# A REQUIREMENT DEVELOPMENT AND RISK REDUCTION ENVIRONMENT FOR MAJOR SYSTEMS APPLICATIONS

J M Wise* and J Allan**

* The University of Birmingham
** The University of Birmingham and Brown & Root

## 1. INTRODUCTION

This paper reviews the current status of a PhD project which aims to create a requirement development and risk reduction environment for major systems applications. The first section of the paper provides an overview of the project and the second discusses the early approaches to the problems involved. The current design ideas for the environment are explained in section 3 and discussed in section 4. Section 5 provides the conclusion.

### 1.1 Origin of the Project

This project has arisen from a perceived gap in the development process of large-scale systems, particularly large railway systems. At present, many methods and tools exist for developing and modelling prospective systems or evaluating changes to systems. However, even a brief analysis shows that none brings together all the various techniques available to provide an integrated development environment for assessing systems. Almost all occupy their own specialist area with little regard for how they might be used in conjunction with others.

The use of systems engineering on projects to ensure that they are 'right first time' should be seen as a comparatively low cost option compared to the overall cost of the system and the cost of retrospective changes. However, its uptake tends to be limited to very high cost or high risk systems. Even in relatively small scale or lower risk systems the cost of failure can be catastrophic, but this is rarely taken into account and the lack of any defined systems engineering process is presented as a cost or time saving measure. However, when the lifetime cost of the system is measured, the lack of that process is likely to be a significant factor in the additional costs faced by those running the system and, should it fail, the potentially huge costs incurred as a result.

At the inception of the project, particular deficiencies were identified in the current range of systems engineering tools. These included the management of risk, the modelling of systems, the progressive validation and re-use of models and the development of requirements as a consequence of modelling. In addition, visualisations of the current state of a system and the system-wide effects of change were not part of any existing approach.

### 1.2 Aim of the Project

The aim of the project is to create the type of development environment unavailable elsewhere. This environment should be simple to manipulate and provide a tool set capable of assessing and managing a system over its entire life-cycle (from initial concept through to decommissioning). It should also be designed so that it allows non-experts to assess a system as easily as those involved in its creation.

The ideal environment would have a very simple purpose. It would allow an interested party to make an enquiry about a system and provide them with an appropriate answer. Ideally, the enquiry should be allowed to be expressed in any reasonable, unambiguous form and the answer returned in a form which best suits the enquiry and the enquirer.

As well as enquiries, the environment should also be able to deal with requests to modify system information. The status of the requester would determine whether they would be given permission to modify the specified information, but a record of the request would be recorded to ensure that it would receive attention or approval. This record could then be used in dealing with future enquiries or modification requests.

The environment should be able to handle static and dynamic information within the system representation and be able to take such information directly from the real system that is being represented so that it maintains an accurate representation for use in answering enquiries.

At this level, there is no reason for the interested party to get involved with the intricacies of either the environment or the system which it presents. The mechanisms involved in translating their intention into a meaningful and understandable result should be as transparent as possible. Any enquiry they make about the represented system should present them with the result they would have obtained from the real system.

The current design provides the basis of an environment that can cope with all the complexities of real systems without transferring that complexity to the user. The basic principle is not to try to simplify the systems under consideration, but to provide users with tools that

filter out the underlying complexity, enabling them to focus on the task that they wish to perform.

The four level structure of the environment consists of a system representation, structures to present that information, a modelling and simulation layer and a user interface. It will incorporate a flexible tool set that combines a representation of a system's architecture and requirements with a set of general (and system-specific) modelling, simulation, assessment and presentation options.

Although the project is being carried out with a particular emphasis on railway systems, the issues it deals with are much more general and can be applied to any system.

## 2. INITIAL APPROACHES

Early work on the systems engineering aspects of the project was aimed at gaining an overview of the subject from which the role of the various tools and techniques could be assessed. These tools would form the basis of the environment and so it was important to make an informed choice. However, experiments to investigate different ways of organising a system, using the hierarchical structures on which most tools relied, found that none offered any lasting benefit.

The original organisational idea was to describe a system as a top-down hierarchy. A set of base requirements for the system could be used to derive further requirements and build up a system description. This could then be mapped to hierarchies to provide structural and functional information about the system. Unfortunately, as the system became more complex the simplicity of this representation could not be maintained. This structure, shown in figure 1, led to the duplication of lower level requirements, incompatibilities in the requirements derived from them, problems in tracing back up the structures and difficulties in organising the hierarchies.

