A scenario-based safety argumentation for CBTC safety case architecture

C. Liu[1], X. Sha[2], F. Yan[3] & T. Tang[1]
*[1]State Key Laboratory of Rail Traffic Control and Safety,
Beijing Jiaotong University, China
[2]Beijing Traffic Control Technology Co., Ltd, China
[3]School of Electronic and Information Engineering,
Beijing Jiaotong University, China*

Abstract

The Communication based Train Control System (CBTC), as a symbol that China has stepped into the stage of rapid urban rail traffic development, is a safety-critical system that guarantees rail traffic safe-operating and high transportation efficiency. The safety case for the CBTC generic product is an essential justification document to prove the system can be accepted as adequately safe. To extract safety requirements implicitly illuminated within the system requirement specification, operational scenarios are widely used to depict the behaviours and interactions of subsystems and components, which becomes a challenge when constructing safety case architecture from the aspect of system function. This paper presents a promising method based on Goal Structuring Notation (GSN) to establish a composition of safety argumentations for managing safety cases. The method introduces the concept of safety argument modules to express rationally encapsulated goal-based safety claim sets that conform to safety requirements, but are deduced in accordance with hazard analysis based on the operational scenarios. An example generic modular safety case architecture for CBTC generic products is presented to illustrate how the whole safety case architecture is structured to be in line with system requirements, and the ease with which module updates and reuse, according to revises for system development, can be performed.

Keywords: CBTC, GSN, safety case, safety argument module.

WIT Transactions on The Built Environment, Vol 114, © 2010 WIT Press
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/CR100761

1 Introduction

1.1 CBTC generic product

As key equipment deployed in urban rail traffic systems, the CBTC system is comprised of Automatic Train Supervision (ATS), Automatic Train Protection (ATP), Automatic Train Operation (ATO), Computerized Interlocking (CI) system, and Data Communication System (DCS), and is conducted to guarantee safe operation and improve the traffic capacity of stations and sections, as well as realize automatic railway traffic control and high transportation efficiency.

The ATP system is the core of the CBTC system, which dominantly serves to guarantee safe operation. The ATP system consists of the Vehicle On-Board Controller (VOBC) and the Zone Controller (ZC), see Fig 1. The VOBC measures and sends location information to the ZC periodically via both trackside Access Points and waveguides. Combining train location with line occupancy supplied by the CI, Database Storage Unit (DSU) and ATS, as well as other trackside equipment, the ZC calculates movement authority for a specified train and sends information back to the VOBC in the same way, with which the VOBC generates service brake and emergency brake profiles to supervise the train movement. The DCS includes a redundant wired backbone network and wireless communication between on-board devices and trackside equipments, both of which can provide protocol-independent data transmission for the functional application.

As a safety-critical system, the CBTC generic product should be certified to meet the requirements in railway standards regarding safety related applications, e.g., the EN5012X series. For specific functional domains, diverse standards are adopted to achieve design targets, for example the LCF-300 CBTC product developed by BJTU, MIL-STD-882C, is applied for semiconductor component

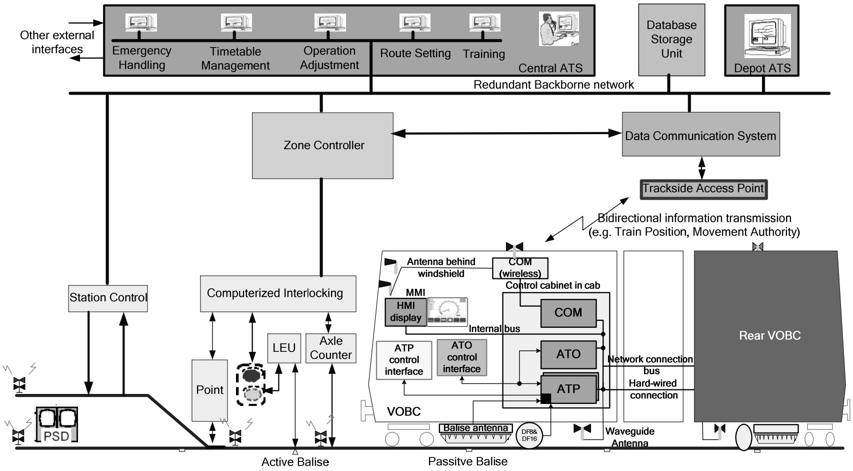

Figure 1: Configuration of CBTC generic products.

design, EN50129-2 and IEEE1474.1-1999 are referred for wireless communication application, etc. However, this paper focuses on CBTC product safety case development, which mainly consults with CENELEC standards, namely EN50126, -8, and -9 [1–3]; other norms are outside the scope of this paper.

