# Active User Designs in Hypermedia for Better Simulation Model Specification

Lesley A. Gardner[1], Simon J.E. Taylor[1] and Nandish V. Patel[2]

[1]*Department of Computer Science and Information Systems at St. John's, Brunel University, Uxbridge, UB8 3PH;* [2]*Department of Business Systems, University of Luton, Park Square, Luton, Bedfordshire, LU1 3JU, UK*

## Abstract

A criticism of simulation modelling is that models are delivered late, over budget and not closely representing the functions of the target system. This paper addresses the issues of living information systems pertaining to the problem of model specification. It builds on earlier work regarding hypermedia storage of lifecycle documents to look at modelling end-user requirements for information system construction. This paper follows very closely the model of living information systems and proposes constructive ideas that may lead to the fuller development of flexible, better specified models for simulation.

## Keywords

Simulation Modelling, Living Information Systems, Tailorable Modeller, Hypermedia, Hypertext.

## Introduction

Many attempts to use simulation modelling as a decision making tool have tended to disappoint the end-user. Even after several decades of experience, model specification techniques still fall short of expectations. It is widely recognised that there are several major problem areas that are associated with the development of simulation models *per se*. This paper examines the problems associated with traditional modelling techniques within the context of more general information systems development and suggests an alternative approach.

Nance (1995) comments that the factors which influence the acceptance of, and trust in, the results of a simulation are extremely complex. Many excellent examples of how user confidence can be increased exist in the literature (Balci, 1995; Law and Kelton, 1991; Robinson and Pidd, 1995). However, the main focus of this work concerns activities *after* the system under study has been conceptualised. It is perhaps the elicitation of requirements from the user group responsible for decision making in the system under observation that is the greatest problem in information systems and simulation model development today. Many of these problems arise through poor communication of actual needs and use of modelling techniques that cannot accommodate systems in a manner which is natural and intuitive to the end-user. Intimidation of the user by the analyst team and general misunderstanding by both the user and the analyst of terms and operating procedures within the organisation are just two examples of the possible pitfalls encountered during the initial stages of system design. Many other examples are reviewed by Holmes and Polymenakou (1995). This is particularly exacerbated when it is unlikely that users do not have the experience to determine what their own requirements are (Paul, 1993). Much of the bottleneck of information systems development occurs here (Gardner and Paul, 1994).

Through the development of living information systems and simulation models it is seen that there is a need for a much faster elicitation of user requirements (Gardner *et al.*, 1995) and thus a more urgent requirement to address the elicitation of user knowledge and experience (Gardner and Paul, 1996). This paper discusses these issues and proposes a hypermedia alternative within the context of simulation modelling.

An introduction is given into the problems and deficiencies encountered in the specification of information systems and simulation models. We introduce the Living Information Systems Thinking paradigm and living information systems. A possible hypermedia solution is considered and a tool called Hyper-Tmodeller is introduced and the functionality of this tool is developed. The paper concludes with a critical discussion and the conclusions to the paper.

## Traditional Information Systems Development and Problems in Simulation Model Specification

There have been many notable failures in information systems and simulation modelling (Paul, 1993) through the last decade. From the literature it is evident that there is a great deal of user frustration. This may be broken down into several areas. Firstly, new simulation systems are delivered late, since the developer may have grossly underestimated the length of time taken to elicit requirements successfully. Secondly, frustrations are also caused by disappointment in what the system will actually do when it is delivered. Finally, the frustration caused by the system not performing the functions initially required, or not reflecting current system organisation.

It is clear that the major cause of frustration is the inappropriate methods used in elicitation of user knowledge. Users feel uneasy, threatened and unable to contribute fully to the design process. Misunderstandings are easily caused when terminology is unfamiliar and needs careful explanation. It is understandable that confusions arise and Paul and Hlupic (1994) summarise the problems associated with using simulation modelling. The themes that emerge have much in common with those associated with more general information systems development.

Many systems to which one applies simulation are poorly defined. The result of this is that the specification of a model based on such a poor definition must be performed in a way which suits the capture of the subtleties and nuances that exist in the real world. A method which forces a decision maker to compromise his or her notion of their real system could lead to inconsistencies which would lead to an incorrect model specification. Care must be taken to allow users to express their conceptual view of the system being modelled in a manner which is natural enough and expressive enough to satisfy their needs.

Any complex problem in a system which is important will more than likely involve conflicting interests and understanding. If the modelling process is going to lead to change in the organisation then one must expect that it is unlikely that all decision makers will see these changes as being favourable. Negative attitudes and spoiling tactics must be anticipated as a territorial reaction to what they perceive as the ultimate aim of the simulation modelling exercise. Care must be taken to use the modelling process in a neutral way to help the participants in the decision making process understand their problem and come to a resolution amongst themselves.

