A model-based framework for the safety analysis of computer-based railway signalling systems

R. Niu & T. Tang
State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiao Tong University, China

Abstract

Ensuring safety in railway signalling systems is always considered as significant as a guarantee of the safe and efficient operation of the whole railway. In fact, safety analysis of the signalling system with distributed computer technique is becoming extraordinarily difficult, because of the frequent and complex interaction between components and the various backup modes. The dominant approaches are subjective, difficult to reuse and not well structured, thus leaving the safety analysis process time-consuming and error-prone. This paper develops a hierarchical methodology for safety analysis based on the failure propagation model and state-transition model. Unlike traditional safety analyses, the proposed approach demonstrates more accurate representation of practical failure behaviour in a computer-based signalling system. Dynamic properties, system structure and failures at the component level are separately modelled in different layers, and connected with synthesis laws. The analysis can be easily refined as the system design progresses and automatically produces safety-related information to help the engineer in making design decisions. The preliminary design of the Communication Based Train Control (CBTC) system for the Yizhuang Line in Beijing is used to demonstrate this approach.
Keywords: signalling system, automatic safety analysis, model-based, FPTN.

1 Introduction

Railway systems have a very low tolerance for accidents, because of the potentially large numbers of injuries and deaths, huge financial losses and even worse social effects. Achieving a high degree of safety is one of the most

WIT Transactions on The Built Environment, Vol 114, © 2010 WIT Press
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/CR100751

important objectives of a railway signalling system. While advanced information techniques have been widely used in new generation signalling systems, safety analysis becomes a genuine challenge. Due to the development of automation, networking and to the general increase of train speed, the number of interacting components or subsystems has increased drastically over recent decades. Transplanting the redundant structure and degrade-recovery technique into a digital system makes the signalling system even more complex (Leveson [1]). It is not sufficient to comprehend the system in its minute details just depending on intuition and experience. What is worse, as the functions are much stronger and the techniques are totally changed, the availability of safety data for the new computer-based signalling systems, such as accident or incident statistics, is limited (Vernez and Vuille [2]).

To cope with the increasing complexity of signalling systems, CENELEC, IEC and many countries have developed several standards and recommendations. These standards regulate the system development process (lifecycle) of signalling systems to design for safety, and also give out technical requirements, such as SIL. Traditional techniques are recommended in the safety assessment process, including HAZOP, FTA, FMEA/FMECA, etc. These specific inductive or deductive methods of analysis are used to identify hazard, trace causation and evaluate their risk at different stages of the lifecycle, and the results are the main basis for design decisions. This methodology has been used by most railway equipment suppliers over the last 20 years, although they obviously lag behind the state-of-the- art engineering practice.

These dominant applied approaches commonly rely on expert opinion. The analysis models explain accidents in terms of multiple events connected by causality relationship. The methods just give out a very simple rule (tree structure or tables) for the description of relationship. There is no limitation for the category of events, and they could be some type of component failure, human error, or energy-related event. However, the selection of these events, the links between the events and even the point of beginning and ending is arbitrary (Khan and Abbasi [3]). In order to reduce the subjectivity, more experts with different academic backgrounds are involved and the results need to be reviewed at least once, which obviously make the safety analysis time-consuming and mentally intensive. Furthermore, the simple rules of most classic safety analysis are not well structured. The forward or backward reasoning is carried out with regard to the hierarchy of failure influences rather than to the architecture of the system (Vaidhyanathan and Venkatasubramanian [4]). So at each stage, if the design of the system has changed, many analyses need to start from the very beginning. Moreover, there are major defects in most traditional safety analysis techniques, so different techniques are chosen at different stages of the lifecycle, and two or more techniques are usually employed at one stage to make up the defects of each other. However, as there is no unifying framework for these techniques, it is very difficult to relate the results of the various safety studies to each other and back to the high level failure analysis.

