INTEGRATED FORMAL METHODS FOR SAFETY ANALYSIS OF TRAIN SYSTEMS

Wolfgang Reif, Frank Ortmeier, Andreas Thums, and Gerhard Schellhorn

Chair of Software Engineering, Department of Computer Science, University of Augsburg, Universitätsstr 14, 86135 Augsburg, Germany

E–Mail: {reif, ortmeier}@informatik.uni-augsburg.de

Abstract In many technical applications the notion of system safety covers different aspects. The first is correct functioning. This means that the system does what it is supposed to do. The second is an analysis of failures and their effects. This answers the question what happens, if components break or fail. And finally a quantitative analysis, which quantifies the risk of the system and its subsystems.

In all three dimension formal methods may help. Temporal logics and formal verification assure functional correctness. Formal safety analysis techniques give rigorous proof of cause-consequence relationships, finally statistical models and mathematical optimization help to minimize risk and give advice for design decisions.

In this paper we sketch an example and illustrate how such an integrated approach can be done and what benefits it provides.

Keywords: safety analysis, formal methods, fault tree analysis, quantitative optimization

1. Introduction

New technologies are emerging in every aspect of railways. Let it be electronic scheduling and speed control, new propulsion techniques like in the German ICE 3, where the engines are distributed among all cars of the whole train or autonomous, intelligent equipment on the track. This all results in great benefits like faster travel, less energy consumption and improved maintainability. But there is also a price to pay for it. More and more software has to be integrated into the control units, control is increasingly decentralized and the complexity of track items like track switches, signals or level crossings rises dramatically.

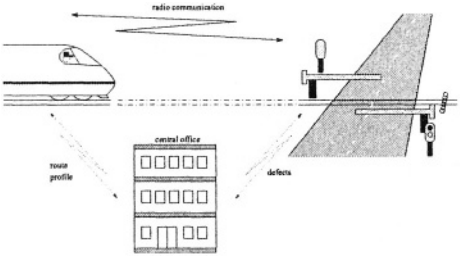

Figure 1. Overview over the Radio-Based Crossing

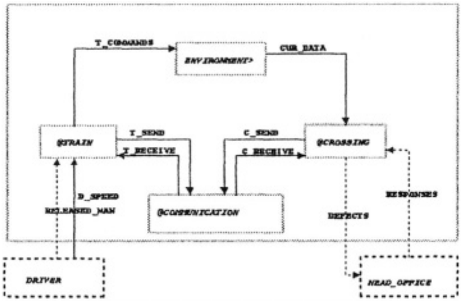

Figure 2. Involved subsystems for the Radio-Based Crossing

This results in an increased risk of failure. For example the U.S. Department of Transportation lists in its 2003 report on transportation [6] more fatalities involving railroads than any other form of transportation besides cars.

The most advanced techniques with respect to safety guarantees are formal methods. Systems are described as mathematical models and safety predictions can be analyzed and rigorously verified. In the following we present the ForMoSA approach for safety-critical systems [9]. It combines formal methods and safety analysis techniques covering different aspects of safety.

2. An Example

In the following example is taken from the German railway organization, Deutsche Bahn, which prepares a novel technique to control level crossings: the decentralized, radio-based level crossing control [7]. This technique aims at medium speed routes, i.e. routes with maximum speed of 160 km/h. The main difference between this technology and the traditional control of level crossings is, that there is no central control unit. Signals and sensors on the route are connected by radio communication. Software computations in the train and in the level crossing decide if the crossing can be passed safely or if an emergency stop must be triggered.

To achieve this, the train computes the position where it has to send a signal to secure the level crossing. Therefore the train has to know the position of the level crossing, the time needed to secure the level crossing, and its current speed and position, which is measured by an odometer. When the level crossing receives this command it switches on the traffic lights - first the 'yellow' light, then the 'red' light - and finally closes the barriers. When they are closed, the level crossing is 'safe' for a certain period of time. After this time the crossing opens the barriers again automatically.

The train requests the status of the level crossing, before reaching the latest point for a safety stop. Depending on the answer the train will brake or pass the crossing. The level crossing periodically performs self-diagnosis and automatically informs the central office about defects and problems. This solution is cheaper and more flexible than a centralized control, but also shifts safety critical functionality towards all involved components.

This example involves a number of complex tasks like calculations of speed, expected time of arrival at the danger zone (DZ) of the level crossing and latest point in time for a safety stop in the train, specific control sequences for the crossing to activate the lights and the bars with respect to different timing constraints and - between crossing and train - a radio based communication protocol. All these components must interact correctly to assure safety.

The main safety relevant questions, that must be answered, are:

- "Is this complex interplay correctly designed, such that the system always closes the barriers before a train passes?" (functional correctness)
- "How failure tolerant is the design? What happens if e.g. the radio communication is out of order" (failure tolerance)
- "What is an upper bound for failure? Is there a better design with less risk?" (risk calibration)

3. Modeling and Verification

First the radio-based level crossing is modeled in the formalism of state charts (of Statemate). This notation has a formal semantics [4] and is very similar to common engineering notations. Figure 2 shows the top level of the model of the example. The system is decomposed into the four parts train, crossing, communication and environment. These parts are again modeled using state charts and activity charts.

Mathematically the system model describes a set of traces over which the safety properties can be expressed in temporal logic. To express safety properties we use interval temporal logic (ITL) [3]. This logic allows to state many safety properties in a very intuitive way. The central safety property in the example - a train must not pass the crossing while the barriers are not closed - is stated in ITL as follows:

$$\Box\,(Train = DZ \rightarrow Barriers = closed)$$

This formula says: "It is always the case $(\Box)$, that a passing train (Train=DZ) implies $(\rightarrow)$, that the barriers are closed (Barriers=closed)." The next step is to verify or falsify the property. Depending on the system we use an interactive theorem prover - the KIV system [2] - or a suitable model checker.

