Formal Assurance of Signaling Safety:

A Railways Perspective

Shri Mahesh Mangal, General Manager/CORE/ALD

Shri Pallab Dasgupta, Professor, IIT/Kharagpur

Mahesh Mangal holds a BE degree in Electronics and Communication from MBM college, Jodhpur, and a M.Tech degree in Reliability Engineering from the Indian Institute of Technology Kharagpur. He has served as a Director/NP&M of RailTel corporation, as the Deputy Regional Manager of Indian Railways at Bangalore, and since April 2010 as the Senior Executive Director of Signaling at the Research Designs and Standards Organization, Indian Railways. Mr Mangal is currently a member of the Railways Board, Indian Railways.

Pallab Dasgupta holds a B.Tech, M.Tech and Ph.D in Computer Science & Engineering from the Indian Institute of Technology Kharagpur. He is a professor in Computer Science and Associate Dean (Sponsored Research) at IIT Kharagpur. His research interests include Formal Verification, Artificial Intelligence and VLSI. He has over 160 research papers and 3 books in these areas. He has collaborated with several industries as a consultant in formal verification, including Intel, Synopsys, General Motors, Freescale, Semiconductor Research Corporation and Indian Railways. He is a Fellow of the Indian National Academy of Engineering and a Fellow of the Institution of Electronics and Telecommunication Engineers, India.

ABSTRACT

The EN50128 guidelines of the European Committee for Electro-technical Standardization (CENELEC) recommend the use of formal methods for proving the correctness of railway signaling and interlocking systems. Considering the safety criticality of railway signaling, the potential benefit of formal safety assurance is of unquestionable importance, but the path towards implementing the recommendations is far from clear. The EN50128 document does not specify how formal assurance of railway interlocking may be achieved in practice. Moreover, the task of setting up an electronic interlocking (EI) equipment involves multiple parties, including the EI equipment vendor, the certification agency which certifies the resident EI software to be correct, and the end user (namely the railway service provider) who must configure the EI equipment with respect to the layout of the signals in the yard on which the EI is to be deployed. Considering the distributed nature of the development process, a feasible approach towards formal certification of the end product (post configuration) is not obvious. This chapter outlines the basics of formal verification technology and presents, from the perspective of the railways, a pragmatic roadmap for the use of formal methods in safety assurance of its signaling systems.

Keywords: railway signaling, electronic interlocking, formal methods, model checking, control table, temporal logic, relay logic, railway yard, route locking

INTRODUCTION

Railway signaling has been one of the most well studied safety critical systems for nearly two centuries. During this period, the notion of railway signaling has evolved in various ways, including the protocol for signaling, the technology used for implementing the signaling system, and most importantly the way in which safety guarantees are assured. The very early form of signaling relied on temporal separation of trains, which was primarily implemented by setting up time tables that ensured that two trains never shared a track at the same time. With the increase in railway traffic, it became necessary to divide the tracks into segments (or blocks), thereby giving birth to the notion of block signaling, where one or more trains can be on the same track, but on different blocks – which effectively means that the trains are spatially separated. Signals guard the entry of the block and implement the spatial separation.

Railway yards also have points where two tracks intersect. The point setting determines whether the train will continue to move on the same track or whether it will move into the intersecting track. A complex yard may contain

many points , with the tracks crisscrossing each other, which significantly increases the complexity of ensuring that a train can safely move from one track to another possibly passing through several intermediate tracks. In signaling parlance such complex passages are called routes. A railway yard can have hundreds of routes, and the signaling system must ensure that conflicting routes are kept free when a train is passing through a route. This is achieved by a mechanism called interlocking.

In the past railway interlocking was primarily manual. The hand-operated point levers shown in Figure 1 are reminiscent of the times when point positions were changed manually using such levers. The interlocking system, implemented using electrical relays, would ensure that signal aspects change only when points are in proper position.

Figure 1. A Point (left) and Hand-operated Point Lever (right)

In recent times, railways use electronic interlocking, where the signals and point positions are controlled by software running on an electronic interlocking (EI) system. Such computer controlled systems are designed to adhere to the signaling standards. Once configured properly with respect to a specific railway yard, they continue to work reliably over time and are not subject to human errors.

The primary concern with EI equipment is in determining with absolute certainty that the system configuration is correct, and that given a correct system configuration, the software is guaranteed to never err in signaling decisions. In current practice this verification task is performed using a high level of certification of the software and through rigorous testing of the configured equipment.

In spite of the best practices followed in configuring and testing of EI equipment, there are known instances of failures resulting out of human errors in the design of the logic for the EI equipment. The following two incidents may be highlighted:

- [The Milton Keynes Incident, Dec 29, 2008] As reported in (Railway Accident Investigation Branch Report, 2010), on December 29, 2008, a serious signaling error was detected at the Milton Keynes Central station on the West Coast Main Line in UK. The driver of a train observed a signal change from red to green, although the track beyond the signal was occupied by another train. An accident was averted by the driver, but the signaling error raised serious questions about the safety of modern EI systems. Investigations revealed that the error was due to incorrect configuration data for the EI equipment, namely that a track segment was missed in a specific part of the control logic. An even more intriguing fact which came out of the investigation was that interlocking data errors had in fact caused several other incidents in the past at Rugby, Glasgow Central, Peterborough, and Shenfield. In all of these cases, data errors had not been detected before the interlocking was commissioned.
- [The Cootamundra Incident, Nov 12, 2009] As reported in (Australian Transport Safety Bureau Report RO-2009-009, 2009), possible collision was averted by the driver of a train approaching No.1 Platform Road at Cootamundra, New South Wales, who noticed that the last wagon of a freight train was obstructing his approach. Investigations revealed that the track segment used for clearing the signal which allowed the train to approach was incorrect, namely that the data used to configure the EI system defined a wrong track segment for clearing the signal.

