9

DISPLAYS AS DATA STRUCTURES: ENTITY-RELATIONSHIP MODELS OF INFORMATION ARTEFACTS

T. R. G. Green
MRC Applied Psychology Unit
15 Chaucer Road
Cambridge CB2 2EF, U.K.
t.green@mrc-apu.cam.ac.uk

David Benyon
Dept of Mathematics
Open University
Milton Keynes MK7 6AA, U.K.
d.r.benyon@open.ac.uk

KEYWORDS

Structure map, entity-relationship, cognitive map, information design, display-based problem-solving, user model.

ABSTRACT

The analytical tools of database modelling can be applied to information artefacts. We extend a well-known technique, entity-relationship modelling, to give models ('structure maps') of information structures in both static (paper-based) and interactive cases. Structure maps can reveal deep similarities between different-seeming devices, set bounds for searching and updating information structures, and reveal likely discrepancies between the 'system image' and the intended user's conceptual model, and they lay bare the resources available for display-based problem-solving.

INTRODUCTION

We present a novel technique for analysing information displays. This technique is built on the observation that the structure of an information display closely resembles the structure of a data base: like a database, a display usually contains several types of item with several instances of each type, and there are relationships between the items. Thus a portion of a menu structure (Figure 1) may be represented in terms of two types of item, 'headings' and 'items', the items being subsumed under the headings. Information displays can therefore be modelled using the tools for the modelling of database structures. We shall show how entity-relationship modelling, a well-developed data base modelling technique, may be applied to information displays to gain improved understanding for the analyst.

Edit	
Cut	Ctrl X
Copy	Ctrl C
Paste	Ctrl V
View	
Outline	Ctrl O
Normal	Ctrl N

Figure 1: a list with headings and items

The ERMIA technique (Entity-Relationship Modelling of Information Artefacts) emphasises the *structure* of the information display, and its relationship to a conceptual model, rather than the actual content of the display. Structure is what can be inferred from the display itself without appeal to the underlying domain represented; if you were handed a torn-up timetable, you could infer the layout of the rest of the timetable without knowing whether the times were for trains, conference papers, or satellite transits, but you could not infer the actual times. We shall call these models *structure maps.*

Traditionally, HCI has focused on interactive devices, while paper-based displays have been part of information design research. ERMIA can be applied to both, and especially to:

(i) the interaction of task procedures with the information structure built into the artefact;

(ii) the relationship between the user's conceptual model and perceptual entities in the information display;

(iii) understanding what inferences may be achieved from a given information display.

In an earlier paper (Green, 1991) ERMIA was applied to interactive devices, such as the Unix filestore and browser, or to clearly defined manipulations upon paper-based notations, such as rewriting a music score in a different key. The focus of that paper was on (iii). The present paper will emphasise (i) and (ii).

ENTITY-RELATIONSHIP MODELLING

Our first example is deliberately simple and familiar, to introduce the technique of entity-relationship modelling (Chen, 1976), (Benyon, 1990) and its adaptation for ERMIA. ER modelling is quickly explained, although the consequences may be more far-reaching than at first is evident. ER models concentrate on the cardinality and complexity of relationships. Consider the relationship between the two entities of Figure 1, *Heading* and *Item*[1]. The author of Figure 1 has assumed that menu items are

K. Nordby et al. (eds.), *Human-Computer Interaction*

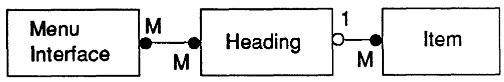

Figure 2a: elementary ER diagram for Figure 1

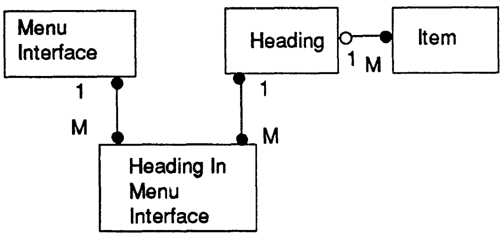

Figure 2b: eliminating the M:M relationship

not duplicated under different menu headings, so each item belongs to a single heading; but a heading can contain several items, so there is a one-to-many (1:M) relationship between Heading and Item. The relationship is clearly mandatory from an Item's point of view (every item is listed under some heading, so we assume that a thing can only be an Item if it belongs to a Heading), but we can envisage a heading which has no items (e.g. a menu "Work"), so for the Heading entity the relationship is optional. In Figure 2a the open circle indicates an optional relationship. A third entity has been introduced, the MenuInterface itself, and we have modelled a situation where the same heading might be found in different menuInterfaces (e.g. the word-processor menuInterface and the spreadsheet menuInterface both contain Edit). In the MenuInterface-Heading relationship both types of entity must participate; an menuInterface must have at least one heading (but it may have many) and a heading must belong to at least one menuInterface. Notice that this is how we, the modellers, define menuInterfaces and headings. The ER model makes our definitions and constraints explicit.