An attempt at improvement involved dividing the structure into more manageable parts. It was hoped that by starting at a higher level than that of the target system it would be possible to get an overview of the target system in terms of the objectives of its containing systems (shown in figure 2). This would provide a view of the system that concentrated on purpose rather than implementation and would build up general structures that could be used as a basis for developing the more specific. Each general structure could then provide a means of comparison or assessment between alternative lower level structures. However, the duplication and conflict problems experienced in earlier structures could not be reduced and the new structures introduced problems of their own, particularly in defining the structure at the top levels. These difficulties led to the need for a different type of organisation for the environment's intended systems.

## 3. PRESENT STATUS OF THE PROJECT

Large, complex systems are, obviously, large and complex, so to represent such systems accurately the inherent complexity must be reproduced in the organisational structures. Using simple top-down hierarchies, the complexity reveals itself by complicating the hierarchical structures to a similar degree. The present design has evolved to provide the basis of an environment which can cope with all the complexities of real systems without transferring that complexity to the user. The principle is not to try and simplify the systems under consideration, but to provide users with tools that filter out the underlying complexity, enabling them to focus on the task that they wish to perform.

One influence on the design has been the desire to make the environment as flexible and extensible as possible. A second has been the growth of Internet technologies, which have influenced some of the modelling, simulation and user-interface implementations for the environment. A third has been increased exposure to the problems experienced by those dealing with the type of systems at which the environment is aimed and the wealth of different types of information required to build up an accurate picture of a system. The challenge was to incorporate all of these features without the result appearing unintelligibly complex.

Within any large, complex system, the huge quantities of information that need to be managed mean that the system representation must be optimised for a computer to manage. A computer-based representation is still only as good as the information provided to it, but, it will be able to check the integrity and validity of its information far more regularly and quickly than would otherwise be possible. Such a representation must be able to deal with all of the information already available and any new information that might become available. To do this it must have an adaptable, modular structure so that new types of information can be incorporated seamlessly into the representation and manipulated as easily as existing types.

Structures must be applied to this representation so that it can present its information in a form that can be understood by the people managing the system. Since the system representation should contain all the complexity and interconnection of the original system, it should be possible to interrogate the database containing the representation in order to generate organisational structures such as top-down hierarchies. These hierarchies need not suffer from the same duplication and conflict problems as those in the earlier approaches because they are constructed from information in the base representation. Each piece of information exists only once in the base representation and while it may be used in many different hierarchies and other organisational structures, all of these structures carry only references to the original. Similarly, conflict problems are avoided because any view of the information can be taken; changing the view

does not change the information, only the way in which it is presented.

### 3.1 General design

A working design for the environment (figure 3) is based on a four level structure containing a system representation, a set of information presentation structures, a modelling and simulation layer and a user interface.

The system representation is the lowest level and manages the collected information about the system as interlinked, unstructured objects. Above this, the information presentation structures use hierarchies and other structures to extract information from the representation to build up particular views of the system and present it in a readily understandable form. The modelling and simulation layer provides a means of evaluating a system, or changes to it, using the information presentation structures.

The user interface is at the top-level, bringing together all the information in a readable, or visual, format for the user. It will, primarily, allow the user to analyse and respond to models or simulations of the information provided by the presentation structures. However, it will also allow the user to look at the presentation structures and, if necessary, the base representation. Changes to the representation and presentation structures (as opposed to changing the information within them) would be considered as an administration function.

### 3.2 System Representation Structure

Figure 4 shows two representations of the information required when dealing with a system. The first shows that the target system has to be considered along with the effects on it from (and its effects on) its contained and containing systems. The second diagram shows how all of these can be represented as individual objects, unordered, but containing exactly the same information. In this view, an element is an object which contains information objects that store all the information required to describe it and its links to other elements. Elements are the lowest level of any organisational representation (the basic building blocks) and the links between them provide either a simple connection or a path for the flow of information

Any system must have a reason for its existence and that reason can be expressed as a function. To fulfil this function, the system requires resources which are expressed as components of the system. Both functions and components can be broken down through levels of increasing detail into their lowest level constituents. Once a function or component is mapped to an element, any additional information on it can be stored within that element. Hence, the representational elements mirror the system under consideration. To differentiate between the elements representing functions and components, different element object types can be used as shown in figure 5. These can then be sub-typed to provide further refinement.

