Supporting Multimedia User Interface Design Using Mental Models and Representational Expressiveness

by

David Martyn Lewis Williams

Doctoral Thesis

Submitted in partial fulfilment of the requirements for the award of

Doctor of Philosophy at Loughborough University

27th September, 1996

Abstract

This thesis addresses the problem of output media allocation in the design of multimedia user interfaces. The literature survey identifies a formal definition of the representational capabilities of different media as important in this task. Equally important, though less prominent in the literature, is that the correct mental model of a domain is paramount for the successful completion of tasks.

The thesis proposes an original linguistic and cognitive based descriptive framework, in two parts. The first part defines expressiveness, the amount of representational abstraction a medium provides over any domain. The second part describes how this expressiveness is linked to the mental models that media induce, and how this in turn affects task performance. It is postulated that the mental models induced by different media, will reflect the abstractive representation those media offer over the task domain. This must then be matched to the abstraction required by tasks to allow them to be effectively accomplished.

A 34 subject experiment compares five media, of two levels of expressiveness, over a range of tasks, in a complex and dynamic domain. The results indicate that expressiveness may allow media to be matched more closely to tasks, if the mental models they are known to induce are considered.

Finally, the thesis proposes a tentative framework for media allocation, and two example interfaces are designed using this framework. This framework is based on the matching of expressiveness to the abstraction of a domain required by tasks. The need for the methodology to take account of the user's cognitive capabilities is stressed, and the experimental results are seen as the beginning of this procedure.

Acknowledgements

Heartfelt thanks go to, and sighs of relief come from:

- My supervisor, Prof. James Alty.

- My funding body, the EPSRC.

- Jo for constant support, encouragement, and a shoulder to cry on.

- Mum and Dad for their implacable faith in my ability.

- Prof. Bob Spence (Imperial College) for motivation and a watchful eye.

- My fellow students in Hut A5, especially Soosy, Ghada, and Mark. All soon to follow.

- The friends who have come and gone over the past three years.

Table of Contents

List of Figures by Chapter

Introduction

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Chapter 8

Chapter 9

Chapter 10

Chapter 11

Chapter 12

List of Tables by Chapter

Chapter 3

Chapter 9

Chapter 10

Introduction

I The Lure of Multimedia Interfaces

Multimedia interfaces in their proper sense are nothing new. Different media have always been combined in communication, whether through gesture and voice, or words and pictures. This is true both of user interfaces and non computer-based communication. However, what is new is the availability, cheapness, combination, and speed of new computer processors and resultant output media. Technologies such as continuous or discrete digital audio and video, 'bit-mapped' graphics, and high speed animation, have now become widely and inexpensively available. However, with this availability comes a wide variety of often unsubstantiated claims about what multimedia interfaces can offer the interface designer and the user. A selection of these are:

- **Increased information bandwidth**

  The variety of media which are available allow the allocation of information to visual, auditory, and haptic channels. Marmollin (1991) argues that this engages the 'whole mind' by stimulating verbal, auditory, and haptic brain centres. In information transmission *parlance*, this increase in the information being attended to is described as increased information *bandwidth*;

- **Improved 'attention getting'**

  The wide variety of media available allow for unusual representations of information. Since attention is drawn to unusual or new stimuli, it is suggested that multimedia can afford some congruence with this sensory phenomena. For example, this may be particularly useful in environments where the 'getting' of a users attention is essential, e.g. warning situations in a process control-room;

• **Increased realism**

The use of realistic media such as still, and moving, video allows the interface designer to represent real-world situations directly. This may be through the use of live video (Tani, 1992) or the depiction of recorded images of objects, e.g as virtual exhibitions (Mannoni, 1996).

Unfortunately, the technology has overtaken the methodology and the selection of media for tasks has become *ad hoc*. Concomitant with this, is that with so much more choice the risk of incorrect media selection is higher, thus the chances of producing an inadequate interface have greatly increased.

To attempt to remedy this situation, research in a variety of fields from social science to computer science has identified a number of important indices to guide this choice. Whilst many of these approaches are not computer-based, in every case the goal of effective communication is paramount. Work concerned with goal and task descriptions for interactive and didactic systems (Remus, 1984; Casner, 1991; Alty et. al. 1992; Maybury, 1993 ; Andre and Rist, 1993), data descriptions (Roth and Mattis, 1990; Mackinlay, 1986; Arens et al., 1993), automatic presentation design (Casner, 1991; Arens et. al., 1988; Mackinlay, 1986), perceptual characteristics of users (Casner, 1991; Buttigeig, 1989), characteristics of media (Bertin, 1983; Alty and Rijkaert, 1993; Hunt, 1989; Lohse et al., 1990; Arens, 1991), hardware constraints (Alty and McCartney, 1991), terminology (Frolich, 1991; Arens and Hovy, 1993; Nigay and Coutaz, 1993), demonstrates the multi-disciplinary nature of multimedia research, each group having its own motivations and goals. One result of such multidisciplinary approaches is a divergence in the definition of terms, such as *medium, modality* and *channel*. These are often used interchangeably depending on the focus of study.