1.2 Safety argument in a safety case

The production of a safety case is an essential part of the safety assessment process for safety-critical system development. The gist is to communicate a clear, comprehensive and defensible argument that a system is acceptably safe to operate in a particular context (Kelly and Weaver [5]).The safety case consists of three principal elements: Requirements, Argument and Evidence, which are composed to convince someone that the system is safe enough (when compared against some definition or notion of tolerable risk). According to the review of some conventional context based safety cases, a common flaw exists, which is that the role of the safety argument is neglected and, instead, many pages of supporting evidence are often presented (e.g. hundreds of pages of fault trees or FMECA tables), but little is done to explain how this evidence relates to the safety objectives. Safety arguments aiming to communicate the reasoning relationship between requirements and evidence are often suggested to be expressed in well-structured texts; such arguments can be efficient to be understood by the involved developers of the safety case, but can be ambiguous and unclear to other engineers who are not familiar with the author's literary manner. Besides, cross-references are necessarily introduced to argue integrity of evidences, however, multiple cross-references in text can be awkward and can disrupt the flow of the main argument. Without a clear and shared understanding of the argument, safety case management is often an inefficient and ill-defined activity.

This paper will introduce a structured technique, Goal Structuring Notation (GSN), to provide an explicit representation of the concepts required to create an argument and to represent the argument inferences linking the requirements to the evidence.

1.3 Incremental safety case for railway applications

In order to obtain safety approval for a generic product, safety case need to well organize the overall documentary evidences to be submitted. Historically, the production of safety cases has often been viewed as an activity to be completed to the end of the safety lifecycle. To initiate safety case development at the earliest possible stage and arrange phrasal evidences incrementally collected in step with system development, a common approach to managing the gradual development of the safety case is to submit a safety case at various stages of project development. For instance, the U.K. MoD Defence Standard 00-55 [7] talks of formally issuing at least three versions of the Safety Case:

- Preliminary Safety Case – after definition and review of the system requirements specification

- Interim Safety Case – after initial system design and preliminary validation activities
- Operational Safety Case – just prior to in-service use, including complete evidence of having satisfied the systems requirements

The EN50129 [3] also recognizes the importance of recording the relationship between partial safety cases and overall safety cases that a section of the recommended safety case structure is reserved for this purpose. As EN50129 talks of safety cases being structure into six parts:

- Part One – Definition of the System
- Part Two – Quality Management Report
- Part Three – Safety Management Report
- Part Four – Technical Safety Report
- Part Five – Related Safety Cases
- Part Six – Conclusions

Part Five of the safety case acts a dual role. Firstly, it should be used to record references to the safety cases of any subsystems or equipment on which the main safety case depends. Secondly, it could be used to present an account of the evidence of satisfying safety conditions from other safety cases, which could embrace those partial safety case carried forward into the bases of the main Safety Case.

This paper will emphasize the role operational scenarios play during the incremental safety case development, and a method upon the scenarios of establishing the traceability between phrasal safety case and main safety case

2 Goal structure notation

The Goal Structuring Notation (GSN) (Kelly and Weaver [5]) – a graphical argumentation notation - 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, see Fig 2.

The principal purpose of a goal structure is to show how goals are broken down into sub-goals, and eventually supported by evidence (solutions) whilst making clear the strategies adopted (e.g. adopting a quantitative or qualitative approach),the rationale for the approach (assumptions, justifications) and the context in which goals are stated (e.g. the system scope or the assumed operational role).

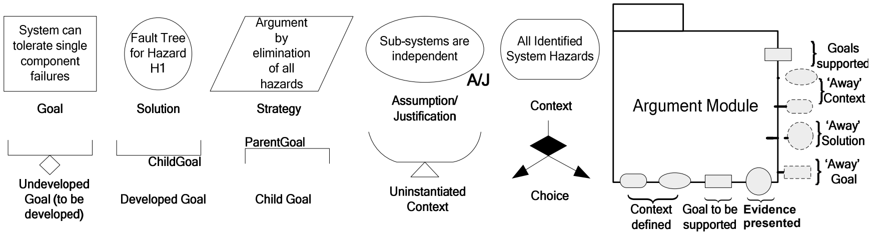

Figure 2: Principle elements in GSN.