Another major problem in the simulation modelling process is that of model confidence. Typically, validation and verification techniques are used to guarantee to the user that the model specification mimics the real system under investigation as precisely as possible. However, models specified against a dynamic, real system cannot be successfully validated. The real world does not hold still. Models must be specified against the real world at a specific point in time. Precise validation of the model specification will therefore attempt to effectively match the past against the future. No model of any size can therefore be successfully validated against the real, dynamic world. Alternatively, the aim must be to use methods that demonstrate confidence in what the model is doing and the way in which it is doing it.

Observations made of so-called traditional information systems are also relevant to model specification. It is understandable that since traditional systems have to be designed and developed there will usually be a cut off point after which no further requirements and modifications may be made (Pressman, 1993). Given that business and the use of information is a dynamic, constantly changing, phenomenon, any static, time related information system will not relate exactly to the user requirements at any particular stage (Paul, 1993).

It is also likely that these requirements and expectations of a model specification will expand as a project continues due to users gaining more insight into their own systems as a consequence of participating in the design process. Traditional information systems development works with the premise that users know what they want from a proposed development and that, consequently, systems analysts are able to extract a systems specification from them. Living Information Systems Thinking (LIST) does not make the same assumption. On the contrary, the premise that LIST works with is that users cannot know what they want from a proposed information systems development. In this context Friedman and Conrford's (1989) observation is pertinent that: 'Someone has suggested that the plaintive cry of the user is "can't tell you what I want, but I'd recognise it if you showed one to me!"'. These comments are also particularly relevant to simulation model specification.

An information system, or indeed a model specification, may be more appropriate if it allows a more dynamic modification and development process where the user and analyst have much more of an understanding of the real world system under investigation and allows modifications and new areas to evolve. Prototypical examples of evolutionary systems (Mayhew and Dearnley, 1987) have been experimented with, resulting in both notable successes and failures.

This section has looked at the shortfalls and partial solutions to the information systems development problem with respect to simulation modelling. In the next section an alternative flexible approach to information systems development and modelling is introduced.

## An Alternative Flexible Approach to Simulation Model Specification

Flexibility within the development process is difficult to achieve whilst maintaining the discipline necessary to complete the development of a simulation model within a sensible time scale. This section looks at living systems theory and tailorable information systems (Gardner *et al.*, 1995) and how experience in these can be passed onto the problem of simulation model specification. It looks at the potential for these ideas to encourage flexibility within system development and the take-up of the developed system to increase model confidence.

Patel and Paul (1995) have proposed the Spiral of Change model (SOC model) as a useful theoretical view of the relationship among information systems, end-users, their organisational tasks and change. All these are considered to be variables in the SOC model, change being the variable that continuously affects the other three. Figure 1 shows the adapted SOC model with respect to model specification. The SOC model supports the view that model specification should be adaptable to the changes that take place in an organisation. Adaptability in information systems is considered to be possible through end-user tailoring of information systems, and Gardner *et al.* (1995) have proposed a theoretical framework for designing living information systems. In terms of model specification, the use of this flexible framework encourages the breakdown of conceptual barriers between the analyst and the decision makers.

## A Possible Hypermedia Solution

We now examine the possibilities that a hypermedia system may have in the elicitation of user requirements and the modelling of systems within end-

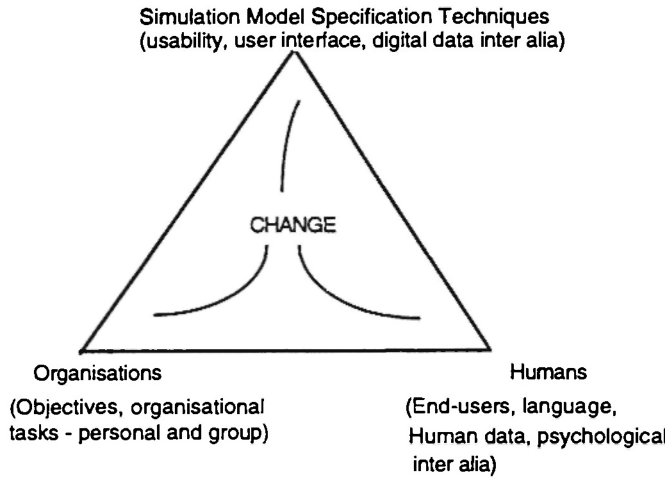


**Fig. 1:** The general spiral of change model of living information systems.

user tailoring. It examines existing work by Gardner and Paul (1994) and attempts to improve on the model through Tailorable Modeller.

