This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Automatic Generation of Route Control Chart From Validated Signal Interlocking Plan

Arindam Das, Manoj Kumar Gangwar, Devleena Ghosh, Chittaranjan Mandal, *Senior Member, IEEE*, Anirban Sengupta, and M. Mubashshir Waris

***Abstract*— Railway signalling is a complex and safety critical problem that has been extensively studied and standardised over a long period of time. The signalling equipment is typically procured from standard vendors and configured with yard specific application logic for which the route control chart (RCC) is a key input. As the yard size increases, the number of routes also increases and accordingly the difficulty of RCC preparation increases rapidly. RCC preparation for big yards may take months and affects the project deadlines adversely. In this work we report computational procedures to address: *a*) capturing of signal interlocking plan (SIP) given on paper and storing it using suitable data structures, *b*) generating the RCC automatically from the captured SIP supported by procedures based on graph theoretic formulation, *c*) storing the SIP graphically in memory, *d*) application of formal methods towards validation of yard structure and *e*) generation of temporal logic properties for formal verification of electronic interlocking (EI) logic. Several important steps towards RCC generation, such as conflict identification and isolation determination and also validation and verification (V&V) covering yard layout and safety property generation based on graph theoretic modelling are the most interesting aspect of this work. The described techniques have been tested successfully on many actual yards.**

***Index Terms*— Railway signalling, control table generation, signal interlocking plan (SIP), route control chart (RCC), yard layout validation, validation and verification (V&V).**

I. Introduction

A KEY step in the design of interlocked signalling systems for railways is the capture of the signal interlocking plan (SIP) and then generating the route control chart (RCC) which lists the routes which will support train movement in the yard. These are safety critical tasks. The RCC also contains a number of useful information which are needed to generate the logic for electronic interlocking (EI) and panel interlocking (PI) systems which control the various yard elements such as signals, points, level-crossings to enable safe movement of trains. RCC generation is an intricate process whose steps need to be well understood to automate the process, both in the interest of safety and quick turnaround, especially important for yard signalling modification. The primary inputs for generating the yard specific application program data for an EI are the front panel diagram (FPD), SIP and the RCC. The SIP broadly describes the layout of the different tracks, the points, the signals and the level crossings. The RCC defines the different routes for train movement in the yard, indicating the signals that must be asserted, the required positions of the points and the other routes that conflict with the given route. The RCC forms the basis for developing the EI application data for the yard. The inputs to the EI are the positions of the trains in the yard (as sensed through the track circuited portion (TCP) of tracks and axle counters) and the route requests asserted by the user of the EI. The role of EI is to operate the yard based on the logic designed from the yard RCC to guarantee that the various signals and points are asserted in a manner that is compliant with the railway signalling principles so that there is no conflicting movement and maintenance of safe-distances (overlap in case the train overshoots the signal) is ensured. The important aspects of this work towards RCC generation are the graph based representation of the captured SIP and creation of functions to support important sub-tasks such as: route, overlap and isolation point enumeration. Subsequently, newer requirements that may arise; one such is presented in this paper to illustrate how the existing algorithms can be augmented to handle the new requirement to support a junction. A tool named LayoutEditor (LE) was developed to capture a given SIP (provided by Indian Railways (IR)) and generate routes and then the RCC for the captured yard. There are two main components of this tool. The first one is the graphical user interface (GUI) for capturing the SIP. The other part houses the back-end algorithms used for the route, overlap, isolation point, and conflicting route enumeration to generate RCC. Validation of yard layout is important for safe operation of a yard. The graph-based representation has been leveraged to achieve this using formal methods (model checking) where desirable or undesirable yard properties are expressed as temporal logic formulae. While all the sub-tasks described here are relevant, route conflicting determination, isolation determination, formal validation of yards, extension

Manuscript received May 7, 2019; revised November 3, 2019 and March 12, 2020; accepted April 21, 2020. This work was supported by the Research Designs and Standards Organisation (RDSO), Indian Railways and DST-SERB under Grant IMP/2018/000471. The Associate Editor for this article was V. Punzo. *(Corresponding author: Chittaranjan Mandal.)*

Arindam Das, Devleena Ghosh, and Chittaranjan Mandal are with the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, Kharagpur 721302, India (e-mail: chitta@iitkgp.ac.in).

Manoj Kumar Gangwar is with the Department of Advanced Technology Development Centre, Indian Institute of Technology Kharagpur, Kharagpur 721302, India.

Anirban Sengupta is with South Eastern Railways, Indian Railways, Kolkata 700043, India.

M. Mubashshir Waris is with the Research Design and Standards Organisation, Indian Railways, Kolkata 700043, India.

Digital Object Identifier 10.1109/TITS.2020.2993794

of the techniques to handle junction yards and algorithm analysis of various tasks are the most interesting aspects of this work. Efficiency aspects are important because the tool is meant to be used interactively for large yards. The rest of the paper is organised as follows: in section II, the mechanism for capturing the SIP and representing it in graph form is discussed; the back-end algorithms required for RCC construction are described in section III; results of experimentation on several IR yards of South Eastern Railways (SER) zone are presented in section VI; in section IV; an extension of these algorithms is presented to handle a situation not described earlier in the paper to illustrate how the back-end algorithms can be extended to handle the newer situation, in this case handling of junction yards with complex train movements. V&V using formal methods is presented in section V. Related work in the context of the work already is presented in section VII. The paper is concluded in section VIII.

II. SIP Capture and Representation

The LE provides a simple GUI for the user. Using this GUI the user can draw a yard layout having different signalling components. The tool uses internal data structures to store the geometric data of the yard which are then used for further processing. Based on this data, the back-end of the tool can enumerate all the possible routes. The GUI permits selection of appropriate routes and also annotate them, as per requirements of railways. The selected routes can then be used to generate RCC for the yard. In the following, we describe the features of the tool in more detail.

A. Graphical User Interface of Tool

The GUI is developed based on the rationale of a free-hand drawing tool. It is developed using Java Swing as it provides ready to use components for GUI development. Through the drawing canvas (gridded), the user can draw different signalling elements. Zooming and panning are available for handling large yards. There is a palette of yard elements, such as: tracks, signals and level-crossings can be drawn or placed on the drawing canvas. These get connected to other elements already on the canvas based on a proximity threshold. While constructing the yard through the GUI, track segments have to be drawn. Those are automatically abutted to nearby track segments if the placement is close enough. Similarly, signal placement is permitted only in the close vicinity of track segments and a placed signal gets aligned to the proximal track segment. Similarly, points must be placed in the vicinity of two track segments. For convenience, the co-ordinate space is discretised with a high enough precision so that the SIP components can be placed without any perceptible impairment of placement information. Proximity detection to placed track segments is achieved through range searches. A convenient data structure for range searches is the extended AVL tree where each range (the horizontal extent of a track segment) is stored in an AVL tree node as the key, along with the right extent. Efficient range searching is especially useful for large yard.