An extension to GSN is an explicit representation of modules themselves. This is required to be able to represent a module as providing the solution for a goal. In order to refer to goals defined within other modules, the concept of ‘away element’ is introduced (e.g. away goal or away solution), which derives a vital feature of modular GSN: argument module interfaces. The argument module interfaces define clearly the visible contents of a argument module including the Objective addressed by the module, public objectives and evidences that support to (or from) other argument modules, assumed context defined within the module together with any dependencies on other cases. Interfaces are specified to provide other argument module developers with sufficient information to allow them use a particular argument module.

3 Safety case architecture

Following the definition of software architecture (Bass *et al.* [8]), Kelly [9] presents a similar terms of Safety case architecture: ‘The high level organisation of the safety case into components of arguments and evidence, the externally visible properties of these components, and the interdependencies that exist between them’. This definition declares equal importance to the dependencies between safety case modules (or ‘components’) as to the components themselves, which means for the incremental safety case development during safety lifecycle, an kind of structures that is able to establish clear and seamless interfaces so that safety case elements can be safely composed, removed and replaced, should be considered from the very beginning stage of constructing safety case.

3.1 High level argumentation

Constructing a safety case architecture for CBTC from high-level requirements during the system development lifecycle allows the low-level requirements for evidence to be identified. Thus the need for testing, analysis and other evidence generation approaches can be determined during system design.

EN 50129 [3] supports the principles of establishing multiple related safety cases in stating a safety case provides evidence that a generic product is safe in a variety of applications. However, an attempt to enumerate and justify all possible configurations is unfeasibly expensive; to establish the safety case for a specific configuration will nullifies the benefit of flexibility (Kelly [10]). A more promising approach is to attempt to establish a modular, compositional safety case that has a correspondence with the modular structure of the underlying architecture (Kelly and McDermid [11]). However, it is more significant that what aspects of system architecture can be classified as basis of partitioning argument modules.

Whilst conceiving a complex safety critical system, designers are prone to scheming safety functionalities that system should achieve rather than constructing system structure, because the structure is just a specific solution of all function requirements. Besides, to discuss how one function relies on another and their interface requirements is more practical than make clear the boundaries between subsystem structures just after system requirement has been defined. Hence, constructing modular safety cases in accordance with system requirements will be easy to operate and make the potential modular change minimized. In addition, this style has one advantage over the subsystem decomposition style in that it promises to be more cohesive from a safety perspective.

3.2 Preliminary safety case

After specifying the system requirements of CBTC generic product, the Preliminary Hazard Analysis (PHA) can be undertaken in order to identify hazards related to design and operation and ensure that the preliminary design is built-in with safety properties from the beginning of the CBTC system development. Consequently, High level argumentation in Preliminary Safety Case covers the functional requirements, as well as the specification of all external interfaces, performance requirements, Electromagnetic Compatibility (EMC) requirements, and Reliability, Availability, Maintainability and Safety (RAMS) requirements, all of which form a framework that safety case architecture has to conform to, see Fig 3.

Next, operational scenarios will introduced to deal with functional division cutting across subsystem boundaries, also help to collect safety goals supported by other modules according to the reference relationships indicated in Fig 3.

4 Operational scenarios and hazard analysis

As panoptic view of functional design, operational scenarios aim to reveal detailed schemes which are constructed by the system designers to fulfil specific

