

DIVISION OF COMPUTER SCIENCE

3-Dimensional Modelling in Knowledge Engineering

Sara Jones

Technical Report No.174

January 1994

3-Dimensional Modelling in Knowledge Engineering

Sara Jones
S.Jones@herts.ac.uk

School of Information Sciences, University of Hertfordshire,
College Lane, Hatfield, Herts, AL10 9AB, UK.

Tel: +44-707-284370
Fax: +44-707-284303

Abstract

Graphical representations are now used in the human-computer interfaces of applications in a wide variety of domains. This paper considers a particular domain - that of knowledge engineering - and discusses the benefits of using 3-dimensional node and link diagrams to support various activities in that domain.



3-DIMENSIONAL MODELLING IN KNOWLEDGE ENGINEERING

S. Jones*

Graphical representations are now used in a wide variety of application domains. This paper will consider a particular domain - that of knowledge engineering - and discuss the benefits of using 3-dimensional node and link diagrams to support various activities in that domain.

What are Graphical User Interfaces? The term 'Graphical User Interface' (GUI) has been given a variety of different meanings in the last few years. In the early days, it was often used to describe an interface in which boxes or lines could be drawn around chunks of text, perhaps to attract the user's attention. More recently, it has been taken to mean an interface with windows, icons, menus and pointers through which the user is able to issue systems with commands. In this paper, however, the graphical representations of interest will be those used to represent parts of the application domains in which such systems are used.

Graphical representations in this class can be placed within a spectrum according to the degree of realism they aim at in modelling elements of the domain of interest. At one end of the spectrum are representations of the kind used by virtual reality systems which allow users to draw quite directly on their knowledge of real world objects in their interactions with objects in the interface. Work on representations of this kind has looked at their use in, for example, plant control [1] and medical imaging [2].

Next along the spectrum come slightly more stylised representations which rely more heavily on the use of metaphors or schemes for representation, allowing users to understand the information of interest without necessarily using a faithful representation of the relevant objects. This approach is exemplified by systems used in molecular modelling (see, for example, [3]). Since molecules are not actually visible to the naked eye, a suitable means of representing them has had to be invented to permit scientists to understand and reason about their structures. Computer-based tools for molecular modelling often support 'ball and stick' representations based on metaphors developed earlier this century.

Finally, at the opposite end of the spectrum are representations of purely abstract knowledge, information or data structures which have no concrete embodiment so that the question of 'realism' does not arise. Visual programming environments (see [4]), databases (see, for example, [5]), and CASE tools all use graphical representations of such structures. The aim in such systems is simply to develop representations which are 'cognitively transparent' [6], i.e. ones which reveal or emphasise what users need to see or know in order to develop productive mental models of the structures represented. Of course, the kind of information which will be useful in developing such models will

* School of Information Sciences, University of Hertfordshire, College Lane, Hatfield, AL10 9AB. Email: S.Jones@herts.ac.uk.

depend on the tasks users need to perform. Users may need one kind of information to develop an initial design for a system, and quite different information to debug it.

The graphical representations of interest below are of the abstract knowledge structures used in knowledge-based systems. The rest of the paper presents some findings regarding the utility of 3-dimensional representations of such structures in supporting various knowledge engineering tasks. Some hypotheses concerning the significance of these findings and their generalisability are briefly discussed.

Graphical User Interfaces for Knowledge Engineers Tools for knowledge engineering - that is, for the development of knowledge-based systems - have for some time provided users with graphical representations of knowledge structures under development (see [7] for a review). Many of these representations took the form of 2-d node and link diagrams with elements of knowledge structures being shown in the diagram as nodes, and significant relationships between those elements being shown as graphical links. More than one kind of relationship can be shown in a 2-d diagram by using simple visual coding schemes employing, for example, solid, dashed and dotted lines. Knowledge structures are, however, typically multi-dimensional, in that many different relationships between knowledge elements may be of interest to the user. If many such relationships are represented simultaneously in the same 2-d diagram, it can quickly become difficult to use. Work at City University (see, for example, [8]) is investigating the extent to which the use of 3 dimensions can ameliorate this problem by permitting larger numbers of relationships between elements of a knowledge structure to be simultaneously perceived.