Since any human-computer interaction relies on mediation of the machine's internal state to the user in a way which suggests its use, the question must be asked; "how do we allocate media at the user-interface, both in the design of interaction widgets and in showing data in order to solve tasks?". It is this question which the thesis addresses. However, it is not concerned with hardware issues such as media synchronisation (Prathakaram and Raghavar, 1994) or relationships between different compression standards. For the purposes of this thesis, it can be assumed that the technologies are available, what is needed is an effective methodology for deploying them.

I.1 The Thesis in Context

To begin with, it is important to place this thesis in the context of other multimedia user-interface research work (Chapter 1 will provide a more detailed discussion of these studies). Figure I.1 shows a three-dimensional framework, developed by the author, for characterising the multimedia literature. To give sufficient leverage for the methodology proposed in this thesis areas other than computer science are investigated. In doing so, the important issue of the relationship between the interface and the cognitive facility of the user is addressed. The three dimensions of the framework (along with their extremities) are:

- **Design** *(features vs. benefits)* : What are the main motivations for design?
- **Implementation** *(conceptual vs. practical)* : How important is implementation?
- **Interpretation** *(cognition vs. performance)* : How important is a consideration of cognition in media selection?

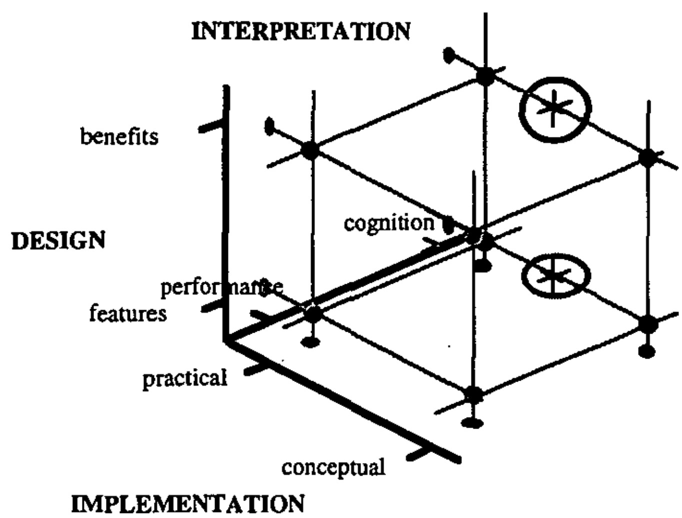


Figure I.1: Three dimensions of multimedia research

The position of this thesis is shown by the circle. The dimensions will now be described in detail.

I.1.1 Design: Benefits vs. Features

Alty et al. (1992) provide a basis for this dimension by stressing the need to separate technological *features* from user interface *benefits*. Examples of technologies which were innovative but were not driven by the need to offer well defined benefits to their users are:

• Bell Lab's video-phone appeared many years ahead of modern video telephones and provided the first example of full-motion video at the user interface. Unfortunately, the technology was not sufficient to push the product into the market. Since there was no target user-population the product failed;

• Xerox's revolutionary Alto system, precursor to the Apple Macintosh, failed to gain acceptance (Card, 1996). This was due to a number of reasons. Firstly, although the interface features provided were the forerunner of WIMP systems[1], they were far ahead of the needs of targeted users and environments. Secondly, there was little opportunity for third-party software development. Finally the cost of the system was prohibitive. Only by addressing the needs of those who would use the systems, and those who would write the software, did Macintosh successfully build on this innovation with the Apple Lisa.

The technology drive in the multimedia literature exhibits similar problems. As Mayes (1992) points out:

"Multimedia systems are not primarily defined by their data-structures, but by the nature of their communication.", (Mayes, 1992b: pp. 2).

Technology-centred research includes:

• Standardisation of compression algorithms, e.g. JPEG, MPEG;
• Modelling of sound and video synchronisation (Sventek, 1992);
• Standardisation of hardware protocols.

Whilst this research is essential it does *not directly* address what the user/designer will gain.

At the opposite end of the *design* dimension are those investigations which are shown to add value to the interface design process or the user-computer dialogue. Value may be an increase in ease of learning, ease of use, task support, accessibility, or market impact. These approaches rely on the existence of an adequate technology-base which allows them to focus on issues which would normally come later in the

[1] Graphical user interface standard using W(indows), I(cons), M(enus), and P(ointers).

design process. It is only by addressing these issues that the prevalent technologies can be integrated into a design *rationale* which considers both users' cognitive characteristics and the nature of their tasks.

