Development of Communication Protocol Verification Tool for Vital Railway Signaling Systems

Jong-Gyu Hwang† , Hyun-Jeong Jo* and Jae-Ho Lee*

Abstract - As a very important part in development of the protocol, verifications for 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, some ambiguity may be contained therein. This indwelling ambiguity in control systems can cause the occurrence of accidents, especially in the case of safety-critical systems. To clear the vagueness contained in the designed protocol, we use the LTS (Labeled Transition System) model to design the protocol for railway signaling. And then, we verify the safety and the liveness properties 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. We verify the safety and liveness properties of Korean standard protocol for railway signaling systems. To perform automatic verification of the safety and liveness properties of the designed protocol, a communication verification tool is implemented. The developed tools are implemented by C++ language under Windows XP. It is expected to increase the safety and reliability of communication protocol for signaling systems by using the developed communication verification tool.

Keywords: Formal Verification, LTS, Protocol verification tool, Railway signaling systems

1. Introduction

A few years ago, most equipment consisted of vital relay-based systems to ensure the safety of railway signaling systems. However, according to the computerization of railway signaling systems, lots of information is exchanged among the computerized railway signaling equipment. By the systemization of railway signaling systems, the communication link is considered more significant than before. Therefore, the communication protocol has to be clearly defined and standardized for the systemized and intelligent railway signaling systems [1].

A new protocol for railway signaling systems has been 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 signaling systems or accidents. Therefore, the communication protocol for vital systems like railway signaling 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 the LTS model to design the communication protocol for railway signaling. The LTS model is an intermediate model for encoding the operational behavior of processes. Next, 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 signaling protocol specified in LTS as well as a developed automatic verification tool that is able to verify formally whether properties expressed in modal logic are true in regards to LTS specifications. The implemented formal checker using model checking method enables verification as to whether deadlock and/or livelock properties are accurate on specifications.

† Corresponding Author : Train Control System Research Team, Korea Railroad Research Institute, Uiwang-city, Kyonggi-do, Korea (jghwang@krri.re.kr)
* Korea Railroad Research Institute, Uiwang-city, Kyonggi-do, Korea
Received March 6, 2006 ; Accepted September 16, 2006

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 the 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. As mentioned in the above section, a formal method is applied in protocol design phase to clear up any built-in faults or ambiguities, thus assuring safety and reliability of protocol for safety-critical control systems, such as railway signaling systems.

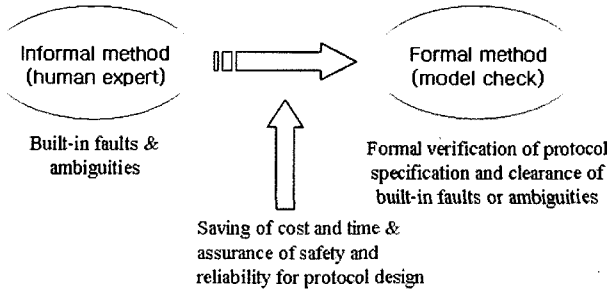


Fig. 1. Introduction of formal method

Fig. 2 shows the procedure of protocol development with formal verification. Formal verification has a formal specification phase by formal description language and verification by the model checking method. The inherent ambiguities or faults can be cleared and assure the safety of designed protocol through the protocol specification and this verification phase. The two yellow colored parts in Fig. 2 signify the above described formal verification phase.