As part of this work, knowledge engineers were interviewed in a series of 6 case studies to determine the extent to which they thought 3-d node and link diagrams might support various knowledge engineering tasks. They were asked to collaborate in the design of 3-d node and link representations of knowledge structures with which they had recently been working. One of the representations developed in this way is shown in figure 1. Finally, after carrying out some simulated editing and manipulation tasks using the representations they had helped to develop, the engineers were questioned again regarding the utility of the diagram they had used. Further details of the studies can be found in [9].

In the interviews, four of the six knowledge engineers said they thought graphical representations would be helpful in knowledge acquisition, when initial models of the knowledge structures to be used are being developed. Two of the engineers reported having used diagrams drawn with paper and pencil at this stage, even in the absence of tool-based support, and a third said a graphical tool would be useful for '*organising your thoughts and getting the structure down*'. Two of the subjects felt that graphical representations would provide good support for knowledge validation activities, in which experts are asked to check the validity of knowledge recorded by knowledge engineers. Three of the subjects thought graphical representations would be useful in developing high-level designs for systems or knowledge structures. One subject (the most experienced by some years) was particularly interested in the opportunities presented by 3-dimensional representations at this stage, for example, in permitting

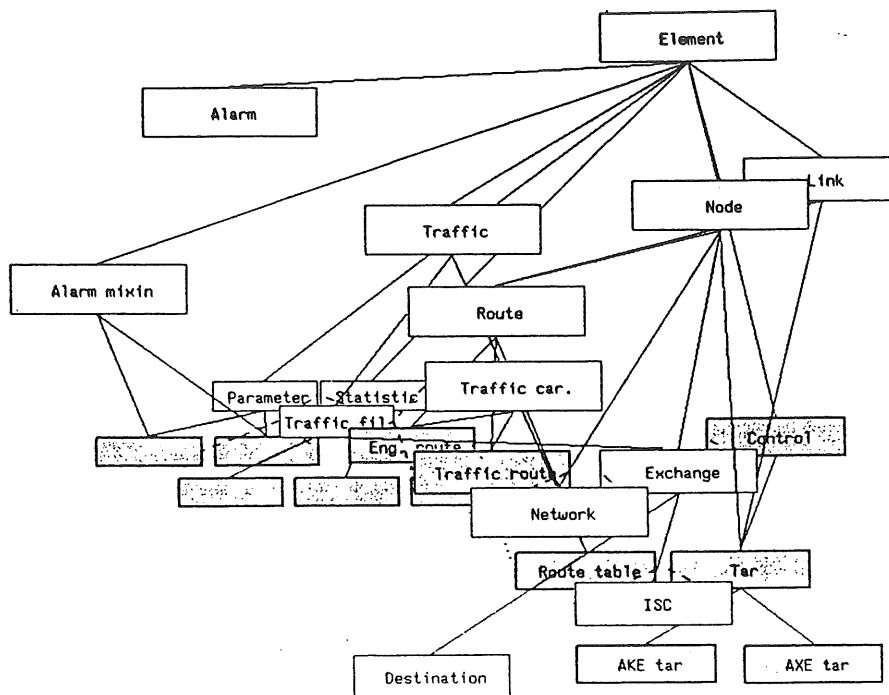


Figure 1. 3-dimensional node and link representation of knowledge used in a network management support tool. Two different kinds of knowledge object are shown (nodes are different colours in the original) and three different kinds of relationship between them: inheritance (top-bottom using solid links), physical containment (in depth using dashed lines) and referential containment (in depth using dotted lines).

knowledge engineers to see the relationship between two different knowledge structures in the same diagram: *'I think it [3-d] might help to do that quite valuably'*.

Considering development (i.e. the writing of knowledge-based system code), one subject felt that it would be useful to be able to see a diagram of the system under development if such a diagram could be generated automatically: *'I like being able to see some kind of graphical representation of the kind of knowledge structures you've built because it's a hell of a lot easier than having to scroll through hundreds and hundreds of lines of code to try and find out'*. Another subject had used a facility provided by KEATS [10], in which code could be generated automatically from 2-d node and link representations of parts of a system, and had found it extremely useful as *'a productivity tool'*. Four of the six subjects were enthusiastic about the possibility of using graphical representations to support debugging activities. All of these thought that it would be easier to spot relevant patterns or paths in the knowledge structures using diagrams rather than text. One of the engineers thought that graphical representations would be useful in supporting system demonstrations in acceptance tests with clients, and finally, one thought that diagrams of a system's knowledge structures would be helpful during system maintenance.

Conclusions On the basis of the findings reported above, we may draw a number of tentative conclusions regarding the kinds of tasks, and users, for which Graphical User Interfaces of the kind considered can provide greatest, and least, support.