In the past ten years, many researchers have devoted themselves to the solution to these problems of traditional safety analysis with model-based

approaches [5–10]. They intend to build precise models for the system architecture and its failure modes, so that computers can help to do the tedious and error-prone hazard sources tracing and probability calculation. One solution of model based safety analysis is extending the system development model with a fault mode. Formal languages are used to describe normal and failure behaviours of the system, and model checking tools or simulation engines are used to do automatic analysis. Some commercial safety analysis software tools/packages based on this idea are available, such as FSAP/NuSMV-SA [5] and SCADE [6]. However, the major portion of this kind of model is still a normal process, rather than a failure process. It is very difficult to plug in detail failure information because of the limitation of model scale from analysis tools. Another solution is to model the failure propagation behaviour directly. The Failure Propagation and Transformation Notation (FPTN) described in [7, 8] is the first component-based failure behaviour model. Kaiser [9] introduced modular concepts for a basic fault tree to analyze complex component-based systems. Based on early researches, Papadopoulos et al. [10] proposed a model-based semi-automatic safety and reliability analysis technique that uses tabular failure annotations as the basic building block of analysis at the component level, called Hierarchically Performed Hazard Origin and Propagation Studies (HiP-HOPS). This tool can automatically synthesise the component failure modes and generate a fault tree. However, the model does not work well in describing the dynamic behaviour of system.

The present study proposes an improved failure propagation approach for the safety analysis of a computer based rail signalling system. In order to describe the complex structure and function, the study has developed an output-guided hazard identification method with a scenario hazard table to ensure the correctness of system understanding and the completeness of hazard identification. A kind of simplified state machine model is used to express the dynamic properties of signalling system structure. The study has also developed an iterative algorithm to combine the dynamic model with FPTN components and compute qualitative results automatically.

The rest of the paper is organized as follows: Section 2 is a description of the dynamics of a computer-based signalling system. Section 3 introduces the hierarchical dynamic safety analysis framework, including methodology hypothesis, definitions of each layer, and the synthesis algorithms of different layers. The case study of a CBTC system in Section 4 demonstrates the application of this approach. The conclusion is drawn in Section 5.

2 Dynamics of computer-based signalling systems

Computer-based signalling systems generally adopt a distributed structure, including a trackside control centre and onboard vital computer systems, which are connected with a wireless communication network. The trackside equipments collect the parameters of trains within a certain area and related information from other trackside systems (such as ATS, interlocking) to compute a safe unoccupied region for each train. The onboard computer systems are responsible

for keeping train speed within the upper limit computed with the safe region from the trackside and train parameters from the onboard computer. The European Train Control System (ETCS) and Communication Based Train Control (CBTC) system applied in urban mass transit are the representative computer-based signalling systems.

Traditionally, the logic relations of different scenarios are expressed by the combination of the trackside discrete electromechanical components, while the function of each signalling system remains unchanged. In computer-based signalling systems, trackside equipments are cut down, and their functions are integrated into onboard computers. In this way, computers should provide different functions and work with different interfaces under different operation scenarios. This kind of system is called a phased-mission system (Alam and Al-Saggaf [11]), which means that the mission served by the system composes of several distinct phases with different objectives (the phased-mission characteristic is called behavioural dynamics). In each mission phase, the system has different service objectives, and therefore the safety constrains may change from time to time, which make the safety analysis error-prone. For example, safety engineers often make the mistake of generally treating the measured value of train distance as greater than the actual value that is safe. In fact, when a train is moving out of a station or a speed-limit section, see fig. 1, a greater measured value of distance will make the calculated permitted speed larger than the real one, which might cause a derailment or train rollover. Not only the structure of the signalling system, but also the function of the onboard computer is different when the operation level or mode changes.

Additionally, some safety measures inherited from the electromechanical system increase the dynamics of the signalling system. In order to apply the powerful and undependable computer technique into a safety critical signalling system, redundant structures are used in almost all of the kernel trackside and onboard processors. Moreover, the control mechanisms and even the whole architectures are designed to be redundant, which are represented in the form of backup modes and system levels. For example, the CBTC system used in the Beijing Yizhuang Line defines three operation levels for the whole system and

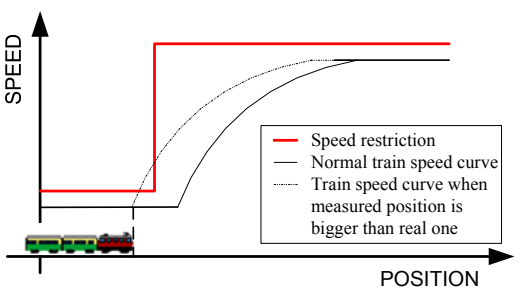

Figure 1: Speed curves when a train is moving out of a speed restriction region.

three operation modes for the onboard system. Therefore, the structure of this subsystem will be changed with time, in case any replications are down.