This thesis addresses the benefits of different media, whether they are new (moving video, bitmapped icons, real-time animation) or old (tables, graphs, bar charts) to both users and interface designers. In doing so, it acknowledges the need for adequate technological support but does not propose anything which is beyond present technologies. An overview of the literature shows that this focus on the benefits of different media is unusual which gives an indication of the timeliness of research of this kind.

I.1.2 Implementation: Conceptual vs. Practical

The second dimension addresses the division between theoretical expositions and the implementation of systems which may be prototypes or products. In general, a higher percentage of research is carried out with a view to implementing systems. This covers automatic media allocation, automatic dialogue design and 'widget' selection (Singh, 1990), compression algorithms, modelling of multimedia interaction, and automatic allocation of output resources (Alty and McCartney, 1991).

The conceptual work is related more to issues of representation (Arens et al., 1993; Gilmore, 1991), terminology (Frolich 1991; Nigay and Coutaz, 1995), and task descriptions (Alty and Rijkaert, 1993). Generally, these studies give no discussion of enabling technologies. What makes such discussions necessary is the lack of theoretical underpinning which accompanies the majority of multimedia interface design methodologies. Here, concern is not with the low-level protocol or compression details since they will always be formally described in mathematical terms. Rather, the cognitive capabilities of users and the conceptual description of tasks are the central issue. Multimedia interfaces should be designed to be congruent with the former, and support the user in the latter.

This thesis describes what is at present a theoretical discussion. Consequently, no attempt is made to present an implementation of an automatic media allocation system. This is because there is still a great deal to learn about media as representational forms, and their relationship with human cognitive processes. The scanty reference to these issues made in the literature is testament to this. The author is conscious that what is presented here only provides the *beginnings* of a sound user and task-centred basis for output media selection.

I.1.3 Interpretation: Cognition vs. Performance

The applicability of cognitive science techniques and results to HCI is an active area of debate (Green et al., 1996). Opposition focuses on the applicability of cognitive theories to social activity, i.e. computer use, arguing that interaction is based more on the *context* of the interaction rather than the *actor (user)*. Thus, 'situated action theory' (Suchman, 1987) focuses on social and ecological issues rather than treating the user as a set of cognitive processes in isolation . However, it is the author's view that whilst not telling the whole story, *cognitivism* does provide a well developed language to describe user activity. More importantly, since the emphasis of this thesis is on the allocation of output media to convey information, rather than the design of interaction, the need for ecological studies of the interaction landscape is considered negligible.

The final dimension is therefore based on a distinction made by Stenning and Oberlander (1995) who suggested the separation of information presentation research (Tufte, 1983; Mackinlay, 1986; Roth and Mathis, 1990) from those concerned with the cognitive effects of different representation methods (Palmer, 1978; Egan and Grimes-Farrow, 1982). Stenning states:

"All of these (Tufte et al.) are concerned with improving the consumption of information. However, there is very little theory which underlies the choice of design and spans a wide range of information expressions.", (Stenning and Oberlander, 1995: pp. 3).

By 'consumption', Stenning means the transference of deterministic domain data to the viewer, or *consumer,* measured by their task performance (thus, this extreme is called performance). For example, Mackinlay (1986) showed that a horizontal axis can be used to encode binary relationships, and he provides a notation to explain how. This, and similar work (Bertin, 1983; Tufte, 1983), has its genesis in the design of decision support systems. These systems are concerned with the representation of information to aid managers in decision making (Remus, 1984). In this literature, the emphasis is on quantitative measures of the use of standard media (e.g. charts and tables) in decision making, rather than investigating the cognitive explanations behind these performance differences. However, it does provide important discussion to accompany work such as Mackinlay's, particularly with respect to those non-cognitive factors which influence the choice of standard representations (Coll et al., 1993).

The other extreme of this dimension, is populated by work that focuses on cognition. These studies go beyond the 'cognitive dimensions' identified by Green (1991) and Gilmore (1991) who are concerned with describing the emergent behaviours of *notations* (e.g. interfaces) and the surrounding environment (e.g. smart tools) in a simple language. Rather, they investigate the efficacy of representations in conveying information with respect to their congruence (or not) with human cognitive processes, e.g. working memory (Stenning and Levy, 1988), long term memory (Bainbridge, 1992), and language centres (Bos, 1995). This literature is cited as important, even though much of it does not make specific reference to computer-based media.

This leads to the essence of the thesis. Multimedia interface designers must learn from the use of non-computer based studies in this area, in order to illuminate the use of media in user interfaces. This is because the user's cognitive apparatus is responsible for processing those media which constitute the interface, making a consideration of these essential in any discussion of media allocation. Thus a study of all of the literature that is relevant to this area, e.g. linguistics, logic, problem visualisation, is required.

Ultimately, interface designers can use the knowledge of the cognitive effects of media to support media selection. This goes deeper than allocating media on the strengths of their form alone, as suggested by Feiner and McKeown (1993), Arens et al. (1993), and Alty and Rijkaert (1993). A discussion of form is important but only in the context of the *cognitive* and *task landscape* of the user's interaction with the computer system.