B. Main Yard Elements

The main yard elements available in the LE for capturing a given yard layout are tracks, points (double ended), level-crossings, axle counters, independent and dependent shunt signals, main signals (2 or 3 or 4 aspects main signals are placed either separately or with associated combination of shunt, and calling-on), four types of trap points and 4-types of derailment points (DP) based on the direction of the yard, stop signal (SS), stop board (SB), sand-hump, hand plunger lock (HPL), dead-end and level-crossings (LC).

C. Graph Representation of Yard Layout

It is necessary to represent the association and connectivity between various tracks, points and signals. It is also necessary to indicate TCP (track node) of tracks. If a TCP has a signal placed on its side, the corresponding signal name is tagged onto the TCP node as a signal attribute; such a TCP with a signal attribute is called a *SignalNode*. Similarly, if a TCP has a point placed on it then the node associated with that TCP is treated specially and marked as *pointNodes*. For this purpose, the yard layout can be represented as a directed labeled graph as shown in Figure 2. This representation is for the yard shown in Figure 1 where the vertices represent various yard components, for example, the vertices can be of classified as tracks, points, signals, level crossings. An edge between two vertices is present if the corresponding yard components are adjacent to each other. The edge label indicates the direction of train movement i.e. up (UP) and down (DN) directions so that a path following a certain direction between two vertices representing valid signal components will give a valid route in that direction.

III. Backend Algorithms

In this section algorithms for route and overlap enumeration, algorithms for identification of isolation points of each route and conflicts between routes are presented. Complexity analysis of these are also carried out. Thereafter RCC generation is elaborated.

A. Route Enumeration

To enumerate all possible routes of the yard we need to find paths in the graph between any pair of TCP nodes associated with same direction signals. A route is defined as a path in the yard layout graph originating at a *SignalNode* and terminating at another *SignalNode* of the same direction and having no *SignalNode* of the same direction in between. To identify routes originating at a *SignalNode*, a depth-first traversal of the yard graph is initiated at that *SignalNode*, progressing monotonically in the direction of that signal. Termination happens on encountering the next *SignalNode* in the same direction (with successful identification of a route) or when a TCP node is reached with no successor (with failure).

B. Overlap Enumeration

A train can move on a path from the entry to the exit signal with adequate overlap, once the route is selected.

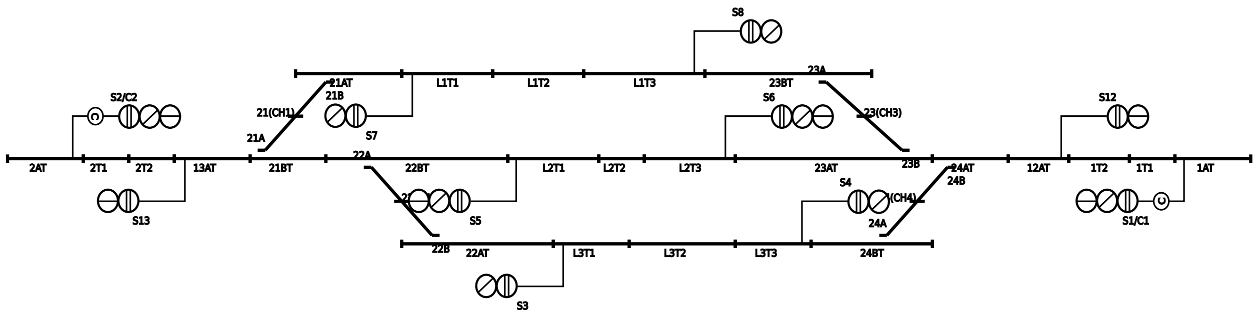

Fig. 1. Captured yard layout.

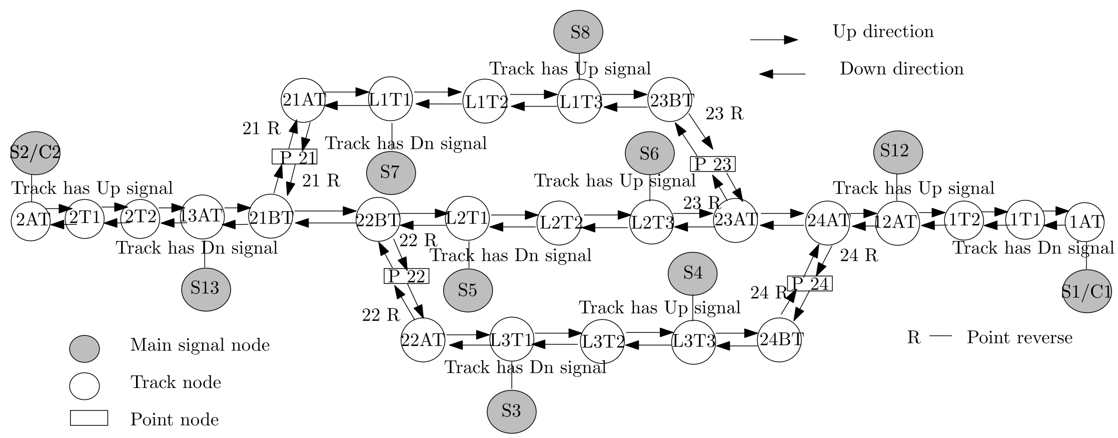

Fig. 2. Graph representation of the captured yard layout.

In the interest of safety, when a train might overrun the exit signal of a route, an overlap is also maintained beyond the route exit signal. Overlaps are also used to support through train movements. Mainline route portions (entry signal being a home signal, a starter signal or an intermediate block (IB) signal) within a yard are suffixed with an overlap segment. As per the railway signalling convention of IR, the minimum length of the overlap region is defined but it varies based on the type of main signal. For routes originating from the home signal, a large overlap is required (sequence of track circuits from the exit *SignalNode* to the next *SignalNode* in the same direction) or a certain number (usually one) track circuit (when after the exit signal TCP node is reached with no track to follow). Routes originating from an IB or a starter signal have an overlap up to the axle counter associated with the exit *SignalNode* of those routes. Overlaps are not present for calling-on and shunt routes, as traffic on those routes are always in controlled speed with extra caution. Overlaps are identified through a DFS starting at the exit signal, going in the same direction as the route; termination happens as route on encountering the next signal or when there are no more tracks; if the distance to the next signal beyond the exit signal happens to be less than the overlap distance, then traversal continues after skipping that signal.

