Model checker for railway signalling communication protocol

J.-G. Hwang, H.-J. Jo & J. H. Lee
Train Control Research Team, Korea Railroad Research Institute (KRRI), Korea

Abstract

As a very important part in the development of the protocol, verifications for a developed protocol specification are complementary techniques that are used to increase the level of confidence in the system functions by their specifications. Using the informal method for specifying the protocol, a little ambiguity may be contained in the protocol. This indwelling ambiguity in control systems can be the cause of accidents, especially for safety-critical systems. To clear the ambiguity contained in the designed protocol, we use the LTS (Labelled Transition System) model to design the standard protocol for railway signalling systems. Then we verify the safety and liveness properties automatically and formally through the model checking method. The modal μ-calculus, which is an expressive method of temporal logic, has been applied to the model checking method. In this paper, we verify the safety and liveness properties of the Korean standard protocol for railway signalling systems. To automatically check the safety and liveness properties of the designed protocol, the formal checker is implemented. The developed tools are implemented by the C++ language under Windows XP.

1 Introductions

A few years ago, most of the equipment consisted of vital relay-based systems ensure the safety of railway signalling systems. However, according to the computerization of railway signalling systems, lots of information is exchanged among the computerized railway signalling equipment. By the systemization of railway signalling systems, the communication link is considered more significant than before. Therefore, the communication protocol has to be clearly

WIT Transactions on The Built Environment, Vol 88, © 2006 WIT Press
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/CR060661

defined and standardized for the systemized and intelligent railway signalling systems [1].

A new protocol for railway signalling systems has designed and standardized in our research. It is expected that the communication protocols designed by experts could have brought about some ambiguities. Provided that there were some ambiguities in the designed protocol by the experts, the ambiguities might provoke fatal flaws in the control of signalling systems or accidents. Therefore, the communication protocol for vital systems like railway signalling systems has to be correctly verified [2, 5]. The primary objectives of protocol standardization are to allow systems developed by different vendors to work together, to exchange and handle information successfully. In recent years, the application of formal methods to standardized protocol design has given rise to a new field called protocol engineering. Formal verifications are complementary techniques that are used to increase the level of confidence in the correct functioning of communication protocol by their specifications. Formal verification can give certainty about satisfaction of a required property, but this certainty only applies to the model of the specification.

For our research, we use LTS model to design the communication protocol for railway signalling, LTS model is an intermediate model for encoding the operational behaviour of processes. And then, we verify automatically and formally the safety and liveness properties through the model checking method, especially modal μ-calculus [3, 4]. This paper presents a model checking method for Korean railway signalling protocol specified in LTS and developed automatic verification tool which is able to verify formally whether properties expressed in modal logic are true on LTS specifications. The implemented formal checker using model checking method enables to verify whether deadlock and/or livelock properties are true on specifications.

This paper is organized as follows: Section 2 describes some key concepts and notations in protocol verification. Section 3 describes specifications of a Korean standard protocol for railway signalling systems. Section 4 gives the functional description of a model checker, and finally conclusions are given in Section 5.

2 Formal verification of communication protocol

2.1 Preliminaries

Communication protocols are developed through several phases, such as user requirement analysis and specification, design, implementation and conformance test phase, respectively. Most of protocol development phases are accomplished by human experts. Thus it is expected that some ambiguities or faults may be applied in protocol specification. Those included ambiguities can come into malfunction of systems or accidents. Above section mentioned, formal method are applied in protocol design phase to clearance of built-in faults or ambiguities, thus to assurance of safety and reliability of protocol for safety-critical control systems, such railway signalling systems.

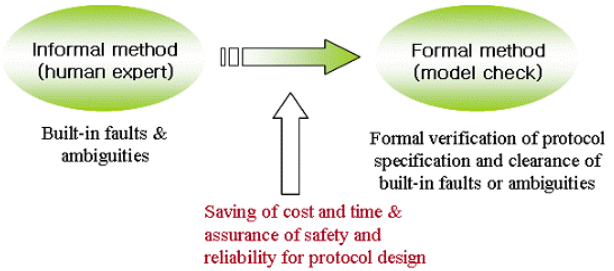

Figure 1: Introduction of formal method.

Fig. 2 shows the procedure of protocol development with formal verification. Formal verification has formal specification phase by formal description language and verification by model checking method. The inherent ambiguities or faults can be cleared and assures the safety of designed protocol through the protocol specification and this verification phase. Two shaded (yellow) colour parts in fig. 2 show the above described formal verification phase.

