*Proceedings of the 1991 Winter Simulation Conference*
*Barry L. Nelson, W. David Kelton, Gordon M. Clark (eds.)*

# A RAPID RAILWAY SIMULATION MODEL DEVELOPMENT SYSTEM INCORPORATING AUTOMATIC MODEL GENERATION

Kyriakos Tsiflakos

The Management School
Imperial College
London SW7 2PG
England

Stephen C. Mathewson

Technical Services Division, BAA
Masefield House, Gatwick Airport
West Sussex RH6 0HZ
England

## ABSTRACT

This paper discusses the development of a railway simulation system incorporating *automatic model specification* and *automatic model generation* techniques. The modelling environment features a GEM-like graphical user interface consisting of a layout editor and an icon editor. The model builder draws only the geometrical layout of a railway network section (e.g. a railway terminal) including the position of the signals, thus constructing a partial railway model. This graphical model consists of arcs representing tracks and nodes that represent junctions. The system assists in completing the specification of the model by automatically inferring the layout of the remaining control installations (i.e. the track circuits and signal blocks) and the possible train routes through the network. Thus, by the end of this process the modeller has come up with a complete railway network data model stored in a database. The final stage of the project involves the development of a simulation code generator that will transform the stored data model to simulation code. The system under development is intended to act as a decision support tool at the operational research (OR) division of the *British Railways Board* (BR). It will replace the currently employed simulation model development method which is much more tedious and lengthy.

## 1 INTRODUCTION TO RAILWAY INFRASTRUCTURE AND OPERATION

Before discussing railway simulation model development, we thought it would be helpful to provide the reader with a basic understanding of what railway networks are and how they operate. First of all, the infrastructure of a railway system consists of a *physical skeleton* complemented by a *control system* and the *rolling stock*. The physical skeleton of a railway is mainly composed of a network of *tracks*, which constitute the physical medium where the trains move on. Tracks intersect at *junctions*. Tracks have their origin or end in *terminal stations*. The movement of trains on the tracks is controlled by *signals*. Each one of the railway's physical elements described above, has certain attributes associated with it. Tracks, for example, are characterized by their gradient (which is expressed as a ratio, e.g. 1:40), maximum train speed (in Britain this currently ranges from 5 to 125 miles per hour) and direction of train motion (single or bi-directional lines). Since some of these characteristics may change several times along the same track, it would be more accurate if we said that their values characterize track sections rather than entire tracks. If we now consider the junctions, these can be either *points* (where a track is intersected by the ending point of another one, e.g. point "a" in figure 1), or *diamond crossings*, also known as *crossovers* (where two tracks intersect as at points "b" of figure 1). Finally, signals can be classified according to their number of *aspects*, i.e. the number of lights each signal has (we may come across 2 or 5-aspect signals, although signals with 3 or 4 aspects are much more common; in figure 1, we only have 3-aspect signals).

A *railway control system*, on the other hand, is implemented by means of physical entities known as *control or safety installations*, and other, abstract entities, known as *signal blocks*. The most important control installations are the signals and the *track circuits*, whose purpose is to ensure the safe and efficient movement of trains on the network (Nock, 1980). A track circuit can be a section of a single track or it may be composed of more than one track sections belonging to different tracks, which might intersect at junctions or not intersect at all (figure 2 shows various possible track circuit shapes). The track circuits constitute the basic control elements of a railway system. Their task is to monitor the current positions of the trains on the tracks. They are implemented by