In any large, complex system there are generally alternative implementations for each of its sub-systems. However, all the alternatives for a particular sub-system will be required to fulfil the same role within the system, so, any could be replaced by a generic, ideal model for the sub-system. Intermediate levels can also be created to generate a path from the ideal requirements for a system to the actual properties of its implementations through several levels of detail. The reverse path is also available, enabling the passing of specific data to the generic system for use in generating typical data. In a real system, several different implementations can be in use simultaneously and, by using generic systems of progressively greater detail, analyses can be performed using a suitable level of detail. At the top levels of the system representation, specialised generic objects provide a mechanism for representing the effects on the represented system from (and its effects on) those systems outside the scope of the representation.

The representation will store every possible view of the system and, hence, every possible route between the element objects. By ensuring that, in any particular breakdown, each element object can only see its immediate parents and children directly, the complexity of the routes can be minimised. A strict naming convention can be employed to provide a basic level of integrity checking and to ensure that the information passing through each element object is valid.

### 3.3 Information Presentation Structures

The information presentation structures are intended to present the information contained within the system representation in an accessible form. While the representation holds every possible view of the system, the presentation structures each show a section of one particular view. The system representation imposes no structure on the elements, but, it must contain all the information necessary to create structures based on the information stored within it.

The situation in figure 6 shows how a set of element objects have been taken and one set of hierarchical links between them have been isolated. In this case, the links represent a path from a generic element object to a set of more specific implementations.

The chosen route can traverse the representation in any order to build up any view of the system with any desired focus or emphasis. Different decompositions can be mixed together and a chosen view may skip levels within a hierarchy which are not relevant. The only limit is that the intended route must be made available by the base representation. If the route does not exist it must be created. The information presentation structures have no direct structural effect

on the representation and changing their structures only results in different element objects being used to provide the information.

### 3.4 Modelling and Simulation

Modelling and simulation capabilities will be provided as an extension to the basic functionality of the environment. The information presentation structures which provide a specific view on the system also contain (via reference to the element objects) the information necessary to generate models and simulations of that view. The task for a modelling or simulation tool is to take the required information from the presentation structure, manipulate it as desired and then present its results to the user for evaluation via an interface. The user will then have the option of storing any new information created within the system representation, storing the information presented along with the criteria used to create it or discarding the results. Changes to be kept within the representation can be made directly, via the presentation structures, or indirectly by commuting them to change requests to be approved at a later date.

### 3.5 User Interface

The user interface for the environment could be as simple or as complex as desired. Many tools support some form of automation, so it is possible to reduce lengthy repetitive procedures to simple, automated tasks, integrating the tools from a single interface. This process can be extended to enable remote automation of the full functionality of the environment.

The most significant recent change in this area has been the adoption of Internet (and intranet) technologies for gathering, analysing and disseminating information. These allow information to be distributed with greater flexibility and promote the ideas of common, platform-independent information standards. Since the environment is aimed at large and complex systems, and because these are commonly distributed over a wide range of locations, it seems sensible that access to this environment should be possible from all locations and by using the in-place systems.

If common formats are used to distribute information throughout the environment then the information can be read directly into any tool, or by a user. For example, HTML could be used to represent objects as pages, with hyperlinks representing the objects' links so that structures such as hierarchies can be represented. Extending this principle using technologies such as VRML allows the user to view the system through 3D visualisations. The ability to return information through similar mechanisms, combined with platform-independent, object-oriented programming languages such as Java, makes it possible to provide a remote user with a complete user interface. An example of such an interface is shown in figure 7.

It is clear that different users will use the environment for different purposes. A customisable interface would allow the individual rights and preferences of users to be imported and would provide each group of users with the capabilities they need to perform their tasks in a form which suits them most.

## 4. DISCUSSION AND FUTURE WORK

This environment is aimed at eliminating the problems which result in systems that are difficult to maintain, difficult to use or unable to fulfil the function for which they were designed. For example, maintainability and usability requirements can be inferred from containing systems of the target system. They can then be designed into the system from the start rather than added retrospectively. Improved visualisation of the system will ensure that compliance with such requirements can be checked more reliably and integrated risk assessment will help to ensure that the system design is correct before it is commissioned.

The effect of the completed system on its users can be judged by representing the users as element objects and designing the rest of the system around them and their interactions with the system.