I.2 Summary

From this introduction the originality of this thesis can be described. Firstly, the work is concerned with supporting interface design based on a sound and empirically validated discussion of representations and users. The foundation of this discussion is the close match of the technological and physical aspects of interfaces to the cognitive characteristics of the user and the tasks they are to perform. By making the user's cognitive facilities central to the allocation strategy, the thesis stands virtually alone in the multimedia literature, adding to the work of the few who have alluded to this focus, e.g. Palmer (1978), Mayes (1992b), Faraday (1995), and Stenning (1995). This approach is also analogous to the goal of the literature on 'cognitive tools' (Mayes, 1992a), which aims to match computer-based teaching strategies to the cognitive strengths of users.

The thesis now continues with an in-depth discussion of the prominent methodological research, whose drawbacks have already been touched on, but which require further elucidation.

II. Thesis Overview

The thesis is divided into four parts;

- Developing a cognitive and linguistic based approach to media allocation;
- Mental models and expressiveness;
- An empirical investigation of the relationship between expressiveness, mental models of task performance;
- Towards a methodology for multimedia user interface design.

Each part will now be described in more detail. Figure 1 gives on outline of the thesis structure.

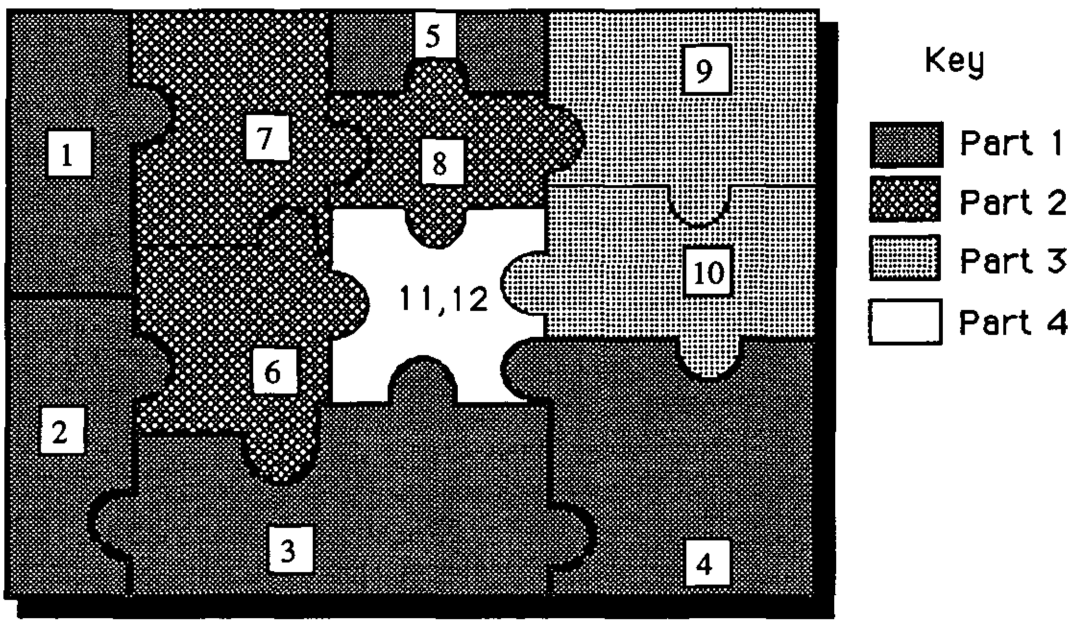


Figure II.1: Fitting the thesis chapters together

Part 1: Developing a Cognitive and Linguistic based Approach to Output Media Allocation

Part one describes candidate methodologies for multimedia user interface design in the literature. The methods address the problem of selecting and/or presenting multiple representations in the user interface. The methods are divided into three groups which view the problem of media selection from a conceptual view point, an architectural view point, or an implementational view point. The lack of proper consideration of the user's cognitive processes is seen as a major drawback in all these methods.

Chapter 2, discusses research aimed at defining a consistent multimedia terminology. A set of standard definitions are introduced which classify a multimedia interface as one incorporating more than one distinct representation system in parallel, i.e. all modern Graphical User Interfaces. A main drawback in this literature is the lack of formal definitions for the wide variety of non-graphical media. To address this issue, the need for a unifying dimension is identified which will allow these media to be compared, and will allow cognitive structures to be discussed. The expressiveneness of medium is offered as such a dimension. This defines the amount of domain information a medium can represent, and is seen as dependent on its encoding mechanisms. To begin to examine these mechanisms in more detail, Chapter 3 makes an anology between interface media and writing systems. These are described from their historical origins, to their manifestation in modern multimedia interfaces. Thus, the chapter draws out the different ways in which different media encode information.