C. Isolation Point Enumeration

Sometimes goods rakes are parked in goods sidings where those are left unattended. Due to vibration from passing trains and abetment due to gradient and wind, such a parked rake may get dislodged and start rolling. The objective of the isolation point is to prevent such parked rakes from inadvertently rolling into a selected route and causing an accident. It is also a legal requirement in some countries. In the tool, signals in the railway yard where trains will come to a halt and remain there for some time are marked as *parking signals*. For isolation identification, parking of (non-passenger) trains at signals (mainline signals such as advanced starter (AS) and IB) where there is no scope of leaving a rake parked may be left out of the scope of parking signals. While finding isolation, sometimes both mainline and loop line signals may show up at a point for isolation. In such cases, preference is always given to non-main line signals for isolation. Thus, if a loop line is to be isolated from a mainline route, the corresponding point would be set to normal.

In order to identify the isolation setting for a route, it is important to identify the vulnerable parts of the route from where a rolling rake can come and hit the train. Obviously, the starting and ending positions are vulnerable. If the route passes through a point set in reverse, then the TCP node on which the point is placed is also vulnerable at the exposed side. All such TCP nodes which have an end in the route and another end exposed are marked as *half-tracks*. In graph theoretic terms the predecessor or successor TCP node of a half track TCP node is not part of the route.

Isolation point identification takes place after identification of half-tracks. From each half track of the route, it is checked if there is a valid path up to a parking signal through one or

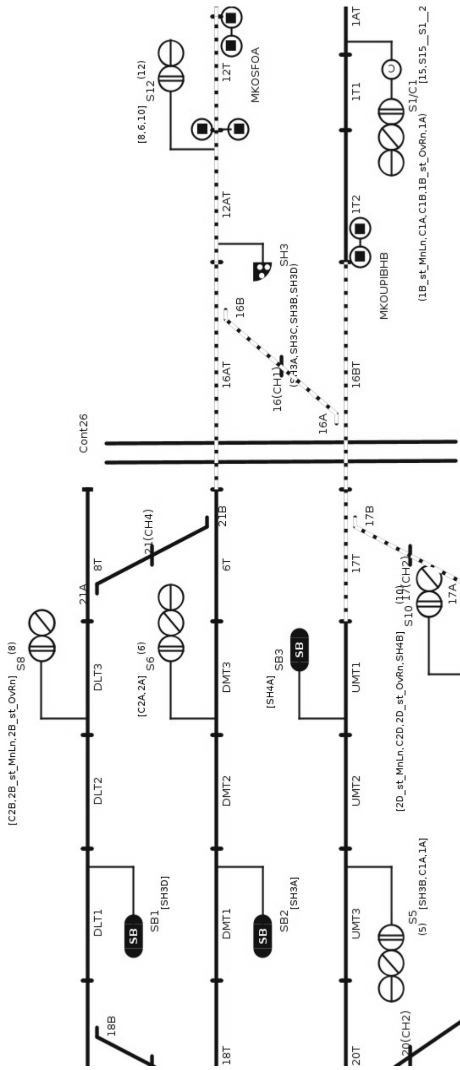

Fig. 3. Up-half track for down directional route.

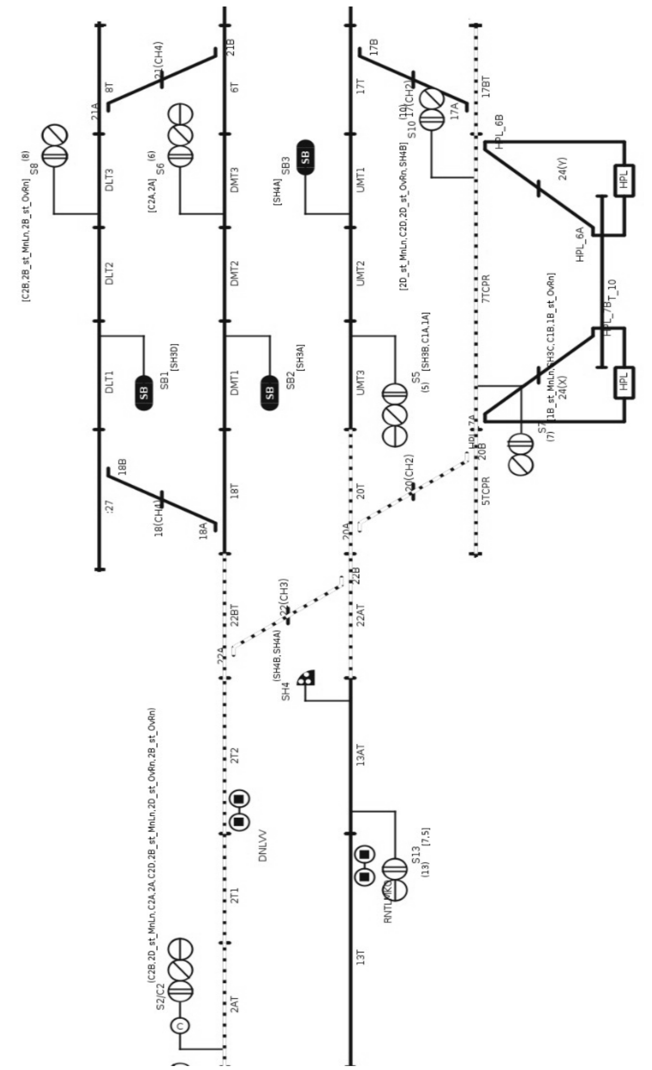

Fig. 4. Dn-half track for up directional route.

more point in reverse. In that case, the first point (near the half-track) is treated as an isolation point.