The next stage of development will provide mechanisms for assembling and integrating the environment. The main activity is related to the core of the environment; the system representation. The current design takes an object-oriented attitude towards the representation of system information. This will provide the base for the application of knowledge representation techniques, which should be able to store, organise and retrieve any information related to the target system. To enable modelling and simulation of the system, static and dynamic information will need to be captured and special cases such as non-linear dynamic behaviour within the target system must be considered.

The generation of information presentation structures and the aggregation of systems from the identification of their interrelated components will rely on dynamic information which will change as the target system (and so, its representation) evolves. Thus, such processes will operate continuously on the representation to ensure that their interpretations of the target system will reflect its current state. The information presentation structures will, then, provide up-to-date views of the systems contained within the target system.

Modelling, simulation and assessment techniques are being investigated throughout the project, the main areas of interest at present being process modelling and risk assessment. Work on producing a user interface for the environment will combine the work on the automation of the environment and the distribution of its information to provide a consistent interface between

the user and the varying set of tools combined within the environment.

An ultimate aim might be to provide a complete, visual simulation of any view of the proposed system at any level of detail, driven by the information contained in the system representation. Approved changes and corrections made to (and on the basis of) the visualisation would be reflected in the representation and result in an updated system specification.

## 5. CONCLUSIONS

This paper has presented the current status of the project, a requirement development and risk reduction environment for major systems applications.

Existing methods of developing large systems have several deficiencies, particularly in regard to the assessment of risks associated with the intended system and visualisation of the intentions of its design. The new approach outlined in this paper will use an object-oriented representation to provide the basis of a full description of such a system. The basic representation will be extended to provide modelling, simulation and visualisation of the system at any level of detail and provide management of the system over its entire life-cycle.

