# Formal Dynamic Operational Model of RIS Components

**Nazir Ahmad Zafar**
Department of Computer Science, King Faisal University, Hofuf, Saudi Arabia

***Abstract*—Primary objective of railway interlocking system is preventing trains from collision and derailing while at the same time allowing efficient and normal movement of trains. Railway interlocking system is a safety critical system and it needs a high practice of its modeling and development. Previous work of the author was on modeling of railway network components and safety analysis of the system. In this paper, we have applied Z notation to describe the operational model controlling the network's components based on their formal definitions. At first the critical components are described then state space is defined. Then components are refined and integrated to define the simplified system. The state space is enhanced by adding trains and controls to describe the entire system. Finally, formal dynamic model is proposed by defining critical local and global operations to guarantee safe and efficient operation of the system. The model is analyzed and validated using Z/Eves tool.**

***Index Terms*—Formal methods, Operational model, Railway interlocking, Z notation**

## I. INTRODUCTION

Railway interlocking is a safety critical system because its incorrect functioning may cause serious consequences, for example, environmental damages, economical losses, human deaths, severe injuries. The use of computers in safety critical systems has increased the safety issues. Formal methods are a promising way of providing confidence in modeling of such systems. Indeed, the usage of formal methods in safety critical and complex systems is recommended [10], [11].

The objective of railway interlocking system (RIS) is preventing trains from collision and derailing while at the same time allowing normal and efficient movement of trains. There are two existing technologies, i.e., fixed block and moving block interlocking. Moving block is getting high important due to some disadvantages in fixed block interlocking. The biggest disadvantage of fixed block interlocking is that a long distance is needed between two trains which limit the capacity of network and speed of trains.

The system under hand is a complex and it requires much effort in its correct modeling. In previous work [13], [25], [26], [27], [28] and [29] of the authors, formal description of network components was given and its safety analysis was provided. In this work, we have described formal definitions by simplifying the network components then the components are integrated to describe the entire network. Trains and controls are added to complete the definition of the whole system. Finally, critical operations are defined over the system for safe and efficient movement of the trains. The overall objectives of this research are: (i) to apply formal methods in developing safe and efficient operation of railway interlocking, (ii) integration of approaches supporting automated modeling, and (iii) to learn and practice the validation and proving techniques.

Modeling and development of railway interlocking system is an open research problem. There exists a lot of work on modeling of railway interlocking. A list [6] of about 300 publications addressing various issues on railway interlocking proves importance of this research and practical problem. It is noted that most of the publications are on fixed block while our work is on moving block interlocking with a different approach. The work of A. Simpson is close to this in which he uses Z, CSP and FDR2 [20] and his work is a starting point for us. Hansen [11] used VDM to model concepts of railway network topology and defined safety criteria for validating through simulation but again his work is for fixed block interlocking. The system is constructed by making control and information coexist for modeling a large-scale system [14]. In [24], it is described railway safety system by focusing on electronic interlocking. In this work, component-based approach is applied using extended and stochastic petri nets. Interlocking rules are validated by instantiating to a specification as described in [8]. It is proposed a model-based generation of interlocking software from control tables using model-driven tool-chain in [18]. She, X. et al, have presented an interlocking using graph search approaches based on heuristic algorithms. Some other relevant work of interest can be found in [1], [2], [3], [4], [5], [7], [12], [16], [17], [19] and [23]. Rest of the paper is organized as follows:

In section 2, an introduction to railway interlocking is given. In section 3, an overview of formal methods is presented. Formal specification of the system is described in section 4. Model analysis is presented in section 5. Conclusion and future work are discussed in section 5.