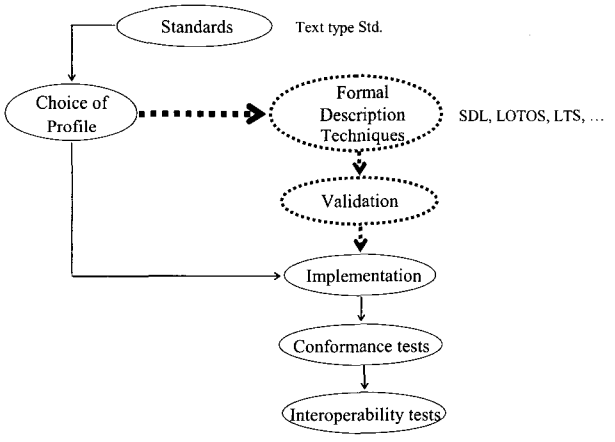


Fig. 2. Protocol design with formal verification

The protocol for railway signaling systems requires more high reliability and safety than other industrial control systems. Formal verification is a very useful method to substantiate the correctness of the designed protocol. In our research, formal verification is applied to designed standard communication protocol for Korean signaling systems. The LTS model is used to design the communication protocol for railway signaling 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 signaling systems.

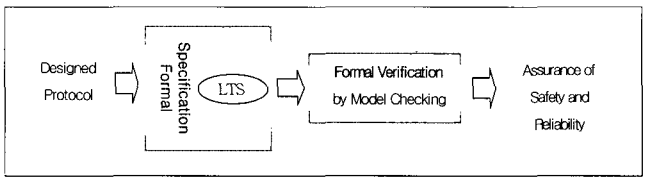


Fig. 3. Formal verification procedure of railway signaling protocol

2.2 Definition of LTS

The formalism of LTS is used for modeling the behavior 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 [6-8].

Definition 1: A labeled transition system is a 4-tuple $<S, L, T, s_0>$ 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.
- $s_0 \in S$ 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 to each other to decide whether the property holds for the system.

We use finite state LTS to specify systems. It has been the most common specification paradigm in recent years. Also, we choose the modal μ-calculus as property specification language. It is a reasonable compromise between expressive power and complexity of model checking. Generally the major obstacle in applying finite state LTS checking to the correctness of large

specifications is the combinatorial explosion of the state space arising when many loosely coupled parallel processes are considered. To address this problem, the modal μ-calculus is originally due to Kozen [3] although in its current incarnation, modalities are parameterized by action names [7]. The modal μ-calculus 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 [8]. More generally practitioners have found Hennessy-Milner logic useful to enable the expression of 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. As such, extra operators at external fixed points are added in modal μ-calculus. The result is a very expressive temporal logic.

In modal μ-calculus, fomulas consist of atomic propositions, ∧ (conjunction), ∨ (disjunction), [ ] (necessity), < & > (possibility), ν (greatest fixed point) and μ (least fixed point). And generalized formulas for modal μ-calculus are as follows:

$$\Phi ::= tt \mid ff \mid Z \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid [K]\Phi \mid \langle K \rangle \Phi \mid \nu Z.\Phi \mid \mu Z.\Phi \qquad (1)$$

where:

- *ν and μ: Operators for expression of least and greatest fixed-point, respectively*
- *tt and ff: Expression of true and false for all state, respectively*
- *Z: Specific variables, that is, process*
- *k: Element of action set*
- *Φ: Formulas for process characteristics*

Where tt and ff are the atomic proposition that is respectively true or false at every state, Z ranges over a family of propositional variables, and K over subsets of A (set of action). The binder νZ is the greatest fixed point operator whereas μZ is the least fixed point operator [3, 4].

2.3.2 Safety and Liveness

The protocol has two properties that 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. If the formula Φ captures the complement of those bad states, then $\nu Z.\Phi \wedge [-]Z$ expresses safety. Where [-] represents all actions.

A liveness property claims 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. If Φ captures the good states then $\mu Z.\Phi \vee (\langle - \rangle tt \wedge [-]Z)$ expresses liveness with respect to state. Where the presence of $(\langle - \rangle tt \wedge [-]Z)$ to ensure that Φ does become true.