ER modelling provides a number of useful techniques

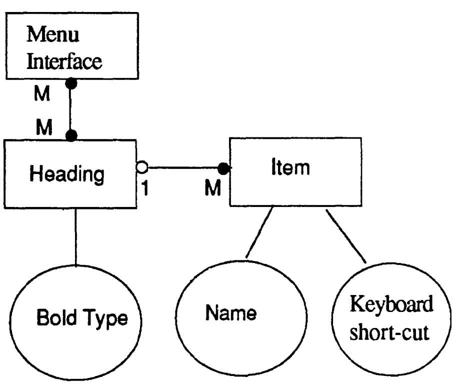

Figure 2c: adding attributes

which help to understand and represent structure. One of these helps to expose hidden entities. Any M:M relationship can be replaced by an entity which will have a mandatory M:1 relationship with each of the entities originally in the M:M relationship. Attention is thus focused on the new entity and the modeller is forced to ask what the uncovered entity portrays. In Figure 2b the M:M relationship has been replaced by an entity representing a heading in a menuInterface. This defines which headings appear in which menuInterfaces (e.g., if there is no view heading in the spreadsheet menuInterface 'View' will be absent).

Entities require defining or illustrating to make their meaning clear. Figure 2b could be accompanied by a natural language description such as *"An example of a menuInterface is given in Figure 1. A heading is a menu heading (e.g. Edit, View) which appears in bold type. Items are named flowers"*. Entities have characteristics or *attributes*, some of which help to define the 'meaning' of the entity. An entity may be defined as an aggregation of its attributes. Attributes are represented (if necessary) on the diagram by circles. Attributes usually have 1:M relationships with their entities; e.g. any given menu item has a single keyboard short-cut (e.g. Ctrl X), but a given keyboard short-cut may be shared by many items (e.g. Ctrl O may be the keyboard short-cut for 'outline' in the word-processor and 'open' in the spreadsheet). A more detailed ERMIA of the menuInterface in Figure 1 is shown in Figure 2c. Constraints other than the degree of the relationship (1:M, 1:1 or M:M) and the participation conditions (mandatory or optional participation in the relationship) may also be included as natural language predicates.

ERMIA ANALYSIS

To adapt ER modelling for our purposes we need a distinction between *types of entity*, a distinction between *conceptual* and *perceptual* entities, and the addition of *perceptually-coded attributes*. (Figure 2c shows the perceptually-coded attribute 'Bold Type' for the entity 'Heading'.) These extensions (plus a few others) define ERMIA. We represent an ERMIA analysis as a 'structure map'.

DEVELOPING A CONCEPTUAL MODEL

An information display is a perceptual representation of someone's conceptual model, in use for various purposes (some of them probably unanticipated). For instance, Figure 1 is a perceptual representation of a conceptual model in which items are neatly divided into distinct headings. ERMIA supplies an analysis of that representation.

In this section we shall develop an ERMIA analysis for a familiar artefact, the telephone directory. We shall start by developing a standard ER version of the conceptual model, although in another case we might well have decided to start with the actual information display.

Evidently the phone book relates subscribers to their phone numbers, but the structure is a little more complex than first appears. The conceptual entities include Subscriber, Name, Address, and Phone-