Hypermedia is an information retrieval mechanism which allows flexible but robust information storage and retrieval via a network of nodes and linkages or relations (Neilsen, 1993). The node is the conceptual container of any media information (single or multiple types of data can be stored at the same node) which the user wishes to incorporate within a system. Conceptually as much or as little information as necessary can be stored at each node. The user moves from node to node via linkages. Linkages join objects together. Any point (word, phrase, graphic, sound, etc.) can be considered as an object and may be linked to another node or point within a node. It follows that any node may have one or more linkages joining it to any other node within the network.

## FIELD

The **F**ully **I**ntegrated **E**nvironment for **L**ayered **D**evelopment (**FIELD**) (Gardner, 1991) is a hypermedia documentation mechanism to hold all information relating to the development of an information system. The overall structure of the system was based around the stages of the life cycle, i.e. Feasibility, Analysis, Design, Coding, Testing, Implementation, Maintenance and Evaluation (Gardner and Paul, 1996). The development of FIELD provides contextual background, by which it was informed, to Hyper-Tmodeller described below.

The purpose of FIELD is to hold all information and documents created during the process of information system development. It was designed with the view that it would be the main depository of information for a particular development. It is well known that during the development of an information system documentation is generated on an enormous scale. The ability to store all documentation in an accessible fashion is often difficult. By storing this information on-line, the resolution of problems becomes more straightforward (Gardner and Paul, 1994).

Each stage of the life cycle has a separate functional unit or layer to support its needs. In some cases there may be several sub-sections which must also be supported. For example, during the stages of analysis and design, data diagramming in terms of data flow diagrams, logical data structures and entity life histories must be supported. The analysis and design stages must support the provision of graphical editing tools to allow the incorporation of diagrams and associated information from traditional CASE tools. Graphics tools should also allow the creation of diagrams within the system so that the user does not have to use another CASE tool (Gardner and Paul, 1995).

Such a tool dynamically shows how information flows through the system. It is a tool that relies heavily on the navigation metaphor based on linkages. Such a metaphor is designed to allow an information systems development tool to hold information. The mechanisms by which this information is to be accessed must also be designed to provide maximum benefit to the developer and maintenance team. The first item that must be decided is the correct types of linkages to be used by the system (Wright, 1989). Hypermedia uses linking to describe relationships between items (Marshall and Rogers, 1992). In terms of information systems development these linkages needed to be defined. The developer is unlikely to want to create these linkages manually, so the system must force such linking to occur automatically. The user intervenes in the process where more complex relationships occur. Automatic integrity checking is appropriate to make certain that each object either has a linkage or a null link attached to it. Such an information system should automatically be able to translate the linking patterns of information and files from other tools and draw up parallel relationships for inclusion into the relevant part of the model.

The linking strategy shows the relationships between the observed systems and the new design system. In many ways this is similar to prototyping, but with the advantage that the user is able to go back to the original observed system and cross-check that all parts of the system are being allowed for in the new system. The overall architecture of FIELD is shown in Figure 2.

There are a series of links which need identification. The first of these is *between-layers linking*, where relationships are drawn between layers of the development stages. These help track and examine the likely effects of major changes within the system. *Within-layer linkages* provide important relationships between certain parts of a subject area, showing more detail of certain features of the logical relationships between certain objects. Within all stages of the information system development tool, there are times when the developer and maintenance people are likely to want to create notes to themselves, perhaps as comment notes for programming. Therefore notation links are included within the system. These are called annotation linkages (Gardner and Paul, 1996). For brevity, these are not shown in Figure 2.

The final stage that this system must address is that of its documentation standards. There is much debate about this in the world of systems analysis. This should, therefore, be designed as a practical tool which allows the work of the development team to be examined in a structured fashion both on-line and on paper. The use of documentation as milestones varies in volume and frequency depending on the development metod used. It is necessary for some documentation functions to be met by query on-line facilities and others must be produced in hard copy. The production of consistent documents is highly important and must be supported (Gardner and Paul, 1994).

The functionality of FIELD is interesting and as far as its inherent life cycle model will allow, it gives the flexibility and structure needed to maintain the development of a traditional information system. It is however unsuitable for a tailorable system. In this light the idea for Hyper-Tmodeller has been developed. Tailorable-Modeller (Hyper-Tmodeller) is discussed next.