Manuscript received September 5, 2011
Manuscript revised September 20, 2011

## II. RAILWAY INTERLOCKING SYSTEMS

It is quite obvious that railway interlocking system is a safety critical system because its failure may cause loss of lives, severe injuries, environmental damages etc. Primarily, the task of railway interlocking system is to guarantee safety while at the same time allowing normal train movements. The objective of railway interlocking is to control signals and points for the safe and efficient operation of system preventing collision and derailing of trains. For this purpose, it needs to ensure that there must be, at most, one train at one component preventing collision. Similarly, the points of railway tracks must be set in such a way that train's direction must be consistent with the point's control, which is in the route of the train. There are various types of interlocking systems, in practice, varying from purely mechanically operated to state-of-the-art computerized moving block interlocking systems. The mechanically operated interlocking systems coordinate the levers controlling the points with the signals governing the sections and linked branches, sidings, or loops at the railway network. In manually operated mechanical interlocking, it is relied on the signalman to move about from signals and a set of points to another combination of signals and points to operate in connection with the route of a train. Electrically operated interlocking systems are advanced schemes in which the points and signals are integrated with complete description, for example, state of the section whether it is occupied or free. These interlocking are sometimes integrated with mechanical systems and called electro-mechanical interlocking. A more advanced form of interlocking system is automatic block signaling in which the signals are automated and operate in conjunction with track circuiting or with other techniques detecting the presence of a train in a block. In such systems, when train enters into a block the state of the signal changes automatically.

There are two main types of railway interlocking systems, i.e., fixed block interlocking and moving block systems. Moving block interlocking is getting more important in railway industry due to some disadvantages in fixed block interlocking. In fixed block interlocking, the railway network is divided into fixed blocks which are separated by signals. At one time, only one train can move in a block and can enter into a block only if the next is clear. The length of block is limited and cannot be shortened below a certain fixed length which is the distance required by a train at full speed to come to a complete stop. As a result, the biggest disadvantage of this system is that a long distance is required between two trains which limit the capacity of railway line and speed of train. Further, position of a train is not recorded which makes the system highly in-efficient particularly in main routes.

In reality, the safe distance between two trains is the distance needs for a train to come to a complete stop which is much less than the length of a fixed block and even shorter if a train is moving at low speed. The idea of moving block is based on this concept, i.e., keeping only safe distance between trains. Instead of cutting piece of railway line into fixed blocks, train's occupying area and some distance in front of it becomes the moving block in which no other train can enter.

## III. FORMAL METHODS

In this section, an introduction to formal methods is given. Formal methods are mathematically based techniques and approaches for describing and analyzing properties of software and hardware systems [9]. The description of a system is written using notations which are based on mathematical expressions rather than informal notations. These formal approaches are typically drawn from areas of discrete mathematics, such as, logic, set theory or graph theory. There are several ways of classification and description of formal methods. For example, model and property oriented methods are two major classes of formal methods [22]. Model oriented formal methods are used to describe model of dynamics of a system. For example, state transition diagrams are used to model the behavior of a system as a set of states and transitions between it. However, the transitions can be deterministic or non-deterministic. Property oriented methods are used to describe system in terms of a set of properties or constraints which must be satisfied thereat. The Z notation [21] is a model oriented approach based on set theory and first order predicate logic. In addition, it is used for specifying the dynamics and behavior of abstract data types and sequential programs. The Z is used in this research for specification because it describes a state space and a set of operations that may be performed on a system.

## IV. DYNAMIC MODEL

In this section, formal dynamic model is described. At first formal definitions of railway network components are given then components are integrated to define the entire network. Finally, most some important operations are defined over the critical components of the system.

### A. Formal Definitions

Formal definitions of network components are given in this section. At abstract level of specification, a track segment of the network is specified by *Track* and the entire network is described by a *Graph* relation as given below.

$$[Track, TrainId];\ Graph == \{\ x: Track;\ y: Track \mid x \neq y \bullet (x, y)\ \}$$

$$State ::= OCCUPIED \mid CLEAR;\ SwitchControl ::= LEFT \mid RIGHT$$

A switch is specified by a schema with four variables that is identifier, state, train occupying switch and control mapping. The control of a switch is either in left or right.

```
┌─ Switch ──────────────────────────
│ switch: Track
│ state: State
│ trainid: TrainId
│ control: SwitchControl
└───────────────────────────────────
```

Railway crossing described by the schema *Crossing* which is constituted by three components which are crossing identifier, train occupying the crossing and state of the crossing as given below.

```
┌─ Crossing ────────────────────────
│ cross: Track × Track
│ trainid: TrainId
│ state: State
└───────────────────────────────────
```

The intersection of railway track and road is defined as level crossing. A level crossing consists of four components level crossing identifier, state, train occupying the level crossing and barriers. There are two states of barriers which are either open or closed.