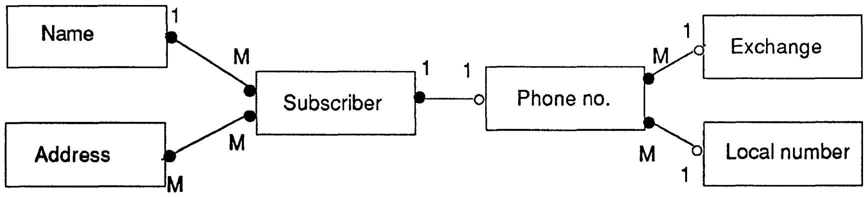

Figure 3: Initial ERMIA of conceptual model

number. In our (the authors') conceptual model of the telephone system, each phone number comprises an exchange and a local number, which together uniquely define that phone number. Every subscriber has at least one address and at least one phone number. Every address which appears in the telephone book has at least one phone number. Most phone numbers are allocated to a unique subscriber (some phone numbers are used for system purposes). The combination of exchange and local number together define the phone number. The phone number determines the subscriber, and the subscriber determines the phone number (thus Phone-number : Subscriber is 1:1). A subscriber must have one name and at least one address, but may have many addresses. The combination of name and address uniquely determines the subscriber (though one subscriber may have more than address, and more than one subscriber may share an address).

The conceptual model, represented as a conventional ER diagram, is shown in Figure 3. Observe here that this is *only* a conceptual model. The reality might be surprisingly different; indeed, if ex-directory subscribers and various special-purpose phone numbers were included, the structure would certainly be different.

However, there is a problem with the conceptual model since it shows a subscriber having only one phone number, when we know it is possible that a subscriber can have more than one phone number. This problem is overcome when we explore the M:M relationship between subscriber and address. M:M relationships should always be investigated further since they often hide an entity which may turn out to be important. In this case we can see our error when we open up the M:M relationship. There are two concepts of Subscriber. One is the person or institution subscribed for billing purposes (called Subscriber) and the other is a Subscriber-at-an-address, each of which may only have 1 phone number. (We are slightly simplifying the world here.) Figure 4 is a more accurate representation.

ERMIA ANALYSIS OF THE INFORMATION DISPLAY

Now we turn to the information artefact which represents this conceptual model as an information display. The British Telecom telephone directory uses this format:

K. Lear, Palace, Glastonbury....... Glastonbury 12345
H. MacBeth, Castle, Dunsinane.......Dunsinane 13243
The Other Castle, Cawdor................. Cawdor 758

To make an ERMIA structure map, showing how this conceptual model is represented by the perceptual display, we note the following perceptual entities:

Directory is the list of phone numbers

Name corresponds to the conceptual entity, 'Name', distinguished by bold type

Address corresponds to the conceptual entity, 'Address'

Exchange is part 1 of the conceptual Phone-number; a place name

Local Number is part 2 of the Phone-number; digits

Block Of Entries gathers all the entries for a single Name, using typographic cues to mark the limits of the block. (Lear's block has only one entry but MacBeth has two entries.)

Entry contains a Name, an Address and a Phone-number (omits the Name, for second and subsequent entries in a block). Normally a single line of text.

Of course the names in the directory are in an ordered list, so that a given name can be found quickly. We shall represent this property of the entity *Name* by marking it with a special symbol. The ERMIA structure map is shown in Figure 5.

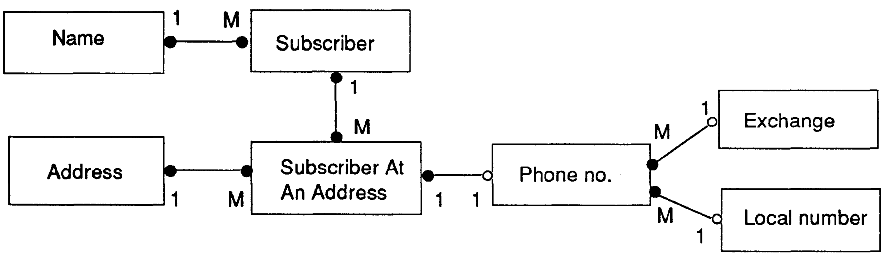

Figure 4: ERMIA of conceptual model after replacing M:M relationship by an entity

