

Experimental validation of train control systems by using a railway model

S. Einer, H. Schrom, R. Slovák, E. Schnieder
*Institute of control and automation engineering,
Technical University Braunschweig, Germany.*

Abstract

New trends in train control systems are determined by an increasing use of software. The established high standards in reliability of such safety relevant systems leads therefore to high requirements in software reliability. However, today's methods in software-engineering are not sufficient to avoid software-failures completely, if excessive tests are not performed. Testing the software in real applications of train control is mostly to large-scaled and also not flexible enough in providing certain operational conditions which are required for testing. Otherwise using only a software environment in testing does not allow realistic results in respect to the implementation. To bridge the gap, testing by using a realistic hardware model could be a golden means for solving these problems efficiently. To evaluate this possibility, a modular model railway has been realised at the Technical University Braunschweig in Germany achieving the purpose of software-validation in train control applications. The operation served by the model railway is the safety process of train passing a radio controlled level crossing. To test the software which realise this operation, the needed behaviour has to be specified exactly. Furthermore an effective concept is needed to meet all requirements in testing, especially the separation of environmental and essential aspects respective the separation in non-realistic and realistic aspects of the railway model. Following the identified tasks, the contribution contains three main parts. At first the specification of the operational process of the specific train control system is given. Secondly a functional concept of hardware railway model for software-validation is introduced and last practical solutions to fit this concept are shown by introducing the state of realisation of the railway model.

1 Introduction

The possibilities of software usage increases the demand for complex functionality in technical systems. However realising complex systems increases also the possibilities of undesired system behaviour. In safety relevant systems each case of the undesired behaviour has to be worked out before implementing the system. The result has to be used in the development of supervision and safety functions, which increases the system functionality again.

The software requirements have to be derived from the functional requirements. To be sure that the software code meets this requirements some formal methods in software verification can be applied. But even if the software is verified, there is still the question, whether the software requirements are sufficient enough to realise a desired and safe operational behaviour in each operational situation. Today there is no possibility to verify this fact, it has to be rather validated by tests.

The problem is, how to test a complex system in a complex operational process. The operational process is a hybrid process consisting of technical and human parts. Supervision- and safety-processes are necessary because of several facts: The obvious one is that as well humans as technical components can fail. Another fact is, that there is an uncertainty as well within human actions as within the essential technical process. Therefore undesired situations may occur also in faultless operation. These situations must not lead to dangerous situations and have to be detected by the supervision and if necessary they have to be corrected by the safety procedures. A third reason for considering safety processes is, that the essential process includes inherent dangerous operations, which become safe, if they are expanded by the safety processes. In this case, the safety-process is also essential.

The control system receives information about the state of the operational process by sensors and influences it by actuators, but because of permitted uncertainty of the operational process, the control system does not determine the operational process completely. Therefore from the view of the control systems the operational process belongs to the system environment, which determines the system input and leads to a special output. For validating the control software, it is necessary to specify all possible classes of scenarios, which could happen in operation and to specify how the control system has to influence them. That means test case specification is necessary.

In order to test under realistic conditions, it is necessary to test on implementation level. But testing the software in real applications of train control is mostly too large-scaled and also not flexible enough in providing certain operational conditions which are required for testing. In other domains, e.g. chemical process technology, an additional step of development, also called "scaling up" is widely accepted to handle this problem. It means to experiment in low scaled processes, before starting the real operation. Such experiments seems to be necessary also in the system engineering, but here experiences are being mostly missed [1].

Within the scope of the priority programme "Integration of Software Specification Techniques for Applications in Engineering" [2] launched by the DFG (central autonomous (selfgoverned) academic institution in Germany serving all fields of science and the humanities by financing research projects), it has been decided to build a physical railway model for validating the research results. This task covers the task of low scale but realistic testing which are considered above.

This paper shows the regular operational process of the system to be realised and point out the failures in operation which should be considered in testing. Further the functional concept of the railway model control is given. The last section describes the essential aspects of the state in realisation of the railway model.