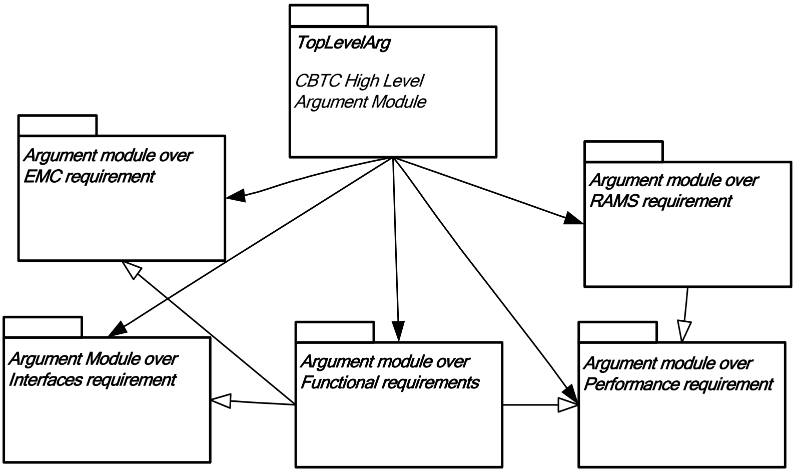
TopLevelArg
CBTC High Level Argument Module
Argument module over EMC requirement
Argument module over RAMS requirement
Argument Module over Interfaces requirement
Argument module over Functional requirements
Argument module over Performance requirement

Figure 3: Modules in high level argumentation.

functionalities, also provide legible process charts for assessors to follow when identifying latent sub-system hazards. In order to recognize the potential causes and consequences for each identified hazards, system safety analysts can find clues referring to the pre-and post-conditions of each step, interactions between sub-systems during single-step execution, as well as the input and output data of components which can awake potential chain-reacting fault states in future interactions.

Fig 4 shows the operational scenario conceived to implement when train starts up in the depot then departs to operate on the mainline. After system requirements has defined, it is more feasible in reason for the designers to decompose function requirements other than deploy subsystem or component, because it is hard to assign the specific function points to corresponding physical divisions especially when correlativity between primary functions has not been clearly discussed yet. Here operational scenarios offer such materials for both

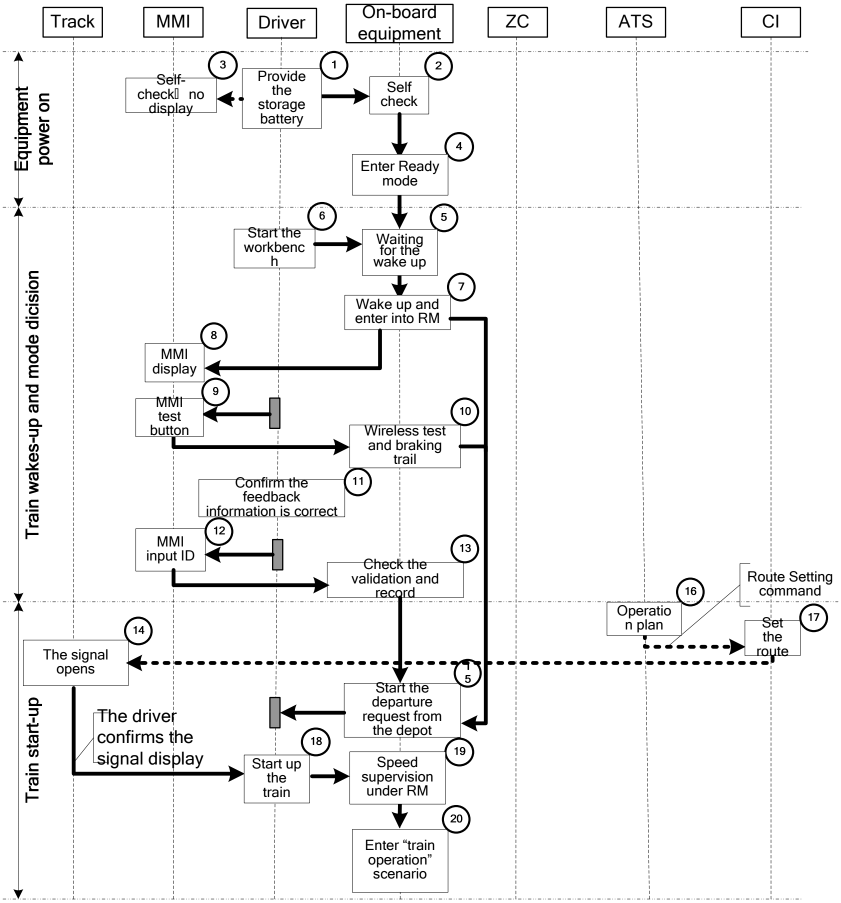
Track
MMI
Driver
On-board equipment
ZC
ATS
CI
Equipment power on
Train wakes-up and mode dicision
Train start-up
1 Provide the storage battery
2 Self check
3 Self-check no display
4 Enter Ready mode
5 Waiting for the wake up
6 Start the workbench
7 Wake up and enter into RM
8 MMI display
9 MMI test button
10 Wireless test and braking trail
11 Confirm the feedback information is correct
12 MMI input ID
13 Check the validation and record
14 The signal opens
15 Start the departure request from the depot
16 Operation plan
17 Set the route
Route Setting command
18 Start up the train
19 Speed supervision under RM
20 Enter "train operation" scenario
The driver confirms the signal display