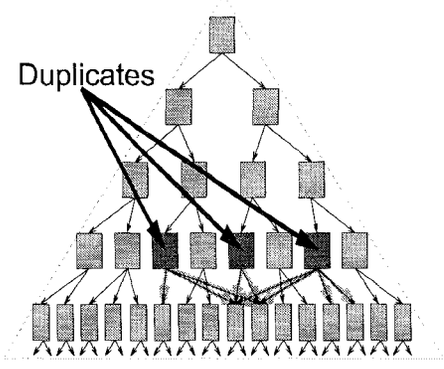


*FIGURE 1. SIMPLE HIERARCHY SHOWING A DUPLICATE*

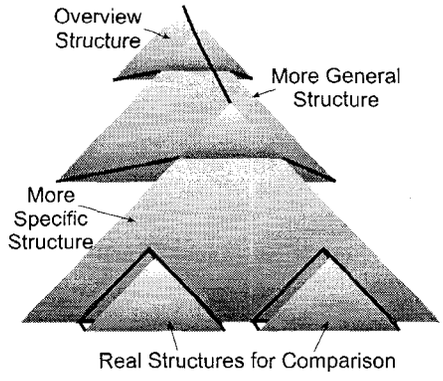


*FIGURE 2. USE OF MULTIPLE HIERARCHIES*

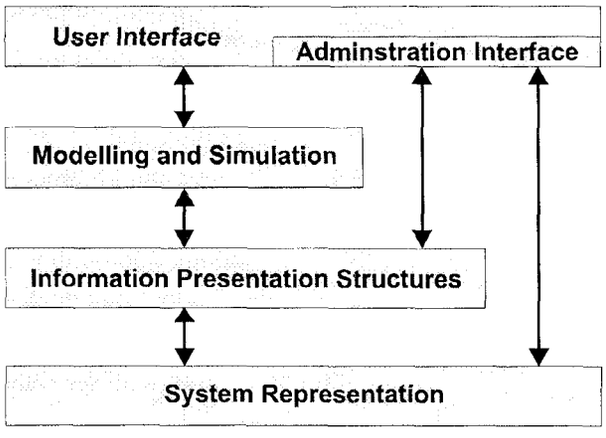


*FIGURE 3. PROPOSED ENVIRONMENT STRUCTURE*

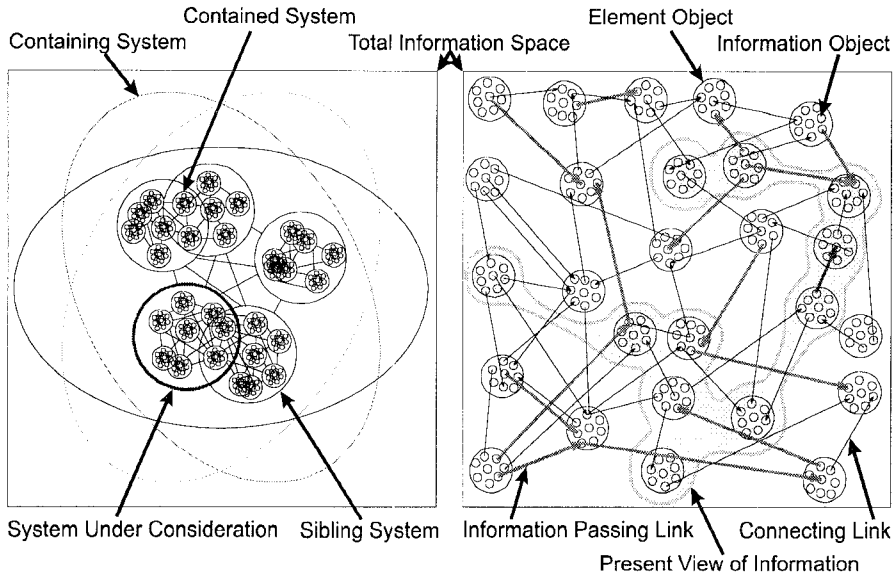


*FIGURE 4. STANDARD SYSTEM VIEW ELEMENT OBJECT SPACE*

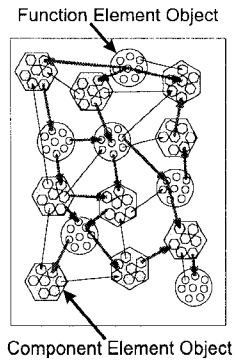


*FIGURE 5. SUB-TYPING OF OBJECTS*

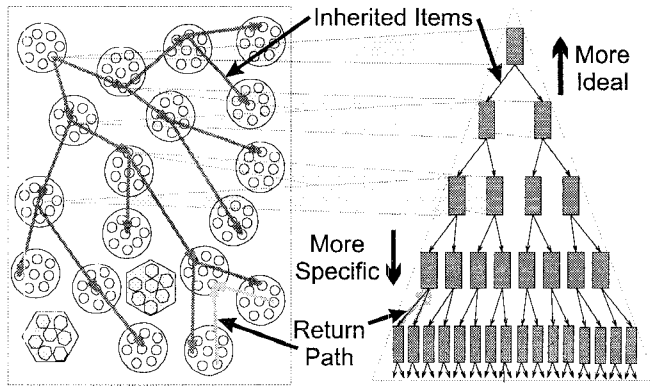


*FIGURE 6. AN INFORMATION PRESENTATION STRUCTURE*

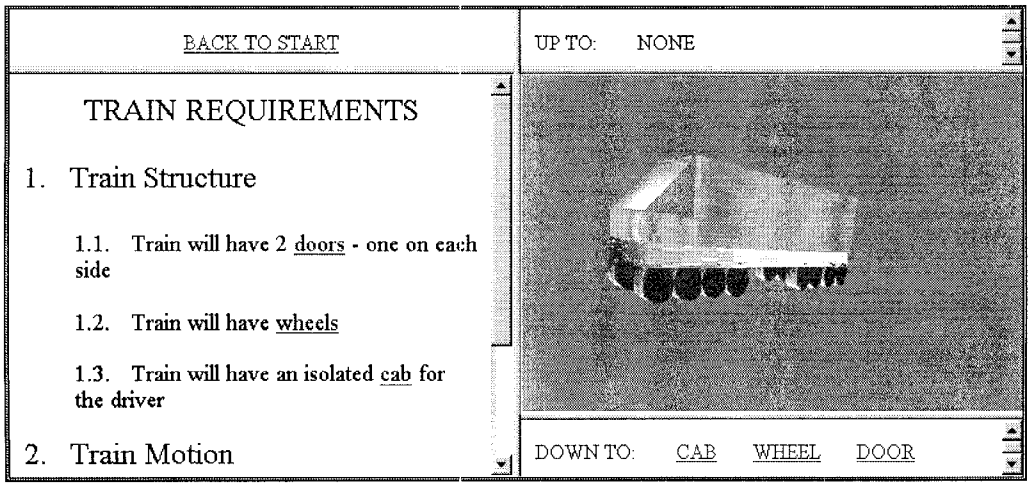


*FIGURE 7. AN EXAMPLE OF AN ACTIVE VISUALISATION SCENARIO*