2 The operational process of the level crossing case study

The operational process of level crossing case study has been published in [3]. Here it is shortly discussed in relation to the general idea of operational processes in engineering systems. The system is a decentralized level crossing which control process bases on radio communication between the train and the track. The target behaviour of the level crossing protection can be roughly described as follows (Figure 1):

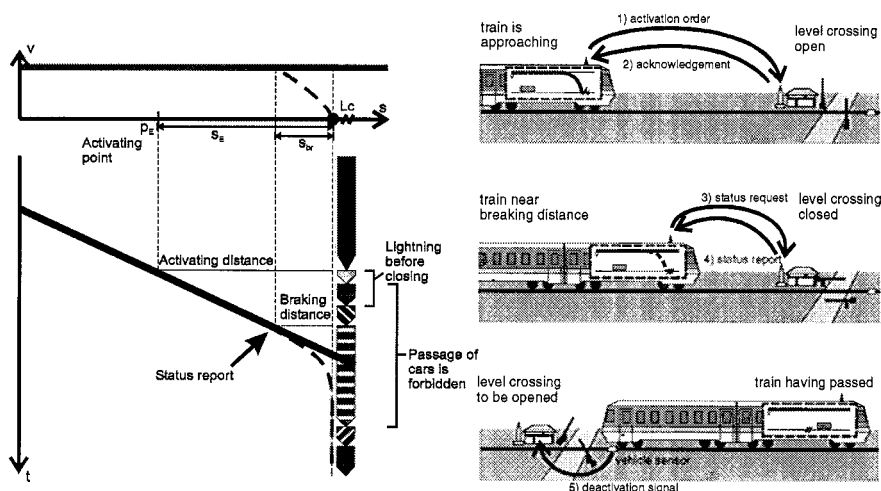


Figure 1: The regular scenario of the operational railway process

A train approaching a level crossing switches on the trackside control unit by radio communication. Consequently the level crossing control first turns on the traffic lights for the road traffic to "yellow" and later to "red". After a defined

time the lowering of the barriers will be started. When the barriers have reached its down position, the level crossing is in the safe state. In a sufficient distance from the danger area, the train interrogates the status of the level crossing. When the train gets the statement that the level crossing is safe, it is allowed to pass the level crossing. After leaving the level crossing, the train activates a vehicle sensor and induces the annulment of the protection.

The described process is the regular one and it includes especially the safety process to prevent collisions between trains and the road traffic. It is an essential safety process, because the level crossing must always be in the safe state when the train is crossing it. Requesting the status is a supervision process. It is needed because a dangerous situation may occur in consequence of faults or especial operational situations. Technical faults, which have to be considered in the operational process are shown in Figure 2. An example for an undesired situation in faultless operation is, that for operational reason the train can not reach the level crossing as fast as usually, even if there is no technical fault. Then after a certain time the level crossing has to be supposed as unsafe. The reason is that car drivers could try to pass the level crossing even if it is closed, supposing no train is approaching, if the waiting time is too long. In all cases, if the train approaches the level crossing and it gets the status report "unsafe" or no status report at all, a safety process has to stop the train automatically. Furthermore operational procedures must be defined to guarantee the train passage even if some safety processes have been applied. The example shows the complexity of the operational process. This leads to a lot of test cases which have to be specified for the validation of the control code.