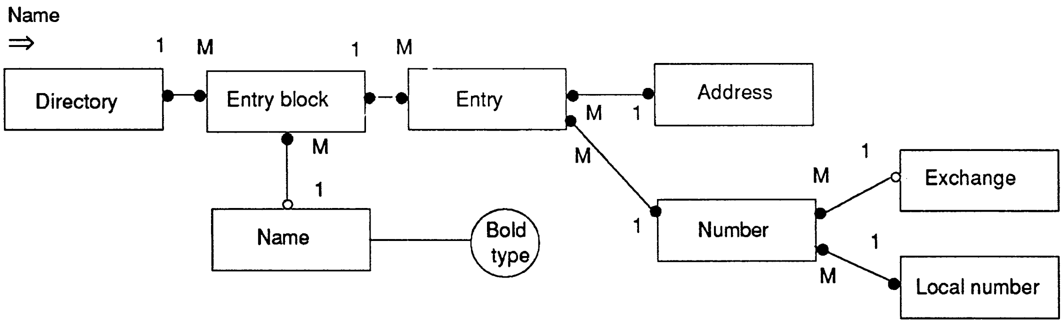

Figure 5: ERMIA analysis of the perceptual display of the telephone directory

The two ERMIA diagrams in Figures 4 and 5 show a very close correspondence: the Entry-block corresponds to a Subscriber, the Entry corresponds to a Subscriber At An Address, and so on. (We chose this example for just this reason. A later example illustrates a divergence between conceptual model and information display.)

SEARCHING THE STRUCTURE

A crucial question in HCI is how task procedures interact with information structures. ERMIA can be used to set minimum bounds on tasks requiring searching or changing entities; more generally, it reveals the time and space complexity of traversing the information structure.

A search procedure can be developed analytically from the ERMIA map, so long as we know the starting entity of the search, which is Directory in our example. The search can follow any route of lines (relationships) that leads ultimately to the target entity. Where more than one route can be followed, more than one search strategy exists. Wherever the search traverses a relationship to an entity via an M, it will be necessary to search instances of that entity to find the right one (which Entry Block has the right Name and is associated with an Entry with the right Address?). This may only demand examining an attribute of that entity, or it may demand searching deeper into the structure to find an entity with a verifiable attribute.

Searches may be aided by an ordering or by perceptual cues, both shown in the ERMIA map. The actual choice of strategy is up to the user (exhaustive enumeration, blind shot in the dark, binary chop, inspired guesswork, etc.) and will depend on circumstances such as how many instances need to be searched, but if the relationship is something:M then the structure of the information imposes an inescapable need to locate one instance among more than one.

The distinction between different types of entity, considered as data stores, is central to the analysis of search and traversal of information structures. The directory is designed to support the task of finding a number for a given name. Any other searching task (e.g., finding the name of the person living at a given address) requires step-by-step search.

Tasks requiring changes to entities will not be considered here, but Green (Green, 1991) shows how the 'viscosity' (Green, 1990) of certain types of information display can be deduced from the ERMIA structure map, by considering how many data elements have to be changed in order to achieve a given result. We shall not pursue this aspect here. (Raymond and Tompa (Raymond and Tompa, 1992) have used a related technique of dependency analysis for this purpose, and considerably extended the analysis.)

CORRECTNESS AND RESPONSIBILITY ISSUES

Lastly, the analysis demonstrates that the modeller must take responsibility for certain decisions in the model, especially for the *grain size* and the *distinctiveness of attributes*. In this case the modeller has chosen a grain size suitable for the regular task of looking up a number, also for unusual tasks such as counting how many Hamlets are listed, but not suitable for finding occurrences of individual letters. The modeller has also decided that the typography of the telephone directory does sufficiently distinguish between one entry and another, one block of entries and another, etc.. Further responsibility lies with the modeller for deciding whether to take into account differences between individual users or groups of users.

COMPARING THE CONCEPTUAL MODEL AND THE INFORMATION DISPLAY

This analysis can readily be applied to one of the recurrent themes of HCI, the discrepancy between the *designer's* model and the *user's* model. A misleading information display (or 'System Image') will lead the user towards a conceptual model which is significantly different from the designer's model (Norman, 1986), (Tognazzini, 1992), as may 'non-standard' models (Thimbleby, 1990). How does the designer determine how well the information display reveals the intended conceptual model? ERMIA analyses are a means to that end, since both the display and the model are represented in the same 'language' and can therefore be combined into one diagram to examine the mapping. Entities which exist only at a conceptual level and have no direct representation as perceptual entities are distinguished by ghostly dotted lines to denote their status.

Most of the textbook examples present cases where the information display fails to capture some aspect of the Designer's Model in a rather elusive, roundabout way. To avoid elaborate analyses, we have chosen for our illustration an unusual case: the card trick. Here we have an information display which is deliberately designed *not* to conform to the Designer's Model. On the contrary, the 'System Image' is intended to induce an inaccurate User's Model, because what the 'designer' wants to do is to deceive the 'user'. Such an example tests the power of our analysis quite stringently, because the 'designer' will use all available means to contribute to the deception, and the question is whether ERMIA has to the power to detect what has happened.