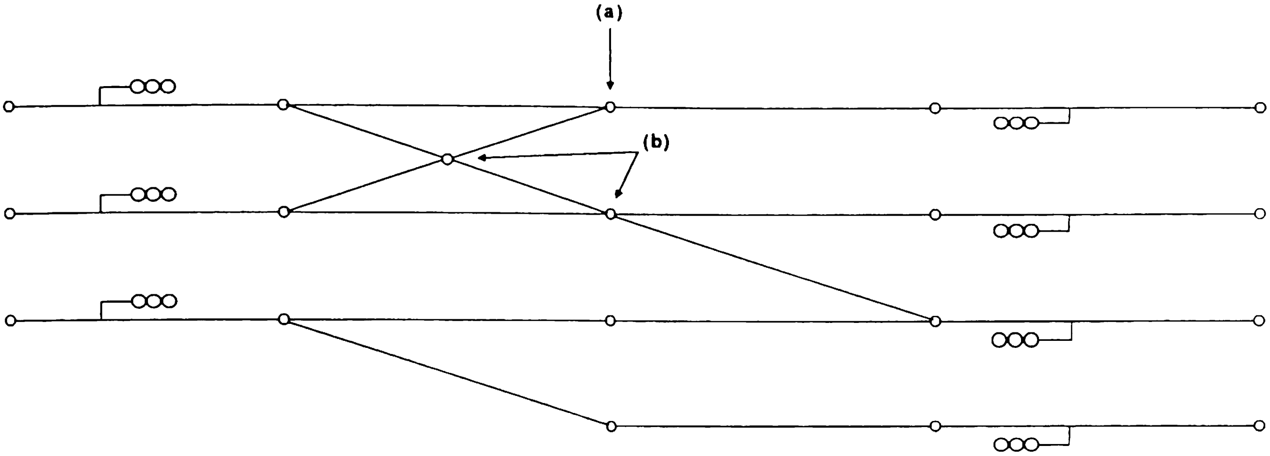


Figure 1: A Section of a Typical Railway Network

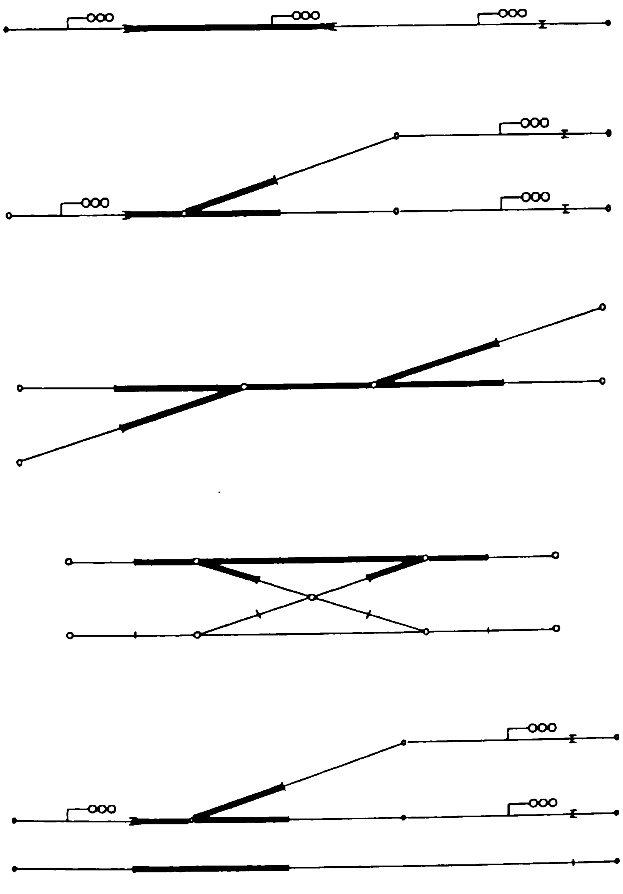


Figure 2: Typical Track Circuit Layouts

electromechanical installations that divide the network in comparatively small areas that are electrically insulated from each other. The presence of a train on a track circuit is detected by the change the train causes on the electrical circuit status. The track circuits implement an extremely important safety rule which demands that "no more than one train may occupy any part of the same track circuit at any particular time". As figure 2 suggests, some of the *track circuit joints* (i.e. the ending points of a track circuit, that separate it from the neighbouring track circuits) that surround a diamond crossing or a point are placed at the *clearance* or *fouling points* between the diverging lines. However, quite a number of rules have to be taken into consideration by the signalling engineers when it comes to positioning these joints. A track circuit may be in one of three conditions, i.e. either *free*, *reserved* or *occupied*. One aspect of the railway control system, known as *interlocking*, ensures that, depending on the status of certain track circuits, the appropriate signals show the right aspect.

Operational safety is the prevailing consideration when signalling engineers decide how to lay down the track circuits. However, there is a compromise between operational efficiency and the cost and complexity of installing the layout. It should be obvious from the discussion so far that the resolution with which the position of the trains on the network can be monitored, is inversely proportional to the average size of the track circuits. We can lay down the safety installations so that the network is divided into a few large track circuits or many small ones, while making sure, with proper interlocking, that both arrangements provide absolute safety. However, the former solution will be less costly but, also, less operationally efficient than the latter. By following some empirical safety and efficiency rules and by taking into consideration the cost factor, signalling engineers come up with a satisfactory compromise.

The interlocking system groups track circuits into larger entities known as *signal blocks.* A signal block is defined as a series of successive track circuits, the first of which is the first track circuit after a signal and the last is the first track circuit after the next signal. There are, however, many exceptions to this definition depending on certain layout features of local nature; a complete discussion of the operation of the signalling system being outside the scope of this paper. However, the fact is that as soon as the signal blocks and the interlocking controls between the signals and track circuits have been established, we can say that the railway control system is in place. Let us now assume that a train has to follow a certain route through the network. Depending on the situation, the control system will try to reserve one or, in some cases, more signal blocks ahead of the train. As soon as the first signal block becomes free of any other train, it will be reserved for the train in question. That will cause the first signal before that signal block to clear to green while the train approaches that signal. As soon as the train starts occupying that signal block, the signal will be set back to red and the control system will ensure that it stays so until the signal block can be reserved by another train heading towards the same route.