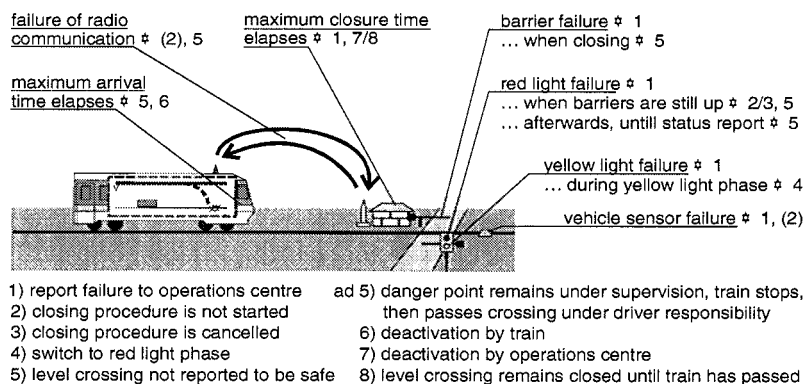


Figure 2: Technical faults to consider within the operational process

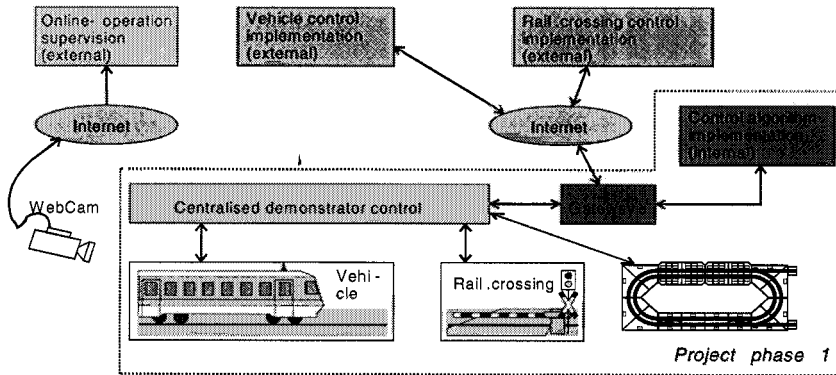


Figure 3: Demonstrator in overview

3 Functional concept of the railway model

The Figure 3 gives a rough overview about the railway demonstrator. Within the first project phase the physical railway model installation consisting of a vehicle and a radio controlled level crossing and the centralised demonstrator control have to be realised. The flexible architecture of the demonstrator model should allow an internal implementation of the formal control algorithms model provided by the external project partners.

In order to meet all at the beginning mentioned requirements a functional structure of the demonstrator control was derived (Figure 4). It consists of three main functional groups. Besides the external control algorithm model, developed according to the case study specification, these are the basic functions of the demonstrator. In the next sections a more detailed description to each demonstrator function block will be given.

The main task of the experiment control is to prepare, configure, start and stop the experiment as well as data recording and processing and finally to visualise the results obtained. Also a suitable mode of the environment control has to be chosen.

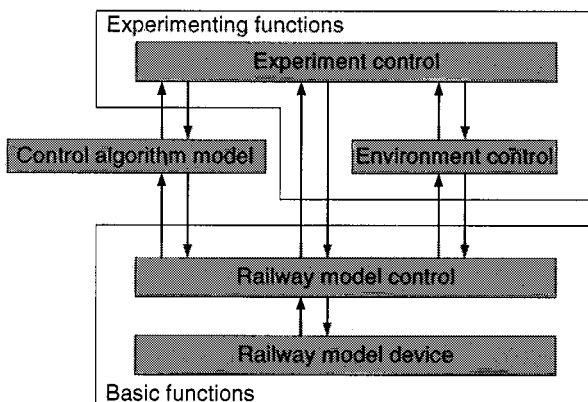


Figure 4: Functional structure of the demonstrator

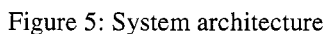
The environment control has to simulate the possible influence of the environment. On the one hand these are the inputs resulting from potential train driver behaviour and from possible actions initiated by the operation inspector supervising the functionality of the level crossing. On the other hand the hazardous occurrence of failures has to be simulated. Failures of yellow or red traffic light (to be regarded separately), barriers, the vehicle sensor and the delay or loss of telegrams on the radio network are considered. The required mode of environment control has to be selected before starting an experiment. Concretely a manual environment control using the command desk or the control computer can be replaced by a concrete scenario or by a hazardous environment influence.

The railway model control stands for a direct interface to the railway model device. Its task is to bring together all control commands from higher functional layers, to test their admissibility and to adapt them to the model device requirements. On the other way the actual operating states of the model device components are to be acquired and transmitted in a suitable form to the experiment or environment control as well as to the external control algorithm model.