Figure 4: Scenario of the train start-up process.

designers and assessors to talk through a particular process: firstly, system designers illustrate their mentalities on the designated function point via each behaviour (the rectangle with circular on the top right corner) on the thread of subsystem objects which they consider as possibly contribution to Hazard log of Train start-up operational Scenario implementation. Secondly, the implementers can negotiate with designers about the function boundaries as criterion to follow with when they later define the subsystem. Most important, all participants will reach an agreement on detailed design, which is of great benefit in case of necessary modification even on a single function point.

For example, the assessors can easily trace the related behaviour sequence and evaluate the side effect on the identified hazards, consequently, decide whether new evidences are needed for this change in relevant phrasal safety cases.

Take the scenario in Fig 4 for example, which elaborates the three stages of train power-on, wake-up and start-up. For each behaviour the assessor will use HAZOP method to question the designers in form of 'Object (direct or indirect) +guideword (no, more or less, etc) +parameter (velocity or voltage or data, etc)'. Designers will follow these questions to investigate the potential causes and consequences in case it happened as a hazard. To complete hazard log, designers need to propose mitigation measures on the purpose of bringing down the risk to a tolerable level, which are essential to form the safety goals in the argumentation. Table 1 gives a fragment of Hazard Log of train start-up operational Scenario, as space is limited, only the reference number and potential causes are presented to explain how the hazards are identified from scenarios.

5 Safety goals decomposition based on scenarios

As has already been discussed, the mitigation measures against each hazard can be treated as sub safety goals under a top safety function representing the safety requirement corresponded with the operational scenario. Before one measure is taken into account of decomposed safety goals, some reduction strategies below will be adopted to avoid unnecessarily duplicated argument work:

- Combine the hazards with same potential causes, which inevitably means identical mitigation measures;
- For the similar measures in different scenarios, if the same supported evidences are needed, can be argued as away goal;
- Eliminate as agilely as possible the human factor hazards from technical safety argument into safety management argument, which will be of great benefit to function argument reuse, as not under all possible scenarios the same human faults happen.
- If one identified hazard serves to be the potential cause of another hazard, then relevant measure could be the sub safe goal of the upper goals derived from that hazard.

With these strategies, the measures of all hazards in Table 1 has been simplified into sub goals which finally construct the whole argumentation under the top goal 'Train safely leaves the depot', see Fig 5. Inside this argument, the

Table 1: Hazard log of train start-up operational scenario.

Scen-Func Ref. No.	Hazard Description	Potential Cause
S1-F1\2	Train cannot supply the power to the on-board equipments	1.Driver skills were sufficient 2.Storage battery was depleted/not regularly maintained/float charged.
S1-F3\4	On-board equipment failed to self-check or check overtime	1. On-board equipment design deficiencies; 2.VOBC functionally failed;
S1-F6\7	On-board equipment failed to wake-up, or overtime.	1.Internal communication failed; 2.Drivers did not choose the head of train;
S1-F8	MMI cannot display train-borne information when VOBC powers on	1.MMI powered down; 2.Communication between train-borne and MMI failed
S1-F9\10	On-board equipment failed to detect rear on-board equipments.	1.Communication between ends of vehicle failed; 2.failed to collect data of rear of vehicle when changing the ends
S1-F11	On-board equipment incorrectly passed braking testing.	1. Train-borne collecting board /collecting channels failed; 2. Drivers considered the wrong feedback information is as the normal;
S1-F12	Driver failed to input IDs.	1. On-board equipment failed to query drivers to input ID; 2.Communication between train-borne and MMI fails; 3.Drivers make human errors.
S1-F13	On-board equipment did not check the validity of drivers' IDs	train-borne software fails
S1-F15	On-board equipment failed to enter corrective mode status which is selected	1. Mode switches failed. 2. Drivers make mistakes; 3. Mode switch is incorrectly wired during building process.
S1-F14	On-board equipment cannot link to wireless	1.No wireless signals, DCS fails; 2.Train-borne equipments fail, cannot receive wireless signals; 3. ZC equipments fail, cannot receive wireless signals.
S1-F15	Signal may not be open yet when train left depot.	1.ATS did not arrange the operation plan or arrange a wrong operation plan; 2. CI did not arrange routes or arrange wrong routes due to failures; 3.Communication between CI and signal failed.