3 Model-based dynamic safety analysis framework

3.1 Framework for the safety analysis of computer-based signalling systems

The construction of the hierarchical structure approach is shown in Fig. 3. Hierarchical modelling is used in our framework, as it fits in well with the system design process and reduces the complexity of system analysis. The system is successively split into subsystems until the level of the basic components is reached following a top-down approach. This kind of approach has been successfully used in recent studies proposed by other authors, such as the successive modelling approach used in HHM to address large hierarchical systems [12], and the MFM approach used in the Safe-SADT method [13].

The block at the top in Fig. 2 represents the operation scenarios of the system, which should be defined at the beginning of its lifecycle. For each scenario, the states definition and state transition of the system/subsystem can be described by the state-transition model. For each state, the safe critical functions can be decided and refined by FPTN models, and it becomes more and more specific when moving down along the system structure. The safety analysis process can be divided into the dynamic layer and the failure propagation layer. The dynamic layer, used to structure and describe the dynamic attributes, is combined with the scenario lists and the state transition models. The failure propagation layer is expressed by FPTN language.

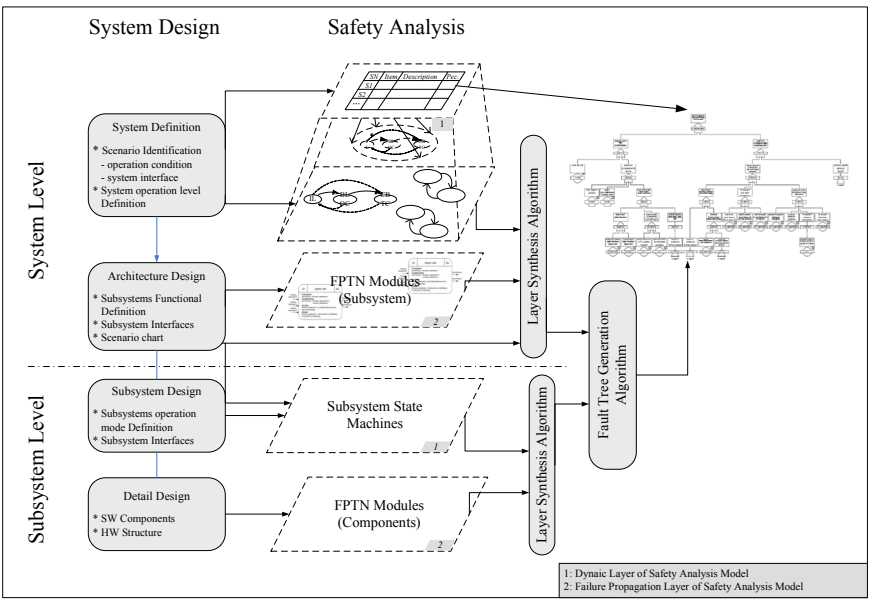

Figure 2: Framework of hierarchical safety analysis.

Ref.	Scenario	Operation Mode	System Structure	β	Function Set	Output	Hazards
SN01	Draw up at a station	CBTC-AM	VOBC,ZC, DCS	0.12	Door open interlocking	DoorO SDoorO	DoorO:c SDoorO:c
…	…			…	…	…	…

Figure 3: Scenario hazard table.

3.1.1 Output-guided hazard identification

Just like all other safety analysis methods, hazard identification is the first procedure in our safety analysis framework. Unlike general automatic control systems, traditional hazard identification methods do not work quite so well for computer-based railway signalling systems. Firstly, as computers are widely used nowadays in signalling systems, most vital functions are processed together by computers and the critical information translation between the trackside and onboard computers becomes much more dangerous. Secondly, the computer-based signalling system is large scaled and its control logic and interactions between components are very complex. Traditional brain-storming methods, such as HAZOP, apparently cannot ensure the correctness and completeness of hazard identification. Fortunately, in railway systems, the signalling system does not control the train directly. Instead, it detects the working conditions of the train, and gives out safe guidance or performs emergency action when necessary. In the other words, the safety of trains is dependent on the correct and prompt output of its signalling system. Therefore, in our safety analysis framework, hazards of the signalling system are defined as abnormalities of system output.