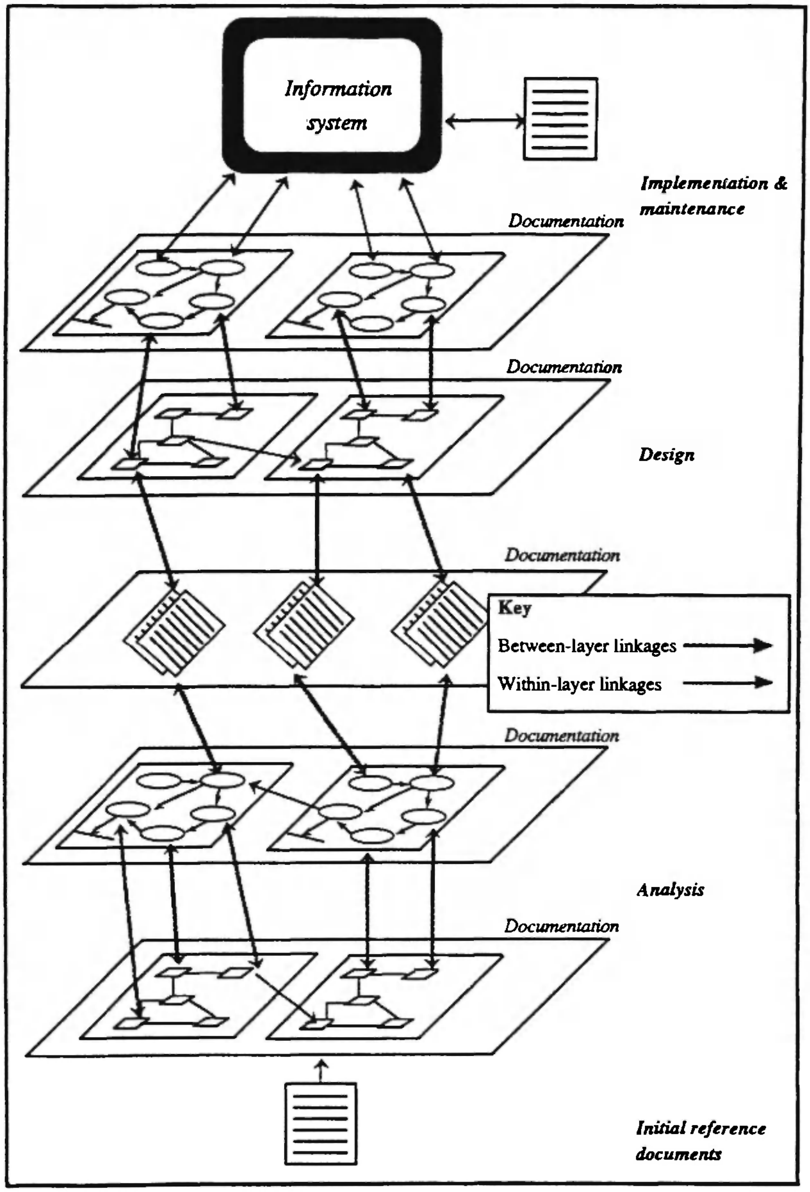


**Fig. 2:** The FIELD system architecture.

## Tailorable Information and the User

The purpose of Hyper-Tmodeller is to enable users to explore their needs from a living information system. The SOC model suggests that users' work environment is continuously changing; such an environment cannot be served by an information system or a modelling technique that is not responsive to changes in information needs. There is thus a need for tailorable information for users, which living information systems seek to provide (Patel, 1995).

Since LIST's basic premise is that users cannot know what their requirements are, let alone what their tailorable requirement might be, we need to provide a living design medium. Such a medium needs to be flexible, hence the use of hypermedia for the design of Hyper-Tmodeller. Hyper-Tmodeller is proposed as a living specification tool that can be used by users and analysts, either individually or jointly.

## Hyper-Tmodeller

Hyper-Tmodeller is designed to enable the end-user to conceptualise the system whose specification and development they participate in. It does this by allowing the end-users to explore their working environment and to have access to a graphical interface in which the organisational structure of their business practice can be seen. From this structure there is the possibility that they may build a better cognitive model and a deeper understanding of their business practice. As the end-user understands what is happening it is postulated that they will be better able to verbalise how they go about their work, who they talk to and what they actually want the system to do. By using such a tool the end-user is more fully involved in the specification process.

Given that the end-user interacts with the system and attempts to draw relationships between the activities they do and other activities and groups within the organisation, it is possible to see that the Hyper-Tmodeller may become a repository for information on the system being specified. If such information could be "picked up" by the modeller, it may be used as the basis for discussion and specification. Given that system detail may be remodelled dynamically it may be possible for the end-user to remodel their ideas and correct earlier ideas, and therefore be possible to get closer to an initial understanding of a potential model specification faster and less painfully than in the past.