First, we have some evidence to support the claim made by Petre and Green [11] that graphics are better than text for providing overviews and relatively abstract models of a complex structure. Five out of six of the engineers in the studies thought that graphical representations would be useful in either knowledge acquisition, high-level design or maintenance: all activities in which it is important to be able to understand and reason about models of the system at a high level of abstraction, without getting bogged down in too much detail.

Graphical representations also seem to be favoured in tasks where it is important to be able to communicate, especially with experts from different domains, or with other members of a development team who are unfamiliar with particular parts of a system. Knowledge validation and verification involve communication with domain experts, and design, development and maintenance activities may all call for interactions between various members of the development team. All subjects felt that graphical representations would be useful in at least one of these tasks. We may speculate that the popularity of graphical representations for use in communications tasks stems partly from their suitability for providing users with overviews and therefore supporting uncluttered communication, and partly from the fact that diagrams often seem less intimidating than mathematical or logical notations, especially for experts in other domains.

Other tasks in which diagrams may provide better support than text are those involving pattern spotting or systematic search. Once again, a similar claim is made by Petre and Green [11]. In the studies reported above, four of the six subjects thought that graphical representations would be useful in debugging activities, where it is often important to be able to spot recurrent patterns of inefficiency, or to follow through particular chains of reasoning used by the system.

Tasks for which graphics may not be able to provide appropriate support include those involving a great deal of detail or complexity. Two subjects noted how in development, detailed coding might best be done using a textual notation. At the stage of high-level design, the use of graphical representations alone - even 3-dimensional versions - is not enough to make the full complexity of some systems manageable. In these situations, further interactive facilities for complexity management are needed.

Before closing, it is worth noting the apparent effects of user experience on the perceived utility of graphical representations. All of the engineers in the studies were able to discuss and use the 3-d node and link representations developed without difficulty. We therefore have some grounds for supposing that no unusual capabilities are needed to be able to take advantage of such representations in tasks such as those identified above. However, the most concrete and imaginative suggestions about the way in which 3-d representations in particular might be used came from the engineer who was the most experienced of the group by several years. It seems likely that different kinds of representations will be appreciated by different user communities, and that users with increasing experience of an application domain may turn to increasingly sophisticated graphical representations.

Acknowledgements This work was funded by a SERC CASE studentship sponsored by the Science and Engineering Research Council and British Telecommunications plc.

References

1. E. Tello. Between man and machine. *Byte*. September 1988.
2. P. Mercurio and T. Erickson. Interactive scientific visualisation: An assessment of a virtual reality system. In D. Diaper, D. Gilmore, G. Cockton and B. Shackel (eds), *Human-Computer Interaction: INTERACT90*. Elsevier Science Publishers BV (North-Holland), 1990.
3. W. Ripka. Computers picture the perfect drug. *New Scientist*. June 1988.
4. B. Myers. Taxonomies of visual programming and program visualisation. *Journal of Visual Languages and Computing*. 1(1) 1989.
5. B. Czejdo, R. Elmasri and M. Rusinkiewicz. A graphical data manipulation language for an extended entity-relationship model. *IEEE Computer*. March 1990.
6. J. Seely Brown. From cognitive to social ergonomics and beyond. In D. Norman and S. Draper (eds), *User Centred System Design*, chapter 22, Lawrence Erlbaum Associates, 1986.
7. S. Jones. Graphical interfaces for knowledge engineering: An overview of relevant literature. *Knowledge Engineering Review*. 3(3), 1988.
8. M. Glykas, U.K. Patel, A.G. Sutcliffe, D.C. Dodson and T. Hackett. Towards interactive explanation by 3-d visualisation. In *Proceedings of the Workshop on Task Based Explanation*, Samos, July 1992.
9. S. Jones. Three-Dimensional Interactive Connection Diagrams for Knowledge Engineering. PhD thesis. Department of Computer Science, City University. 1993.
10. M. Eisenstadt, J. Domingue, T. Rajan and E. Motta. Visual knowledge engineering. *IEEE Transactions on Software Engineering* 16(10), 1990.
11. M. Petre and T.R.G. Green. Where to draw the line with text: Some claims by logic designers about graphical in notation. In D. Diaper, D. Gilmore, G. Cockton and B. Shackel (eds), *Human-Computer Interaction: INTERACT90*. Elsevier Science Publishers BV (North-Holland), 1990.