top safety goal is marked as public goal, as it will be representatively cited to support other argument modules. G2~G11 are sub goals decomposed with the strategy of ‘all hazards have been handled’, namely the safety requirements separated out from mitigation measures using the strategies mentioned above, which can be referred in the context of Hazard Log. For those safety goals need to be supported by specific evidences are presented as undeveloped goals which will be finished in operational safety case. Particularly, G4 is related to the RAMS performance of CBTC system, has to gain testimony from argument module over RAMS requirement, consequently, it is expressed as an away goal.

In order to establish the traceability, an incidence matrix including the relationship of safety requirements, safety functions in scenarios, hazard Log and requirement specification is necessary not only for safety argument, but also for the safety case reuse and maintenance.

As the safety case architecture was built on operational scenarios, the incidence matrix, also called verification matrix, is designed referring to the operational scenarios likewise. As a fragment of such matrix listed in Table 2, one record of a sub goal needs to contain the full information during its period of validity, that is, how it is generated, what it affects, and where it is stated. In case that change happened, e.g. designer have to modify his thought, or implementer have to update the definition of system boundary, the operational scenarios bear the brunt to recompose synchronously. So it is quite vital to recognize the range of influences for a single safety goal as well as functional interaction that this goal will take with other safety goals. To combat this, verification matrix is created to ensure that function interactions are recorded and considered as a separate ‘interactions’ sub safety case, which is obviously less comprehensible but easier to maintain. For someone wishing to investigate all of the possible issues surrounding the maintenance of a particular safety goal, they will find them largely addressed within such a single sub safety case.

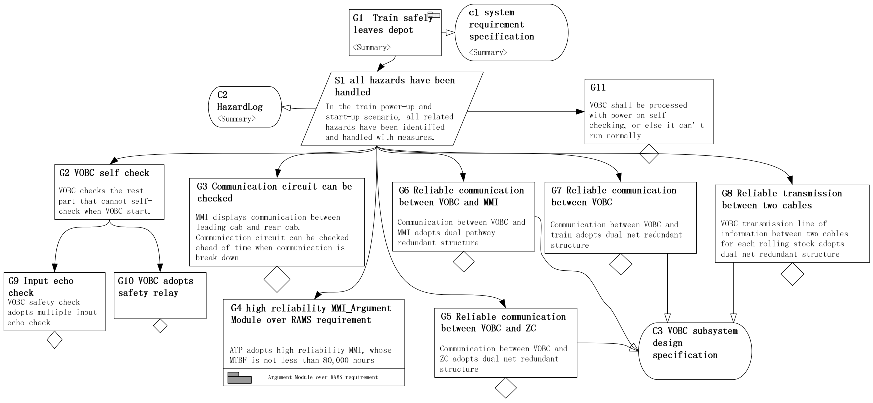
G1 Train safely leaves depot
<Summary>
c1 system requirement specification
<Summary>
S1 all hazards have been handled
In the train power-up and start-up scenario, all related hazards have been identified and handled with measures.
C2 HazardLog
<Summary>
G11
VOBC shall be processed with power-on self-checking, or else it can’ t run normally
G2 VOBC self check
VOBC checks the rest part that cannot self-check when VOBC start.
G3 Communication circuit can be checked
MMI displays communication between leading cab and rear cab. Communication circuit can be checked ahead of time when communication is break down
G6 Reliable communication between VOBC and MMI
Communication between VOBC and MMI adopts dual pathway redundant structure
G7 Reliable communication between VOBC
Communication between VOBC and train adopts dual net redundant structure
G8 Reliable transmission between two cables
VOBC transmission line of information between two cables for each rolling stock adopts dual net redundant structure
G9 Input echo check
VOBC safety check adopts multiple input echo check
G10 VOBC adopts safety relay
G4 high reliability MMI_Argument Module over RAMS requirement
ATP adopts high reliability MMI, whose MTBF is not less than 80,000 hours
Argument Module over RAMS requirement
G5 Reliable communication between VOBC and ZC
Communication between VOBC and ZC adopts dual net redundant structure
C3 VOBC subsystem design specification

Figure 5: Train start-up safety function argumentation.