In our output-guided hazard identification process, it is necessary to identify the abnormal condition of system output in each system state and each operation scenario, because the output of the system and the safe range of the output value vary with the scenarios and system states. Information is recorded in the table shown in fig. 3. Factor β is used to synthesis hazard events under different scenarios in quantitative analysis. This procedure, although a little tedious, makes it much easier to find out the unexpected system output when the system working conditions are specified. The completeness of hazard identification can be ensured as all operation scenarios are analyzed. The synthesized failure propagation models can decide whether a hazard will or will not be in the hazard list. However, if there is change in the operation scenarios or the system state models, the hazard list needs to be regenerated.

3.1.2 State-transition model

Fig. 4 illustrates the five primitive elements of the simplified state machine notation. In this model, dynamic behaviour is expressed as a set of different states of the system (operation mode and system level) and a set of transitions between those states (the mode change condition). State transitions occur for two reasons: either the state changes are induced by some other events, or are triggered by the state change in other mode-chart. The mechanism enables a transition in one mode-chart to trigger other transitions at higher or lower layers of the dynamic model, which allows us to represent situations where failures of

sub-systems may lead to losses of function at system level. It also allows us to represent situations where a change of function at system level should be followed by a number of necessary functional or structural transformations at lower levels.

3.1.3 FPTN

FPTN (Failure Propagation Transformation Notation) modules describe how failure modes of incoming messages, together with internal faults of the components, propagate to failure modes of outgoing messages. The basic entity of the FPTN is a FPTN-module. This FPTN-module contains a set of standardized sections. In the first section (the header section), for each FPTN-module an identifier (ID), a name and a criticality level (SIL – Safety Integrity Level) is given. The second section specifies the propagation of failures, the transformation of failures, the generation of internal failures and the detection of failures in the component. These failures are denoted as incoming and outgoing of the FPTN-module. This paper gives out a modified failure categorization for

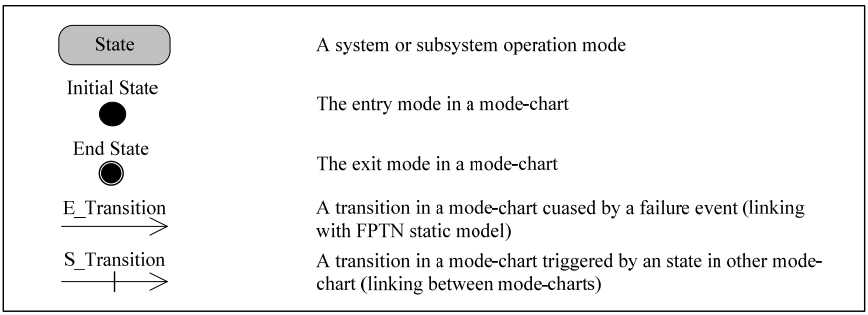

Figure 4: Notation of the state-transition model.

Table 1: Failure class definition.

Categories	Failure Class	Sign	Explanation
Provision Failure	Commission	c	Unexpected output
	Omission	o	No output
Value Failure	High	h	The value is higher or bigger than the normal range
	Low	l	The value is lower or smaller than the normal range.
	Stuck	s	The value is stuck to a certain number.
Time Failure	Delay	d	Later than intended.
	Early	e	Earlier than intended.
Communication Failure	Insertion	is	Wrong message destination
	Masquerade	ms	Wrong message source.
	Corruption	cr	The data is error with uncertain tendency.
	Repetition	rp	Message is send more than once.
	Resequence	rs	The sequence of message is changed.
	Deletion	dl	Message is lost.
Handle limit	Limit	limit	Limits of deviation handler.