2.3.3 Protocol Specification and Model Checking

In this section, as a reference model for verification, we concentrate on a Korean railway signaling protocol between the CTC communication server and SCADA (Supervisory Control And Data Acquisition). Recently, according to the computerization of railway signaling systems such as CTC, EIS, ATC, and so on, the importance of protocol for railway signaling systems is increased by greater exchange of information among computerized signaling systems. For this reason, a protocol with high reliability for railway signaling systems is required. Among these, in this research, we concentrate on the interface link between CTC and SCADA. The CTC system is a major railway signaling system and the SCADA system performs the role of control and monitoring of the railway catenaries system.

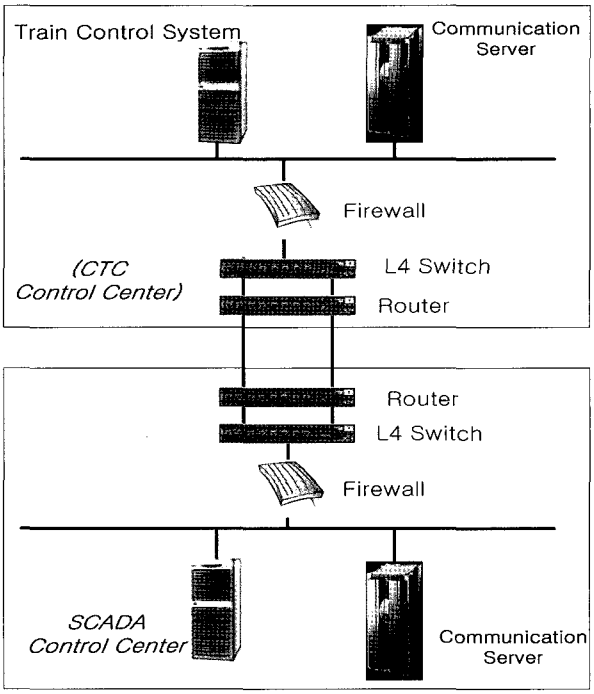
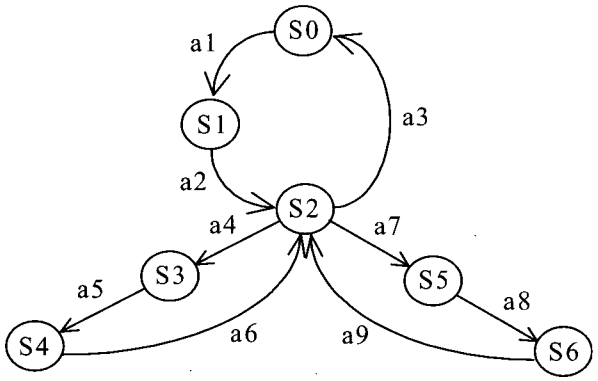


Fig. 4. Configuration of SCADA ⇔ CTC link

Only a few years ago, there was no interface link between these two systems in Korea, so these two systems had been operated separately. However, the interface between SCADA and CTC has been important according to the upgrade of train running speed such as the Korean high-speed train, because CTC for railway signaling has needed



S0 : idle	S1 : ack_awaited
S2 : TCP_con_estab	S3 : T1=P1
S4 : resp_awaited	S5 : T2=P2
S6 : ack_awaited	
a1 : TCP_con_req	a2 : ack
a3 : release	a4 : operate_T1_timer
a5 : train_state_msg	a6 : ack
a7 : operate_T2_timer	a8 : SCADA_state_msg
a9 : ack	

Fig. 5. LTS model generated from designed protocol

the catenary's information for safety operation of high-speed trains. Fig. 4 shows the configuration of interface link between these two systems by network router equipment. The SCADA systems send the states information such as relating to powered catenaries sections, and reversely the CTC system sends the location information or train number of running trains to the SCADA systems.