Another railway feature which is worth mentioning is the coordinate system used. A railway network, no matter how much complicated, is modelled as an one-dimensional structure. The rationale behind this approach is that, although a section of a track layout might look like a two-dimensional structure, if we consider sufficiently large areas, one dimension (the one extending along the tracks) is by far greater than the other. Another reason is that tracks have one dimension anyway. The origin of this coordinate system is a reference line known as the *datum.* This is vertical to the railway tracks and is usually positioned near the buffer stops in the main stations. On the railway layout maps, the linear distances of the most important elements (as, for example, the signals) from the datum are recorded.

## 2 THE CURRENT STATUS OF RAILWAY SIMULATION MODEL DEVELOPMENT

OR analysts of the British railways industry have been developing simulation models for the last twenty years. Most of these models have been aimed at infrastructure investment appraisal and capacity planning. Their development currently involves a manual and lengthy modelling procedure, which is based on the HOCUS (P-E Inbucon Ltd., 1989) simulation software package. The process starts by gathering data concerning the network topology, control installations, safety rules and operational procedures from numerous sources (e.g. railway layout maps, filing cabinets and experts' empirical rules). After analyzing and classifying this information, BR analysts model the system in terms of entities (e.g. trains and track circuits), activities (e.g. that reserve signal blocks) and queues (e.g. of reserved track circuits) and construct an *activity cycle diagram.* The first stage of the diagram construction process involves filling information in special cards. For each entity, activity and queue, a different card has to be filled in with an identification code (ID) and, where appropriate, some attributes (e.g. the maximum speed limit on a track). On the activity cards, the modeller has to list all the conditions that must be met for an activity to be initiated. These cards are then placed on large boards and arrows are drawn among them, representing the flow of logic. According to the HOCUS modelling logic, activities move entities from one queue to the other. As we mentioned earlier, in order for an activity to be initiated, certain conditions must be satisfied. In the railway paradigm, for example, if we consider an activity whose purpose is to reserve a signal block for an approaching train, the initiating conditions should include the presence of all track circuits that comprise this signal block in the track circuit queue that is labelled as "free". Also, when this activity is executed, its actions should include the movement of all track circuits involved from the queue labelled "free" to the one labelled "reserved". Similarly, an activity whose purpose is to move a train over a signal block should update the position of the train. Where two or more different routes can be established between the same two points of the network (this occurs frequently in areas of high network complexity, such as outside terminal stations), route preferences and criteria for selecting routes have to be stated. In general however, by far the greatest amount of modelling effort goes into identifying and listing all possible train routes through the network. For each such route, all track circuit reserving, releasing and train movement activities have to be identified.

After having constructed the activity cycle diagram, the analysts state the initial conditions of the simulation on separate cards and, finally, code the simulation model in HOCUS. Coding can be performed manually, with the low level simulation language that HOCUS supports, or alternatively the user can enter the model specification via a higher level, menu-driven, user interface that is included in the HOCUS modelling environment. Both ways require significant HOCUS-specific modelling and coding expertise on the part of the user. For large models the process is very time consuming, repetitive, tedious and subjects the analyst to significant tension and pressure. Under such circumstances, the method becomes error prone. Therefore, it is highly unlikely that the analyst will end up with an entirely valid model in a single attempt. The motive that led BR to initiate the project was a growing concern about the lengthy development times required for the railway model building process. Development periods of three to six months have been a commonplace for complex large stations (e.g. Paddington or Cannon Street).

Apart from its cumbersome nature, this modelling approach can also be described as rather superficial, for two reasons. First, the resulting model looks more like an exhaustive list of all routes through the network that are used in practice, rather than a representation of the network structure (from which structure, of course, the routes, among many other things, could be inferred). Second, as a result of this, the modelling approach is application-specific. It is only useful for constructing the simulation models in question. As we shall see in the next paragraph, the proposed modelling method on the other hand, includes, at some intermediate stage, the creation of what we call a *generic model.* (As opposed to a *specialised model* which is supposed to meet the requirements of a specific OR technique, e.g. simulation, we use the term *generic model* to refer to a system-centred and OR technique-independent model that encapsulates a description of the main structural and operational characteristics of a system. If we now assume that this generic model is stored in a data base, various specialised models, serving different purposes, can be automatically formed by the selective extraction of appropriate information from the generic data pool).

Moreover, if we refer to the railway simulation model development literature, we shall see that the techniques employed elsewhere, also suffer from various drawbacks. Most of the already developed railway simulation models, of course, do not suffer from the complexity of detailed network modelling, because either the purpose that these models are supposed to serve does not require modelling at this level, or such modelling is simply avoided for the sake of convenience. Howard, Gill, and Wong (1983), for example, present a review of models that concentrate on the detailed modelling of the performance characteristics of trains rather than the network itself. Some models, like the one discussed by Reich (1966), deal with simulating traffic on the much simpler layouts of the urban rapid transit railways, while the ones that model complex intercity networks (Jones, Lach, and Metsos, 1968) usually follow a coarse approach, leaving out most of the detail that BR requires for its network models. In spite of these simplifications, however, the level of computer literacy that these models require on the part of the modeller is usually remarkably high.