In Chapter 3, effective support of problem solving by the interface is described as the main purpose of HCI. This is related to studies in the cognitive psychology, artificial intelligence, and the scientific visualisation literature. A key factor identified in effective problem solving, is the right level of abstraction of task domain information. It is suggested that expressiveness can determine how much abstraction a medium can provide. In Chapter 5, the importance of the cognitive effects of using different interface media are discussed, in terms of mental models. The study of the literature shows the importance of mental models in effective task performance or problem solving. Moreover, by addressing the affect of the interface representation on a user's mental model and the effectiveness of their subsequent task performance, the thesis stresses the need for investigating this area. Finally, expressiveness is suggested as a link between effective interface activity, interface media and mental models of application domains.

Part 2: Mental Models and Expressiveness

Part two describes the theory which links task domains, representations, and mental models. Chapter 6 discusses media as linguistic systems in terms of morphology, lexicon, syntax, semantics and pragmatics. However, it is stressed that interface media present a different case from the study of natural language in that they represent closed application domains , rather than the open world. This difference affects how much of the linguistics literature is applicable. This discussion provides the representational foundation for a full definition of expressiveness .

Chapter 7 describes the notion of media expressiveness in detail. Expressiveness is described as the domain-dependent, abstractive ability of a medium and is a product of its encoding mechanism. Specifically, it defines the number of different abstractions of a domain the encoding mechanism of a medium allows. The chapter ends with a description of a variety of media in terms of expressiveness. Chapter 8 describes the relationship between expressiveness and mental models, and postulates that it is the correct level of expressiveness which ensures a mental model which will effectively support task behaviour. Given this investigation, the expressiveness definition is extended to account for the expressiveness afforded by emergent perceptual properties of low expressiveness media. These are termed 'perceptual pragmatics'. The chapter ends with a description of a variety of media in terms of the mental models they would induce.

Part 3: An Empirical Investigation of the Relationship between Expressiveness, Mental Models and Task Performance

Part three develops the options for a preliminary experiment to investigate the relationship between mental models and expressiveness in the solution of tasks. In chapter 9, possible experimental designs are investigated. The chosen method involves the use of a complex domain requiring subjects to control traffic flow in a road network. The proposed method of elucidation of the subject's mental model is described. This involves domain knowledge that subjects may learn being classed into two types; declarative (obtained from subject verbalisations and divided into a number of general categories) and procedural (obtained from an automatic performance log). These categories are also intended to allow results from the study to be generalisable to other application domains which may exhibit similar characteristics. Five media are chosen as representative of common output media; animation, static video, table, bar chart, and graph. The media cover two levels of

expressiveness; low and higher. These are defined by the preceding theory.

Chapter 10 describes the actual experimental method and the discussion of the results. The experiment investigates the effect of expressiveness on a range of problem complexities in the system, and the types of models induced by the different classes of media. The results suggest that the right level of expressiveness does help subjects in the performance of harder tasks and within the comprehension of abstract domain concepts.

Part 4: Towards a Methodology for Multimedia User Interface Design

Part 4 describes conclusions and further work. Chapter 11 discusses a tentative framework for media allocation and two example interfaces are designed using this framework. This framework is based on the matching of expressiveness to the abstraction of domains required by users to achieve tasks. The need for the methodology to take account of the user's cognitive capabilities is stressed and the experimental results are seen as the beginning of this procedure. Finally, Chapter 12 concludes the thesis by suggesting that modern interfaces should be based on the notion of representational expressiveness and its relation to the correct mental models of task domains. This would allow the wide range of non-graphical media to be used in the more socially defined tasks which are becoming prevalent. To this end, further experimentation is suggested using a wider variety of expressiveness.

Part 1

Developing a Cognitive and Linguistic based Approach to Output Media allocation

Chapter 1

Multimedia Interface Design Methodologies

1.1. Introduction

As a starting point for a discussion of the relevant multimedia literature, there follows an overview of the notable approaches to the problem of media allocation. This discussion is divided along similar strands as mentioned in the prologue, top-down design, conceptual system descriptions, and implemented systems. Representative examples of each strand are presented, and the commonalties and drawbacks are discussed. Particular emphasis will be placed on the consideration that the candidate methodologies give to the user's cognitive characteristics in the selection of output media.

As a basis for this discussion, Hutchin's 'Model World view' (1989) will be adopted. In this view, the user is seen as *acting* on some *world* using the representation and functionality provided by the user interface.

1.2. Top-down Design Approaches

A large body of research approaches the problem of interface design using a *top-down* design approach. What distinguishes these approaches is their *a priori* analysis of some or all of the three most important aspects for the designer, the user, the interface, and the tasks domain. The work can be grouped into four categories which will be discussed in more detail.

- The *type* of data that the represented world contains;
- The selection of a presentation medium;
- The combining of different media temporally and spatially;
- The descriptions of user goals in the world.

