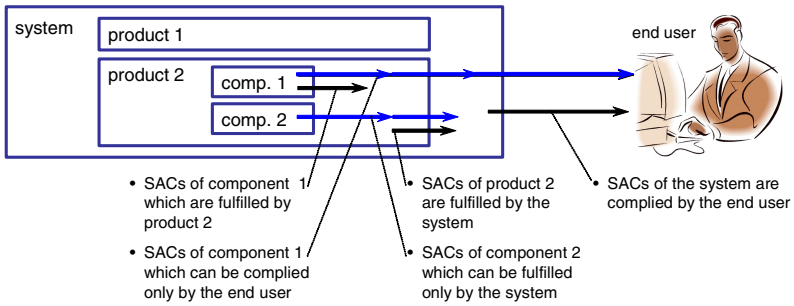

Fig. 1. Examples on which levels SACs are forwarded for fulfillment

by the railway standard EN50129 [1]. SACs can clarify safety responsibilities when using a system or a product in the phases after the development and the safety case have been completed, e.g. who of the end users has which safety responsibility. SACs clarify which safety responsibilities the maintenance staff, the rail traffic controller and the operating company have. SACs from subordinate items can clarify which safety responsibilities are on component, on product and on system levels. In Fig. 1 examples are given on which levels SACs could be forwarded to superior levels.

A typical example of a SAC which has to be fulfilled at development of a superior item, here a generic platform: *The application must ensure that a restart is possible only after the hardware has been reset. Reasoning: A soft reset is not sufficient for a safe restart. As the generic platform is designed the hardware has to be reset for a safe restart. It is the task of the application development to ensure this.*

A typical example of a SAC for an end user for any device is: *At least once within 12 months, the maintenance engineers have to check the device outputs with a certain test program. Reasoning: The calculated hazard rates are valid, only if the user complies with the Failure Detection Time of 12 months for the output circuits.*

Furthermore SACs can also be contributions to an economical development. SACs can allow the deployment of an item by definition of rules for safe application also with inexpensive design decisions. This is the case if easily to fulfill SACs can be defined instead of cost-intensive complex design solutions which are hard to realize.

2.2 Definition of SACs

SACs can be defined as followed which complies also EN50129 [1]. SACs are:

- regulations, that must be observed during the usage of an item in a superior context due to safety reasons,
- regulations, whose compliance lies in the responsibility of the user and
- regulations, which can avoid hazards, that are not covered through internal characteristics of the item itself, but which can be covered through the adherence of external measures or conditions during the usage of the item.

An example of a SAC is: *A point has to be switched once in 12 months*. The superior context for this example is the application of the point by the railway operator. The hazard is that the point switch is not in the correct position when it is run over because

of an undetected failure in the switchover circuit. For avoidance of the hazard the external measure is the passing of the point in the test cycle of 12 months.

3 Related Works and State of the Art

A well-known accident which demonstrates the meaning of SACs is the Chernobyl disaster in 1986. This accident and the consequences of violating safety rules (SACs) for end users have been analyzed in detail in [3]. In this context psychological factors for violating safety rules are in focus and have been investigated in detail.

According to [3] an essential part of the accident causes were human failures. But "everything the [plant] operators did they did consciously and apparently with complete conviction that they were acting properly". [4] explains: "The plant operators, […] however, thought in terms of linear networks of causation rather than considering potential side effects of their decisions and actions" . To handle these kinds of problems the consideration of safety regulations is absolute necessary. The human errors of Chernobyl were the consequences of the contempt of safety related regulations.

One reason for violation of safety rules according to [3] is that safety reasons for the rule are unclear for the operators. Furthermore, safety rules often bring a special effort for application. Therefore the violation of rules can lead to a simplified application. If a safety rule has been violated sometimes without any negative consequences then the tendency is in succession that the rule would be violated regularly. Then actions are based on own estimation of the situation. But this is hazardous because the user does not know the internal system states and the side effects.

IEC 61508 [5] only requires mandatorily that there must be operational and maintenance instructions to avoid mistakes during operation and maintenance procedures. In addition it is stated that all instructions must be easily understood. Explicit requirements for instructions related to safety are missing.