*Barriers* ::= *CLOSED* |*OPEN*

```
┌─ LevelCrossing ───────────────────
│ levelCrossing: Track
│ levelState: State
│ occupiedBy: TrainId
│ barriers: Barriers
└───────────────────────────────────
```

## B. Railway Network

The railway network is composed of four components that is topology, switches, crossings and level crossings. All of the components are defined as a sequence type because order is required in their interlocking.

Invariants: (i) Different switches have different identifiers, i.e., switch identifier is unique. (ii) Railway crossing identifier is not in the topology relation. (iii) A switch identifier is not connected to a track segment of a crossing identifier. (iv) The track segment of a level crossing cannot be a segment of a crossing identifier.

The *Controls* is also a mapping from control identifier to Control. The Control is composed of two components, i.e., section for track segments in a section and trains moving in that section. The formal description of the control is given below by using the schema *Control*. It is noted that moving block of a train is contained in the section of a control.

```
┌─ Network ─────────────────────────
│ topology: 𝔽 Graph
│ switches: seq Switch
│ crossings: seq Crossing
│ levelcrossings: seq LevelCrossing
├───────────────
│ ∀sw1, sw2: Switch | sw1 ∈ ran switches ∧ sw2 ∈ ran switches
│   • sw1 . switch ≠ sw2 . switch
│ ∀cr: Crossing | cr ∈ ran crossings • cr . cross ∉ topology
│ ∀cr: Crossing; sw: Switch | cr ∈ ran crossings ∧ sw ∈ ran
│ switches • (sw . switch, cr . cross . 1) ∉ topology
│     ∧ (sw . switch, cr . cross . 2) ∉ topology
│ ∀lx: LevelCrossing; sw: Switch | lx ∈ ran levelcrossings ∧ sw ∈
│ ran switches • lx . levelCrossing ≠ sw . switch
└───────────────────────────────────
```

[*ControlId*];

*Controls* == *ControlId* ⟶ *Control*

```
┌─ Control ─────────────────────────
│ section: ℙ Track
│ trains: TrainId ⟶ ℙ Graph
├───────────────
│ ∀t: TrainId | t ∈ dom trains
│   • ∃sarea: 𝔽 Graph
│     • trains t = sarea
│       ∧ (∀s1, s2: Track | (s1, s2) ∈ sarea
│         • (s1 ∈ section ∧ s2 ∈ section))
└───────────────────────────────────
```

## C. Railway Interlocking

In this section, formal description of the system is given by putting the formalization of railway network and controls. The relationship between these components is established in terms of invariants defined in the second part of the schema. It is noted that every track segment of the section under a control must be in the network topology. And every track segment of the network topology must be in the section of a control.

```
┌─ RIS ─────────────────────────────
│ Network
│ controls: Controls
├───────────────
│ ∀s1, s2: Track | (s1, s2) ∈ topology
│   • ∃cid: ControlId | cid ∈ dom controls
│     • ∃control: Control | controls cid = control
│       • s1 ∈ control . section ∧ s2 ∈ control . section
│ ∀s1, s2: Track
│   | ∃cid: ControlId | cid ∈ dom controls
│     • ∃control: Control | controls cid = control
│       • s1 ∈ control . section ∧ s2 ∈ control . section
│   • (s1, s2) ∈ topology
└───────────────────────────────────
```

### *D. Dynamic Operational Model*

Formal dynamic operational model is presented in this section. At first, local operations defining network components controls are defined then global operation defining entire control of the system is described.

The operation to interlock a switch is defined below. In the schema given below, the symbol Δ is used stating that state of the component *Switch* will be changed. In the operation, two variables are used as input. The symbol *?* after a variable states that it is an input variable. To interlock switch the identifier is given as input for the switch to interlock. The state of the switch is changed to be in the occupied state. The train occupying the switch is recorded. Finally, control of the switch is moved in the left direction.

```
InterlockSwitch
ΔSwitch
sw?: Track
state: State
tid?: TrainId
control: SwitchControl
----------------------
switch = sw?
state = OCCUPIED
trainid = tid?
control = LEFT
```

In the schema given below an operation to make free the switch is defined. The variable Δ has same meanings as defined above. In the schema the state of switch is changed to make it free. The other variables will remain unchanged.

```
FreeSwitch
ΔSwitch
----------------------
state = CLEAR
```

The state of the railway crossing is changed by the schema *OccupyXing* given below. The meanings of symbols Δ and *?* are same as defined above. In the operation, two variables, that is, crossing identifier and train occupying the crossing are given as input and then state of the crossing is changed. In the operation, the state of the crossing is changed to be in the occupied state. And then the train occupying the railway crossing is recorded.