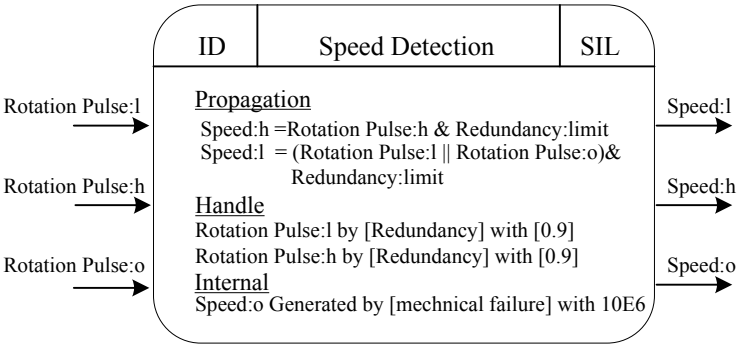

Figure 5: A simplified FPTN-module of the train speed detection component.

computer based railway signalling systems, in order to include the seven kinds of threats (deletion, repetition, resequence, delay, corruption, insertion, masquerade) brought by the general network, see table 1.

Fig. 5 provides an example of a FPTN-module of the train speed detection component. The incoming failures are Rotation Pulse:l, Rotation Pulse:h and Rotation Pulse:o, and the outgoing failures are Speed:l, Speed:h and Speed:o. The propagation and transformation of failures is specified inside the module with a set of equations or predicates (e.g. for propagation: Speed:h=Rotation Pulse:h and for transformation Speed:l=Rotation Pulse:l || Rotation Pulse:h). Furthermore, a component can also generate a failure (e.g. Speed:o) or handle an existing failure (e.g. Rotation Pulse:l and Rotation Pulse:h). Consequently, it is necessary to specify a failure cause or a failure handling mechanism and a probability.

3.2 Safety analysis process

In order to analyze the cause of each hazard, this study designs an algorithm for automatic fault tree generation. Firstly, the layer synthesis algorithm is used to integrate the FPTN-modules under different modes. Then, a kind of depth first search algorithm is used to draw a fault tree for each hazard.

3.2.1 Layer synthesis algorithm

1. Scenario synthesis

The hazard events in different scenarios are generally separated by time and space, which means they occur in different times and different places. In fact, it is not necessary to synthesize these scenarios in qualitative analysis. In quantities analysis, the probability of a hazard event appearing in several scenarios can be calculated by the weighted summing-up of the number of each scenario with factor β of the scenario hazard table as the weight coefficient.

2. Mode synthesis

The state-transition model and FPTN-modules are synthesized with the algorithm shown in fig. 6. The E_Transition of the state are added to the FPTN-

```
SynthesisStateLayer (csro, cnode) {    //layer synthesiser
    cstatesList= FindStates (csrc, cnode); //Traverse the model and find the states
                                          //which relates the output failure (cnode)
    For each element in cstatesList      //For each element in cstates set
        If  element is not (the initial state)
                                          //If element is not the initial state of the state transition model
            Loop
                ctrans= FindInputTrans (element);
                If the input transition of the element
                        is not (a s_transit)
                                          //If the input transition of the element is not a s_transit
                 AddEConditionList (stran);
                                          //Add the input transition into EConditonList
                 Break;
                 Else  element= ctrans;
                                          //Otherwise continue to find the input transition of the state
    For each event in EConditionList
        ANDFailureCause (event);   //Each event in EConditonList are added  into
                                   //FPTN-module as input failure, and AND the equation of cnode
    }
```

Figure 6: Layer synthesis algorithm.

modules as the input failures and the cause of the output failures is the E_Transition AND original Boolean expression. S_Transitions of the state connect this model with other state-transition models. Find out the E_Transition of the state indicated by the S_Transition, and run the above steps again.

3.2.2 Fault tree generation algorithm

The synthesis algorithm translates the system (or sub-system) failures to component failures, and translates the failure propagation formula of the FPTN module to the Fault Tree. When a sub-system is encountered during the traversal of the hierarchical model, the causes of its output failure are always traced first at the sub-ordinate hierarchical level of the design, which describes the architecture of the sub-system. A simplified pseudo-code representation of the proposed fault tree synthesis algorithm is presented in Fig. 7.