In these discussions, particular emphasis will be placed on the pointers towards a robust methodology.

1.2.1. The Type of Data that the Represented World Contains

A number of authors see the most pressing problem in providing multimedia representations as delivering an adequate representation of domain data. In general, these approaches provide a formal definition of domain data *types.* This canonical description can then be used as a template for any form of data which may need to be represented. Database theory provides a rich basis for this classification, and a number of data taxonomies have been suggested. For example, work by Arens et al. (1993) provided a comprehensive taxonomy of data types, i.e. ordered/unordered integer and real values. At this fundamental level, the term *knowledge* was used to suggest that the *data* defined by the types, was required for the solution of some specific problem. This type definition was part of a comprehensive description of both the types and their *instantiations* (see footnote [1]). This is shown below (the superscipts are used in the discussion)

The Data Type

Type[1]: Nominal, Ordinal, Quantitative;
Sampling[2]: Discrete, Continuous;
Meaning Density[3]: Continuous, Discrete;
Distribution of Data in the World[4]: Sparse, Dense.

Data

Dimensionality[5]: 1..infinity, i.e. Is the data a unary, binary....value?
Transience[6]: Permanent, Transient.

---

[1] This describes the notion of a value being declared from a type, e.g. in 'C' the value A will be instantiated as an integer by the statement int A; from then on, A can only take on integer values.

An example application of this taxonomy is the discussion of a photograph. Arens et al. view the photograph as a collection of nominal data points [1] which have infinite dimensionality [5] (pixel intensity), are permanent [6], whose meaning varies continuously [2,3], and are densely populated [4]. This description is intended to help the interface designer match the photographic medium to some *world.* Ideally that world would be populated by densely packed points which took on some value which varied continuously.

A second view of data characterisation is provided by Alty and Rijkaert (A&R, 1993). Their data types are once again *nominal*, *ordinal*, and *quantitative*, but the work offers a different perspective on the building up of *complex* data elements. Knowledge is treated in an *object-oriented* way, which includes data *objects* and *inheritance* (see footnote [2]). Three levels of data are defined:

- Low-level: **primitive** elements e.g. temperature;
- Medium-level: **derived** elements e.g. density (mass divided by volume);
- High-level: **complex** elements e.g. triple co-ordinates {x, y, z}.

These variables are further described, like Arens et al.'s work, in terms of a comprehensive type and instantiation description with the addition of derivation and inheritance ([7, 8] below, [1, 2] are definitions matching Arens et al.'s):

Name: A String
Type[1]: Nominal, ordinal, quantitative
Cardinality: Single or double valued
Range - Maximum and minimum values
Ordering - Ascending or descending
Stability - Static or dynamic
Continuity[2] - Continuous or discrete
Directionality - Scalar or vector
Derived from what?[7] (for derived variables)
Using what variables?[8] (for complex variables)

A&R's taxonomy provides a robust description of data, but struggles when a more complex structure is to be described, e.g. a photograph.

[2] The instantiation of types from other types, taking on their data definitions and processes.

A third view of knowledge is provided by Roth and Hefley (R&H, 1993). Like Arens et al. they make the distinction between fundamental data types:

`Nominal, ordinal, quantitative;`

and the *meta*-dimensions of

`Coordinate, amount.`

The question of *domains* was addressed further by Roth and Mathis (1990) as manifested in automatic diagram generator, SAGE (also Roth et al., 1994). In these studies, *domains* are simply immutable types based on the four fundamental measures of the physical world, i.e.

`Time, space, temperature, mass.`

Unlike A&R, both Arens et al., and R&H described *relationships* within their data framework. This borrowed from relational database theory in the following definitions:

Binary relations: modelled as diatonic attribute objects;
Complex ternary relations: modelled as multi-attribute objects.

The further distinction of data-set size was seen as particularly relevant to represented domains which are data-rich, e.g. solid modelling systems, weather forecasting systems. Roth and Mathis suggested the following should be considered:

• Holding intra-data meta-information to allow aggregation and partitioning along chosen dimensions. This allows the visualisation of specific aggregations of large data-sets in question (Earnshaw and Wiseman, 1992; Keller and Keller, 1993);

• Showing dependencies between elements which allows the effects of changing one value to be traced throughout the data-set.

Of these three approaches, only Roth & Mathis and Arens et al. make any allusions to the relevance of a data description to a task. However, what all the methods lack is a *detailed* description of how the data is to be used. For example, if a domain of the type identified earlier (infinite dimensionality, densely populated), were to be studied for clusters of high values then the photograph would be ideal (see Figure 1.1). However, if the number of points with a value of 99 were to be known, a graph would be more useful, assuming that there are too many points to count manually (see Figure 1.1). Two issues are apparent in this example:

- Data has its own properties, e.g. density, which can affect which media should be chosen to represent it;

- Data is inert, and therefore to be useful to a task must become knowledge; an output medium should be chosen which allows this.