In the example the first proof attempts failed and revealed a safety gap. The gap is, that the original system design had some flaws involving different timers, responsible for reopening the barriers after the train has passed the crossing. This problem could be fixed, and then the safety property could successfully be verified.

4. Formal Safety Analysis

Verification showed, that the complex interaction between all components works correctly. This means, there are no inadvertent flaws in the design of the system. The next step is to analyze, what happens if for example the brakes fail or the barriers get stuck. Cause-consequence techniques like fault tree analysis (FTA) [5] or failure modes and effects analysis (FMEA) try to answer this problem.

FTA stepwise reduces hazards to primary events that may cause the hazard. For example an unsafe crossing may be due to either a failure of the brakes **or** the barriers getting stuck.

Originally FTA and FMEA operate on informal models. In ForMoSA both techniques have been formalized [10]. FTA gates resp. the connection of the FMEA columns are expressed in interval temporal logic. Then it is possible to use the same reasoning techniques as before to verify the correctness of the analysis. So if the fault tree has the form, that a collision may only occur if either the brakes failed or the barriers got stuck, this could be translated into the (ITL) formula:

$$\Box\ (Collision \rightarrow Brakes = failure \vee Barriers = stuck)$$

If this formula is proven correct, then a universal theorem - the minimal cut set theorem - ensures, that no other causes for a potential collision have been forgotten (at least with respect to the mathematical system model).

5. Quantitative Analysis

In practical application no safety analysis is complete without a quantitative assessment on the risk. Most traditional safety analysis techniques incorporate a quantitative part as well [11]. This part may easily be adopted for formal safety analysis.

But the quantitative analysis may also be augmented. For example traditionally quantitative FTA is simply done by using static probabilities for all leaves of the fault tree. In many applications, this often not good enough. Systems frequently have free parameters, which have various effects on the system. In the example, allowed speed of the train or run-times of the timers are such parameters. This makes it necessary to give up the requirement of static probabilities but rather use continuous distribution functions. This allows a far more elaborate analysis, as the effect of free parameters on the system may be examined. It is even possible to find optimal configurations, that minimize risk [8]. Another possibility to use formal methods to improve quantitative analysis is the use of probabilistic model checkers [1]. This approach allows to analyze the difficult problem of common cause failures. But it requires more elaborate system models as input.

6. Summary

Railways is not only traditionally but also nowadays highly safety-critical. This makes an increase in R&D effort for safety necessary. Increasing complexity and more and more software in all parts of railways systems result in very difficult safety analysis tasks. This problem may be addressed using formal methods. They are very well suited to deal with software/hardware interaction. Formal methods are used for safety critical applications in three flavors. Verification is used to assure correct functioning. Formal safety analysis gives answers about qualitative safety-critical, cause-consequence relationships. Mathematical and statistical models help in finding upper bounds for risk and to calibrate antagonistic safety requirements.

Altogether, the approach helps to achieve a very high level of safety with acceptable effort.

References

[1] Christel Baier, Edmund M. Clarke, Vassili Hartonas-Garmhausen, Marta Z. Kwiatkowska, and Mark Ryan. Symbolic model checking for probabilistic processes. In *Automata, Languages and Programming,* pages 430–440, 1997.

[2] M. Balser, W. Reif, G. Schellhorn, K. Stenzel, and A. Thums. Formal system development with KIV. In T. Maibaum, editor, *Fundamental Approaches to Software Engineering,* number 1783 in LNCS, pages 363–366. Springer-Verlag, 2000.

[3] A. Cau, B. Moszkowski, and H. Zedan. *ITL – Interval Temporal Logic.* Software Technology Research Laboratory, SER Centre, De Montfort University, The Gateway, Leicester LE1 9BH, UK, 2002. www.cms.dmu.ac.uk/˷cau/itlhomepage.

[4] W. Damm, B. Josko, H. Hungar, and A. Pnueli. A compositional real-time semantics of STATEMATE designs. In W.-P. de Roever, H. Langmaack, and A. Pnueli, editors, *COMPOS'97,* volume 1536 of *LNCS,* pages 186–238. Springer-Verlag, 1998.

[5] J. Fragole J. Minarik II J. Railsback Dr. W. Vesley, Dr. Joanne Dugan. *Fault Tree Handbook with Aerospace Applications.* NASA Office of Safety and Mission Assurance, NASA Headquarters, Washington DC 20546, August 2002.

[6] Marsha Fenn, editor. *Transportation Statistics Annual Report.* U.S. Department of Transportation, Bureau of Transportation Statistics, October 2003.

[7] J. Klose and A. Thums. The STATEMATE reference model of the reference case study 'Verkehrsleittechnik'. Technical Report 2002-01, Universitat Augsburg, 2002.

[8] F. Ortmeier and W. Reif. Safety optimization: A combination of fault tree analysis and optimization techniques. Technical Report 5, Institut für Informatik, Universität Augsburg, 2004.

[9] F. Ortmeier and A. Thums. Formale Methoden und Sicherheitsanalyse. Technical Report 15, Universität Augsburg, 2002. (in German).

[10] G. Schellhorn, A. Thums, and W. Reif. Formal fault tree semantics. In *Proceedings of The Sixth World Conference on Integrated Design & Process Technology,* Pasadena, CA, 2002.

[11] W. E. Vesely, F. F. Goldberg, N. H. Roberts, and D. F. Haasl. *Fault Tree Handbook.* Washington, D.C., 1981. NUREG-0492.