As the system is modelled by interaction between end-user and analyst, it is possible for recent changes to be identified and, through dynamic re-modelling, updates to the specification to be created. Where changes to these models become minor the specification may be developed and documented (using a hypertext documentor, e.g., FIELD) as a living information system. The following section discusses the specification of end-user modelling and Hyper-Tmodeller.

## End-User Modelling

The concept of end-user modelling is based on the Spiral of Change model (SOC model) and can be implemented in several ways. One, it may mean that the actual modelling of a potential living information system is done by potential users of a system. Two, it may mean that analysts make models of end-users in their living situations, where all things are dynamic or likely to change sooner or later. Three, both users and analysts could work together, either starting from scratch or working from an initial model provided by analysts. End-user modelling encompasses the following:

- the process of modelling brings about greater understanding for modellers, analysts and users;
- there is an eventual, amalgamated model; the potential living information system;
- users learn about their own work environments while modelling; understanding processes and organisational issues, thus providing a learning environment;
- models provide suggestions of ways of proceeding; strategies;
- enables building of models interactively for users (managers, other employees);
- uses simple structures, diagrams, icons and other things familiar to or easy to learn for end-users.

**Tailorable Modeller**

Tailorable modeller (Hyper-Tmodeller) is based on the SOC model, and is one form of end-user modelling. The SOC model asserts that digital technology, organisational tasks, and end-users, the constituents of a tailorable information system, continuously change. Hyper-Tmodeller is conceived to cater for such change and allows both users and analysts to model a potential application area flexibly. Hyper-Tmodeller should improve the process of learning and understanding what needs to be designed into a living information system, by allowing users to be the actual designers.

The purpose of Hyper-Tmodeller is to help with the design and assessment of living information systems. Hyper-Tmodeller would enable analysts to understand users' needs from a living information system. The aim is to create models or representations of living situations and to use these to understand living information system architectures. The use of Hyper-Tmodeller would provide material, in the form of drawings, graphs, comments, and documentation, for analysts to investigate further an application area. This material would be modelling-users' own supplied material. Figure 3 shows an example of this.

Modelling-users can model their living situations using Hyper-Tmodeller, thus allowing users to be involved in the modelling (design) process. Hyper-Tmodeller should therefore be easy to use free from non-intuitive specification techniques. This can be achieved by use of visual reasoning/thinking, and graphical representations can provide this in Hyper-Tmodeller. The Hyper-Tmodelling process may be started off by either providing an initial tailorable model (Tmodel), to the modelling-user, modelled by analysts, to start the modelling process or requiring them to begin from scratch. This modelling process would be done in the Originator-Creator module in Hyper-Tmodeller.

The use of Hyper-Tmodeller should result in a visible Tmodel to base discussions around and exchange ideas of the required living information system. This would be a stage before actual model specification begins. It therefore provides a platform for everyone involved in the modelling process, in their multiple roles, to discuss the issues involved, while simultaneously working towards an actual specification. Thus no distinct phases are envisaged in this process – this is important to understand, since Tmodelling is not part of any methodology. Tmodelling is itself part of the living situation.