All the half-tracks are classified as UP or DN half-tracks. A track circuit which has another track circuit to its left which is not the part of the route is marked as a UP half-track, a rake may move into it from the UP direction. Consider Figure 3; the route from signal S10 to S12 with adequate overlap has been selected (shown as dashed line). Here, signal S10 is a starter signal and S12 is an AS signal. The selected route contains two points (points 17 and 16) in reverse direction. A train can move on to a selected route only after it is selected and locked. In the indicated route 17T, 17BT, 16BT and 16AT contain points and among them 17T and 16AT are open from the left-hand side. So, at signal S8, a parked rake suffering brake failure can roll down through the point 21 (set in reverse) and collide. We can prevent such a collision by setting point 21 to normal. For a UP half-track, search for a parking signal is initiated in the down direction. Similarly, a TCP which has a TCP to it's right which is not the part of the route is marked as a DN half-track, a rake may move into it from the DN direction. Consider Figure 4; the route from signal S2 to S10 with adequate overlap has been selected (shown as dashed line). Here, S2 is a home signal and S10 is a loop line starter signal. The selected route contains two (points 22 and 20) in reverse direction. In the indicated route 22BT, 22AT, 20T and 5TCPR contain points and among them 22BT and 20T are open from the right-hand side. So, at the stop board SB1, a parked rake suffering brake failure can roll down through point 18 (set in reverse) and collide. Such a collision can be prevented by setting point 18 to normal. Similarly, for a DN half-track, search for a parking signal is initiated in the up direction. In order to achieve isolation, a depth first traversal is carried out from each of the half-track in the route to find out the isolation points for every routes containing that half-track. All parking signals are placed in a structure called globISOSig (signals for handling isolation). A separate data-structure has been used to store all the selected isolation points for each route.

D. Route Conflicts Enumeration

Certain routes cannot be enabled together without compromising safety (so that head-on or sideways collision might occur). Such routes are said to be conflicting. The objective of finding a set of conflicting routes for every selected route is to prevent such collisions. A pair of routes may be in conflict for many reasons. The primary source of conflict between routes is that they have some common track circuits. However, other factors are also present. Identification of conflicting routes is based on some interlocking rules as given below.

i) Determination of conflicts may be easily done by examining all the track circuits of Ri with all the track circuits of Rj, for Ri and Rj ranging over all the routes and $Ri \neq Rj$. This simple procedure has time complexity $\mathcal{O}(\sum_{\substack{i,j \\ i \neq j}} (|Ri||Rj|))$. This may be improved by observing that it is sufficient to examine the routes passing through certain critical track circuits. It may be noted that points are present only between the home signal and the AS. There may also be DPs. Between the home signal and AS, it is sufficient to consider routes passing through track circuits adjacent to points or DPs. At other places, routes passing through track circuits associated with entry and exit signals. Since at most even routes may originate at a signal, the complexity of this scheme has an upper bound of 7(|signals| + |points|).

ii) Two routes in opposite directions separated by a sequence of track circuits conflict with each other. For each DN direction route, check whether there is a sequence of tracks leading from the entry signal of that route to the entry signal of another UP direction route.

iii) We introduce the concept of the first crossover to identify conflicting routes for shunt and calling-on signals. Trains on shunt and calling-on routes are driven with extreme caution; for this reason, railway (such as IR) do not impose the requirement for such routes to have an overlap. A point beyond the exit signal of a shunt and calling-on route is called a crossover for that route as some other route may be expected to pass through that point when it's set in reverse. Note that crossovers are not important for mainline routes because they have an overlap. For a shunt or calling-on route, any route has the crossover point on its route is marked as conflicting with that route and vice-versa.

iv) Two routes are consecutive when the exit signal of the first route is the entry signal of the next route. So, a prefix of the control tracks of the second route is a suffix of the overlap tracks of the first route. A train may be permitted to run through a consecutive route if there is no need for it to stop. For such routes their conflicts are dropped. However, run through movement is not allowed for calling-on routes (or, for that matter shunt routes).

v) A route starting at an AS will conflict with a route cutting across it's directly trailing points as these two routes may never be utilised simultaneously. These conflicts will be detected in constant time for each route cutting across the trailing point of an AS.

E. Estimation of Conflicts

In this section, we discuss the time complexity of important back-end operations starting with the route and overlap enumeration. This is relevant from the end-user experience perspective and also an understanding of the structure of the yard routes. For, route enumeration, each *SignalNode* is visited as a possible entry signal of a route. Thereafter, traversal takes place up to the next *SignalNode* in the same direction or until a terminal TCP node is encountered. The sum total of TCP nodes visited ($|TCP|$) for identifying routes in one direction is the total number of TCP nodes. For both directions it is $2|TCP|$. Similar consideration applies for overlap direction identification. Thus, the time complexity of route and overlap identification is bounded by $4|TCP|$. For isolation point computation, the identification of half-track is needed. From each half-track we traverse the graph up to the next main signal. So the complexity for isolation point identification is $\mathcal{O}$ (|points| + |signals|) (|TCP|).

The number of conflicts may be estimated according to the topology of the yard and combination of signals with consideration of the number of routes emanating from a signal post. As per signalling conventions (of IR), the first stop signal is the home signal and it consists of a main and a calling-on signal. The maximum number of routes for UP direction k_3 emanating from a home signal is usually 7; generally, $k_3 = k_1 + k_2$ (main routes with k_1 number of overlaps + k_2 calling-on routes). Similarly, the next signal called starter signal contains a main and a shunt signal. So the maximum number of routes k_3 emanating from a starter signal is 7; generally, $k_5 = k_6 + k_7$ (k_6 main route + k_7 shunt routes) with terminating AS or IB. The incoming main route from the home signal conflicts with the shunt routes of starter. Another, incoming calling on route conflicts with all the routes emanating from the starter signal. Thus, overall conflicts are up to $k_8 = k_2 + k_5$. AS or IB signal is the last stop signal and a maximum number of routes emanating from that signal with the routes emanating from the opposite home signal is $(k_9 + 1)$. Where, $k_i \in$ [1,..,7] for i $\in$ [1, 2, 6, 7, 9, 10] and $K_i > 0$ for i $\in$ [3, 5, 8].

Opposite direction route conflicts may be present between the routes emanating from home signal to the route emanating at an AS/IB. In each berthing track, there may be two opposite direction route conflicts giving rise to 2|berthing tracks| conflicts. So, overall route conflicts in worst case is: $(((k_1 + k_2)(k_1 + k_2 - 1))/2 + ((k_6 + k_7)(k_6 + k_7 - 1))/2 + (k_8(k_8 - 1))/2 + 2|$ berthing tracks$| + ((k_1 + k_10)(k_1 + k_10 - 1)/2)|$routes emanating from home signal i.e. (k_1) and opposite starter signal i.e. $(k_10) + (k_9(k_9 + 1))/2|$ routes emanating from opposite home signal i.e. k_9 and AS signal|). It may be noted that the number of TCP does not figure in the computation of route conflicts.