Such incidents reveal the uncomfortable fact that human errors in preparing the interlocking data may not be detected at the time of commissioning a EI equipment. The problem is that post-configuration testing may not sensitize the data error, particularly if the error is manifested only in very few corner case scenarios, say involving a very specific pattern of train movements in the yard. The commissioned equipment may then work perfectly for as long as providence does not create the scenario in which the error is manifested.

The problem of exhaustive testing hits what is known as the complexity barrier, that is, the number of possible test cases multiplies with every signal, every point, and every track segment. This combinatorial growth causes state explosion, that is, the number of states of the signals and points exceeds the capacity of any computer, and exhaustive testing becomes infeasible beyond that point. A deadline for commissioning the EI equipment further limits the number of tests that can be run. Therefore, most railways have their own well defined lists of tests, which are performed and then the equipment is commissioned. The above incidents show that such test plans are not adequate to cover all relevant scenarios.

The EN50128 guidelines (European Committee for Electrotechnical Standardization, 2001), issued by Comité Européen de Normalisation Électrotechnique (CENELEC), is the main reference for railway signaling equipment manufacturers in Europe, and also a widely referred standard in other railways around the world. The EN50128 document uses the concept of Software Safety Integrity Level (SSIL) to define the criticality of the components of a signaling system – specifically, for the software for railway control and protection systems. The SILs are: 4 (very high), 3 (high), 2 (medium), 1 (low), and 0 (not safety-related). While the guidelines do not contain any clear prescription of the software development methodology, they classify some of the commonly used techniques as forbidden, highly recommended, and mandatory.

One of the remarkable aspects of the EN50128 guidelines is in classifying a class of techniques, loosely called formal methods, as highly recommended for components with higher levels of SIL. Intuitively, formal methods are techniques for mathematically proving the correctness of software or a system, where the correctness requirement is formally specified using formalisms like mathematical logic. The basis of this recommendation lies in the realization that formal methods can comprehensively prove the correctness of a system using an arsenal of mathematical techniques, where as conventional testing fails to provide the desired level of assurance on those systems within feasible time. The EN50128 guidelines are however silent on how formal methods may be applied to arrive at better safety assurance for the high integrity software/system components, thereby posing a very pertinent research problem before the signaling and formal methods communities.

Any study on formal assurance of signaling safety must not only dwell upon the key technical challenges in verifying a configured signaling system, such as EI equipment, but must also focus on a deeper understanding on the commissioning process of such equipment. EI equipment is a sophisticated system, manufactured by only a handful of companies around the world. The software residing in such systems is a proprietary property of the EI equipment vendor. On the other hand, the yard specific data for configuring EI equipment is developed and provided by the signal engineers of the railway company. The vendor configures the equipment on the basis of the data received from the customer and then the configured equipment is subjected to rigorous testing by the end user. The software residing in the EI equipment is of the highest SIL, and may have been verified formally (to be compliant with railway signaling principles), but if the yard specific data has human induced errors, then the configured system will have errors.

The railway company has no means for formally verifying the configured equipment without having code-level access to the vendor's software. The vendor, on the other hand, has to rely on the data provided by the railway company and therefore all its assurances (formal or otherwise) are under the assumption that the data is correct. Therefore, the best intent of the railway company should be to ensure that the data provided to the vendor is correct, and that too with formal assurance, as recommended by the EN50128 guidelines. This chapter outlines the suggested steps towards achieving this goal, with important ramifications in ruling out future incidents similar to the Milton Keynes and Cootamundra incidents.

A PRIMER ON RAILWAY INTERLOCKING

This section outlines the traditional development approach for a railway interlocking system, and prepares the reader for the main focus of this article, namely the formal assurance of signaling safety. We begin with the definition of the key components of a signaling system, and then walk through the main steps in developing

the signaling logic.

Signaling Terminologies

The components of a signaling system and the terminologies used to define them are readily available from various sources. In this section we outline a small fragment of these definitions – only those that are used to explain the formal assurance approach.

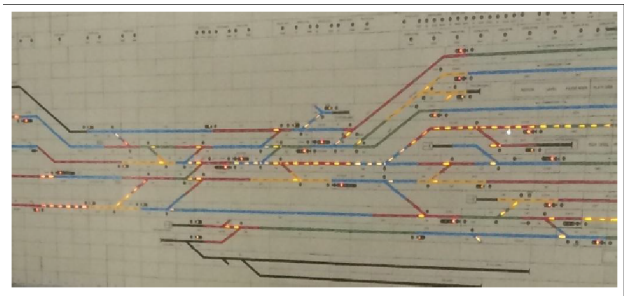

Figure 2. Section of a Control Panel

Figure 2 shows a typical control panel in a railway station. Observe the layout of the various lines and their intersections (points). The operator of this panel can issue route requests to the EI equipment, as we shall explain shortly, and the EI equipment allows the route to be locked only if conflicting routes are not locked at that time. In order to understand these key terms, let us walk through the steps in setting up a EI equipment.