## 3 THE PROPOSED RAPID MODEL DEVELOPMENT APPROACH

In the previous paragraph we had the opportunity to evaluate the scope and limitations of the currently employed simulation modelling methodology by the British railway service industry. In practice, however, these are not the only short-comings in the traditional process. In many cases, for example, as soon as an analyst has completed the development of a simulation model of a terminal station, another starts developing the model of another station almost from scratch. The process of trying to establish an alternative model development framework started by noticing that much of the data and knowledge required to build a model, is shared among different model development sessions. For example, although the geometry of the track layouts of two different railway stations may vary considerably, there are some common features that need to be modelled (e.g. the signalling rules). These remain, more or less, the same. Moreover, as soon as the modelling of the network topology (i.e. the track layout), including the positions of the signals, has been completed, the layout of the remaining control installations (i.e. the track circuits) can be automatically inferred from the network topology and a set of safety regulations. Also when a model needs to be modified because, for example, a new platform has been constructed in the station, the modeller has to go through a very tedious and error-prone procedure in order to make the necessary changes in the program code. Therefore, we can now see how railway model development could benefit from the incorporation of automatic model specification and generation techniques. In short, automatic model specification contributes in arriving from an initial, user-specified partial model to a complete railway model specification, while with automatic model generation we can have the complete model specification automatically converted to a simulation model coded in one of the available computer languages.

The value of incorporating computer graphics in simulation modelling and execution has been widely acknowledged. Hurrion and Secker (1978), Withers and Hurrion (1982), Bell (1985), and Kirkpatrick and

Bell (1989) are among the ones who have written extensively on this issue. Hurrion (1989) also presents a survey of simulation systems that support graphical interaction techniques. Indeed, many of today's integrated simulation environments, like GPSS (Minuteman Software, 1986) and HOCUS (P-E Inbucon Ltd., 1989), support the graphical execution of simulations. Some go even further by incorporating graphical model specification capabilities (Mathewson 1985, 1987; Raczynski 1990). However, these tools are too general to address the specific needs of railway network model development. In fact, although most of them support simulation code generation facilities, they lack any automatic model specification capabilities; the main reason for this being the application-specific nature of such techniques. It became obvious therefore, that a more customised approach was required in this case. As we shall see in the following paragraphs, this approach is implemented through a model development environment composed of a suite of software modules, namely a *layout* and an *icon editor*, a *data/knowledge base* (DKB), a *track circuit generation module*, a *route generation module* and a *simulation code generator*.

### 3.1 The Layout Editor

The layout and icon editors are used for capturing the network topology. When we were faced with the problem of laying down the functional specifications of the layout editor, we tried to think what would be the most natural modelling approach from the modeller's point of view. Thus we came up with a modelling framework which allows the modeller to draw a network layout onto the computer screen, with a simple set of basic graphical functions. During this process, the modeller may use a standard railway layout map as his/her main source of topological information. By the end of this drawing session, he/she has come up with a graphical model on the screen, which can then be stored in the DKB. So far the modeller has managed to capture only the topology of the track layout, constructing thus a *partial generic railway model.*

A convenient and natural way of approaching the problem of generic railway network modelling, is to analyze it in terms of three different *layers* or *levels of detail.* The coarser layer involves modelling the topology of the track layout in terms of *arcs* and *nodes*, with the nodes representing either points, diamond crossings or sudden changes in the attributes of a track. At the second level of detail the user is able to add signals along the arcs. Finally, the third layer is automatically generated in two steps that are implemented by the automatic model specification modules: first the track layout is automatically track circuited and then the signal blocks and train routes are inferred. After each step, of course, the modeller is given the opportunity to intervene and edit the automatically generated elements of the layout, if he/she thinks that something has to be corrected. During this incremental modelling process, the initial partial model successively evolves into a *complete generic railway model.*

When modelling the first *layer*, the user can draw an arc in a GEM-like (Digital Research, 1986) fashion, i.e. by pointing to its origin and dragging it wherever its terminal point should be. Two nodes are then automatically generated at its two ends, unless one or both of them are already there. For the needs of the last two layers, we provide an editing facility that allows the modeller to move the cursor so that it traces an arc. As the cursor advances along the arc, he/she can add signals or edit the track circuit joints.

Attributes may be associated with system elements such as arcs, nodes, signals or track circuits. The modeller is able to add or edit (e.g. delete) system elements and to modify their attributes (e.g. the maximum train speed on a piece of line represented by an arc), by using a user friendly editing facility. In order to select the item he/she wishes to perform an editing operation on, the modeller only has to move the cursor near the desired element and perform an operation that selects the "nearest element". The system subsequently responds by identifying and highlighting that item. The user can then act on it. *Zoom-in* and *zoom-out* facilities are also provided, in order to allow for the viewing and accurate editing of those parts of the layout which exhibit a higher degree of complexity.