As an example, we illustrate number of conflicts arising in the yard shown in Figure 1. The overall number of route conflicts in worst case is computed as follows: (number of route emanating from home signal (S2) is 8 i.e. (5 number of routes from S2 + 3 number of calling-on routes from C2) + (these 8 number of routes will be in conflict pairwise with

SlNo	Route	Signal/GN	UN	Destn	PointRoute(CH)	PointOverlap(CH)
15	SH4A	SH4	CL/1orCL/2	S10	108R(CH6), 107R(CH5)	
	BLTCO	**Xing/SDG**	**ORLTC**		**ORLTO**	**OVSET**
	108BT, 108AT, 107BT, 107AT					

PointIsolation(CH)	Track Circuit on Route	Track Circuit on Overlap
106N(CH4)	108BT, 108AT, 107BT, 107AT	
ConflictingRoutes		**Remarks**
1A, 1B_st_UM, 1E, 2A, 2B_st_DM, 2B_st_OV, C1E, C2A, C2B, S10, S11, S13, S15, S7, SH3B, SH4B, SH4C, SH4D, SH5		

Fig. 5. Auto-generated RCC sections in PDF.

the routes from starter to AS or IB, giving rise to (7×4)) + (the number of route emanating from starter signals (S8, S6 and S4) is 3) + (routes emanating from the home signal (S2) and routes from opposite starter signal (S5, S7 and S3) to AS signal (S13), giving rise to (7×4)) + (from two berthing tracks L1T and L3T total number of conflicting routes is 4 in opposite direction)). So the auto-generated maximum number of route conflicts from the yard given Figure 1 is 178. This figure is much less then $\binom{N}{2}$, where N is the number of routes.

F. Route Control Chart Generation

For RCC generation, the route definitions of all the routes are traversed to generate the row entries one route at a time. As mentioned earlier, the RCC is generated in XML. It may be noted that the XML tags used here are not conformant to railML [1], [2] which is a generalised XML schema to capture a wide range of information pertaining to railways. In the future, conformance of the XML schemas used here to railML would be desirable. The RCC contains tags for the different columns of the RCC are categorised and enumerated below.

RCC tags related to route definition
Those are ⟨controlTracks⟩: list of all the control tracks a route, ⟨approachTrack⟩: the track contains the entry signal of a route, ⟨backlockTracks⟩: list of all the tracks from first control track to the track contains point of a route, ⟨overlapTracks⟩: list of all the overlap tracks of a route, ⟨entrySig⟩: the entry signal of a route, ⟨exitSig⟩: the exit signal of a route, ⟨overlapcleared⟩: the previous track of the track contains opposite directional signal of a route and ⟨overlapoccupied⟩: list of all the tracks from the track contain opposite directional signal to the track contain exit signal of a route. For example, all the control tracks t_i are defined in XML RCC is given below: ⟨controlTracks⟩ t_i ⟨/controlTracks⟩ Similar type of XML tag declaration is done for other columns associated with route definition.

RCC tags related to point position
Those are ⟨pointNormal⟩: list of all the points that are in their normal position for a particular route setting, ⟨pointReverse⟩: list of all the points that are in their reverse position for a particular route setting, ⟨overlapPointNormal⟩: list of all the points in the overlap section that are in their normal position for a particular route setting, ⟨overlapPointReverse⟩: list of all the points in the overlap section that are in their reverse position for a particular route setting and ⟨crankHandle⟩: list of all the crank handles involved in a particular route setting. For example, all points p_i are defined in XML RCC is given below: ⟨pointNormal⟩ p_i ⟨/pointNormal⟩. Similar type of XML tag declaration is done for other columns associated with point definition.

RCC tags that are derived using other route definitions
Those are ⟨isolationPoints⟩: list of all the identified points in normal or reverse state for some particular routes and ⟨conflictingRoutes⟩: list of all the routes conflicting to a particular route. Here the tag declaration is similar as the previous declaration for point and route definition.

Other attributes related to the concerned yard
⟨levelCrossingGates⟩: list of all the the level-crossing gates interlocked with a particular route setting. Tag declarations in XML RCC follow on similar line as discussed earlier.

The RCC is a tabular representation of the configuration of various objects in a railway yard for safe usage of the routes. For convenient viewing, the RCC is generated in PDF. The fields included in the PDF for the SER zone of IR are as follows: route, entry signal, route button, exit signal, route point (reverse/normal), overlap point (reverse/normal), back locked track circuit occupied, level-crossing, overlap track circuit cleared, overlap track circuit occupied, overlap set, point isolation, track circuit on route, track circuit on overlap, conflicting routes.

It may be noted that fields to be represented in the PDF of the RCC may vary from one railway to another. In case of IR there are variations among the railway zones also. For this reason, it is important to have a flexible scheme to generate the PDF. We have developed an XML processor based scheme to generate the LaTeX file which, on compilation, generates the necessary PDF. An XML scheme is defined for specifying how the contents in LaTeX format are to be generated by querying the XML RCC; on running the XML translator the requisite LaTeX is generated. Excerpts of the generated PDF are given in Figure 5.

IV. Handling Junction Yards

Yards usually have only two ends even though they may have several lines, where as a junction has three ends or more. We have already considered yards such as A connecting B to the west and C to the east (say). Now consider the situation where yard A serves as a junction for B, C and, additionally, D to the south. Let D be connected to E further south. From the perspective of the station master at D, let trains leaving D towards E be DN trains and trains leaving D towards A be UP trains. Now an UP train from D may proceed towards B or towards C. Thus, such an UP train from D may be going either UP (towards B) or DN (towards C) in yard A. What this really means is that the UP trains from yard D will sometimes move towards the UP direction signals and sometimes towards the DN direction signals in yard A. This physical reality needs to be incorporated into the algorithms that have been presented, especially for the algorithms for identifying the routes and the overlaps. The new feature to enable this is a logical direction reversing point. While traversing through the direction reversing point, signals of the opposite direction are sought on crossing the point. Also, traversal happens through edges of the opposite direction. As an example, the graphical representation of triple-ended yard is shown in Figure 6. Note that the end corresponding to yard D is still drawn horizontally. However, note the presence of the regular point (P 21) on the left and the direction reversing point (P 22) on the right. The point (P 22) has been marked as *direction reversing* meaning that it is a regular point (physically) but has the logical role of reversing the UP/DN sense of a route as that point is crossed. Traversals from D through the regular point happen exactly the same way as before. However, when traversing through the reversing point, a search along UP edges will progress as searches along DN edges and the search for UP signals will proceed as search for DN signals, and vice-versa.