Table 2: Traceability between sub safety goals and migrating measures in the verification matrix.

Sub safety goals	Scen-Func Ref. No.	Source	Name of Requirements	No. Requirements in specification
G2	S1-F1\2\3\4	HL-029[1]	VOBC Drive Module Design Specification	3.2drvSystemSelfTest description 3.6drvSelfTestResult description
G3	S1-F6\7\8	HL-031[2] HL-154[2]	VOBC Subsystem Requirement Specification	3.3.4-Communication status check among subsystems 9.1.1Information display function
G4	S1-F8	HL-032[1] HL-053[3]	HW User Manual-INC-70.xx hardware manual	7.2Electrical specification
G5	S1-F14	HL-098[2] HL-136[2] HL-149[3] HL-192[3]	ZC subsystem architecture specification	3.2.2.2-Redundancy design principle
G6	S1-F6\7\8	HL-032 [2] HL-053 [4]	VOBC subsystem architecture specification	3.1-Subsystem division 5.2.2-Logical Interface between ATP and MMI

6 Conclusion

Rather than organizing the safety case architecture in accordance with the existing system structure, another style is to decompose the case according to safety functions. This style has one advantage over the subsystem decomposition style in that it promises to be more cohesive from a safety perspective. This paper constructs a modular safety case architecture following the system requirements, then introduces operational scenarios as skeleton to guide the safety goal decomposition, and records safety argumentation in function-independent modules with GSN method. With such method, the dependences between argument modules can be explicitly expressed in module interfaces and be directly traced in verification matrix, which will obviously bring the potential benefits of changeability and reusability compared to a monolithic safety case. In future work, we intent to use the extended GSN concept of safety contract to record such traceable cross-references between argument modules to preferably help manage the dependences.

Acknowledgements

The bulk of the work reported here was supported by the project of Natural Science Foundation of China (NSFC): the basic theory and key technology research of train operation control and organizations (Ref.60634010).

We are grateful to Beijing Subway Yizhuang Line Project and Department of safety and quality assurance of Beijing Traffic Control Technology Co., Ltd for the information and explanations which formed the basis of our case study.

References

[1] *EN 50126 Railway Applications - the Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS) - Part 1: Basic requirements and generic process.* European Committee for Electrotechnical Standardisation, 1999.

[2] *EN 50128 Railway Applications – Software for railway control and protection systems.* European Committee for Electrotechnical Standardisation, 2001.

[3] *EN 50129 Railway Applications – Safety related electronic systems for signalling*. European Committee for Electrotechnical Standardisation, 2003.

[4] Railtrack: *the yellow book: Engineering Safety Management Volume 1 and 2:Fundamentals and Guidance Issue 4,* Rail Safety and Standards Board,2007

[5] Kelly, T., Weaver, R. The Goal Structuring Notation – A Safety Argument Notation. *Proc. of Dependable Systems and Networks 2004 Workshop on Assurance Cases,*2004

[6] *MoD Defence Standard 00-56 Safety Management Requirements for Defence Systems,* Ministry of Defence.1996

[7] *MoD Defence Standard 00-55, Requirements of Safety Related Software in Defence Equipment,* Ministry of Defence.1997

[8] Bass, L., Clements, P. and Kazman, R. *Software Architecture in Practice,*Addison-Wesley,1998

[9] Kelly, T. Using Software Architecture Techniques to Support the Modular Certification of Safety-Critical Systems. *Proc. Eleventh Australian Workshop on Safety-Related Programmable Systems (SCS 2006),* Melbourne, Australia. CRPIT, 69. Cant, T., Ed. ACS. pp53-65, 2006.

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

[11] Kelly, T.P., McDermid, J.A., A Systematic Approach to Safety Case Maintenance, *Reliability Engineering and System Safety* vol. 71, Elsevier, pp271-284,2001.