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

For the reference LTS shown in Fig. 5, we will verify the general correctness of the LTS specification by applying the explained concepts proposed in Subsection 2.2. Now we wish to determine which states of this process satisfy the formula Lμ where $\{B_1, B_2\}$. Where Lμ is

$$\nu Z.\ (\mu Y.A \vee (<->\ tt \wedge [-]Y)) \wedge [-]Z \text{ if, } A=\{S_0\} \qquad (2)$$
$$B_1 = \min\{Y = A \vee (<->\ tt \wedge [-]Y)\}$$
$$B_2 = \max\{Z = Y \wedge [-]Z)\}$$

Recall that states satisfying this formula have the safety property that along all paths emanating from S_0 state, it is always the case that A eventually holds. Fig. 6 contains the translation of this block set into blocks having only simple right-hand sides; it also shows the dependency graph corresponding to these new blocks. Note that variable X_1 corresponds to variable Y, while X_7 corresponds to Z.

$B_1 \equiv \min\{X_1 = X_2 \vee X_3$	$B_2 \equiv \max\{X_7 = X_1 \wedge X_8$
$X_2 = A$	$X_8 = [-]X_7\}$
$X_3 = X_4 \wedge X_5$	
$X_4 = [-]X1$	
$X_5 = <->X_6$	
$X_6 = tt\}$	

Fig. 6. Max block, min block

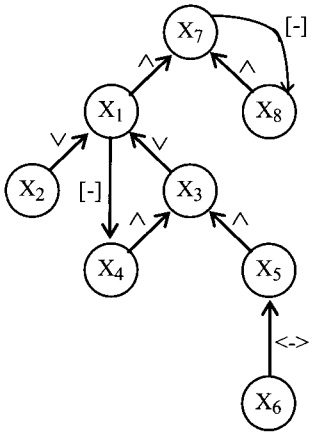


Fig. 7. Edge-labeled directed graph G

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 as a model-checking algorithm is applied to this equation (2) formula. From this process the verifying results can be obtained. The details pertaining to algorithms are found in [4, 5].

3. Development of protocol verification checker

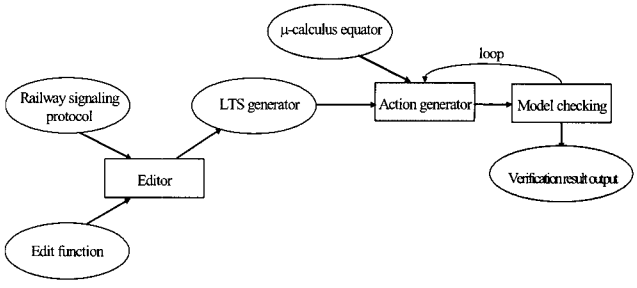


Fig. 8. Components of the integrated environment

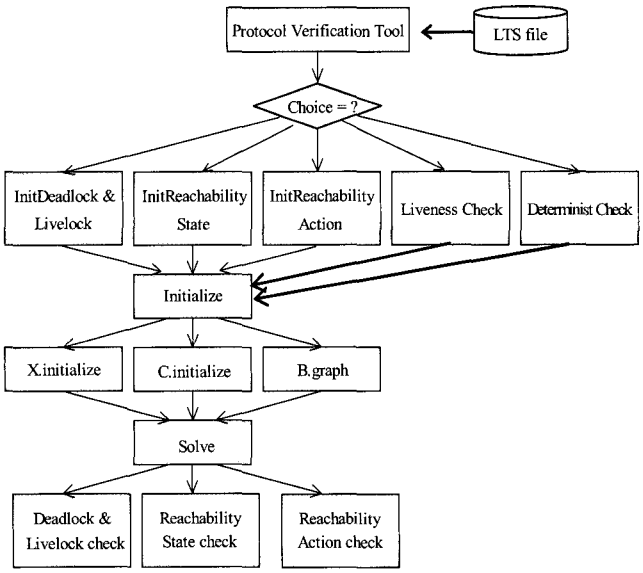


Fig. 9. Operation flow of developed model checker

We developed the formal verification tool called model checker for the protocol design of railway signaling systems by the 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 is presented in Figs. 8 and 9.