We begin with a brief explanation of the layout of a simple railway yard from a remote village in India. Figure 3 shows the layout diagram of the yard. Trains may approach this station from both directions (left or right) along a single (bidirectional) line. At the line approaches the station from either side, it branches into three lines as shown in red. The upper and lower lines are called loop lines, which are adjacent to platforms shown in yellow. The middle line is called the main line. The two blue lines represent two level crossings located at the two sides of the station.

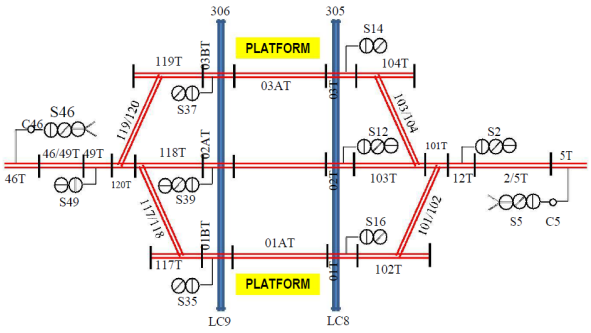

Figure 3. Layout Diagram of a Railway Yard

The main components of the layout from a signaling perspective are as follows:

1. **Track segments:** The tracks are divided into segments, and each segment has a label. In Figure 3 the labels ending with T represent track segments. The track segments have circuits to sense whether it is occupied by a train. Various types of track circuits are used in practice, but the purpose of all of them is to sense the presence of a train on the track. Note that a train may occupy more than one track segment at a time, for example, when it is passing from one segment to another. Track circuits provide inputs to the signaling system.
2. **Points:** The points are located at the intersections of two tracks. The position of a point determines the direction of the passage of the train. When the point is in normal position, the train continues on the same track; when it is in reverse position, the train moves to the adjacent track. For example, consider the point connecting segments 119T and 120T in Figure 3. When this point is in the normal position, a train approaching from 49T will pass towards 118T. On the other hand
   if the point is in reverse position, then a train approaching from 49T will change track from 120T to 119T along the point 119/120. Point positions are controlled by the signaling equipment.
3. **Signals:** The signals are asserted by the signaling system. In Figure 3 the signals are shown beside the tracks. The names of the main signals start with S (for example, S35). Since these are bidirectional tracks, there are signals on both sides of each track in Figure 3, but for a train moving in a particular direction, only the ones on its left hand side of the track are applicable. For example, the signal S35 is applicable only for trains leaving the platform segment 01AT towards 117T, and not for trains approaching the platform from the point 117/118. The signal C46 shown with signal S46 in Figure 3 is a special type of signal called calling-on signal. The calling-on signal asks an approaching train to pass a red signal of the main signal at a low speed. For example, when C46 may be used for calling-on a train even when S46 is red. Calling-on signals are used for various purposes, for example, to attach a coach at the end of an existing train.

4. Routes: Routes are sequences of consecutive track segments that define the path of a train from one part of the yard to another. For example in Figure 3, if a train approaching from the right has to be admitted on the upper platform, we need to clear the path from 5T to 03BT consisting of the track segments 2/5T, 12T, 101T, 103T, 104T, 03T, 03AT, 03BT. The point 101/102 has to be in normal position, and point 103/104 has to be in reverse position. The route is characterized by the starting and ending signals, which in this case are S5 and S37 respectively. Point positions may exclude one route when the other is locked. For example, when the route from S5 to S37 is locked, the point 103/104 is in reverse position, which excludes the route from S5 to S39, since it requires the same point to be in normal position. If two routes share one or more track segments, then they are said to be conflicting. For example, the route from S5 to S39 and the route from S12 to S2 are conflicting. The request for locking a route is made from the control panel of the yard, and it is the responsibility of the interlocking equipment to guarantee that conflicting routes are not asserted at the same time.

The locking of routes in a yard must adhere to the international railway signaling principles, which consists of a set of universally adopted rules. Any signaling system that follows these rules under all situations is guaranteed to be safe from signaling errors. In the EI equipment, adherence is ensured by the correct development of the application logic and the resident software that interprets this logic. We shall explain the development of this logic in the next section.

The Route Locking Logic

The basis for locking a route for the passage of a train is to guarantee that all track segments in that route and overlap are free, all points in that route and overlap are in proper position, and most importantly, the guarantee remains valid until the train has exited the route. In order to develop the logic for locking and releasing routes in a systematic way, a table called a control table or a route locking table is created.