Another important feature of the layout editor is the incorporation of two separate coordinate systems: a conventional two-dimensional coordinate system with its origin at the bottom-left corner of the layout and a railway-specific one-dimensional system with the *datum* serving as its origin. They both provide a way of referencing each point of interest on the network (e.g. the junctions, signals, track circuit joints, etc). However, while the former is used when it comes to processing these points graphically and drawing the layout on the screen, all other non-graphical internal processing is based on the latter. Analysts who construct simulation models take most of the railway-specific coordinates the model requires, directly from railway maps; they then calculate the remaining, by interpolating between points with known coordinates. This can be done more easily with one-dimensional coordinates, than the case would have been with the conventional two-dimensional coordinate system.

### 3.2 The Icon Editor

The functionality of the layout editor is complemented by an icon editor. The purpose of the icon editor is to allow users to define, create and edit icons as elements of a railway graphical model. The term "icons" encompasses all those features of the graphical model that present a complexity which can not be dealt with the editing functions that the layout editor provides. Magnified bit-maps of these layout features can be edited with the more powerful functions (e.g. *draw* a line or a circle, *copy*, *move*, *clear*, *fill*, *flip* vertically or horizontally a region of an icon, *mix* two images, etc.) of the icon editor. Once in their final form, these images can be loaded and integrated with the network layout by using the layout editor. Such icons may represent both static (e.g. signals) and dynamic (e.g. trains) entities, which are supposed to complement the primitive network elements (i.e. the arcs, nodes, track circuits, etc.) in the graphical model of the railway system in question. The editor is flexible enough and can handle images of variable shape and complexity. Figure 3 gives an idea of the modelling environment that the combination of the two editors offers.

As a conclusion to the discussion of the editing modules, we can say that the structural and functionality philosophy of the modelling environment's user interface is founded on a set of *primitive graphical constructs*. Depending on their nature, these can be classified in different groups, namely a set of *primitive physical constructs* (i.e. the *arcs*, *nodes*, *icons* and *track circuits*), a set of *primitive logical constructs* (i.e. the *signal blocks*) and a set of *operators* (i.e. *add*, *modify*, *delete*, *select* and *zoom*). These are supported by an underlying editing infrastructure (i.e. the two-dimensional and linear coordinate systems, a *machine-independent hardware interface*, a *windowing environment*, a *cursor* and a *help facility*). The main objective in building this module was to transform the railway model building from a tedious and error-prone task into an enjoyable game. We consider user friendliness and reliability to be two essential features of this module, whose significance stems from the fact that it constitutes the main "direct access tool" to the DKB, which in turn is the core element of the system.

### 3.3 The Data-Knowledge Base (DKB)

The DKB is a data/knowledge pool where different generic railway models may be stored. The generic models stored in the DKB may be complete or partial, at any intermediate development stage. A complete model contains all the data, rules and general knowledge that constitute what we call a *generic description* or *generic model* of the system in question. The DKB plays a central role in the system, since all other modules may interact with it. It thus provides a way of integrating all useful and relevant, to railway model building, information into a single data base. This way it provides the means of increasing the accessibility and potential usefulness of this information.

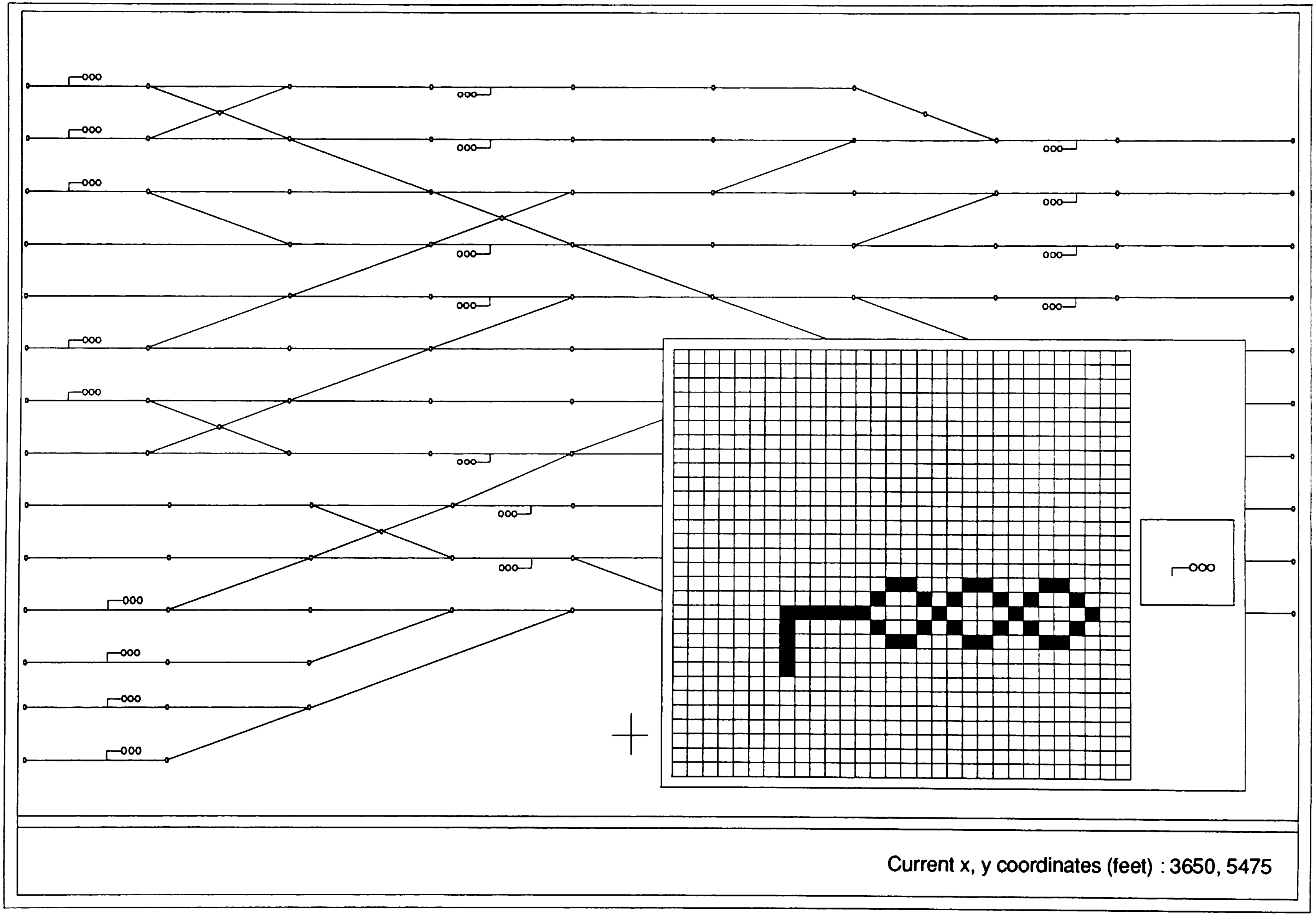
Our main considerations while designing the DKB were the following:

- To achieve a reasonable compromise between access times and storage requirements, with the emphasis placed on the former. The rationale behind this approach is that, although speed is useful in computationally intensive areas such as automatic control infrastructure generation and route selection, the total storage requirements are likely to be limited, since the system is to be mainly used for the detailed modelling of relatively small areas of high complexity (e.g. terminal stations, complicated junctions), as opposed to large railway networks.

- To provide the capability to store as many of the geometrical and operational characteristics of a railway network as possible, no matter how many of them will actually be used for the purposes of the current project. This way it will be much simpler to extend the system later to cover other areas of railway-related decision making without having to make extensive modifications to the database structure.

- To come up with a data definition which presents an "open" and clearly defined data interface to the other modules and application programs, while maintaining a lower level structure which is totally independent from theirs. This will provide the flexibility required in establishing the network database as the heart of a railway decision support system (DSS), where software modules servicing different decision support needs can be easily "plugged in".

- Since the system is aimed at being used and, if necessary, amended or modified later by BR OR staff, it is equally important that the structure is easy to understand and maintain from both the system developer's and the user's point of view.



Current x, y coordinates (feet) : 3650, 5475

**Figure 3: The Network Modelling Environment Provided by the Layout and Icon Editors**

A detailed description of the DKB structure being outside the scope of this discussion, we shall only note that, at the lower lever, the DKB is managed by a relational data base management system (RDBMS), while at a higher level it presents an *object-oriented* flavour, in accordance with the *entity-relationship model* (Chen, 1976). Indeed, an object-oriented model development environment is well suited to this particular application, due to it being inherently object-oriented (the system elements such as the track sections, the junctions and the signals, and the corresponding modelling primitives such as the arcs, nodes, etc, are readily identifiable *objects* or *entities*). The benefits that object-oriented model development brings to simulation, have been already discussed by some researchers (Thomasma and Ulgen, 1988; Petty, Moshell, and Hughes, 1988; Bezivin, 1988).

The data structure model chosen for the representation of the network structure follows the *arc-node* paradigm (Star and Estes, 1990). When an entity is created in the data base, is given a unique ID which is, in most cases, automatically generated. An entity may also have a number of attributes (a signal, for example, will have a number of aspects associated with it) and may selectively enter into relationships with other entities. Relationships can be of various types. The relationship between an entity and its unique ID, for example, is an *one-to-one* relationship. Relationships between entities, however, are usually more involved. Although a signal is always tied to a specific arc, for example, an arc may have more than one signal sitting on it. This represents a *one-to-many* relationship. Finally, we may also have *many-to-many* relationships, as in the case of the tracks and the track circuits: a track circuit may lie on more than one arcs, but we may also have parts of different track circuits lying on the same arc. It is very convenient for the application programs that access the DKB, if the data base automatically maintains *inverse relationships*, so for every *owns* link there is an *owned-by* link pointing to the reverse direction. One way we represent relationships is to treat them as attributes or *slots* in record structures, where the value in a slot of one entity's record structure is the ID of another entity. Although this approach works satisfactorily for one-to-one and one-to-many relationships, it can not handle efficiently many-to-many cases, particularly where inverse relationships come into question. In this cases we adopt a different approach that involves the introduction of a separate file structure which, by storing all interacting instances of entities in pairs, reduces the many-to-many complexity to a much simpler one-to-one problem. Most of the file structures are also indexed, usually on an ID field, in order to increase the speed of the file searching operations. An alternative hierarchical approach for representing the structural interrelationships between the different elements of a railway network is discussed by Giger (1987).