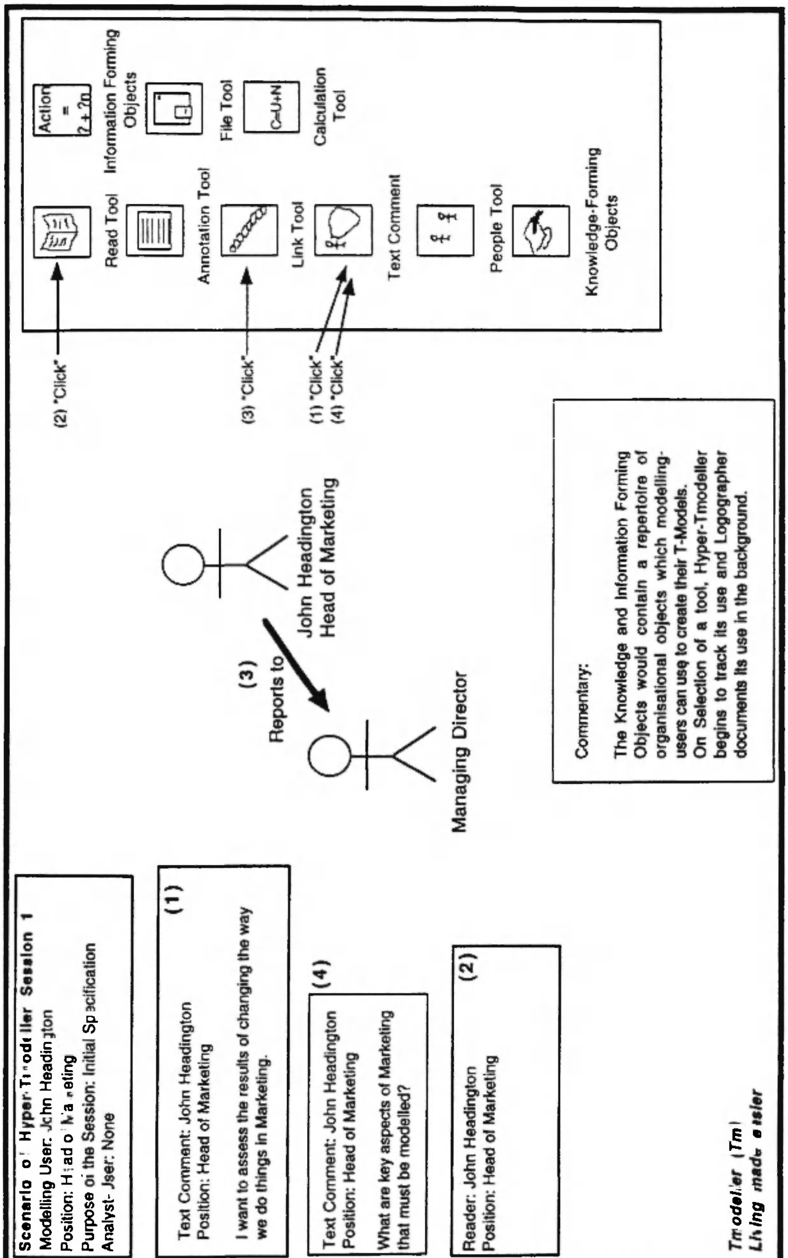


**Fig. 3:** An example of the use of Hyper-Tmodeller.

Hyper-Tmodeller may be used simultaneously by modelling-users and analysts, and within each user type many such users can simultaneously use

Hyper-Tmodeller may be used simultaneously by modelling-users and analysts, and within each user type many such users can simultaneously use Hyper-Tmodeller. All these different views of a potential living information system need to be amalgamated to provide an overall view. This would be done by the living information system analyser module. The module would provide documentation of all the tailoring activity.

## Hyper-Tmodeller's Modules

### *Originator-Creator (O-C)*

This module creates the initial tailorable model (Tmodel) through discovery. An initial model may be postulated by analysts, which modelling-users can amend to suit their perceptions of what they think is happening in the potential application area; the living situation. Modelling-users can add or delete aspects of the Tmodel by using the various types of links suggested or they could create their own link types. If an initial model is provided, users could be invited for their opinions or comments against this model thereby providing the basis for modification to the Tmodel. Alternatively, users could begin by drawing their own diagrams (initial Tmodel).

### *Discover-Designer (D-D)*

Once the initial Tmodel has been created, users and analysts can add or delete to the Tmodel further, according to their perceptions of their living situations. This is the module where all the tailoring would happen, so all the required tailoring tools (Ttools) would have to be provided here.

The thinking underlying the SOC model is that most aspects of a living information system will change – changing users, organisation, and technology. Since change is all-pervasive in a living information system environment, Hyper-Tmodeller should contain a functionality to cater for change and its modelling. Some features are listed below:

- appropriate icons
- moving of icons – as in desktops
- each move, if causal, connects in the background
- talk-links; I-talk to X for such and such a reason
- need-links; I-need X, Y from him or her
- browse-links; I-browse through this or that document for information

- information-links; I-need this or that information, now and then, regularly, etc.
- file-links; I-file this or that here or there
- text comment
- diagrams – diagramming
- graphs – graphing
- drawing palettes – either floating or fixed or an option for either.

*Text*

Text in Hyper-Tmodeller could be used in various ways. It could be used to provide a comment facility to capture users' thoughts as they are designing. Users could make text comments. Analysts could provide text comments to the initial Tmodel, if used.

When modelling, users may model something that is not technically feasible. Text comments would ask the modeller-user or analysts to reconsider their designs or suggest alternatives. This facility could be used in Hyper-Tmodeller to let users know the technical limitations or capabilities of living information system technology or of what they are modelling.

*Diagrams*

Diagrams allow users to view, modify and create pictures of their living situations (Tmodels). Visual thinking is useful to provide in Hyper-Tmodeller because modelling-users would be able to "see" their living information system. Visual images also enable sharing ideas publicly so that misunderstanding is minimised.