4 Implementation of the railway model

4.1 Train equipment and communication

The railway model is made of three trains of the type “Cargo Sprinter” on a ten meters round double track in the scale of 1:22,5 (IIIm).



The realisation of the train-borne equipment can be seen in Figure 6. On the chassis of a "Cargo Sprinter" the electronic boards are mounted. On the bottom the motor module with the plugged in micro-controller is placed. Above, the communication module with its two antennas of the DECT devices and another plugged in micro-controller is located. On the top a pentium PC together with a hard disk and the PC power supply is put on. The power for the operation of all units is supplied from the metal rail track. This compact unit is therefore a nearly autonomous train vehicle.

4.2 Level crossing equipment

The relevant track-side equipment is shown in Figure 7. The railroad crossing module consists of two barriers, four traffic lights, four axles counters, one building for the wireless communication and one command panel. Every component is placed in a slot which are spread around the module. The slots communicate with each other by the use of a fieldbus system that provides the local communication.

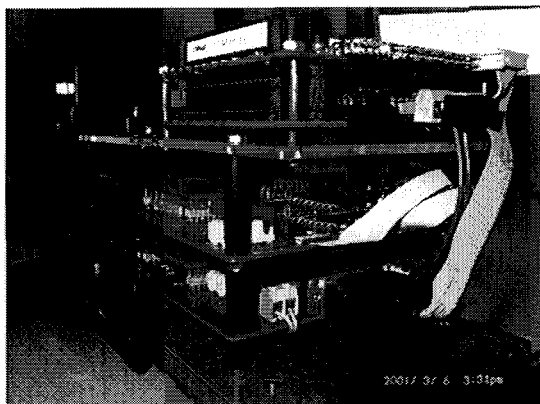


Figure 6: Electronic equipment (trainborne)

The barriers can be controlled separately. They can move up and down and can be stopped at every position. When reaching the end position, the motor is stopped and a message is generated. The motor can be set into a disturbed state what stops the barriers immediately and produces an error message. This error message holds the same information, no matter if the message is generated because of a hardware error, e.g. a truck is standing under the barrier and blocking it, or generated by a software induced disturbance. This makes it possible to run independent error scenarios. An hardware error is detected by checking the motor torque.

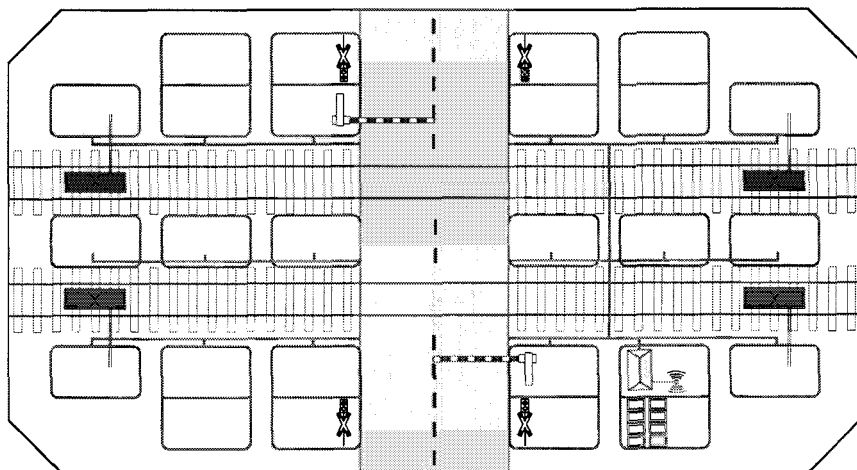


Figure 7: Railroad crossing track module

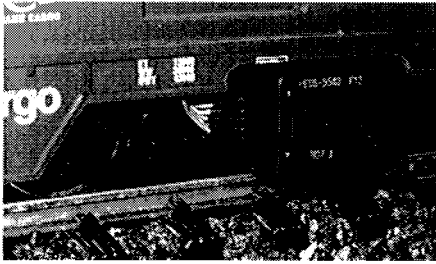


Figure 8: Odometer

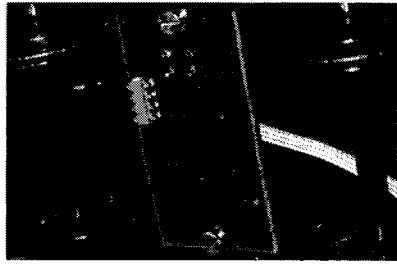


Figure 9: Balise antenna

The traffic lights can be set into red, yellow or off status. The combination of red and yellow light contemporary is not allowed. The correct function of the traffic lights is checked and disturbance messages can be received and generated as already described for the barriers.