In the schema *CloseLXing* given below, the state of level crossing is changed to be in the occupied state. In the operation, two variables, level crossing identifier and train occupying the level crossing are given as input and state of the level crossing is changed to be occupied. The barriers are closed in case a train is approaching to the level crossing. Finally, the train occupying the level crossing is recorded. When the train is passed the barriers are open.

```
OccupyXing
ΔCrossing
xid?: Track × Track
state: State
tid?: TrainId
----------------------
cross = xid?
state = OCCUPIED
trainid = tid?
```

```
CloseLXing
ΔLevelCrossing
lxing?: Track
tid?: TrainId
----------------------
levelCrossing = lxing?
levelState = OCCUPIED
occupiedBy = tid?
barriers = CLOSED
```

```
FreeLXing
ΔLevelCrossing
----------------------
levelState = CLEAR
barriers = OPEN
```

When a train enters into a new section then the computer based controls observing trains must be updated. In the schema *AddTrain* given below, train entering into the section and tracks occupied by the train are given as input and the control is updated by adding the train in the trains mapping defined in the control schema by using the union operator.

```
AddTrain
ΔControl
tid?: TrainId
added?: 𝔽 Graph
----------------------
trains' = trains ∪ {(tid? ↦ added?)}
```

In the schema given below, the main global operation is defined to update the entire interlocking system. This update is required, for example, when a train occupies some new tracks, leaves the already occupied tracks, enters into a new section and changes its control monitoring the trains and interlocking components. In the added and removed tracks, switches, crossings, level crossing can be there. In the schema, seven inputs are given and the state of interlocking is changed. In the input, first one is control identifier. The second is train identifier. Third and fourth variables are tracks added and removed from the moving block of the train. The fifth and sixth are sequences of switches and railway crossings. And the last one is a sequence of level crossings. The inputs are given in the first part of the schema and change in the state space is defined in the second part of it in terms of predicates on the variables and interlocking components.

$$
\begin{array}{l}
\textbf{UpdateRIS} \\
\hline
\Delta RIS \\
cid?: ControlId \\
tid?: TrainId \\
added?: \mathbb{F}\ Graph \\
removed?: \mathbb{F}\ Graph \\
sws?: \mathrm{seq}\ Switch \\
xngs?: \mathrm{seq}\ Crossing \\
lxngs?: \mathrm{seq}\ LevelCrossing \\
\hline
topology' = topology \land added? \subseteq topology \\
removed? \subseteq topology \land \mathrm{ran}\ sws? \subseteq \mathrm{ran}\ switches \\
\mathrm{ran}\ xngs? \subseteq \mathrm{ran}\ crossings \land \mathrm{ran}\ lxngs? \subseteq \mathrm{ran}\ levelcrossings \\
switches' = switches \frown sws? \land crossings' = crossings \frown xngs? \\
levelcrossings' = levelcrossings \frown lxngs? \\
\exists control: Control \mid (cid?, control) \in controls \land tid? \notin \mathrm{dom} \\
control\ .\ trains \bullet control\ .\ trains = control\ .\ trains \cup \{(tid? \mapsto \\
added? \setminus removed?)\} \\
\forall sw: Switch \mid sw \in \mathrm{ran}\ switches \\
\quad \bullet (\exists t1, t2: Track \mid (t1, t2) \in added? \\
\qquad \bullet sw\ .\ switch = t1 \land sw\ .\ switch = t2) \\
\quad \land sw\ .\ state = OCCUPIED \land sw\ .\ control = RIGHT \\
\quad \land sw\ .\ trainid = tid? \\
\forall sw: Switch \mid sw \in \mathrm{ran}\ switches \\
\quad \bullet (\exists t1, t2: Track \mid (t1, t2) \in removed? \\
\qquad \bullet sw\ .\ switch = t1 \land sw\ .\ switch = t2) \\
\quad \land sw\ .\ state = CLEAR \\
\forall xs: Crossing \mid xs \in \mathrm{ran}\ crossings \\
\quad \bullet (\exists t1, t2: Track \mid (t1, t2) \in added? \bullet xs\ .\ cross = (t1, t2)) \\
\quad \land xs\ .\ state = OCCUPIED \land xs\ .\ trainid = tid? \\
\forall xs: Crossing \mid xs \in \mathrm{ran}\ crossings \\
\quad \bullet (\exists t1, t2: Track \mid (t1, t2) \in removed? \bullet xs\ .\ cross = (t1, t2)) \\
\quad \land xs\ .\ state = CLEAR \\
\forall lx: LevelCrossing \mid lx \in \mathrm{ran}\ levelcrossings \\
\quad \bullet (\exists t1, t2: Track \mid (t1, t2) \in added? \\
\qquad \bullet (lx\ .\ levelCrossing = t1 \lor lx\ .\ levelCrossing = t2)) \\
\quad \land lx\ .\ levelState = OCCUPIED \\
\quad \land lx\ .\ occupiedBy = tid? \land lx\ .\ barriers = CLOSED \\
\forall lx: LevelCrossing \mid lx \in \mathrm{ran}\ levelcrossings \\
\quad \bullet (\exists t1, t2: Track \mid (t1, t2) \in removed? \\
\qquad \bullet (lx\ .\ levelCrossing = t1 \lor lx\ .\ levelCrossing = t2)) \\
\quad \land lx\ .\ levelState = CLEAR \land lx\ .\ barriers = OPEN \\
\hline
\end{array}
$$