However in the railway standard EN50129 [1] SACs creation and compliance is prescribed compellingly. But there is little guidance related to handling and quality of SACs. The meaning of SACs is explained and it is prescribed in which parts of the safety case SACs have to be handled. SACs are defined as rules, conditions and constraints which shall be observed in the application of the system/sub-system/equipment. SACs from the current item to the superior items are part of the current item Technical Safety Report. Beside possible general topics the following specific topics are named and explained which shall be addressed in SACs: Sub-system/equipment configuration and system build, operation and maintenance, operational safety monitoring and decommissioning and disposal. In "Part 5" of the Safety Case with the topic "Related Safety Cases" it shall be demonstrated that all the safety-related application conditions specified in each of the related sub-system/equipment Safety Cases are either fulfilled, or carried forward into the safety-related application conditions of the item under consideration.

[6] describes a concept, which divides a safety case into modular safety cases according to modular architecture designs. Safety case 'contracts' are used to record the interdependencies that exist between safety case modules – e.g. to show how the claims of one module support the arguments of another. Safety contracts constrain

the interactions that occur between objects, and hence can ensure system behavior is safe. These contracts are broken down into individual requirements placed on the parts of the system. In difference to safety contracts SACs as a rule are directed bottom-up in a system architecture, i.e. an item addresses rules for a safety-related correct application to the superior architectural level.

As explained in section 7 a related topic is the specification, the fulfillment and the evidence of safety requirements. According to [1] safety requirements specifications contain functional safety requirements and systematic and random failure integrity requirements. Functional safety requirements concern all safety relevant control and monitoring functions of the system. Failure integrity requirements are the requirements regarding systematic and random failures.

Safety requirements are as other requirements part of requirements engineering. According to IEEE requirements engineering has to be divided into requirements elicitation, requirements analysis, requirements specification and requirements validation [7]. [5] gives criteria fort he quality of safety requirements. They have to be clear, precise, unambiguous, verifiable, testable, maintainable and feasible; and written to aid comprehension by those who are likely to utilize the information.

Generally there are two strategies to fulfill safety requirements [8], p. 398. The first strategy is to avoid safety critical faults and failures. The second strategy is the avoidance of hazardous consequences from faults and failures. The fulfillment of functional safety requirements is demonstrated by requirements tracing, verification of the several development phases, testing and validation. The compliance with random failure integrity requirements (quantitative safety targets) is shown by hazard analyses. The fulfillment of systematic failure integrity requirements is based on the evidence that adequate means of quality and safety management have been performed and that techniques and measures have been used to reach the necessary level of confidence in the development (Safety Integrity Level) [1] [9].

In comparison to conventional safety requirements the peculiarity of SACs is their origin and the kind of addressees. The origin of SACs are item safety cases. SACs are relevant for other development projects or the users of the customers. They are directed bottom-up to the superior architectural levels while conventional safety requirements concern top-down relations.

4 What Is Necessary for Defining and Handling SACs?

4.1 Challenges and Risks with SACs

Beside the benefits of SACs problems have to be considered which may arise in connection with SACs. Furthermore at SAC formulation and handling the purpose of SACs can be missed if some difficulties with SACs are not dealt with and are not avoided. A consequence could be that the SACs are only handicaps in the development of the concerned items instead of being useful for safe application. In the following those problems and difficulties are listed:

- Poor comprehensibility of SACs for the user.
- Declaration of SACs, which in fact are no SACs. This could lead to a large quantity of unnecessary SACs. That would be hardly manageable and could lead to the possibility of individual SACs not being taken seriously.
- Declaration of SACs that could have been avoided during product development.
- Missing or late information about SACs which must be fulfilled.
- High time investment for the proof of compliancy with SACs.
- Unrealizable SACs for the user, so that SACs counteract against the introduction of a product in the market.
- SACs as unreasonable demands for appliers, when the required safe application of a system or product is very expensive, very complex or highly restricted.
- Uncertainties: At what time do SACs arise in the Development Life Cycle? When are SACs necessary? In which documents should SACs be located and verified? How are SACs fulfilled? Who is jointly responsible for the compliancy and its proof?
- SACs that seem to be fulfilled but are not e.g. because they are ambiguous or misinterpreted or the compliance with the SACs or the evidence has been insufficient.