Table 1 shows a portion of the control table of the yard shown in Figure 3. Each row of the table corresponds to a route in the yard. Each route is defined by the starting and ending signals of the

RT	MOVEMENT		IN THE ROUTE			OVERLAP			REMARKS
			POINTS		TRACK	POINTS		TRACK	
	FROM	TO	NOR	REV	CIRCUITS	NOR	REV	CIRCUITS	
R2	S2	-	-	-	2/5T	-	-	-	LOCKS S5
R5	S5	S39	101/102, 103/104	-	2/5T, 12T, 101T, 103T, 02/02AT	117/118, 119/120	-	118T, 120T, 39T	LOCKS S2, S12, S46, GATES 305, 306
R5b	S5	S37	101/102	103/104	2/5T, 12T, 101T, 103T, 104T, 03/03A/03BT		119/120	119T, 120T, 39T	LOCKS S2, S14, S46 GATES 305, 306
R12	S12	S2	101/102, 103/104		103T, 101T, 12T	-	-	-	LOCKS S5, S39
R14	S14	S2	101/102	103/104	104T, 103T, 101T, 12T	-	-	-	LOCKS S5, S37
R37	S37	S49	-	119/120	119T, 120T, 39T	-	-	-	LOCKS S14, S46
R39	S39	S49	117/118, 119/120	-	118T, 120T, 39T	-	-	-	LOCKS S12, S46
R46	S46	S12	117/118, 119/120	-	46/49T, 39T, 120T, 118T, 02/02AT	101/102, 103/104	-	103T, 101T, 12T	LOCKS S5, S39, S49 GATES 305, 306
R46 (a)	S46	S14	-	119/120	46/49T, 39T, 120T, 119T, 03/03A/03BT	101/102	103/104	104T, 103T, 101T, 12T	LOCKS S5, S37, S49 GATES 305, 306
R49	S49	-	-	-	46/49T	-	-	-	LOCKS S46

Table 1. A portion of the Control Table

route. For each route the control table specifies the points that must be locked in normal position, the points that must be locked in reverse position, and the track circuits that must be clear. The last column of the table highlights the conflicting routes and whether the level crossing gates need to be closed. There are various other aspects in the control table which are not shown here for simplicity.

It is important to introduce the notion of overlap at this stage. When a train approaches a signal, it may not be able to stop before the signal, because the distance between the signal post and the location at which the driver sees the signal may be too less for the driver to stop the train before the signal post. Therefore as a safety precaution, the railway signaling principles require the signaling system to clear the routes up to the next signal before clearing (making green/yellow) a signal. It also requires the point positions in the routes ahead to be locked in a way that no other train can come into those track segments. For example, consider the route R46(a) in Table 1, which starts from signal S46 and ends in signal S14 (see Figure 3). The overlap requires the point, 103/104, to be in reverse position so that the train can roll forward past signal S14. The track segments 104T, 103T, 101T and 12T must be free. Also the point, 101/102, must be in normal position to prevent a train from rolling past signal, S16, towards this direction. The overlap requirement does not apply for calling on signals (such as C5 and C46 in Figure 3) and this is useful for allowing more than one train to approach this specific yard. A called-on train must stop before the red signal (that is, no overlap is allowed for the called-on train).

The development of the control table is a highly safety critical activity, since this table forms the basis for developing the application logic. A busy railway station has more than a thousand routes, and it is close to impossible to manually guarantee that all entries in the control table are correct. Also the task of validating the application logic generated corresponding to a control table is a non-trivial task.

The software in typical EI equipment consists of the executive software which is factory installed, and the application software which is programmed for every yard. The application software represents the yard specific logic, which is interpreted by the executive software while regulating the signals and points. The application data prepared by the end user (namely, the railway company) is used to define the application software.

The application data consists of a set of Boolean equations that define the signaling logic for the yard. For historical reasons, most types of EI equipment continue to accept a form of logic called relay ladder logic. In order to ensure that the failure of a signal or a point always leaves the system in a fail-safe state, traditionally railways around the world have been using different types of relays which guarantee such fail safe operations. The logic which determines when a relay will be energized (or de-energized) is developed in the form of ladder logic, which became popular because of its visual simplicity and verifiability through visual inspection.

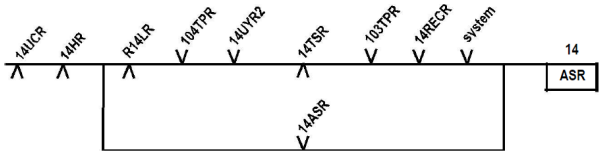

Figure 4. An illustration of Ladder Logic

Figure 4 illustrates the notion of ladder logic in the context of the yard of Figure 3. ASR relays are used to lock routes. Here the logic is shown for the relay, 14ASR, which is responsible for locking the route from the signal S14 to the signal S2. The route is locked when the ASR relay is de-energized. The logic shows that the 14ASR relay can be energized by a combination of the relays shown in the upper rung of the ladder. The relays shown with the ? symbol must be energized, while the relays shown with the ? symbol must be de-energized for the 14ASR relay to be energized. Some of these relays may have ladder logics of their own. For example, the logic of 14UCR guarantees that the points 103/104 and 101/102 are locked in their appropriate positions, and this is necessary for the route represented by 14ASR to be locked.

The lower rung of the ladder has the 14ASR relay itself, which allows the relay to hold its state once it is energized. The ability of a relay to hold its state is an important feature of relay logic, since it shows that relay logic describes sequential circuits with an underlying state transition system. In a real yard, the ladder has many more rungs, which allows the relay to be energized in other

ways. It may be noted that 14ASR can be de-energized (say) by energizing the relay14UCR.