Invariants: (i) The topology is unchanged. (ii) The added and removed tracks are subset of topology. (iii) The added switches, crossings and level crossings are in the domain of sequences defining these components in the topology. (iv) The switches, railway crossings and level crossings are updated by adding the new components by concatenation operator. (v) The control is added by adding the train in it. The moving block of the train is also updated at the same time. (vi) The states of all the switches are updated and defined as in occupied state. The identifier of the train occupying the switch is also recorded. The controls of the switches are shifted to the required direction. (vii) After the train has passed, the states of all the switches are made free to be occupied by any other train. (viii) The states of all the railway crossings are updated to be in occupied state. The identifier of the train occupying the crossing is updated in the railway crossing control. (ix) After the train has passed the railway crossings, the states of all the crossings are made free. (x) The states of all the level crossings are made in occupied state. The identifier of the train occupying the level crossing is recorded in the level crossing control. The barriers are closed to prevent collision between road traffic and train. (xi) After the train has passed the level crossing, the states of all the level crossings are made free for normal operation of road traffic by putting barriers in the open state.

## V. Model Analysis

Although formal specification has various advantages over the traditional specifications, however, any specification written in a formal notation does not mean it is correct and meaningful. The remarkable feature of formal specification is that it can be checked and analyzed for the presence of errors. The Z/Eves tool provides various exploration techniques for analyzing and reasoning the specifications written in the Z notation [15].

In this paper, Z/Eves is used in each paragraph for syntax, type and domain checking of the model. The syntax and type checking does not require to interact with theorem provers, however, it can be used incrementally such as each paragraph of a specification is written; it can be immediately checked and corrected if necessary. The domain checking was used to make sure that all expressions appearing in the specification are meaningful for which type checking does not guarantee. In domain checking each paragraph was examined as it was entered and each function application and definition description for meaningfulness was checked. Reduction commands available in Z/Eves to simplify and reduce the predicates were used. The reduction commands in Z/Eves traverse the current goal accumulating assumptions and performing reduction on the predicates and expressions of the specification in the goal.

Prove by reduction is one of the techniques used for analyzing the Z specification which provided a highest level of proof automation. It repeatedly applies as combination of other commands to reduce the goal until reduction has no effect. It was observed that the Z/Eves theorem prover is quite simple and user friendly. The complex proofs with high level automation can be carried out in a linear way. A snapshot of the Z/Eves tool for analyzing the specification is given in Fig. 1. In the figure, the symbol "Y" in the first and second columns of the window indicates that all the schemas are well written syntactically and proved automatically. Some schemas were proved with the proof assistance of the tool using some reduction techniques.