4.3 Location

The axles counter is realised by the use of three light barriers. The building for wireless communication holds the communication module equipped with DECT devices as described before.

To obtain the location information a balise antenna and an odometer are used. Figure 8 shows the odometer mounted on the side of "Cargo Sprinter" chassis. It counts the axis revolutions and together with the balise antenna shown in Figure 9 the location information is generated. The balises are setting fixed point on the track. The odometer accuracy is decreasing as the train gets moving. The fixed points are used to compensate this decreasing accuracy especially near to critical operational points (e.g. activation point).

4.4 Fault induction

Figure 10 shows the course of events in the case of software induced disturbances. Disturbance inductions can be made by the use of the operator panel or by the environment control through the railway control. In every case the induction message reaches the fieldbus system and the node placed at the installation element reads this message and disturbs the hardware. In the next step the same node detects a hardware disturbance, no matter which event has caused this disturbance. This disturbance report is send via the fieldbus system to every node. The operator panel indicates this state by lightning a display and the railway control or the control model may react on this condition.

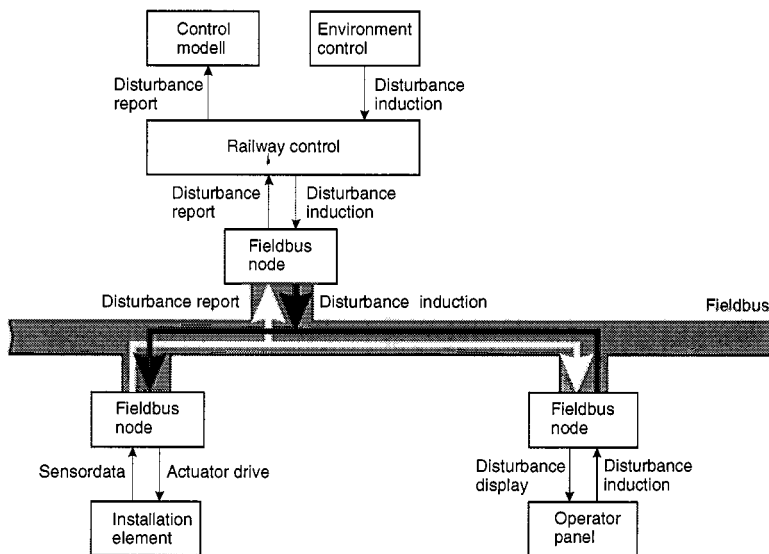


Figure 10: Induction of disturbance

5 Conclusion and perspective

Realisation of the operational process is a very complex task. A physical railway demonstrator model seems to be a suitable means to depict all aspects of real operational system behaviour. Considering the software validation real operational conditions can be offered. The railway model has been conceived and is under construction. After a complete hardware implementation a good base to experimental software validation will be established.

References

- [1] Snelting, G., *Paul Feyerabend und die Softwaretechnologie*, Informatik-Spektrum, pp. 273-276, 1998.
- [2] Ehrig, H., Geisler, R., Klar, M., *Integration von Techniken der Softwarespezifikation für ingenieurwissenschaftliche Anwendungen*, Informatik - Forschung und Entwicklung, 13(1):43-46, 1998.
- [3] Schnieder, E., Jansen, L., *Traffic Control Systems Case Study: Problem Description and Note on Domain-based Software Specification*, Integration of Specification Techniques with Applications in Engineering; extended abstracts, Forschungsbericht des Fachbereichs Informatik der TU Berlin, 2000.