V. Verification and Validation Aspects

While the algorithms described so far identify routes, conflicts between them, isolation requirements and so on, for proper yard operation specific rules need to be followed for developing the yard layout. The location of each type (main, shunt and calling-on) of signal plays an important role. Combination of signals may vary based on the yard specific requirement for operation such as speed control, siding, shunting, through-line movements and movements towards the platform. As per signaling convention (of IR), first, a speed control signal is present as a distant signal. Following that, the first stop signal is a home signal or a secondary home signal. Based on the topology of the yard, it may be necessary to have a block section before a distant signal. In such cases, before a distant signal, an IB signal is present. Next, the signal between a home signal and an AS is present a starter signal and the last stop signal is an AS. Usually, the next stop signal after an AS is the next home signal. However, as discussed, for some yards needing an IB, an IB signal is placed in lieu of the AS signal. The proper organisation of signals in the yard layout needs to be validated to guard against inadvertent flaws. The graph-based representation of yard layout can be

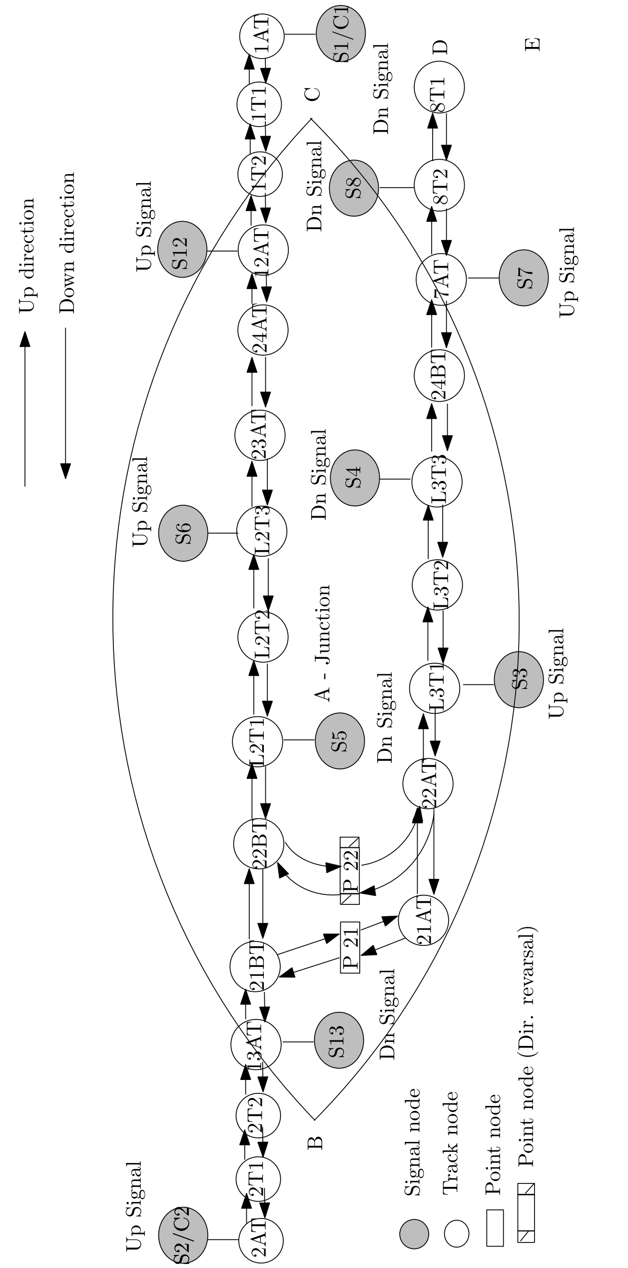

Fig. 6. Graph representation of the complex yard layout with a junction.

leveraged for formal yard validation with respect of yard based linear temporal logic properties (LTL). Five yard properties are discussed below. Such properties are expressed as temporal logic formulae for verification using a model checker.

i) *The distant signal must be placed before the home signal to check speed limit*. The corresponding LTL property for UP and DN direction is given below:
```
LTLSPEC G (((direction = DN)
& DownDistSignal) → X (( X
(DownHomeSignal)) R (!DownSignal)))
LTLSPEC G (((direction = UP)
& UpDistSignal) → X (( X
(UpHomeSignal)) R (!UpSignal)))
```
ii) *Once the home signal is encountered then eventually the starter signal will appear.*

TABLE I
EXPERIMENTATION OF THE LAYOUTEDITOR ON TEST YARDS

Yard	#Routes	#Points	#Signals	#Tracks	#LC	#SDG	#HPL	#DP	#IB	#SB	#SS	#Conflicts (pairwise)
MKN	26	6	17	41	1	–	2	–	2	3	–	149
BRD	27	6	19	42	1	1	–	1	–	1	3	169
BLD	28	6	17	49	4	1	3	–	2	3	–	115
BST	23	6	18	42	3	–	2	–	2	3	–	121
GRP	30	6	22	31	–	–	2	–	–	3	–	219
GDN	34	6	29	51	2	3	5	1	2	4	–	240
LTB	31	6	18	28	1	2	–	–	–	2	–	257

iii) *Once the starter signal is encountered then eventually the AS signal will appear and once the AS has been encountered then eventually IB or NextHome signal will appear.*

iv) *An up direction DP is never eventually followed by a down direction main signal.*

v) *Beyond the up direction de-railment point then there must be a down direction signal for exit to avoid a locomotive getting stuck at a dead-end, in the absence of an exit route.*

Not all violations can be captured in the graphical model, such a track being drawn through a point. However, such violations can be checked and prevented geometrically. The safety properties for EI logic V&V are generated following railway signalling principles [3]. The auto-generated NuSMV [4] model is verified with respect to these safety properties using the NuSMV model checker. This approach provides an automated solution to validate the logic circuits. Constructing of safety properties are described. The corresponding LTL representation of the described safety properties are given below.

i) *The interlocking is free:* The `INT_LR` relay (internal relay of the route initiation relay (LR)) is a dependent relay of `LR` relay. The energisation of `INT_LR` depends on elementary pre-defined conditions which are; i) all the emergency button relays and the sequential proving relay are in de-energised state ii) all the route points are free iii) all the conflicting routes of a specific route are not asserted. The corresponding `LTL` property is:

```
LTLSPEC G ((X INT_LR) →
!(Emergency_Button_Relay) &&
!(sequential_proving_relay) && (All
the route points in free condition)&&
!(Conflicting routes LR))
```

ii) *All the points in route and overlap are locked and detected.*

iii) *The selected route is locked.*

iv) *The track circuits in the route and the overlap are cleared.*

v) *Inside the route and overlap, all the interlocked level crossing gates are locked against the road traffic.*

Supporting routines are available to generate formal models for yard validation and also for logic verification corresponding to the captured yard on which the properties described above can be verified. It may be noted the logic verification is not a substitute for yard layout validation which helps to identify structural flaws in a yard layout which would not create errors for logic verification.