Challenges bring also the different view points and objectives of the different roles involved in the SAC topic and there are role specific thinking pitfalls. E.g. a safety manager may tend to the view that many SACs increase the safety of the item. With this point of view it can easily be overseen that there could be avoidable SACs which make the amount of SACs unmanageable (see explanation of avoidable SACs in section 1). E.g. a product responsible person easily tends to the view point that SACs are unreasonable demands for the clients. Here, the problem could be missing SACs which would be safety critical. A third view e.g. is this of the project which focuses on efforts and costs. It might seem to be more comfortable to define a SAC which has to be solved in the project of the superior item instead of solving the issues within the own project by technical measures. But it has to be considered, also, that often it is easier to solve safety issues in the own project than in the project of the superior item.

These kinds of problems arise if the conditions are not specified in which cases SACs have to be defined and what the quality criteria of the SACs of an item are.

4.2 Demands for Defining and Handling SACs

The problems and difficulties listed in the last section already lead to needs related to defining and handling SACs. The described different objectives of the different roles in projects can be useful for SAC quality, if there are defined rules for SAC formulation. SAC rules for compliancy must also be available. Rules have to be laid down for: Which aspects are SACs and which will not? What are SAC quality criteria? What are the processes of SAC formulation, compliance and demonstration of SAC fulfillment? Who is responsible for what in these processes? How shall SACs and their compliance be documented? When shall SACs be fulfilled? What is important to achieve efficiency? For Thales Rail Signalling Solutions GmbH these demands have lead to the development and introduction of a process instruction which is the basis for this paper.

5 Conditions for Defining SACs

In EN50129 [1] SACs are prescribed between items with separate Safety Cases. But if for a compound system only one safety case is used, then there could be the problem that the safety responsibilities between the items are unclear in detail. For that reason, it is meaningful that the SAC principles are used, this is also true for a compound system using only one Safety Case.

In the following, criteria are listed, stating in which cases SACs must be formulated. Criterion 1 must always be fulfilled together with criterion 2, 3 or 4.

1. Safety risk for non-compliance with an application instruction
A SAC must be created if the reasoning in the safety case or in corresponding documents is dependent upon the compliancy with certain safety rules. If the safety-related argumentation of the safety case requires certain activities of users then these activities will have to be described in SACs. Precondition for a SAC is that the internal attributes of the item are not sufficient for safety argumentation. A SAC should be defined, only if the hazard for which the SAC is a countermeasure for has not already been mitigated by another measure.

A SAC must be formulated if a safety risk occurs as a result of a regulation being ignored by the user, stipulated in a handbook (e.g. Operation Manual or Maintenance Handbook). The evaluation of the risk may result directly from the standards (e.g. demands for channel separation), or the gravity and the frequency of the particular case must be evaluated. In the best case, the degree of risk of the event which requires certain application rules should be examined within the scope of a hazard analysis. SACs should only be generated if the safety aim would fail without it.

2. SACs are reasonable demands for the appliers
SACs often mean that during the application of the considered item, special expenditures or special restrictions are necessary (examples for special expenditures: maintenance expenditures or development expenditures in the project of the superior item; examples for special restrictions: project planning restrictions and operation constraints). If such application expenditures or restrictions are to be avoided, then on the one hand higher development expenditures can be implicated for the own item. For example, there might be application cases which are not required by the customer but which are safety critical and must be excluded by certain SACs (e.g. the use of an interface for a safety related purpose). On the other hand also the benefits have to be considered, which SACs can have in the total Product Life Cycle.

For example, if a generic platform has a watch dog timer for which it is unknown and un-probable that any application will ever use this timer, a reasonable SAC would be: *The safety analysis shall be extended, if the watch dog timer of the hardware is used for safety related functions.* But if it is not expensive to involve this topic also in the generic safety analysis, then the SAC can be avoided.

SACs similar to all other requirements have implications on expenditures concerning realization and proof. Therefore, on the one hand it has to be checked, if a planned SAC is acceptable and reasonable for the user. On the other hand SACs can enable concept and design decisions, which altogether allow an economic development or deployment of a system if the SACs are reasonable for the users.

SACs are only useful if benefits in the whole Product Life Cycle justify the acceptance of special application expenditures and application restrictions. The result could be that SACs must be avoided by changes or extensions of the item. In other cases SACs can avoid extensive analyses in projects of superior items. E.g. a SAC which specifies that an item is not usable for open networks according to EN50159-2, avoids analyses on higher levels if the item is usable for open networks. Such SACs, which are justified in the item concept or design, can be avoided by early planning, about which SACs are necessary, compare with section 6. The requirements and the architecture of the considered item can be changed most easily at an early phase.