The input of this model checker is the LTS model generated from designed protocol such as that in Fig. 5. The above sections describe the verification algorithm, such as formal description technique, modal μ-calculus logics and Solve algorithm as implemented in the model checker. The implemented formal checker is able to verify whether properties expressed in modal logic are true in specifications using modal μ-calculus. The suggested tools are implemented by C++ language under the MS-Windows NT environment. In the way of Fig. 10, the text-based LTS modeling file, 'lts.lts', is inputted in the model checker. By clicking what we want to verify using the Check button, the verification results windows pop up. If the 'Solve Algorithm' is clicked, the executable process of verification by Solve algorithm becomes visible in the 'Solve.txt' text file. Fig. 10 indicates the formal verification process of some part of standard protocol for Korean railway signaling systems using the developed model checker. This Fig. makes one module of a protocol modeled into LTS, inputs it into the developed model checker, and verifies the results. From this, we can validate that in the developed model of protocol, no deadlsock and livelock states exist.

Fig. 11 indicates that we are in generated arbitrary deadlock and livelock state in the inputted LTS model to verify the developed tools. According to these results, we could identify that in the case that deadlock and/or livelock states exist arbitrarily in the inputted model, their status was detected. By applying the developed model checker, the safety of the standardized protocols is verified.

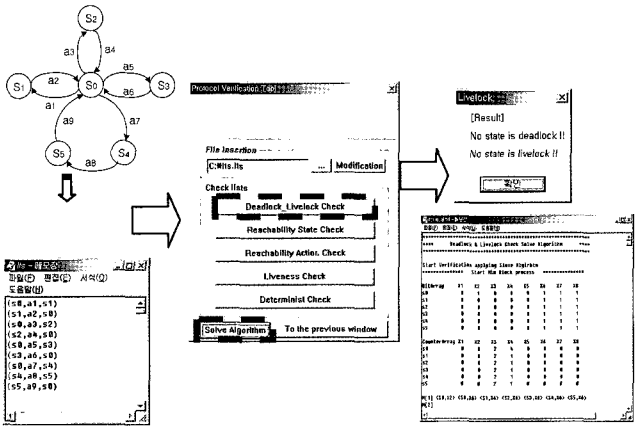
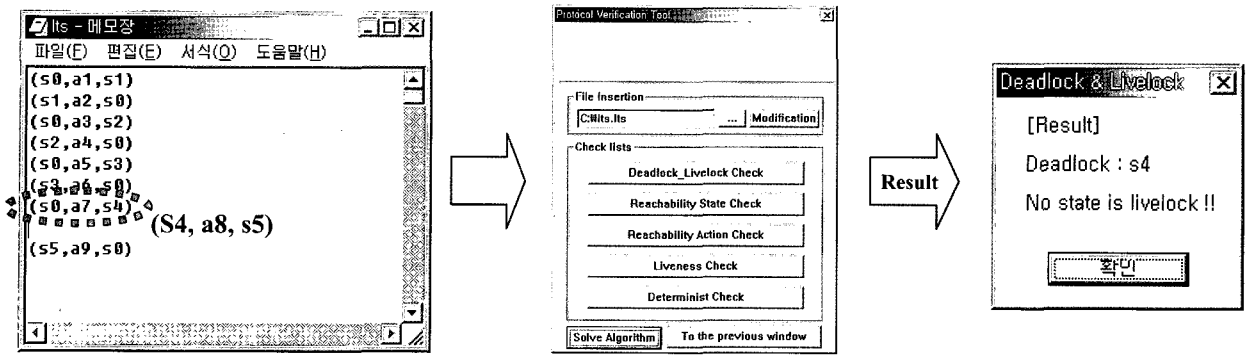