It is possible to add questions concerning changeability of things being modelled; how likely are they to change? Data thus collected would be useful to design required adaptability and tailorability. Rough measures (or precise measures if available) could be stated: high, medium or low, and modelling-users would select one for each of the things they model, giving technical designers an idea of how to write the relevant programs.

*Policies and issues*

At present four types of structural links are envisaged which modelling-users can use to model:

- Procedural links: the procedures of their organisational tasks

- Process links: what processes are involved in their organisational tasks
- Causal links: what events lead to other events
- Policy links: what policy is used to do a particular organisational task.

These are more important in living information systems. Policies are typically realised as inflexible constructs in a model specification. When policies change, as they do quite frequently, traditional techniques are unable to cope with the changes. In Hyper-Tmodeller such policies would be identified and made tailorable.

*Graphs*

Graphs allow modelling-users to view, change and comment on the given graph (Patel, 1995) or make their own graphs (other measures of their own) of what is important to them. Modelling-users can map the various living information system concepts or their own concerns onto the graph. This is based on the notion of end-user modelling. These concepts may actually be key variables for organisational survival (and in terms of changes). Examples of concepts that might be mapped are:

- interfaces
- inputs
- process X, Y, Z
- outputs
- policy A, B C

*Living Information System Analyser*

This module would collect all the designed material, configurations, junctures of tailoring (Patel and Paul, 1995), and configure them into an amalgamated version of a proposed living information system. This is an initial starting point, not a specification for living information systems. It is the present living situation of the modelling-users. This point may be thought of as: I do this at present so the living information system must support that. The analyser module would produce an amalgamated living situation.

The amalgamated living situation could be a concurrent procedure, and available to modelling-users and analysts through windows. Alternatively, when many modelling-users are simultaneously using Hyper-Tmodeller, their individual views could be made available to each other through windows such as captured snapshots of the system. That way modelling users can learn what others think they do in relation to each other; they can/ may base their Tmodels around the Tmodels of other modelling-users.

*Logographer*

This module would be the documentor, documenting all the actions in the Originator-Creator and Discoverer-Designer modules. The logographer would also collect all the tailoring made by modelling-users. It would have to collect all the links, moves and drawings done by modelling-users.

The logographer would collect all the text comments, forming a kind of documentation of Tmodelling sessions. (Documentation in LIST has to be dynamic, just like the living information system itself.) This documentation should be available dynamically to analysts or users. Hypertext would be an excellent medium for this.

The logographer documentation could be used by technical systems designers and programmers to translate modelling-users' ideas of the specification they want.

## Operating Hyper-Tmodeller

To operate Hyper-Tmodeller, modelling-users and analysts would be provided initial training. This would involve training on the actual use of the Hyper-Tmodeller and education in LIST. While operating Hyper-Tmodeller, various types of help would also be available:

- General help – would explain the various functionalities of Hyper-Tmodeller, providing tutorials and examples of Tmodels
- Contextual help – would be made available when a Hyper-Tmodeller session is in progress. Appropriate help would be given whenever asked for during a session.

## Specification of Hyper-TModeller

The specification of Hyper-Tmodeller is split into two parts: Input mode and Information Gathering mode. In input mode end-users may input graphically how their job works and how they interact with the other organisational processes. Hyper-Tmodeller is based around the tasks that end-users perform for their job. These inputs may be used as the basis of a discussion with analysts and may therefore be modified in the light of discussion and further thought. It has two modes in which end-users may use this part of the system autonomously or with the analyst.

Information Gathering mode is designed to be a passive process operating in the background whilst information is input and modified within the system. It is designed to operate intelligently to gather any new information input/modified since a model was created. Given this information may be held in object form from the graphical interface, it may be possible to model this information to produce a data model of the system and thus the basis for a specification.

## Conclusions

This paper has discussed the emerging problems identified with the specification of simulation models within the larger context of information systems and has attempted to shed some light on the potential reasons for these occurring. It has attempted to provide a possible solution to this through both the historical development of FIELD and the current specification for development of Hyper-Tmodeller.

By looking at the problem of the elicitation of end-user knowledge, it is hoped that one of the more significant barriers to the specification of simulation models may be facilitated by Hyper-Tmodeller. The use of Hyper-Tmodeller may allow the specification of simulations models which are fit for their required purpose.

This paper adds to the debate on the understanding of simulation modelling using the medium of hypermedia. It provides views towards an integrated system for developing models (FIELD) and the fuller elicitation of user knowledge (Hyper-Tmodeller). Increasingly more complex systems are being requested thereby increasing the importance of problems addressed in this paper. It is hoped that this paper is a contribution to the solution.

It is now planned to develop Hyper-T modeller further, and to implement a prototype version. It is possible that in the future it will provide an excellent mechanism for the elicitation of user needs and requirements as well as the generation of model documentation.