3. Eliminating defects in the scope of a project is no longer possible
The formulation of a SAC to bypass defects in the considered item is only acceptable if a change of the item is no longer possible within the scope of the project and an emergency solution (workaround) is reasonable. A precondition is of course that the defect can be adequately bypassed with the issue of a SAC. In this case, an entry in the defect management system is always required. As long as this entry exists, the issued SAC is necessary.

Such SACs can be avoided through careful planning early enough in the project, about which SACs are required, see last but one point in section 6.

4. Acceptance of SACs
If a SAC is addressed to a superior item, in which the requirements specification is already completed, then the SAC has to be placed there in form of a change request, compare with section 7.2. For such change requests, a voting process is required if the SACs in the superior project can still be fulfilled or if it is easier to avoid them in the original project. E.g. there can be the case that a superior item could have already been approved and a new release would be possible, only with very high effort. Consequently, the acceptance of a SAC is a condition of SAC creation.

6 Quality Criteria for SACs

The following items lists and explains criteria for the quality of SACs:

SAC character
A SAC must have SAC eligibility and fulfill criteria listed in section 5.

User addressing
The author of a SAC must always take care to whom the SAC should be addressed to. The phrasing must be correspondingly chosen and the user (according to the listed addressees in section 7.1) must always be explicitly named in the SAC. He must be able to understand and apply the SAC.

Context independent comprehensiveness
A SAC must be able to be understood from the SAC addressees, without the reader having to know the source document from which the SAC has been derived (e.g. a Technical Safety Report). Therefore the SAC must be formulated in such a way to allow the user to understand and fulfill the SAC.

Explanations
Explanations about a SAC are important in addition to the formulation of the SAC:

- *Background information:* Background information is useful for the context independent comprehensiveness. A SAC must be described so that it can be understood without special knowledge of the project in which the SAC was issued, even with or without explanations.
- *Cause and source document:* Even after, e.g. personnel fluctuations, it must be clear why the particular SAC was required. E.g. the safety manager of the next product release must know, what was the cause, origin and what the source document for the SAC is or from which SACs from a subordinate item did the SAC derive from (traceability).
- *Hazard / safety reference:* There must always be a comprehensive safety reference in the SAC. This reference should be clarified through explanatory notes or through a link to the hazard logbook or to the document where the reference is stated. It must be clear what hazard will be avoided through the SAC. Example:
  - *SAC-Formulation:* "If the system is in regular operation the diagnosis device must not be plugged in the diagnosis interface."
  - *Explanation:* „The maintenance staff has to observe that the safe operation is not guaranteed in case of diagnosis. If the diagnosis interface is used, there will be no channel independence of the system which is a basic safety principal of the system."
- *Relation to the defect management system:* For SACs that have been defined because certain product faults could not be corrected, due to hard project constraints (this kind of SACs should be avoided), there must be a reference to the defect management system. It must be clear which defects must be corrected in a consecutive release, so that the SACs will be corrected and thus, made irrelevant.

Feasibility
The demands that are set in the SACs must be able to be realized by the addressed users. The requirements must always be in the responsibility of the addressed user, so that he can fulfill the requirements according to the means available to him and the knowledge that is expected of him. Often, feasibility can be improved, if it is clear for the creator of a SAC, how the SAC can be fulfilled in general. If the SAC creator already records such hints, the expenditures of SAC handling could be reduced.

No overlaps between SACs
Overlaps between SACs must be avoided. This is why it must be checked in the SAC formulation process, if the topic has been covered already through existing SACs.

SAC amount
The amount of SACs is dependent upon the type of considered item. Generic items require typically a larger number of SACs than application specific items. However, it must be made sure that only necessary SACs are defined:

- A large amount of SACs is difficult to manage during the development of superior systems, so that the effort for compliancy proof is too large and difficult to control.
- A large number of SACs has the danger that the important SACs are lost in the bulk and are not taken seriously enough. If SACs are incomprehensive and there

are too many (unnecessary) SACs then nothing will be taken into consideration anymore!
- A large number of SACs delays the entry of the product into the market.