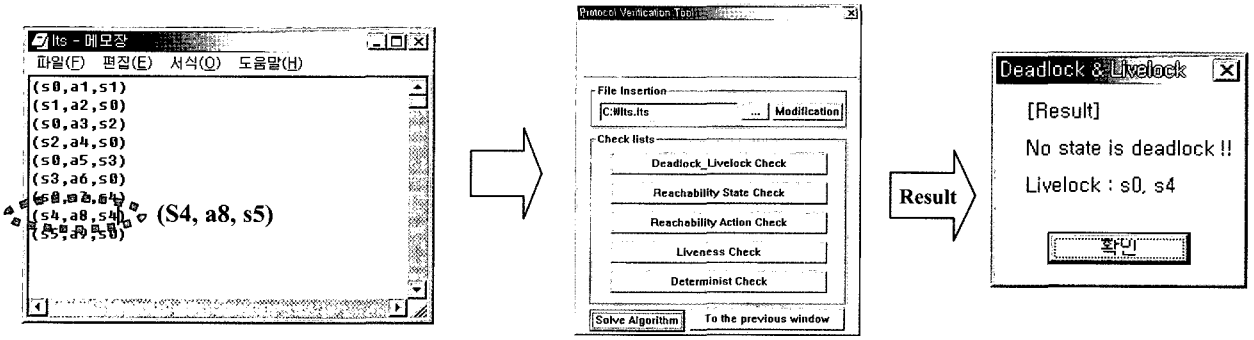
Fig. 10. Formal verification process of communication protocol by developed model checker

4. 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 verifies 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 the MS-windows environment. The safety and liveness properties of two standard protocols for Korea railway signaling systems (point-to-point and network-based protocol) are verified by the developed tools, which are enacted as Korean standards. The standard protocols, which are proved by safety and liveness properties verification by the currently developed tools, are being applied in the work sites by KORAIL, and problems such as the verified properties of the protocol itself has not been identified so far. The effectiveness of the tools developed in this study has also been confirmed from the work sites. Furthermore, 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 signaling systems by using the designed protocol for railway signaling in Korea.



(a) Deadlock occurrence



(b) Livelock occurrences

Fig. 11. Deadlock & livelock checking by developed tool

References

[1] J. G. Hwang and J. H. Lee, "A New Data Link Protocol for Korea Railway Signaling 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 M-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.

[6] R. Milner, Communication and Concurrency, Prentice Hall International, 1989.

[7] R. Cleaveland, B. Steffen, "A Linear-Time Model-Checking Algorithm for the Alternation-Free Modal Mu-Calculus", Formal Methods in System Design, Feb. 1993.

[8] M. C. B. Hennessy and R. Milner, "Algebraic Laws for Non-determinism and Concurrency", J. ACM, 32(1): 137-161, Jan. 1985.

Acknowledgements

The authors would like to acknowledge the counsel and discussion concerning formal verification of protocol communication provided by Professors Sung-Un Kim in Pukyung National Univ. and Mi-Seon Seo in Samsung Electronics, in improving the quality of this paper.

Jong-Gyu Hwang

He received his B.S., M.S. degrees in Electrical Engineering from Konkuk University, Korea 1994, 1996 respectively, and Ph. D degree in Electronic and Computer Engineering from Hanyang University, Korea 2005. Since 1995, he has been a senior researcher with the Train Control System Research Team of the Korea Railroad Research Institute (KRRI). His research interests are in the areas of railway signaling, protocol engineering, communication and computer network technology.

Hyun-Jeong Jo

She received the B.S. degree from the Hankuk Aviation University, Goyang, Gyonggi-do, Korea, in 2003. She worked toward the M.S. degree at the Gwangju Institute of Science and Technology (GIST), Gwangju, Korea. Since 2005, she has been engaged with the Train Control System Research Team of the Korea Railroad Research Institute (KRRI).

Jae-Ho Lee

He received the B.S. and M.S. degree in electronics engineering from Kwang-woon University, Korea, Ph. D. degree in Mechatronics Engineering from Korea University, in 2005. Since 1995, he has been with Korea Railroad Research Institute (KRRI), where he is principle researcher of train control research team, doing research in the areas of railway signaling and telecommunication. His research interests include analysis and design of railway signaling systems and application of mechatronics to train control system.