The ladder logic can be expressed in Boolean logic as well. For example, the logic of Figure 4 is equivalent to the following Boolean function for the next state of 14ASR:

```
~14UCR * ~14HR *
        ( 14ASR +
                (~R14LR * 104TPR * 14UYR2 *
                        ~14TSR * 103TPR * 14RECR * system ))
```

The primary focus of this chapter is to outline how the application logic developed by the signal engineers can be proven to be correct using formal methods, so that data errors that can potentially escape testing are ruled out before the commissioning of the logic. It may be prudent to point out that formal verification of the application data should not be viewed as a measure for replacing the testing of the EI equipment after it is configured. This is because, even if both the executive software and the application software are proven to be correct, we still have to make sure that the EI equipment has been correctly integrated into the yard, namely that the relays have been correctly connected to the EI equipment. The goal of formal methods in this case is to rule out data errors prior to commissioning, and understanding that in its entirety is our goal in this chapter.

FORMAL VERIFICATION

The notion of formal verification has its origin in proving the correctness of programs. In general the domain of the inputs to a program need not be finite, and hence a software program is typically tested using only a subset of the possible input combinations. One of the most fundamental results in Computer Science establishes that in general the task of deciding whether a program is correct for all input scenarios is undecidable, that is, it is mathematically proven that no algorithm can read a program and decide whether it is correct for all inputs. This is not true for the special class of programs where all variables have finite domains, because finiteness allows us to enumerate all input combinations and prove the correctness in each of them. For most systems of moderate complexity however, the theoretical possibility of enumerating all input combinations is not practically possible within feasible limits of time, and hence the objective is to prove the correctness of the system without explicit enumeration. Formal verification techniques achieve this objective using an arsenal of smart abstractions and intelligent decision procedures that work on succinct representations of large state spaces.

Formal verification encompasses several types of approaches – in terms of its objective, and in terms of the methodology used to achieve that objective. For example, equivalence checking and model checking are two popular notions associated with formal verification. An equivalence checking problem consists of the design-under-test and an abstract golden model, and the task is to determine whether for every input sequence the output of the design-under-test is identical to that of the golden model. A model checking problem consists of the design-under -test and a set of formal properties, and the task is to determine whether the properties are satisfied for every execution of the design-under-test.

The railway signaling principles define the safety properties that a signaling logic must satisfy at all reachable states of the signaling system. This verification task has been attempted in the past using several types of formal methods as discussed in the next section, which work with various degrees of user intervention. The approach that we advocate in this chapter is a fully automated one, namely one that uses automated model checking techniques.

Figure 5 illustrates the model checking approach. The design-under-test (DUT) represents the logic that we wish to verify. In the context of the signaling validation problem, our DUT is the application logic developed by the signal engineer. A model checking tool has a front-end pre-processor that extracts a finite state machine representation from the DUT. The formal properties represent the specification – we shall explain how temporal logic can be used for this purpose. In our case, formal properties are gleaned from the railway signaling principles as interpreted on the specific yard. The core model checking engine receives the formal properties and finite state representation of the DUT as inputs and determines whether the formal properties are guaranteed by the DUT. If any property fails in the DUT under some scenario,

then the model checker shows a specific scenario under which the DUT fails to satisfy that property, namely, it finds a counter-example scenario.

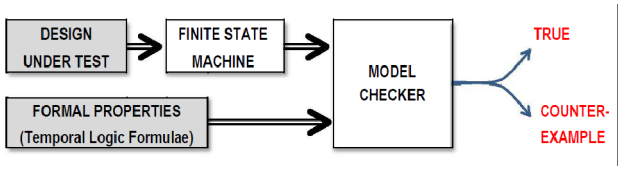

Figure 5. Model Checking

Let us consider a principle that says: A signal is not cleared (made green/yellow) until it is proven that the track circuits in the route up to the next signal and its overlap are clear. If we interpret this principle for the signal S14 in Figure 3, then we wish to prove that the signaling logic never clears S14 until tracks 104T, 103T, 101T, 12T are unoccupied. This property can be formally

LTLSPEC G(X 14HR ⇒ 104TPR **&** 103TPR **&** 101TPR **&** 12TPR)

specified in terms of the relays involved as:

The relay, 14HR, is the relay controlling the signal S14. The relay, 104TPR, indicates whether the track segment 104T is unoccupied (the other TPRs are similarly defined). It may be noted from Table 1, that the route R14 does not have any overlap tracks.

The above property is expressed in Linear Temporal Logic, which extends Boolean logic with temporal operators. This property uses the standard Boolean operators like & (logical AND) and ~ (negation), and two temporal operators, namely the G operator and the X operator. The meanings of these operators are intuitively described as follows:

- A property Gϕ is true in a finite state machine, if and only if the property ϕ is true at all reachable states of the machine. Thus, the G operator (also called the always operator) is used to define properties that are globally true.
- A property Xϕ is true at a state, if and only if the property ϕ is true at all next states of the state. Thus, the X operator (also called the next-time operator) is used to specify what must hold in the next state of the system.

The above property specifies that a prerequisite for 14HR to be high at a time (that is, for the signal S14 is clear) is that the TPR relays for 104T, 103T, 101T, and 12T must all be high. The property is violated if we reach a state where the following holds (where the symbol, **|** , represents the logical OR operator):

(X 14HR) & (~104TPR | ~103TPR | ~101TPR | ~12TPR)

The above expression can become true in broadly two ways:

1. 14HR goes high in some time point, even though the prerequisite condition (namely the conjunction of the TPR relays) is not true. This is in direct violation of the signaling principle stated earlier.