As a conclusion we may note that all information about an entity (apart from some of its entity-to-entity relationship information) is logically clustered together and can therefore be retrieved as a whole. This contrasts with the typical relational data base organisation, where information about an entity may be fragmented over several tables, and can only be gathered together by a series of *join* operations.

### 3.4 The Automatic Model Specification and Generation Modules

Once a graphical model of the network topology, including the positions of the signals, has been constructed and stored in the DKB as an initial partial model of a railway network, we can apply the first of the two automatic model specification procedures in order to include all the safety installations into this partial model. This procedure is implemented by the track circuit generation module. Heaton (1975) and Gill (1986) have already addressed the issue of automatic safety installations positioning on the much less complicated layouts of the rapid transit railways.

The process of dividing a network in a number of track circuits is based on a set of safety and operational efficiency rules that signalling engineers take into consideration when they manually perform this task (Nock, 1980). Although they do not compromise on safety at all when doing so, there is always a compromise involved between operational efficiency and installation costs. Let us consider, for example, an extreme case where the whole network is served by one single, huge track circuit. Although this arrangement offers absolute safety while requiring minimal installation costs, it is a nuisance from the efficiency point of view, since it always allows only one train to move on the network (because of the extremely coarse resolution that this arrangement offers, there is no way for the control system to determine where exactly the moving train is positioned on the network at any particular time; therefore, the signalling system has to ensure that no other train is allowed to enter the network). Let us now consider the opposite extreme case, where the network is divided into a huge number of track circuits, with each one of them occupying a tiny track section. The interlocking system can again be adjusted so that it ensures that no train collision is possible. Moreover, the system efficiency will this time be satisfactory. However, the costs and complexity of the installation will now be extremely high. Thus, what we are looking for and what is actually being done in practice, is the positioning of the track circuits so that a "good" compromise between these two extreme situations is achieved. As we can imagine from the preceding discussion, the algorithm that tries to achieve the "best" track circuit layout is essentially heuristic and is based on rules that are employed in practice by the railway practitioners.

After the track circuits have been put in place, we can initiate the execution of the second automatic model specification procedure, i.e. the route generation module. This module infers the signal blocks and the potential train routes from the, now enhanced, partial model. The result of this process is a complete generic railway model. The main technique that this module incorporates in order to achieve its objectives, is a recursive tree search through the network representation stored in the DKB. Finally, the complete generic railway model is converted to a simulation model coded in HOCUS with the help of an appropriate code generator.

## 4 CONCLUSIONS

The main objectives of the computer modelling approach presented in this paper, are to simplify, speed up and integrate the computer-assisted development of railway simulation models, and to overcome the deficiencies of the traditional modelling methodology, which was briefly presented. The incorporation of the generic model generation phase in the model development cycle, also provides a basis for the rapid development of models serving a variety of decision support needs. These models do not even need to be related to simulation. The realization of these targets provides a framework for improving the cost-effectiveness of developing and using models within an organization.

Speed and simplification of the model development process are interrelated and can be thought of as direct outcomes of the incorporation of intuitive man-machine interaction facilities. The proposed approach introduces flexibility in the simulation model building process and allows for rapid and easy model construction and amendment. The user avoids the tedious conventional model building methods by having the opportunity to define the model specification in a much more natural and less abstract way. This brings the additional advantage of improved model robustness and verification. Rapid model development also enables early experimentation to take place on a prototype model. At the same time, the ease and speed with which models can be modified allows for a greater number of design options to be evaluated.

Integration comes as a result of the incorporation of a central DKB that contains entries with different generic models encapsulating all the structural and operational characteristics of the railway systems concerned. The dynamic nature of the DKB and its open data interface contribute towards making these generic models readily available for modification, extension or conversion to specialised ones. With their help, the starting point for the application of any specialised modelling process (e.g. simulation) is much advanced towards the complete model; therefore allowing for much faster and, in some cases, even automatic modelling. Another advantage that the intermediate construction of generic models presents, is the fact that these models accurately represent the structure of a railway, as opposed to the traditional HOCUS models which, as we saw earlier, look more like exhaustive lists of all possible train routes through the network; not least that these lists have to be manually constructed by the modeller, with the realistic possibility of missing out some valid routes, especially where complex networks are concerned.