## References

Avgerou, C., Cornford, C. *Developing Information Systems Concepts Issues and Practice*, Macmillan, Basingstoke, 1993.

Balci, O. Principles and techniques of simulation validation, verification and testing. In: *Proceedings of the 1995 Winter Simulation Conference*, C. Alexopoulos *et al.*, (eds.), Washington DC, USA. ACM Press, 1995; pp. 147-154.

Friedman, A.L., Conrford, D.S. *Computer Systems Development: History, Organisation and* Implementation, Wiley, Chichester, 1989.

Gardner, L.A. Hypermedia for prototyping and system integration in information systems development. Unpublished PhD thesis, University of London.

Gardner, L.A., Paul, R.J. Teaching cartographic design using HyperCard, *Infor (Information Systems and Operational Research)*, 29(3), 225-239, 1991.

Gardner, L.A., Paul, R.J. Developing a hypertext geographic information system for the Norfolk and Suffolk Broads Authority. *Hypermedia*, 5(2), 119-143, 1993.

Gardner, L.A., Paul, R.J. A fully integrated environment for layered development (FIELD) in information systems. *International Journal of Informaton Management*, 14(6), 473-484, 1994.

Gardner, L.A., Patel, N., Paul, R.J. Moving beyond the fixed point theorem with tailorable information systems. *3rd European Conference on Information Systems*, Athens, June, 1995; pp. 183-192.

Gardner, L.A., Paul, R.J. The development of a documentation mechanism for the creation of maintainable information systems using hypertext. *The Journal of Document Management* (forthcoming).

Garg, P.K., Scacchi, W. A hypertext system to manage software life-cycle documents. In: *IEEE, Developing a Hypertext Geographic Information Software*, Gardner, L.A., Paul, R.J. (eds.), May, 1990; pp. 90-98.

Holmes, A., Poulymenakou, A. Towards a conceptual framework for investigating information systems failure. In: *Proceedings of the 3rd European Conference on Information Systems*, Doudikis, G. *et al.* (eds.), June, 1995; pp. 805-823.

Law, A.M., Kelton, W. *Simulation Modelling and Analysis*, 2nd ed., New York, McGraw-Hill, 1991.

Mayhew, P.J., Dearnley, P.A. An alternative prototyping classification. *The Computer Journal,* 30(6), 481-484, 1987.

Marshall, C., Rogers, R. Two years before the Mist: Experiences with AquaNet. In: *Proceedings of the ACM Conference on* Hypertext, Milano, Italy, Baltimore, ACM Press, November 30th–December 4th, 1992; pp. 53-62. Also in *Management,* 14(6), 473-484, 1992.

Nance, R.E., Overstreet, C.M. Computer simulation: Achieving credible experimental results in virtual environments. *Simulation Digest,* 24(3), 40-49, 1995.

Nielsen, J. *Hypertext and Hypermedia,* Academic Press Inc., 1993.

Patel, N.V. The meaning of LIST! – Using the life metaphor to inform living information systems research. Paper presented at *Living Information Systems Thinking Workshop,* Department of Computer Science and Information Systems, Brunel University, 20th July, 1995.

Patel, N.V., Paul, R.J. Using living information systems thinking to inform the design of end-user tailorable information systems. Submitted to the journal: *Information and Management.*

Paul, R.J. An O.R. view of information systems development. In: *Tutorial Papers of the Operational Research Society Annual Conference* (Lawrence, M., Wilsdon, C., eds.), held at the University of Kent, 12-14 September, 1995). Operational Research Society, Birmingham, 1995; pp. 46-56.

Paul, R.J., Hlupic, V. The CASM environment revisited. In: *Proceedings of the 1994 Winter Simulation Conference,* held at Orlando, Florida, from 11-14 December, 1994, Tew, J.D., Manivannan, S., Sadowski, D.A., Seila, A.F., eds.), Association for Computing Machinery, New York, 1994; pp. 641-648.

Paul, R.J. Dead paradigms for living systems. In: *Proceedings of the First European Conference on Information systems,* held at Henley Management College, 29-30 March, 1993, The Operational Research Society, Birmingham, 1993; pp. 250-255.

Robinson, S., Pidd, M. Service quality in the management of simulation projects. In: *Proceedings of the 1995 Winter Simulation Conference,* held at Washington, D.C., USA, 4-6 December, 1995, Alexopoulos, C. *et al.,* eds.). ACM Press, 1995; pp. 952-959.

Wright, P. Interface alternatives for hypertexts. *Hypermedia,* 1(2), 146-166, 1989.