2. 14HR and the TPR relays are high, but one (or more) TPR relay goes low before 14HR becomes low (that is, a track circuit fails before S14 becomes red). This is also in violation of a signaling principle.

A formal proof must guarantee that the above expression is not satisfied in all reachable behaviors of the signaling system.

Traditional testing will typically examine whether 14HR is energized when the track segments represented by the above TPR relays are unoccupied. On the other hand, formal verification aims to prove that the 14HR relay is not energized in any other circumstance. It may be noted that the relay logic for 14HR is not directly defined in terms of these relays, but indirectly through several steps involving other relays (such as 14ASR, which was defined earlier). If the logic for any of those relays has an error, it may be manifested through a failure of the above property under some peculiar sequence of train movements. Formal verification helps in finding such counter-example scenarios if they exist.

FORMAL VERIFICATION IN RAILWAY SIGNALING

Railway signaling has been a problem of interest to the formal verification community for many years, because of its safety critical nature and its legacy of being treated as a logic design problem. However the adoption of formal methods in actual practice has been scarce, even after the

publication of the EN50128 guidelines which has a clear recommendation for the use of formal methods in SIL4 components. This is primarily because the path towards adoption of formal methods in not well understood by the railway companies and their vendors, and secondly because the investments needed to set up the tools and best practices for this purpose have not been taken up.

Formal verification methods have been researched in the railways context broadly for interlocking systems and train control systems. While interlocking systems control the movement of trains by operating the signals and points along railway tracks, train control systems aim to control the movement of trains directly based on the knowledge of the position, velocity and other parameters of the train. For example, in driverless metro railways, signals have no purpose and trains have to be directly controlled. Train control systems and interlocking systems may coexist, and in such cases the driver may sometimes be overridden by the train control system (for example, a train may brake automatically when an approaching signal is red and the train is travelling too fast). One of the primary references for train control systems is the ETCS standards (European Train Control System, 2002).

Several modeling and formal verification techniques have been studied for train control protocols and subsystems. This includes the use of formalisms like Petri nets (Myer zu Horste & Schnieder, 1999), (Zimmermann & Hommel, 2005) and state charts (Damm & Klose, 2001).

Railway interlocking, which is the main focus of this chapter, has been extensively studied by the formal verification community. A recent survey on the use of formal methods in railway signaling can be found in (Fantechi, Fokkink, & Morzenti, 2012). For example, researchers have studied the problem of gleaning the precise and complete set of yard specific rules from the railway interlocking principles (Fokkink, 1996).

As discussed earlier, EI equipment has executive software, which is the factory set software for driving the signaling system, and application software which is configured in a yard specific manner from the application data/ logic provided by the signaling engineers. For a given application data, the guarantee that the EI equipment will faithfully drive the signals as per the specified logic, must come from the EI equipment vendor. For this purpose, manufacturers of EI equipment have considered a wide variety of formal approaches, some of which can be found in (Fringuelli, Lamma, Mello, & Santocchia, 1992), (LeGoff, 1996), (Bernardeschi, Fantechi, & et. al., 1998). Model checking techniques have also been attempted, for example, (Cimatti & et. al., 1998), (Eisner, 2002). An approach based on theorem proving for verification of signaling data can be found in (Morley, 1993).

Formal Verification of Application Data/ Logic

Research on formal verification of railway signaling has primarily focused on the verification task of the EI equipment vendor. It is important to understand the benefits of the formal verification techniques for the end user, namely the railway company. We summarize the benefits under two broad categories:

1. The model can be validated against yard specific properties generated from layout based on railway signaling principles. This helps in eliminating data errors before the logic is passed on to the EI equipment.
2. The model can be used as the scientific basis for generating test scenarios for the configured equipment. When a yard is modified, the EI equipment has to be reconfigured. In a fully operational yard, the downtime resulting out of the reconfiguration has to be within tolerable limits, and hence the level of testing done prior to commissioning EI equipment cannot be repeated for every modification. Formal analysis can help in prioritizing the test scenarios.

The norm of every verification problem in engineering design is to ensure that the task of verification is carried out independent of the design team, so that the logical bias of the design team (and the logical errors thereof) does not get carried over to the verification team. Figure 6 shows the design tasks and the verification tasks in the mandated flow for commissioning EI equipment. The tasks shown inside blue clouds are the ones that are performed today under the following responsibilities:

1. Rigorous Logic Development. This task is performed by the signal engineers in a manual/ semi-automated way. The kinds of practices followed vary from one company to another.

The logic so developed may have redundancies, may be divided into multiple EI equipment, and may be represented in various functional forms. The internal steps of this task, which may involve the development of control table by the signal engineers, can be found in the manuals of individual companies and are not discussed here. We shall refer to this task as the design phase.

2. EI Equipment Configuration. This task is typically done by the EI equipment vendor, possibly with participation from the end user. This step may involve verification of various forms (including formal methods) which are performed by the EI equipment vendor. Since the executive software is typically proprietary in nature, the railway company has no role in the verification performed under this task.
3. EI Equipment Testing. The testing of the EI equipment is the responsibility of the end user, namely the railway company. The railway company has well defined procedures for creating the yard specific test plan and executing the tests on the configured equipment. This phase is a very critical phase, since this is the last step before commissioning the equipment. We shall refer to this task as the testing phase.