The system under development not only has the potential to enhance the decision support capacity of existing and well established decision support tools (like computer simulation of railway operations and computer-aided train routing optimisation) by drastically reducing the time required to develop the necessary network models, but also introduces computer-assisted decision support in areas such as safety installations positioning and network design of complex intercity railway networks. Apart from their theoretical interest, these techniques become even more effective and practical from the decision support viewpoint if we consider their integration with information technology advances such as interactive graphical modelling, relational database management and object-oriented system development. Such a research framework may present an opportunity for the development of a broad railway DSS expanding incrementally around an integrated DKB kernel that presents an open data interface, where different OR modules could be "plugged in and out". Such a system could address most aspects of railway related decision making and make a substantial contribution to the railway manager's decision making capability in areas such as investment appraisal and capacity planning.

## ACKNOWLEDGMENTS

We would like to thank the Operational Research Division of BR for initiating and financially supporting the project, and for their contribution in providing invaluable data and feedback during the system development. We would also like to thank the Science and Engineering Research Council for their financial support.

## REFERENCES

Bell, P.C. 1985. Visual interactive modeling as an operations research technique. *Interfaces* **15, 4**:26-33.

Bezivin, J. 1988. Design and implementation issues in object oriented simulation. *Simuletter* **19, 2**: 47-53.

Chen, P.P. 1976. The entity-relationship model - toward a unified view of data. *ACM Transactions on Database Systems* **1, 1**: 9-36.

Digital Research Inc. 1986. *GEM Draw Plus*, Second Edition. Monterey, California: Digital Research Inc.

Giger, P. 1987. A data concept for simulation of railway networks. In *Computers in Railway Management*, eds. T.K.S. Murthy, L.S. Lawrence, and R.E. Rivier, 67-76. Berlin: Springer-Verlag.

Gill, D.C. 1986. Computer-assisted design of optimised signalling layouts for rapid transit railways. PhD Thesis, Department of Electronic and Electrical Engineering, University of Birmingham, Birmingham, England.

Heaton, M.W. 1975. Computer aided design of signalling circuits. In *Proceedings of the Institution of Railway Signal Engineers*, 92-110.

Howard, S.M., L.C. Gill, and P.J. Wong. 1983. Review and assessment of train performance simulation models. *Transportation Research Record* **917**: 1-6.

Hurrion, R.D. 1989. Graphics and interaction. In: *Computer Modelling for Discrete Simulation*, ed. M. Pidd, 101-119. Chichester, England: John Wiley & Sons.

Hurrion, R.D. and R.J.R. Secker. 1978. Visual interactive simulation - an aid to decision making. *OMEGA* **6, 5**:419-426.

Jones, R.D., D.C. Lach, and J.W. Metsos. 1968. The CN network model. *2nd Conference on Applications of Simulation*, 66-69. New York.

Kirkpatrick, P. and P.C. Bell. 1989. Simulation modelling: a comparison of visual interactive and traditional approaches. *European Journal of Operational Research* **39, 2**:138-149.

Mathewson, S.C. 1985. Simulation program generators: code and animation on a PC. *Journal of the Operational Research Society* **36, 7**:583-589.

Mathewson, S.C. 1987. Draft/Draw/SSIM - an integrated network based toolkit for simulation. In *Proceedings of the 1987 UKSC Conference on Computer Simulation*, ed. R.N. Zobel, 154-159.

Minuteman Software. 1986. *GPSS/PC Reference Manual.* Stow, Massachusetts: Minuteman Software.

Nock, O.S. 1980. *Railway signalling.* London: Adam & Charles Black.

P-E Inbucon Ltd. 1989. *HOCUS Modelling Techniques.* Egham, Surrey, England: P-E Inbucon Ltd.

Petty, M.D., J.M. Moshell, and C.E. Hughes. 1988. Tactical simulation in an object oriented animated graphics environment. *Simuletter* **19, 2**: 31-46.

Raczynski, S. 1990. Graphical description and a program generator for queueing models. *Simulation* **55, 3**:147-152.

Reich, S. 1966. Computer study aids transit signaling. *Railway Signaling and Communications* **Feb**: 15-28.

Star, J. and J. Estes. 1990. *Geographic Information Systems - an introduction.* Englewood Cliffs, New Jersey: Prentice-Hall, Inc.

Thomasma, T. and O.M. Ulgen. 1988. Hierarchical, modular simulation modeling in icon-based simulation program generators for manufacturing. In *Proceedings of the 1988 Winter Simulation Conference*, eds. M. Abrams, P. Haigh, and J. Comfort, 254-262.

Withers, S.J. and R.D. Hurrion. 1982. The interactive development of visual simulation models. *Journal of the Operational Research Society* **33, 11**:973-975.

## AUTHOR BIOGRAPHIES

**KYRIAKOS TSIFLAKOS** is a research student in the Management School at Imperial College, London. His research interests are simulation modelling environments, graphical user interfaces and decision support systems. He is a student member of TIMS and SME, and a professional member of the Technical Chamber of Greece.

**STEPHEN C. MATHEWSON** formerly in the Department of Management Science at Imperial College, is now an operational research project manager at the BAA.