In the following trick (taken from (Harbin, 1983), but known in the early 19th century), the magician offers the punter 10 pairs of cards. The punter chooses a pair, without revealing the choice, and the magician then deals the cards face up in a tableau, laying them out in a seemingly random arrangement. The punter has to name the two rows in which the two chosen cards lie (or possibly they both lie in the same row). Surprise surprise – from that meagre information, the magician can identify the chosen cards.

The secret, of course, is that the tableau is far from random; it has the very special property that each of the original pairs of cards ends up in a different pair of rows. One such arrangement is illustrated. The first pair of cards (aa) goes one into row 1, one into row 2. The second pair of cards (bb) both go into row 1. The third pair (cc) goes into rows 1 and 5, and so on. (The columns are immaterial, but this layout, which is traditional, helps disguise that fact.) The vital detail is that *every card pair is associated with just one row pair*, so that identifying the row pair identifies the card pair.

Now, the punters are not likely to have the concept of a row-pair in their conceptual model, and the information display (the 'System Image') gives no indication of it: in fact, as the magician lays out the tableau, the cards go into 'random' positions in just such a way as to destroy any hint of a system and to suggest instead that the magician is choosing where to put each card at the instant of laying it on the table. So the act appears baffling.

Figure 6 shows the ERMIA structure map, and introduces two extensions to the basic model. First, an explicit natural language constraint is included with the diagram because there is no easy way of showing it diagrammatically. Second, the degree of the relationship can be made explicit in this case, instead of using the usual indeterminate 'M' symbol, since the actual numbers are known – e.g. each Column defines 4 Cards. The trick works because of the 1:1 relationship between the conceptual entities Card Pair and Row Pair. Combining the two levels of ERMIA, perceptual and conceptual, reveals whether the information display supplies a good cognitive map of the conceptual model.

a	b	c	b	d
e	f	a	g	e
h	f	h	i	d
k	g	k	i	c

Although card tricks are not the usual meat of HCI, this very simple form of analysis neatly illustrates how far the information display captures the conceptual model, and thereby lays bare the essential discrepancy between two conceptual models, the punter's and the magician's.

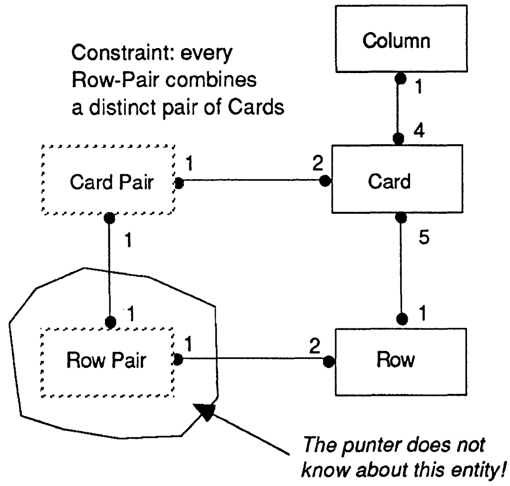

Figure 6: ERMIA for the tableau of the card trick. Entities in dashed line boxes are purely conceptual.

CONCLUSIONS, OR, WHY USE ERMIA?

The problem of the information display is the problem of supplying a cognitive map of the 'Designer's Model'. ERMIA helps by allowing both the conceptual model and the information display to be expressed in the same language, so that discrepancies can be clearly seen (see card trick analysis).

Strengths Other HCI formalisms do not address the 'cognitive map' between the System Image and the conceptual model. Instead, they typically define how the system responds to user activities (Alexander, 1987), (Dix, 1991) or define the user's task procedures (Kieras and Polson, 1985). And of course, if the user's conceptual model is not represented, there is no way to determine whether the System Image or information display matches it! Failing a common language in which both can be represented one is left with *ad hoc* natural language assertions that the system image fails to match the user's model, like Norman (Norman, 1986) and Thimbleby (Thimbleby, 1990), instead of a constructive analysis of the differences between the two.

ERMIA also encourages deeper understanding of devices. In HCI we often find ourselves able to describe the details of an artefact very closely, and able to perceive that it is somewhat similar to another artefact, but unable to describe the two artefacts at a sufficiently abstract level to make the similarities apparent. ERMIA is sufficiently abstract to show the structural similarities between artefacts which may at the surface be very dissimilar.