In some cases, the application data may be simulated for verifying its logical correctness. It is infeasible to achieve 100% coverage of scenarios in simulation or testing, and this is the reason why there are numerous instances where data errors have made their way into the commissioned equipment.

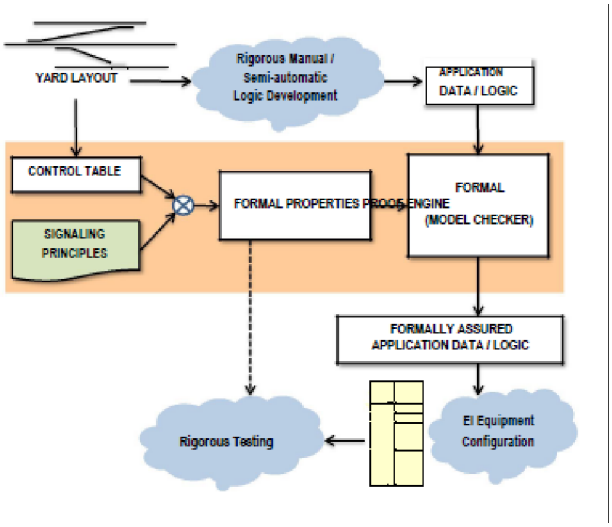

Figure 6. Formal Assurance of Application Data: Railway's perspective

Our prescription for railway companies is to introduce the tasks shown on the saffron banner between the design and testing phases. This is a fully automated flow and can achieve 100% coverage in validating the application data in a matter of few minutes. The main tasks in enabling this flow are:

1. Generating a complete sound and complete suite of yard specific formal properties using the governing railway signaling principles. We shall explain this step in the next subsection. We shall refer to this task as specification development.
2. Reading the application data / logic developed by the signal engineer and extracting a finite state machine (FSM) model from the design. In keeping with traditional practices, many railways continue to use legacy formalisms for expressing the logic (such as ladder logic), and it is fairly straight forward to extract the FSM from such representations.
3. Developing a model checker that can scale to the size of logic entailed by large complex yards. This step requires some insights into the nature of the properties to be proven, and tailoring the model checker for such properties. Elaborating this step is beyond the technical scope of this article.

Before we elaborate the task of specification development, it is important to point out that the properties generated in this step can also be used to ascertain whether relevant test scenarios have been exercised during testing. In other words the property suite also acts as a metric for coverage analysis during the testing phase. This is a very important when partial test coverage is achieved following logic updates after incremental changes in the yard.

Formal Specification Development

Specification development is the task of interpreting the railway signaling principles on a given yard and capturing the resulting requirements through formal properties. The signaling principles are almost the same in all railways and govern the general set of rules that must be followed by any signaling system. The signaling principles are independently published by different countries – for example, see (Paul Woolford, 2003) for the British standards, (IRISET, 2009) for the Indian standards, and

(Warwick Allison, 2013) for the principles followed by RailCorp in Australia.

The signaling principles are comprehensive documents covering all aspects of signaling and safe movement of trains. The principles start with a set of broad requirements, and then present further details on how these requirements are expected to be achieved. For example, the following is an excerpt from one of the earliest versions of the Indian Railways Signaling Principles:

1. Lever frames and other apparatus installed for the operation and control of signals, points etc. must be so interlocked and arranged as to comply with the following essentials:
   a) It must not be possible to take 'OFF' a running signal unless the following are ensured not only in the actual portion of the route on which the train has to travel, but also in the overlap:
      i. All the points are correctly set.
      ii. All the facing points are locked (at site).
      iii. All the interlocked level crossing gates are closed and locked against road traffic.
      iv. The isolation is ensured.
   b) Once the signal has been taken OFF, it must not be possible to do any of the following unless the signal has first been put back to the 'ON' position:
      i. Alter the position of the relevant points.
      ii. Unlock the relevant facing points.
      iii. Unlock and open the relevant level crossing gate.
      iv. Disturb the relevant isolation.
   c) It must not be possible to take 'OFF' at the same time any two fixed signals, which may lead to any conflicting movements.
   d) Where feasible, the points shall be so interlocked as to avoid any conflicting movement.

The above statements provide the guidelines for route locking, route holding as well as route release. Each of these subtasks has more intricate requirements in terms of how the above guarantees are met. In this chapter we consider the signaling principles for route locking to illustrate the task of formal specification development.

Let us consider the route R14 (that is, the route from signal S14 to S2) of the yard shown in Figure 3. We shall highlight some of the properties that are relevant for this route. Note from Table 1, that the conflicting routes for R14 are R5, R5b and R37. The following signaling principles are considered here:

Prove that interlocking is free	We need to prove that all conflicting routes with the same point setting are free. At the logic level this translates to checking that the LR relays for these routes are in the de-energized state. The following temporal properties are generated when this is interpreted for route R14: G(X R14LR ⇒ ~R37LR) G(X R14LR ⇒ ~R5bLR)
Prove that all points in the route, isolation and overlap are operated to the required position, locked, and detected	One way to verify this requirement is to check whether the point position is supported by some route's LR relay. For example, we have the following properties for the points 101/102 and 103/104: G(X 101_102NCR ⇒ R14LR | R5bLR | R12LR | R5LR) G(X 103_104RCR ⇒ R14LR | R5bLR) The NCR relay for a point is energized when the LR relay of any route that requires the point in the normal position is energized. Likewise, the RCR relay for a point is energized when the LR rely of any route that requires the point in the reverse position is energized. Note that we have only described some of the routes in Table 1, and therefore, the above properties are only partially complete.
Prove that the selected route, overlap and isolation is locked	A route is locked when its ASR relay is de-energized. The UCR relay is energized when all points in the route and overlap are detected to be in the required position. This requirement may be verified for route R14 by checking the following property: G (X~14ASR ⇒ 14UCR)
Prove that the track circuits in the route up to the next signal and its overlap are clear	This may be verified by checking whether track occupancy has been correctly considered in the HR logic. For example, for route R14, we have the following property: G(X14HR ⇒ 104TPR & 103TPR & 101TPR & 12TPR)
Clearing of signals	Before clearing the signals (that is, making them green/yellow), we check whether the points have been locked. For route R14, we have the following property: G(X14HR ⇒ ~101_102WLR & ~103_104WLR)

We have highlighted only some of the requirements. In reality, a yard has many other types of routes and objects, including shunt routes, calling-on routes, level crossings, etc., and there are specific signaling principles for each of these.

Most railway companies have their versions of signaling principles, all of which are broadly in agreement in terms of safety requirements. Following the EN50128 guidelines, it becomes the responsibility of the company to develop formal safety rules governing the application logic and proving that the application logic is bug-free. Indian Railways in collaboration with the Indian Institute of Technology Kharagpur has established the broad outline and proof-of-concept for the approach presented in this chapter. It is envisaged that initial investment in this area will translate to more structured verification practices and compliment the testing procedures for the configured EI equipment.

The properties illustrated in this section are called safety properties in formal verification parlance. Intuitively, safety properties are those that assert that the bad behaviors of the system

will never happen. For example, consider the following property:

$$G(X14HR = 104TPR \,\&\, 103TPR \,\&\, 101TPR \,\&\, 12TPR)$$

This property states that a state where 14HR is high (energized) will never be reached from a state where one or more among 104TPR, 103TPR, 101TPR, 12TPR are low. It does not say that 14HR has to be asserted when all the above TPRs are high.

Safety properties are only responsible for assuring the safety of the signaling system. The signaling principles, on the other hand, are not only concerned with safety, but also with progress. For example, a signaling system which keeps all signals red at all times is safe, but it is not useful. Signaling principles also require that progress in passage of trains happen under enabling circumstances. When trains are operated manually, the guarantee of progress involves a human being, and therefore the guarantee of progress must make some assumptions about the admissible behaviors of the driver. In formal verification parlance, such assumptions are also captured using formal properties which are called fairness constraints or assume properties.

On the other hand, the train control system must guarantee the progress when trains are operated under automatic control (as in driver-less metros). As various degrees of automated control gets adopted by various railway companies around the world, what is needed is a unified standard for railway signaling and train control, and formally laid out rules governing the safety and progress in such unified systems. This will help in better integration of formal verification practices in the next generation signaling system design flow.

CONCLUSION

Safety and efficient utilization of resources are the primary concerns in railway signaling. Efficient utilization aims towards facilitating movement of trains in and out of a yard with minimum waiting time. Safety criteria aim to introduce appropriate constraints on the movement of trains so that accidents are avoided with adequate margin of error from the human elements involved in the system. As we move into the modern era of automated control, some of the archaic signaling principles will have to be revised and made more specific and non-ambiguous, so that the ideal balance between efficiency and safety is achieved. Formal methods have an important role in this direction, since both safety and progress criteria can be proven using formal methods. Eventually, the end users, namely the railway companies will have to assume leadership on this subject.

In the present system, guaranteeing the correctness of the application data is an important task for the signaling engineers. For large and complex yards it is very difficult to conceive all possible sequences of train movements in the yard, and the task of achieving adequate coverage of scenarios through simulation and testing has become infeasible in practice. As a consequence, in spite of the best practices of the railway companies and their vendors, incidents like Milton-Keynes and Cootamundra have taken place in not too distant past. Formal verification of the application data, as professed in this chapter, is not only feasible in practice, but is of great value in ruling out data errors.

Having a formal model of the signaling system also opens up other interesting possibilities. We already mentioned the benefits towards formally assisted identification of relevant test scenarios for incremental testing following (minor) changes in the yard. Formal properties may also be monitored online over configured EI equipment for enhanced debugging capabilities. In conjunction with data-loggers (also called historians) which record the signal aspects and movement of trains, monitors can be used to catch violations of assume properties (for example, abnormal behavior of the driver or the train) and provide real-time warnings to the entities on the approach path of the train. All these point to the benefits of the initial investments in developing a tool flow for formal capture and validation of signaling properties.

KEY TERMS AND DEFINITIONS

Formal Verification: *A verification methodology for mathematically proving the compliance of a design against a formally defined logical specification.*

Interlocking: *A procedure for ensuring separation of trains in a railway yard through acquiring locks on signals, points, and track segments on a route.*

Model Checking*: A formal decision procedure for proving a given formal specification on a design implemented as a state machine.*

Railway Signaling Principles*: Internationally adopted guidelines governing signaling rules and*