A sensible amount of SACs can be obtained by paying attention to the listed conditions for SAC formulation, listed in section 5. All SACs written according to the criteria mentioned in section 2.2 have SAC justification. SACs as a temporary means (workaround) for product defects should be avoided. The requirement for the limitation on the amount of SACs should not lead to important SACs being omitted and not defined. If the amount is too large however, the project must examine, which SACs could be avoided through improvements to the item. For understandability and for traceability of SACs it can be useful to define several smaller but understandable SACs, rather than having one extensive SAC. The number of SACs increases on the one hand but on the other hand, smaller SACs are easier to be fulfilled.

Earliest possible definition and distribution of SACs
An earliest possible definition of SACs is useful for different reasons:

- *Avoiding SACs:* If SACs are defined already in early development stages of an item it can be decided easier if the SAC can be withdrawn by changes in the concept, the specifications or the design of the item.
- *Complying with SACs in other projects:* Normally, SACs are embedded in the development process of concerned projects by taking them over as safety requirements (compare with section 7.2). Therefore SACs should be recognized already in the requirement phase of the respective project. It is inevitable that in the case of projects running in parallel those SACs or SAC concepts are made known to the other projects as early as possible. Then potential users in the other projects may react easier and quicker. This can also be reached by involving potential users in SAC consolidations (compare with section 7.1). If SACs are forwarded to superior projects after the requirements phase has been finished, then these SACs can be introduced in the respective project only by using change requests.

Compliancy with guidelines for the structure and description of SACs
SACs should be described, named and structured in a uniform manner. This can be set by company guidelines. Also a SAC should be recognized through a uniform layout.

7 Procedures for SAC Formulation and Handling

7.1 SAC Formulation

Fig. 2 gives an overview on possible procedures for SAC creation. To ensure that SACs have safety relevance a SAC draft shall be defined, only if a respective hazard has been defined for which it is not possible or sensible to counteract with item internal measures. The SACs are collected by the safety manager. In the best case a database is used for SAC storage, compare with the benefits explained in section 7.2.

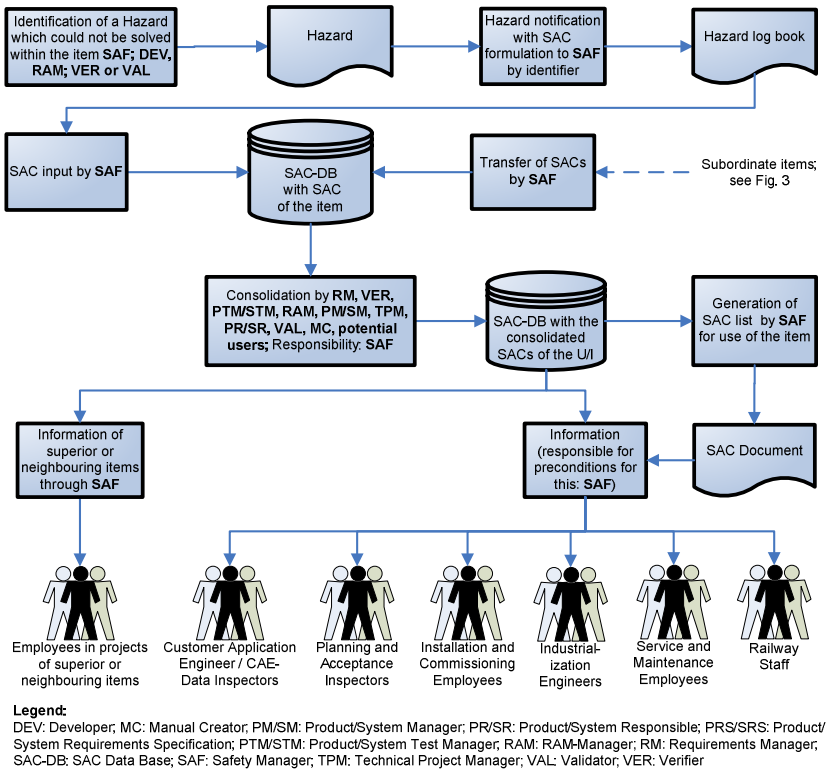

Fig. 2. Process overview for SAC creation

As explained in section 4.1 the consideration of the different role specific views is important for SAC quality. All relevant roles (compare with Fig. 2) must be involved in this process. Known potential users who will have to comply with the SACs e.g. in the development of the superior system should also be involved. They can give important feedback on unambiguity and feasibility. In this way the consolidation of SACs is essential for SAC quality.