Entity-relationship analysis is a well-defined system of mathematics, which has been tested in many situations. Correspondingly, ERMIA diagrams (which if preferred can be expressed as an algebra) have a very exact semantics, as is required for a usable formalism. Of course, ERMIA has less expressive power than predicate calculus or natural language, but paradoxically that is an advantage for many purposes. Not only is it less forbidding than predicate calculus, and more precise than natural language, but also there is no way to descend into infinite detail and complexity, as can happen in some formalisms.

In comparison to most other formalisms, ERMIA is easy – even fun – to use. All the same, you have to work at it. We have applied it to quite a few artefacts, and we find that although the formalism is easy in itself, the first analysis takes some time to think out and is often incomplete. Reading and grasping the import of an ERMIA analysis also takes time, and we have found that our analyses are best accompanied by English-language commentaries. In fact, we believe that some of the virtue lies in 'discovery through process' – one gains as much insight into an artefact by working out the analysis (or by following through someone else's working) as one does by studying the finished analysis (Whitelock, et al., 1994).

Limitations As with other graphic notations, one must question whether ERMIA scales up to large systems. One solution is to use an algebraic formalism (and indeed, we have written a Prolog program to trace structure maps in such a formalism), but at present we have little experience of large-scale analysis. Nevertheless, the experience from database modelling is encouraging: systems of considerable complexity can usefully be analysed with ER diagrams (see examples in (Benyon, 1990)).

It must also be noted that ERMIA is not 'self-performing'. It does not take an image of the screen and automatically compute a structure; instead, the modeller has to take responsibility for certain decisions such as grain size or noting perceptual codings, as observed above. But the purpose of ERMIA is to abstract away from perceptual details and concentrate on structure, so it is inevitable that the modeller should have to take responsibility for those issues. ERMIA aims to *increase our understanding* of artefacts, rather than to provide a total description of them

ACKNOWLEDGEMENTS

We wish to thank Darius the Magician for his help and Darrell Raymond for his advice. The ERMIA project is funded by the joint councils initiative on Cognitive Science and Human-Computer Interaction. Co-workers on the project are Denise Whitelock and Marian Petre.

REFERENCES

Alexander, H. (1987) *Formally-based Tools and Techniques for Human-Computer Dialogues.* Chichester: Ellis Horwood.

Benyon, D. R. (1990) *Information and Data Modelling.* Oxford: Blackwell Scientific Publications.

Chen, P. P-S. (1976) The Entity-Relationship model: Toward a unified view of data. *ACM Trans on Database Systems,* 1, 9-36.

Dix, A. J. (1991) *Formal Methods for Interactive Systems.* London: Academic Press.

Green, T. R. G. (1991) Describing information artefacts with cognitive dimensions and structure maps. In D. Diaper and N. V. Hammond (Eds.) *People and Computers VI.* Cambridge University Press.

Green, T. R. G. (1990) The cognitive dimension of viscosity: a sticky problem for HCI. In D. Diaper, D. Gilmore, G. Cockton and B. Shackel (Eds.) *Human-Computer Interaction – INTERACT '90.* Elsevier.

Harbin, R. (1983) *Illustrated Teach Yourself Magic.* 2nd edn. London: Treasure Press.

Kieras, D. and Polson, P. (1985) *An approach to the formal analysis of user complexity.* Int. J. Man-Machine Studies, 22, 356-394.

Norman, D. A. (, 1986) Cognitive engineering. In D. A. Norman and S. W. Draper (Eds.) *User-Centered System Design.* Hillsdale, NJ: Erlbaum.

Raymond, D. and Tompa, F. (1992) *Applying normalization to notations.* Dept. of Computer Science, Waterloo Univ., Ontario, Canada.

Thimbleby, H. (1990) *User Interface Design.* Reading, Mass.: Addison-Wesley.

Tognazzini, B. (1992) *TOG on Interface.* Reading, Mass.: Addison-Wesley.

Whitelock, D. K., Green, T. R. G., Benyon, D. R. and Petre, M. Understanding Discourse thorugh Structural Models In Oppermann, R., Bagnara, S. and Benyon, D. R. (eds.) *Human-Computer Interaction in Work, Leisure and Everyday Life.* GMD publications, Bonn 1994 (to appear in a Special Issue of *Acta Psychologica* on Cognitive Ergonomics, 1995)

---

[1] Entity names will be capitalised when we mean the class, and left uncapitalised for instances.