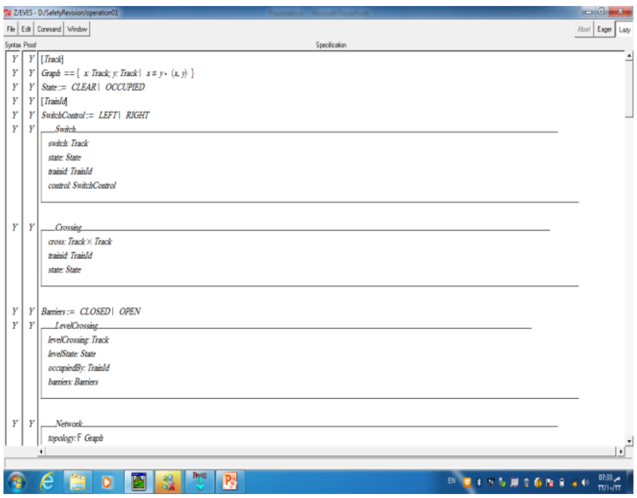

Fig. 01: A Snapshot of the Z/Eves Tool

## VI. CONCLUSION

An operational dynamic model for controlling railway interlocking system is proposed in this research. Initially, the critical components of the railway network topology are described then a state space analysis is provided. Based on the previous work of the author, the components are refined and integrated to define the simplified model of the system. The state space is enhanced by adding the moving objects that is trains. The controls to monitor the trains are also considered to define the entire system. Finally, formal dynamic model is proposed by defining local and global operations based on the definition of interlocking to guarantee safe and efficient operation of the system. The model is analyzed and validated using Z/Eves tool.

Few assumptions were taken in modeling of this system. For example, it was assumed that there does not exist any level crossing containing more than one railway tracks. It was supposed that no two switches or crossings are connected in the network topology. Further, it was supposed that a switch is not connected to any railway crossing. These assumptions were made for simplicity of the model. In the future work, these assumptions will be relaxed and a more general and detailed model will be proposed.

Application of formal methods in a safety critical system was one of the major objectives of this research because the power of formal methods cannot be realized when these are applied to simple cases. As moving block interlocking is an emerging technology therefore its modeling using formal techniques was another objective. Moving block interlocking is getting importance in railway industry and there does not exist any real operational formal model for it, which proves the originality of this research.

Although this work does not represent to a particular interlocking but it is useful for researchers interested in formal methods and applications because of the successful use of formal methods in terms of Z notation in a safety critical system. We believe this research will be useful for academic and railway industry because it focused on general concepts for modeling the system. This model can be implemented for any interlocking system. Finally, the specification is analyzed using Z/Eves tool.

## REFERENCES

[1] Akillilar, M., and Koppe, C.: Increase of Effectiveness and Efficiency in the Interlocking System, Signal and Draht, 100, (4), pp. 26-30, 2008.

[2] Bacherini, S., et al.: 'A Story about Formal Methods Adoption by a Railway Signalling Manufacturer', LNCS, Springer Berlin, 2006, 4085, pp. 179–189.

[3] Banci, M., and Fantechi, A.: 'Geographical vs. Functional Modeling by Statecharts of Interlocking Systems', Ninth International Workshop on Formal Methods for Industrial Critical Systems, 2004.

[4] Banci, M., and Fantechi, A.: 'Instantiating Generic Charts for Railway Interlocking Systems', Proceedings of the 10th International Workshop on Formal Methods for Industrial Critical Systems, 2005, pp.134-143.

[5] Banci, M., Fantechi, A., and Gnesi, S.: 'The Role of Formal Methods in Developing a Distributed Railway Interlocking System', FORMS/FORMAT, 2004.

[6] Bjorner, D.: 'The FME Rail Annotated Rail Bibliography', Department of Information Technology, Technical University of Denmark, 1998.

[7] Calame J. R., et al.: 'Testing of Hoorn Kersenboogerd Railway Interlocking', Canadian Conference on Electrical and Computer Engineering, 2006, pp. 620–623.

[8] Chevillat, C., et al.: 'Model-Based Generation of Interlocking Controller Software from Control Tables', LNCS, Springer Berlin, 2008, pp. 349–360.

[9] Chou, C.T.: 'A Formal Theory of Undirected Graphs in Higher Order Logic', 7th International Workshop on Higher Order Logic Theorem Proving and Application, 1994, pp.144-157.

[10] European Committee Electro Technical Standardization: 'Railway Applications Communications', Signalling and Processing Systems Software for Railway Control & Protection Systems, 2001.