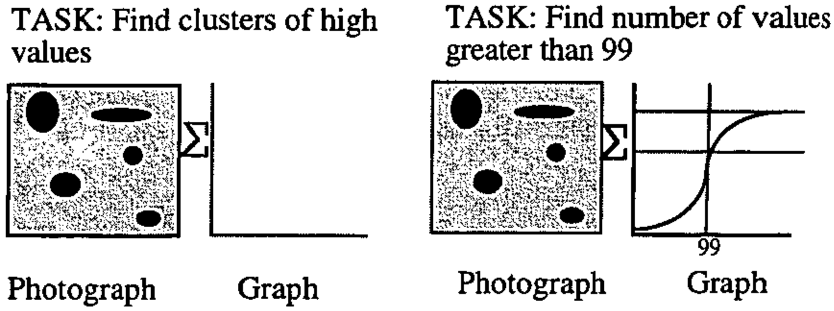


Figure 1.1: Matching data representation to task requirements: Photograph vs. graph.

1.2.2. The Selection of Presentation Media

Given the data descriptions of the previous section, there are a number of studies which define techniques for the mapping of domains to particular output media. The data-driven approach is exemplified by Mackinlay's A.P.T. system (1986). This was based on the ability of media to represent data of a given application domain *expressively* and *effectively*. These terms are defined below:

Expressiveness : The ability of a medium to represent a relational type[3] with its associated source and target domains, viz. can the output medium display the data, e.g. a one-dimensional axis cannot represent an instance of a ternary relation type;

[3] An example of a relational type is 'numerical->numerical'.

Effectiveness : The ability of a medium to encode all the members of the domain data-set so that human information processing can take place effectively.

Comparisons between different candidate representations were also investigated based on empirical studies of human perceptual value judgements. For example, relative positions of points can be judged more accurately than relative areas, e.g. different sized circles. The *effectiveness* criteria was the closest Mackinlay came to considering the relationship between the requirements of the task and the perceptual capabilities of the user.

A second area of interest is the design of 'Management Information Systems' which are used by managers as aids for making decisions. Here, the presentation of information in a way which allows numerical and logistical tasks to be achieved quickly, is essential. The majority of research compared textual-tabular to graphical methods, since the latter used technologies that had become readily available to managers in the mid nineteen-seventies. The main contention in this work was summed up by Remus (1984)

"Although there have been major efforts to develop decisional [sic.] support systems to aid managerial decision making, there has been little research into how best to accomplish this. One basic question is how best to present the data to the decision maker.", (Remus, 1984: pp. 533).

Most studies make the assumption that graphical methods such as functional graphs and bar charts are inherently better in the support of decision making (Dickson et al., 1986). However, Pearce (1983) states:

"Many extraordinary and unsubstantiated claims are made about the educational potential of computer generated Cartesian graphs.", (Pearce, 1983: pp. 41).

Thus, many assumptions made in this literature about how to best represent information are based on factors which are independent of the task and the user and are not empirically validated.

Further studies demonstrate how media can be allocated regardless of task or user specifics. A number of system implementations rely on the conceptual description of data to allow the definition of allocation rules which *hard-wire* certain data-types to particular media. This type of allocation is shown in the WIP system of Andre et al.

(1993). Here, text and graphics are classified as media and the following mappings are defined:

```
Concrete information, e.g. objects-USE GRAPHICS;
Spatial information-USE GRAPHICS;
Temporal information-USE TEXT;
Covariant information, i.e.cause and effect-USE TEXT
AND GRAPHICS;
Quantification-USE TEXT;
Negatives-USE TEXT.
```

The *hard-wiring* of representations to data types suggests a presentation knowledge-base, i.e. what a medium presents best. Work on the PROMISE system (Alty and Rijkaert, 1993) demonstrated this by providing an interface management system with a knowledge-base of the presentation qualities for the available media. As with the management information systems work, these qualities were described independently of other environmental factors. Thus, by tagging media with inherent qualities, the task of matching data to media becomes a pattern-matching exercise.

Finally, there has been the development of strategies have been developed for choosing media based on more than just the data description or perceived inherent advantages of a medium (Coll et al., 1994). Again, this work stems again from the design of management information systems. In this study, Coll et al. state:

"It is becoming apparent that the utility of a particular display type is contingent on the values of the many variables surrounding the target problem", (Coll et al: pp. 77.).

These 'many variables' are defined further, as shown below:

Task Independent Variables:

User Variables:

Personality/psychological type, e.g. analytic vs. heuristic;
Education/training speciality, e.g. engineers vs. artists;
Level of expertise with available system., e.g. novice vs. expert.

Presentation Medium Variables:

Acceptable input;
Quality/richness of the display;

Capacity for restructuring the display;

Intuitiveness of interaction with the display;

Time between the rendering request and display presentation.