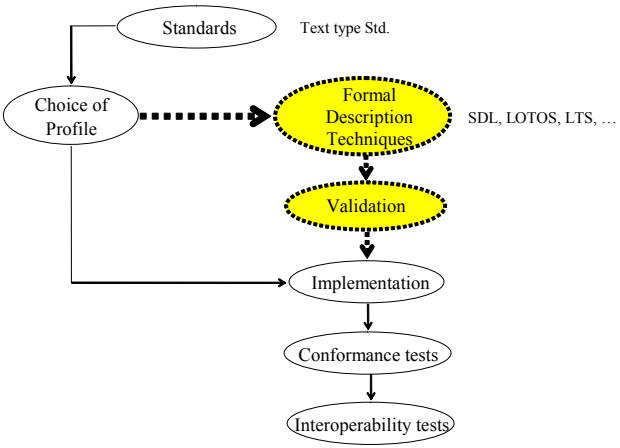

Figure 2: Protocol design with formal verification.

The protocol for railway signalling system requires more high reliability and safety than other industrial control systems. Formal verification is very useful method to verify the correctness of designed protocol. In our research, the formal verification is applied to designed standard communication protocol for Korean signalling systems. LTS model is used to design the communication protocol for railway signalling in this paper, and we verify the safety and liveness properties through the model checking method, especially modal μ-calculus. Fig. 3 shows the formal verification procedure of designed standard communication protocol for Korean railway signalling system.

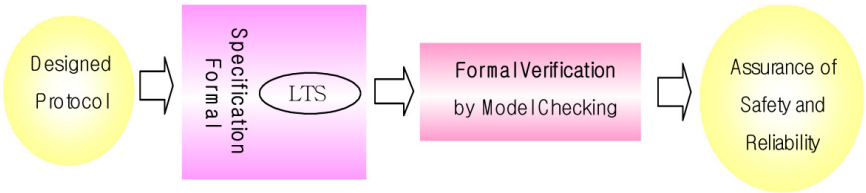

Figure 3: Formal verification procedure of railway signalling protocol.

2.2 Definition of LTS

The formalism of LTS is used for modelling the behaviour of processes, systems and components. LTS serves as a semantic model for a number of protocol specification languages, e.g. CCS (Calculus of Communication System), CSP (Communicating Sequential Processes), and LOTOS [3, 4].

Definition 1: A labelled transition system is a 4-tuple $<S, L, T, s_o>$ with

- S is a (countable) non-empty set of states.
- L is a (countable) set of observable actions.
- $T \subseteq S \times (L \cup \{\tau\}) \times S$ is the transition relation.
- $s_o \in S$ is the initial state.

2.3 Model checking for verification

Model checking is a verification technique that uses formulas of a temporal logic to express properties of a system expressed in some other kind of specification language, and then matches them each other to decide whether the property holds for the system.

We use finite state LTS to specify the designed protocol specification. It has been the most common specification paradigm in recent years. Also, we choose the modal μ-calculus as property specification language. The modal μ-calculus, originally due to Kozen [4], is proposed as a highly expressive logic that can be used to specify properties of concurrent systems represented as LTS.

2.3.1 Modal μ-calculus

The modal μ-calculus can alternatively be viewed as the logic obtained by adding recursion to Hennessy-Milner logic. More generally practitioners have found Hennessy-Milner logic useful to be enabling to express temporal properties of concurrent systems. However, as logic it is not very expressive because formulas of the logic are not rich enough to express such temporal properties. Thus several operators are added in modal μ-calculus. The result is a very expressive temporal logic.

2.3.2 Safety and liveness

The protocol has two properties which it have a safety without deadlock and livelock, and a liveness with some good state and action. A safety property states that some bad feature is always precluded. Safety can either be ascribed to states, that bad states can never be reached, or to actions, that bad actions never happen. A liveness property states that some good feature is eventually fulfilled. Again it can either be ascribed to states, that a good state is eventually reached, or to actions, that a good action eventually happens.

3 Designed protocol specification

In this section, as a reference model for verification, we concentrate on a Korean railway signalling protocol between CTC communication server and EIS (electronic Interlocking System). The CTC communication server located in the centralized control centre receives the control commands from CTC main computer for the control of field signalling equipment such as signal aspects, point machines, and others. Conversely, the EIS transfers state information of the field signalling equipment CTC computer. If this link contains any faults or errors, they may lead to a severe accident because the interface link is the essential hub-link for controlling and monitoring railway signalling, so the interface link is a significant link from the point of view of safety of the railway signalling operation.