At consolidation the fulfillment of the quality criteria introduced in section 6 has to be checked. The consolidation is especially necessary for clarity, feasibility and identification of contradictions and overlaps. Here also the question should be treated if there are possibilities to avoid the SACs by realization of internal measures.

The SACs identification and creation should be done as soon as possible in a project (compare with section 6) e.g. either during development of the Safety Concept or while performing the Preliminary Hazard Analysis. But generally at anytime during a project, SACs are possible, e.g. even during the creation of the validation report.

The product of the described process is the SAC document, compare also with section 7.2. It must be part of the Technical Safety Report according to EN50129 [1]. It can be administered as a separate document and must contain all SACs of the

considered item. As a consequence it is regulated unambiguously, where SACs of an item can be found exclusively.

The first version of the SAC document should be created with the Technical Safety Report because it is strongly related to the argumentation in this document. Furthermore the SACs must be available in a form that they can be transferred into the user handbooks. The end version of the SAC document must be created after completion of the validation report and together with finalization of the Safety Case. Fig. 2 lists also all potential kinds of SAC users who have to be informed about the SACs.

7.2 Compliance with and Evidence of SACs of Subordinated Items

For an item, it must be specified which SACs of other items are relevant and have to be fulfilled. It is obvious to specify this in the Safety Concept, System Concept or in the Preliminary Architecture document.

It can be very time consuming if the SACs in all their origin documents have to be searched for and gathered. If all SACs of one item are listed in its SAC document then it is not necessary to go through all documents where SACs could be specified and there is no uncertainty if all relevant SACs have been found. Another important step is to use a database over the SACs of all items as it is depicted in Fig. 2 and Fig. 3. Then the SACs can be simply queried from the SAC database and time and costs can be saved.

In EN50129 [1], complying with SACs is separated from fulfilling safety requirements. But a separated treatment in projects with different responsibilities for evidence of compliancy leads to additional project efforts. Therefore, if possible, SACs from subordinate items are usually taken over as safety requirements, forming a basis for verifications, being considered in test cases and being treated in the validation, compare with Fig. 3. Consequently, the techniques and measures, according to EN50128 [9] and EN50129 [1], to be used for the compliancy with SACs are the same as for safety requirements. They must always be determined project specifically.

If additional SACs from subordinate items appear after requirements specification has been finished, the safety manager must introduce the SACs to the project as a change request. Then, it has to be discussed, if it is easier to add a safety requirement in the affected project or if it is easier to avoid the SAC by concept or design changes in the source project, compare with section 5.

According to EN50129 [1] the safety management has to describe the relationship to the subordinate safety cases, which is normally in the document “Related Safety Cases”, part 5 of the safety case. In this document the safety management confirms the process of SAC compliance with references to verification and validation reports. Also it has to be judged if all SACs of subordinate items have been fulfilled and proved or forwarded to a further level. SACs that could not be fulfilled within the scope of the own development project, but that are directed to the superior application level (compare with Fig. 2 and Fig. 3) must be passed on. Part 5 of the safety case contains a list of these SACs.

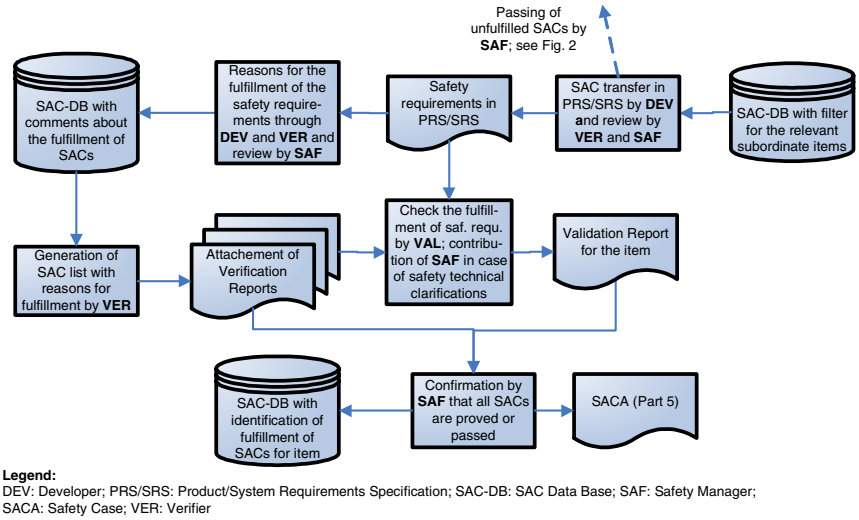

Fig. 3. Process overview about the handling of SACs from subordinate items

The proof must be justified for every SAC which has been identified as fulfilled. This can be achieved through:

- Reference to requirements specification with corresponding safety requirements and argumentations in a verification report.
- If the above point is not possible, a direct argumentative reason and, if required, a reference to safety analysis, to test results or to other documents is possible. A testable SAC must, however, be covered with a test case at all means.
- It is possible that SACs from subordinate items are only valid under certain preconditions or only for a certain context (e.g. customer specific, only valid for certain hardware or only for certain configurations). If these preconditions are not fulfilled, the SAC is not applicable and can be set as fulfilled.
- It can be reasoned, that another solution has been realized than this one which has been required in the SAC. Precondition in this case is that the solution is sufficient to reach the safety targets.

The decisive element for efficiency of SAC compliance and evidence is the quality of the SACs. If time has been invested in the quality of the SACs, then this would have a favorable effect. In addition the systematic cooperation between the roles involved with clear responsibilities is essential for this efficiency.

8 Conclusions

This paper addresses benefits and challenges of SACs. SACs are the necessary means if hazards cannot be avoided by internal attributes of a item itself, but through the adherence of certain regulations during the usage of the item in the intended superior context. We analyzed what are difficulties of formulating SACs, complying with

SACs and providing evidence of the compliance with SACs. The paper describes what is essential for SAC quality and for efficient handling of SACs:

- Rules have been introduced which define what are SACs and also what are not SACs. These should lead to a manageable amount of SACs which are taken seriously for safety. SACs always must have relevance for safety. Whenever a SAC is defined then the relation to a hazard has also to be specified.
- We defined quality criteria for SACs. A good quality of SACs simplifies and supports compliancy with SACs and its evidence. Therefore the SAC quality support to achieve the safety targets.
- We proposed a management process for formulating SACs, compliance with SACs and evidence of fulfilling SACs.
  - The creation of the SACs in early development phases is essential. It gives opportunity to avoid SACs and to react in time on SACs in affected projects.
  - Consolidation is fundamental to check the compliance with the quality criteria.
  - The SAC document and the use of a SAC database are important for a clear SAC storage and management so that efforts can be saved for gathering SACs. Clearly defined processes and responsibilities for SAC creation and handling of SACs on the one hand support the fulfillment of time and budget requirements. On the other hand it is important for safety as the amount of SACs must remain manageable and that the SACs give clear safety instructions.

The result is a process instruction which affects many other processes in the system life cycle and which therefore is complex. To support the handling and the compliance of this process instruction trainings are established.

References

1. CENELEC: Railway applications – Communication, signalling and processing systems – Safety related electronic systems for signalling, EN50129:2003-05-07 (2003)
2. Reuters: Geldstrafen im Transrapid-Prozess verhängt, 2008-05-23 (2008)
3. Dörner, D.: The Logic of Failure: Why Things Go Wrong and What We Can Do To Make Them Right. Metropolitan Books. Henry Holt and Co., New York (1996)
4. Hewison, N.S.: Book Review: The Logic of Failure: Why Things Go Wrong and What We Can Do To Make Them Right. Group Facilitation: A Research and Applications Journal 3, 86–89 (spring 2001)
5. International Electrotechnical Commission: Functional Safety of Electrical/Electronic/ Programmable Electronic Safety Related Systems, IEC 61508. Geneva, Switzerland (2000)
6. Bate, I., Bates, S., Hawkins, R., Kelly, T., McDermid, J.: Safety case architectures to complement a contract-based approach to designing safe systems. In: 21st International System Safety Conference, System Safety Society (2003)
7. Abran, A., Moore, J.W. (eds.): SWEBOK: Guide to the Software Engineering Body of Knowledge. IEEE Computer Society, Los Alamitos (2004)
8. Lauber, R., Göhner, P.: Prozessautomatisierung II. Springer, Heidelberg (1999)
9. CENELEC: Railway applications – Communications, signalling and processing systems – Software for railway control and protection systems, EN50128:2001-05-15 (2001)