VI. EXPERIMENTAL RESULTS

The tool was tested on several yards obtained from the SER. The yards for which results have been given are MKN, BRD, BLD, BST, GRP, GDN and LTB. The SIPs provided were first captured using the tool. Thereafter, possible routes were automatically enumerated within the tool. Routes to be retained in the generated RCC were chosen and the RCC was generated. The specifications of the captured yards are given in Table I which indicates the scalability of the tool.

The results for the first six yards were compared to the RCCs received from IR and found satisfactory. Where there were mismatches, results of the tool were accepted to be consistent with the current practices of SER or adequately compliant with the safety requirements. LTB is a new yard for which the RCC generated by the LE was found to be satisfactory. Experimentation was also done on a much larger yard having about 250 routes for which the RCC was generated in a comparable time. However, those results have not been tabulated as the reference RCC is not available for comparison. Yard layout validation was done by introducing anomalies in the yard layout and those situations were identified via model checking. Similarly, EI application logic was also checked by introducing faults and those too were identified.

VII. RELATED WORK

Railway signaling is a well-known example of a safety-critical system. While there are general principles, specific signalling conventions vary from one railway to another; our work is aligned to signalling for SER of IR. A number of documents have been received from Research Designs and Standards Organisation (RDSO) to support this work. Documents S1 [5] and S2 [6] provide an introduction to signaling and railway interlocking systems with instruments and different track-side equipments. Relay-logic circuits are documented in EIS [7]. A railway yard consists of several components including track segments, points, signals, crank handles, level crossings and axle counters which are relevant to signaling and may be represented in graph form. An initial graph representation of railway yards was developed by Ghosh *et al.* [8]. This tools generates RCC compatible with IR standards [9].

Mirabadi *et al.* [10] present a technique for generation and verification of the control table with respect to signalling principles. However, their technique does not appear to be general enough to handle zonal requirements, such as obtaining route isolation and handling multiple overlaps. Yldrm *et al.* [11] and Mutlu *et al.* [12] explain the automatic generation of interlocking table for a model railway station. However, the role of approach locking and backlocking is not accommodated. Tombs *et al.* [13] describe challenges of RCC construction principles; RCC editing is also considered to address the limitations of the RCC construction procedures. The RCC construction mechanisms reported in the current work are more aligned to IR. While RCC editing, is not supported, safe editing of overlap TCPs was taken into account. Their tool does not encompass isolation.

Haxthausen *et al.* [14] considers the generation of safety requirements from railway interlocking table (or RCC). Our primary observations are that the presented interlocking requirements fall short of those considered in our work, possibly because the requirements were different or because those details were not considered. For example, IR requirements of conflicts between routes and isolation of routes from parked rakes are significantly more involved. Automation of RCC generation without manual intervention is a challenging task. In the literature, we have not come across any technique on such fully automated RCC generation.

Flamming *et al.* [15] present a method for RCC based testing of route formation/selection with respect to track circuit occupancy, correct point setting of the route portion and subsequent signal activation. It may be noted that IR has some additional requirements of checking track circuit occupancy of the overlap portion, checking conflict freedom and establishing route isolation, where possible and prescribed the RCC. The approach adopted by them is equally applicable to our tool suite and may be considered in future in the context of certification.

Marrone *et al.* [16] present a model-driven approach for effectively generating models for V&V, using ERTMS/ETCS as a test case. Although, their techniques are relevant, their test case is presently beyond the scope of our work. Timetabling of railway services is an important (but different) problem for optimum utilisation of infrastructure and providing satisfactory service to customers. Corman *et al.* [17] describe a methodology to resolve conflict detection and resolution problems related to timetabling through a graph theoretic formulation – that graph may be constructed using the tool generated RCC along with other inputs.

The LE has a yard verification component to check for flaws in the yard structure. As a by product, the LE, during RCC generation, produces yard specific LTL safety properties which must be satisfied by the EI systems for the yard. The verification is done by means of the NuSMV model checker, in line with the requirements of EN50128 [18].

VIII. Conclusion

In this paper, we present the development of a tool for capturing a given signal interlocking plan (SIP) (provided by Indian Railways (IR)) and then verifying the yard layout and generation the route control chart (RCC) for the yard. The important contributions of the work are as follows: requisite data structures for representing a yard, a front-end with adequate drawing primitives and other interfaces, in particular, a menu for selecting required routes from all possible identified routes and a back-end with necessary algorithms for realising required sub-tasks for RCC generation. Some of those are route enumeration, overlap enumeration, conflict determination between routes, isolation determination and exporting the yard and the RCC in extensible markup language (XML) and portable document format (PDF). The XML representations of yards and RCCs (in lieu of PDFs of SIP and RCC) which enable easier interfacing with other tools (e.g. tool for logic generation), are new for IR and serve a useful purpose. As already observed, railML has been used as a comprehensive data representation standard in XML for railway signalling applications. In the future, it would be desirable to align the XML standards being used for the tool reported here to railML. The graph-based representation supports graph theoretic formulation of the important tasks and also allows for easy enhancement of features. Extension of our graph based algorithms to support junction yards has been demonstrated. The tool is implemented in Java. The tool has been successfully evaluated on the seven yards received from IR. Formal methods have been adopted for yard layout validation and electronic interlocking (EI) logic verification. To the best of our knowledge, our tool is the first publicly reported work on the comprehensive handling of RCC generation with support for validation and verification (V&V). The graph theoretic formulation used here makes the framework flexible and amenable to easy modification for changing requirements. In the future the LayoutEditor (LE) may also be augmented to generate factory acceptance test (FAT) and site acceptance test (SAT) schedules.

The methods presented here are biased towards IR. It appears that the requirements of IR cover the requirements of other railways surveyed so far. A simple mechanism to satisfy the requirements of other railways would be to apply many features conditionally, for example, the used of overlaps, route isolation and also the several conditions for identifying route conflicts.