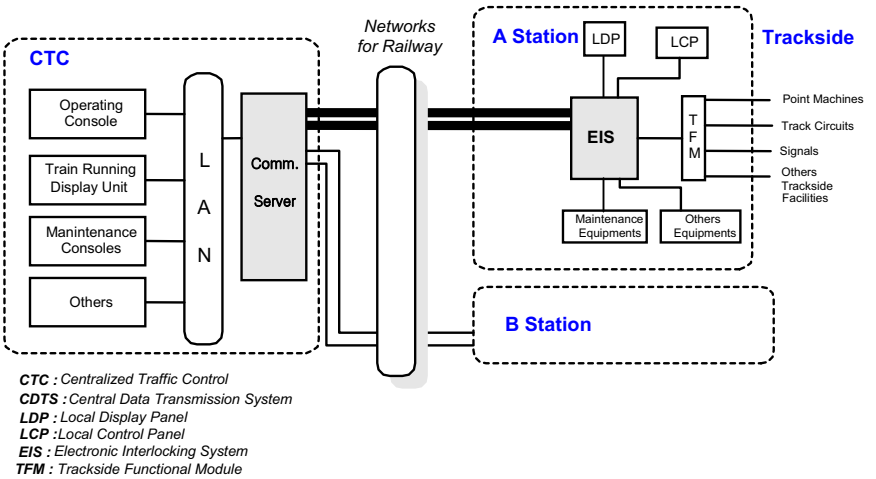

Figure 4: Configuration of railway signalling systems.

Fig. 5 specifies LTS for modelling the behaviour of railway signalling process. This LTS model has 6 states and 9 transitions. The details of standard protocol can be identified in reference [1].

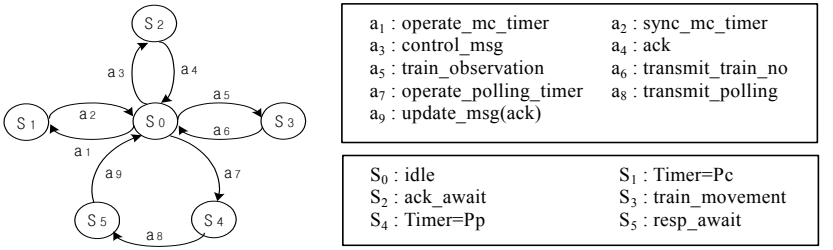

Figure 5: LTS model generated from designed protocol.

4 Model checking by formal checker

For the reference LTS shown in Figure 5, we will verify the general correctness of the LTS specification by applying the explained concepts proposed in section 2. We verify the designed protocol using the modal μ-calculus formula, which means we examine the correctness properties of the protocol. For example, the equation (1) modal μ-calculus formula has to be "true", if the LTS of the designed protocol consists of the non-existence of deadlock and livelock. The Solve algorithm [4] as a model-checking algorithm is applied to this equation (9) formula. From this process the verifying results can be obtained.

$$\nu Z.\ (\mu\, Y.A \vee (\ <->\ tt \wedge [-]Y)) \wedge [-]Z,\ \ A=\{S0\} \qquad (1)$$

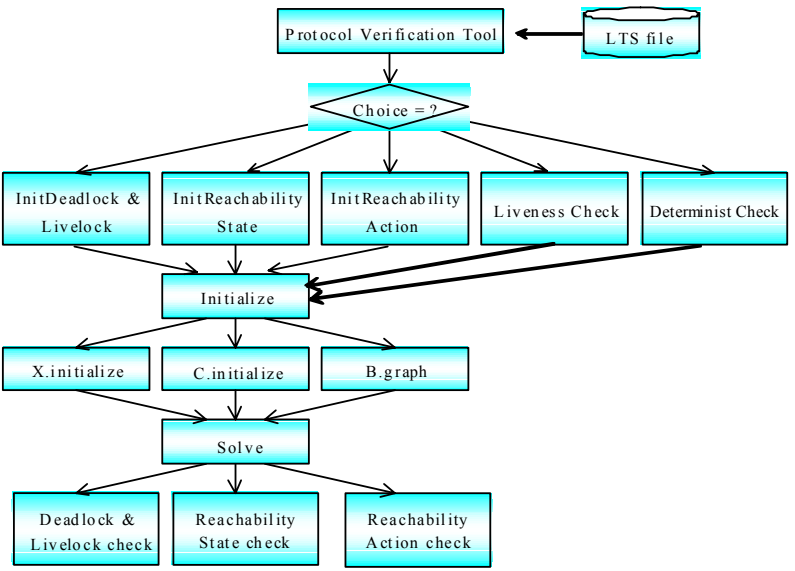

Figure 6: Operation flow of developed model checker.