[11] Foschi, U., et al.: 'The Role of Formal Methods in Software Procurement for Railway Transportation Industry', Symposium on Formal Methods for Railway Operation and Control Systems, Hungary, 2003.
[12] Hansen, K. M.: 'Validation of Railway Interlocking Model', Industrial Benefits of Formal Methods, 1994.
[13] Khan, S. A., and Zafar, N. A.: Improving Moving Block Railway System using Fuzzy Multi-agent Specification Language, International Journal of Innovative Computing, Information and Control, 7 (7), 2011.
[14] Masayuki, M., and Keisuke, B.: 'Application of Assurance Technology for Railway Signalling System', Eighth International Symposium, 2007.
[15] Meisels, I., and Saaltink, M.: 'The Z/EVES Reference Manual', TR-97-5493-03, ORA, CANADA, 1997.
[16] Nakamatsu, K., et al.: 'Intelligent Railway Interlocking Safety Verification based on Annotated Logic Program and its Simulator', International Conference on Networking, Sensing and Control, 2004, pp. 694 – 699.
[17] Peltekova, M., Bonev, T.: 'Method for Modelling Real Failures in Railway System', 6th Symposium on Formal Methods for Automation and Safety in Railway and Automotive Systems, 2007.
[18] She, X., Sha, et al.: 'The Application of Graphic Theory on Railway Yard Interlocking Control System', Intelligent Vehicles Symposium, 2007, pp. 883-887.
[19] Shigeyuki, O., et al.: 'Formal Verification of a Railway Interlocking System by the SPIN Model Checker', Research Reports on Information Science and Electrical Eng., Kyushu University, Japan, 2006, pp. 33–38.
[20] Simpson, A.: 'Towards Mechanical Verification of Moving Block Signaling Systems', Technical Report CMS-TR-99-06, School of Computing and Math. Sciences, Oxford Brookes University, 1999.
[21] Spivey, J. M.: 'The Z notation: A Reference Manual', Englewood Cliffs, NJ, Prentice-Hall, 1989.
[22] Wing, J. M.: 'A Specifier's Introduction to Formal Methods', IEEE Computer, 1990, 23 (9), pp.8-24.
[23] Winter, K., Robinson, N. J.: 'Modelling Large Railway Interlocking and Model Checking small ones', Proceedings of the 26th Australasian Computer Science Conference, 2003, 16, pp. 309–316.
[24] Xinhong, H., et al.: 'Improving Reliability of Railway Interlocking System with Component-based Technology', Journal of Reliability Engineering Association of Japan, 2006, 28 (8), pp 557-568.
[25] Zafar, N. A.: 'Modelling and Formal Specification of Automated Train Control System using Z Notation', IEEE, INMIC06, 2006.
[26] Zafar, N. A.: 'Formal Model for Moving Block Railway Interlocking System based on Un-directed Topology', International Conference on Emerging Technologies, 2006, pp.217-223.
[27] Zafar, N. A.: Formal Specification and Validation of Railway Network Components using Z Notation, Software, IET, 3, (4), pp. 312-320, 2009.
[28] Zafar, N. A., and Araki, K.: 'Formal Model for Moving Block Railway Interlocking System using VDM-SL', International Symposium on Information Science and Electrical Engineering, Japan, 2003.
[29] Zafar, N. A., and Araki, K.: 'Formalizing Moving Block Railway Interlocking System for Directed Network', Research Reports on Information Science and Electrical Engineering, Kyushu University, Japan, 2003.

**Nazir A. Zafar** is with College of Computer Sciences and Information Technology, King Faisal University, Saudi Arabia.
He was born in 1969 in Pakistan. He received his M.Sc. (Math. in 1991), M. Phil (Math. in 1993), and M.Sc. (Nuclear Engineering in 1994) degrees from Quaid-i-Azam University, Islamabad, Pakistan. He earned his PhD degree in Computer Science from Kyushu University, Japan, in 2004.
He has served at various universities and scientific organizations in Pakistan. He is on leave from University of Central Punjab (UCP), Pakistan and has a position of Full Professor in the Faculty of Information Technology. He is the founder and Chair of Formal Methods Research Group at UCP. His current research interests are modeling of systems using formal techniques, integration of approaches, reverse engineering, databases, safety critical systems, etc. He is an active member of Pakistan Mathematical Society.
Dr. Zafar has lectured at national and international level promoting use and applications of formal methods at academics and industry. He is a member of various organizations and technical committees organizing conferences and scientific activities, and reviewer/editor of various computer science international journals.