Acknowledgment

Mr. R. K. Jain, Mr. S. Mathur, and Mr. A. Katiyar are thanked for their support and participation in this work from RDSO.

References

[1] A. Nash, D. Huerlimann, J. Schuette, and V. P. Krauss, "RailML—A standard data interface for railroad applications," *WIT Trans. Built Environ.*, vol. 74, p. 8, Jan. 2004.

[2] *RailML*. Accessed: Oct. 22, 2019. [Online]. Available: https://www.railML.org/en/user/subschemes.html

[3] S. H. Safdar, *S36, Relay Interlocking Systems, Indian Railway Standard Specifications, Ministry of Railways*. Lucknow, India: Research Designs and Standards Organisation, 1996.

[4] A. Cimatti *et al.*, "NuSMV 2: An OpenSource tool for symbolic model checking," in *Computer Aided Verification (CAV)*. Berlin, Germany: Springer, 2002.

[5] M. Dixit and S. V. Rao, *S1, Basics of Signal Engineering*. Telangana, India: IRISET, 2009.

[6] A. V. R. S. Rao and J. Ray, *S2, Principles of Interlocking*. Telangana, India: IRISET, 2009.

[7] S. Khandelwal, M. M. Waris, V. K. Pandey, and S. Yadav, *Report of Standardisation of the Typical Circuits for Electronic Interlocking*. Lucknow, India: Research Designs and Standards Organisation, 2018.

[8] D. Ghosh and C. Mandal, "Layout validation using graph grammar and generation of yard specific safety properties for railway interlocking verification," in *Proc. Asia–Pacific Softw. Eng. Conf. (APSEC)*, Dec. 2015, pp. 330–337.

[9] M. Dixit and S. K. Nanda, *S11, Control Tables and Selection Circuits*. Telangana, India: IRISET, 2010.

[10] A. Mirabadi and M. Yazdi, "Automatic generation and verification of railway interlocking control tables using FSM and NuSMV," *Transp. Problems*, vol. 4, pp. 103–110, Jan. 2009.

[11] U. Yıldırım, M. S. Durmuş, and M. T. Söylemez, "Automatic interlocking table generation for railway stations using symbolic algebra," *IFAC Proc. Volumes*, vol. 45, no. 24, pp. 171–176, Sep. 2012.

[12] İ. Mutlu, U. Yıldırım, M. S. Durmuş, and M. T. Söylemez, "Automatic interlocking table generation for non-ideal railway yards," *IFAC Proc. Volumes*, vol. 46, no. 25, pp. 55–59, 2013.

[13] D. Tombs, N. Robinson, and G. Nikandros, "Signalling control table generation and verification," in *Proc. Cost Efficient Railways Through Eng. (CORE)*. Wollongong, NSW, Australia: Railway Tech. Soc. of Australasia/Rail Track Association of Australia, 2002, pp. 415–425.

[14] A. E. Haxthausen, "Automated generation of safety requirements from railway interlocking tables," in *Leveraging Applications of Formal Methods, Verification and Validation. Applications and Case Studies*, T. Margaria and B. Steffen, Eds. Berlin, Germany: Springer, 2012, pp. 261–275.

[15] F. Flammini, N. Mazzocca, and A. Orazzo, "Automatic instantiation of abstract tests on specific configurations for large critical control systems," *Softw. Test., Verification Rel.*, vol. 19, no. 2, pp. 91–110, Jun. 2009.

[16] S. Marrone, F. Flammini, N. Mazzocca, R. Nardone, and V. Vittorini, "Towards model-driven V&V assessment of railway control systems," *Int. J. Softw. Tools Technol. Transf.*, vol. 16, no. 6, pp. 669–683, Nov. 2014.

[17] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo, "Bi-objective conflict detection and resolution in railway traffic management," *Transp. Res. C, Emerg. Technol.*, vol. 20, no. 1, pp. 79–94, Feb. 2012.

[18] *Railway Applications—Communication, Signalling and Processing Systems—Software for Railway Control and Protection Systems*, Standard EN 50128:2011, CENELEC, European Committee for Electrotechnical Standardization EN50128, 2014.

Arindam Das received the B.Tech. degree in electronics and instrumentation engineering from the Haldia Institute of Technology, India, in 2008. He is currently pursuing the M.S. degree (by research) with the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur. He is also associated with a research project as a Senior Project Assistant with the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur.

Manoj Kumar Gangwar received the B.Tech. degree in computer science and engineering from the Dr. Ambedkar Institute of Technology for Handicapped, Kanpur, India, in 2016, and the M.Tech. degree from the Department of Advanced Technology Development Centre, IIT Kharagpur.

Devleena Ghosh received the B.E. degree in computer science and technology from the Indian Institute of Engineering Science and Technology, Shibpur, and the M.Tech. degree in computer science and engineering from the Indian Institute of Technology Kharagpur. She is currently a Ph.D. Research Scholar with the Department of Computer Science and Engineering, IIT Kharagpur. Before this, she worked as a Software Engineering at Microsoft India Pvt., Ltd. She has worked in logic simulation and applications of formal verification in railway interlocking systems.

Chittaranjan Mandal (Senior Member, IEEE) received the Ph.D. degree from the IIT Kharagpur, in 1997. He was a Reader with Jadavpur University before joining IIT Kharagpur in 1999, where he is currently a Professor with the Department of Computer Science and Engineering. He has handled several sponsored projects, both from Indian government agencies and from private industries. He has supervised several Ph.D. students and has more than 130 publications. His research interests include formal modelling and verification. He became an Industrial Fellow of Kingston University, U.K., in 2000 and received the Royal Society Fellowship for conducting collaborative research in the U.K.

Anirban Sengupta has been working as a Senior Signal and Telecom Engineer with the Drawing and Design Office of HQ SE Railway. He is experienced in installation, testing, commissioning of all types of panels, and electronic interlocking. He has been closely involved in the commissioning 800 route Kharagpur electronic interlocking system, the largest in Asia.

M. Mubashshir Waris received the B.Tech. degree in electronics from Aligarh Muslim University. He is currently an IRSSE Officer of 2006 exam batch, Indian Railways, South Eastern Railway cadre. He is serving as a Joint Director, Signal at Research Designs and Standards Organisation under Ministry of Railways, Government of India. He handles the design and development of some of the modern signalling equipment for Indian Railways specially the EI, Data-loggers, and High Speed and EMI/EMC for Railway Signalling.