4 Case study

The Yizhuang line of Beijing is composed of a large number of equipments and highly interactive subsystems of various natures (see Fig. 9) (electro-mechanical, electrical, infrastructure, hard-/software, electromagnetic) and locations (tracks elements, control centre, embarked systems), most of which are still under development. The signalling system of the Beijing Yizhuang Line employs the CBTC system design by the Beijing Jiaotong University. The system consists of a Vehicle On-Board Controller (VOBC), Zone Controller (ZC) and Data Communication System (DCS). The DCS includes a wired backbone network and wireless communication between on-board devices and trackside equipments. The DCS transmits data packages in a manner transparent to the application. Secure Devices (SD) are installed as the safe guard between the safety critical part (e.g. ATP) and the non-safety related part (DCS) of the CBTC.

```
SynthesisFPTN(sys, op){                          //Recursive FPTN synthesiser
      module = FindFPTNModule(sys, op);          //Travers the modules within sys and find
                                                 //the module with output deviation op.
     PropagatonToFaultTree(module, op);          //Transform the propagation bool formula of op to Fault Tree
     If leafnode is not (a handler limit) or (a internal deviation) or (a deviation of   system input)
              SynthesisFPTN(sys, leafnode);      //If the leafnode is not a basic event then call the recursive FPTN
                                                 //synthesiser
}


FaultTreeGeneration (scenario, failure){
      system = Findstructure(scenario);     //Travers the scenario hazard table and find the
                                            //system module array of  the scenario
      SynthesisFPTN(system, failure );      //Call the recursive FPTN synthesiser
}
```

Figure 7: Fault tree generation algorithm.

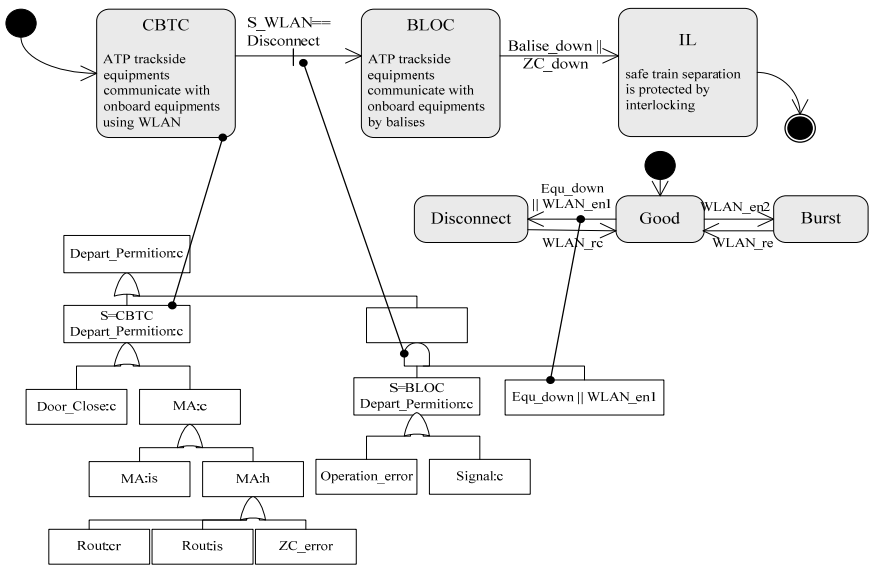

Figure 8: Analysis results of the "draw up at a station" scenario.

4.1 System modelling

The first step in the safety analysis is to identify operation scenarios of a particular application and elaborate a scenario hazard table of the system. The table helps to identify the system functions and interfaces in each working condition. Now 11 scenarios are identified for the whole CBTC system operation process and 21 system level safety related functions, including 15 functions for ensuring traffic safety and 6 functions to protect passengers. The deviations of the system output treated as the hazard events will be used as the top event of the fault tree (see description in Section 3.2.1).

The CBTC system of the Beijing Yizhuang Line defines three operation levels for the whole system and three operation modes for the onboard system. The system levels are divided into the CBTC Level (ATP trackside equipments communicate with onboard equipments using WLAN), the BLOCK Level (ATP trackside equipments communicate with onboard equipments by balises) and the IL Level (onboard equipments cannot be controlled by ATP trackside equipments, safe train separation is protected by interlocking). The operation modes of onboard system are RM (Restricted Manual) mode, CM (Controlled Manual) mode and EUM mode (i.e. Bypass mode). The state-transition model of the "Draw up at a station" scenario is shown in Fig. 9 as an example.

Starting from the top function for which the system is designed (''trains follow successively their optimal route"), the system is successively broken down into sub functions, individual elements/components, and then the FPTN-modules can be elaborated by analyzing the failure propagation/transformation behaviour of each module. These modules are connected by component interfaces.