We developed the formal verification tool called model checker for protocol design of railway signalling systems by above described formal specification and verification method. The verification items are deadlock, livelock, reachability, liveness and determinist properties of designed protocol. The configuration of developed model checker shows fig. 6.

The input of this model checker is the LTS model generated from designed protocol such as fig. 4. Above sections described verification algorithm, such as formal description technique, modal mu-calculus logics and Solve algorithm is implemented in model checker. The implemented formal checker is able to verify whether properties expressed in modal logic are true in specifications using modal mu-calculus. The suggested tools are implemented by C++ language under MS-Windows NT environment. In the way of fig. 7, the text-based LTS modelling file, 'lts.lts', is inputted in model checker, and clicks one of Check button what we want to verify, and then the verification results windows are popped. If the 'Solve Algorithm' is clicked, the executable process of verification by Solve algorithm becomes visible in 'Solve.txt' text file. Fig. 6 shows process of formal verification of standard protocol for Korean railway signalling systems using developed model checker.

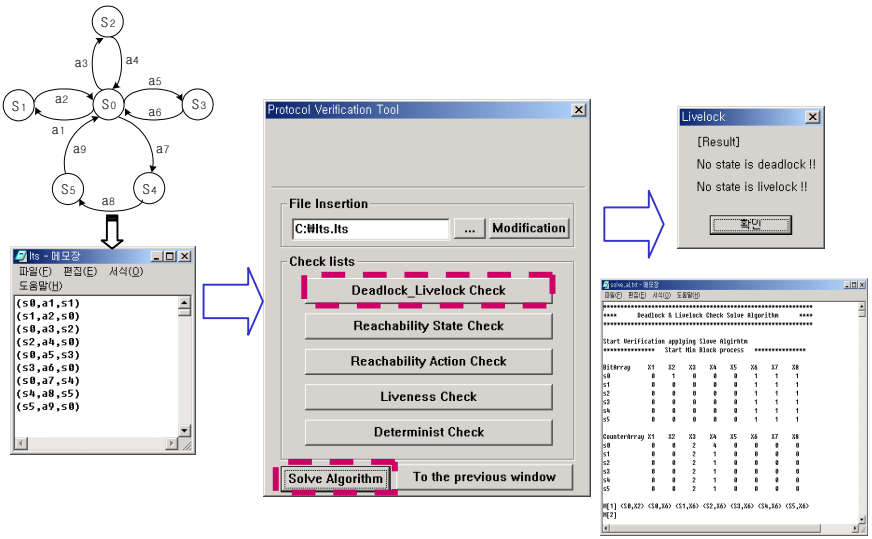

Figure 7: Deadlock and livelock checking by developed model checker.

5 Conclusion

Using the informal method in specification of the communication protocol, ambiguities are generally contained in the protocol. To clear up the ambiguity contained in the designed protocol, the protocol is specified in LTS and verified the safety and liveness properties by the model checking method. This formal verification process for designed protocol requires elaborate and difficult efforts.

In this paper, we have developed the user friendly model checker by GUI (Graphic User Interface) under MS-windows environment. It is expected to reduce the time and cost for protocol design by using this verification tool, and to increase the safety, reliability and efficiency of maintenance of the signalling systems by using the designed protocol for railway signalling in Korea.

References

[1] J. G. Hwang and J. H. Lee, "A New Data Link Protocol for Korea Railway Signalling Systems", KIEE Int'l Trans. on EMEC, Vol.3-B, No.4, pp. 195-201, Dec. 2003.
[2] D. Schwabe, 'Formal Techniques for the Specification and Verification of Protocol', Ph.D Thesis, Univ. of California Los Angeles, 1981.
[3] O. Burkart and B. Steffen, Model Checking the Full Modal Mu-Calculus for Infinite Sequential Processes, LFCS Report ECS-LFCS-97-355 (1997).
[4] Kozen, 'Results on the prepositional μ-calculus', Theoretical Computer Science, 27:333-354, December 1983.
[5] J. H. Lee, J. G. Hwang and G. T. Park, 'Performance Evaluation and Verification of Communication Protocol for Railway Signalling Systems', Computer Standards & Interfaces in Elsevier, vol. 27, pp. 205-219, Feb. 2005.