4.2 Results

For each potential threat, the output deviation of safety related system functions in every scenario, we have tracked down the causes and evaluated the corresponding occurrence probability. The results are expressed as Boolean expressions of component failures as a column of the Hazard Log, and also can be shown as fault tree figures to make them easier to understand. Thirty seven FPTN-modules were built and 183 hazard events have been identified, which is obviously too large to lay out in a single piece. Therefore, this paper only shows the results of the "Draw up at a station" scenario as a demonstration.

5 Conclusion and future work

This study has addressed some of the pitfalls pointed out in the literature (lack of system overview, conflicting objectives) and offers some solutions to overcome some of the difficulties. In this study, a hierarchical framework, based on the Failure Propagation Transformation Notation (FPTN), has been developed to perform safety analysis and risk management of large and complex computer based railway signalling systems. This approach is based on the data flow among components rather than the hardware description of the system, which enables failure behaviour modelling in various stages of the design lifecycle.

Some further notation developments for FPTN are still needed in this direction to allow for a better expression of the time properties of failure events. Enriching FPTN with Temporal Logic is part of our current research. The Temporal Logic should cover all kinds of sequential relations of failures, and should not make the model too complex to solve. Another interesting research is the more accurate description of the deviation of continuous data. These continuous data are usually affected by several different factors. How to express the influence of each factor in the failure propagation model and how to decide the synthetic variation tendency are the problems that need to be solved urgently.

Acknowledgements

This paper is sponsored by the National Natural Science Foundation of the P. R. China under grant No.60634010, with the title "The Theory and Key Technology Research of Train Control System", and is also supported by the Urban Rail Transit Automation and Control Beijing Municipal Government Key Laboratory.

References

[1] Leveson, N. G., A New Accident Model for Engineering Safer Systems, *Safety Science*, vol. 42, pp. 237-270, 2004.
[2] Vernez, D. and Vuille, F., Method to assess and optimise dependability of complex macro-systems: Application to a railway signalling system, *Safety Science*, vol. 47, pp. 382-394, 2009.
[3] Khan, F. I., Abbasi, S. A., TOPHAZOP: A knowledge-based software tool for conducting HAZOP in a rapid, efficient yet inexpensive manner, *Journal of Loss Prevention in the Process Industries*, vol. 10, pp. 333–343, 1997.
[4] Vaidhyanathan, R., Venkatasubramanian, V., Diagraph-based models for automated HAZOP analysis, *Reliability Engineering and System Safety*, vol. 50, pp. 33–49. 1995.
[5] Bozzano, M., Cavallo, A., Cifaldi, M., et al, Improving Safety Assessment of Complex Systems : An Industrial Case Study, *Proc. of the Formal Methods 2003*, Vol. 2805, Springer-Verlag, pp. 208-222, 2003.
[6] Abdulla1, P. Deneux1, A., et al, Designing Safe, Reliable Systems Using Scade, *Leveraging Applications of Formal Methods*. vol. 4313 of LNCS, Springer-Verlag, pp.115-129, 2006.
[7] Fenelon P., McDermid J., et al, Towards integrated safety analysis and design, *ACM Computing Reviews*, pp. 21-32, 1994.
[8] Fenelon P., McDermid J., An integrated toolset for software safety analysis. *Journal of Systems and Software*, 21(3):279–290, 1993.
[9] Kaiser, B., Extending the Expressive Power of Fault Trees, *Proc. of the 51st Annual Reliability & Maintainability Symposium (RAMS05),* 2005.
[10] Papadopoulos, Y., Mcdermid, J., et al, Analysis and Synthesis of the Behaviour of Complex Systems in Conditions of Failure. *Reliability Engineering & System Safety*. Vol 71, pp. 229-247. 2001.
[11] Alam, M., Al-Saggaf, U. M., Quantitative Reliability Evaluation of Repairable Phased-Mission Systems Using the Markov Approach, *IEEE Transactions on Reliability*, R-35:498-503, 1986.
[12] Bozzano, M. & Villafiorita, A., Integrating Fault Tree Analysis with Event Ordering Information, *Proc. of the ESREL 2003*, pp. 247-254, 2003.
[13] Kaiser, B., Liggesmeyer, P., Mackel, O., A New component Concept for Fault Trees, *Proc. of the 8th Australian Workshop on Safety Critical Systems and Software (SCS'03)*, Adelaide, 2003.