Task Dependent Variables:

User Variable:

Level of expertise with the target problem.

Data Variables:

Type of data;

Amount of data needed;

Amount of data available;

Variability of data available, i.e. closely packed vs. spread;

Complexity of data, i.e. dimensionality.

Task Variables:

Problem requirements, e.g. skills, data sources;

Problem complexity, i.e. solution structure;

Amount of knowledge available for solving the problem;

Time available for solving the problem.

Work Group Variables:

Work group consensus, i.e. group think.

Thus, the whole of the user-system environment was considered. This work is echoed by Jarvenpaa (1989):

"The effectiveness of computer graphics on decisional support tools varies as a function of the task environment in which the user is operating.", (Jarvenpaa , 1989: pp. 286).

The above work shows that considering the whole range of factors involved in the selection of media is prohibitive, therefore it is necessary to establish which are most important. A number of approaches have been adopted to take into account at least some of the factors outlined by Coll et al. WIP (Andre et al., 1993) uses a database of user characteristics which include mother-tongue, and expertise. This information is used to influence the final rendering. Also, the BOZ system of Casner (1991) bases the choice of presentation medium on both task and data constraints. The conceptual task description is a series of PASCAL-like, primitive operations (called Logical Operators , or `LOPS`. See Figure 1.2). These are mapped to perceptual properties (called Perceptual Operators or `POPS`) of graphical media, which will allow the task to be carried out as a perceptual operation, e.g. the LOP, `COMPARE (A, B)` is mapped to the POP, *comparison of the length of two lines.* This represents a tightly coupled presentation process where the link between representation, task and data is made explicit. However, the task descriptions are limited to simple mathematical operations which produce correspondingly simple representations. There is clearly more to computer-based tasks than unary or binary mathematical operations, but these are beyond the scope of Casner's system.

```
(DOMAINSETS
        (flight NOMINAL 50)
        (origin NOMINAL (pit hou dal ord alb mex gdl qto paz bga)
        ...

(PROCEDURE
        (let ((found nil))
        (while (and found (findFlightWithOrigin FLIGHT 'pit')) do
                (if (available? flight 'T) then ...
(OPERATORS
        (LOP findFlightWithOrigin (<FLIGHT> <origin))
                (ASK (Origin <FLIGHT> <origin>) ...
```

Figure 1.2 : Air-flight booking task description in BOZ Casner (1991)

Marks (1991) introduced the concept of *communication pragmatics* which are more abstract descriptions of user goals than Casner's logical operations. Marks implemented a system for displaying network topologies. Since it was identified there was no explicit, data-related reason, why one network configuration should be shown rather than another, Marks introduced *pragmatic* operations to explain why users preferred different configurations. These *pragmatics* were taken from linguistic theory in an attempt to capture *meta*-aspects of the human-computer dialogue in an analogous way to the use of *intentions* and *beliefs* in human communication. Thus, Marks captured general aspects of the user's goal in the rendering of the network. For example, users may have required

- The description of a particular route through the network;
- The differentiation between inputs and outputs;
- The representation of cycles.

The *pragmatics* were converted into corresponding variations in the network topology, such as:

- Make a certain node more apparent;
- Make inputs and outputs clear;
- Emphasise cycles.

These *pragmatics* were an attempt to tailor output to vague communication aims, rather than the explicit task descriptions exemplified by Casner. The approach was developed further by Seligmann and Fiener (1991) and Feiner and McKeown (1993) with the COMET system. Here *communication intents* were introduced. These were more specific than Marks's *pragmatics* but did not approach the fine level of granularity advocated by Casner . Examples of *intents* are:

Show location : show the location of an object in context ;
Show relative location : show the relation of two or more objects.
Show property : show the texture, colour, size and shape of an object;
Show state: show the state of an object;
Show change : show a difference between a set of object states.

COMET produces representations which allow these goals to be reached by following a series of *design rules* which map media to *communication intents*. Thus, media selection is closely linked to the purpose of the communication. Clearly, COMET assumes that communication is *intent*-based, and only by articulating these intents to the interface design software, can a suitable presentation be generated.

In summary, since the selection of the presentation media depends on a number of factors, the usefulness of certain media can only be discussed when given an operating context. As Norman (1986) points out:

"The same information may be represented in a different form for different tasks. With the appropriate choice of representation, hard tasks become easy." (Norman, 1986)

An important corollary to this statement is that an inappropriate choice of representation can make easy tasks hard.

1.2.3. Combining Media

The power of multimedia communication lies in the co-operation of multiple output resources to meet a communication goal. However,this raises the question, "How does the user interface designer ensure the sum of the parts is better than using each part individually, or in series?". To answer this question, the relationship between different parallel representations must be investigated. Key issues identified in the literature are *completeness*, *consistency*, *redundancy*, and *coherence*. A comprehensive example which touches on each of these cases was shown in the WIP system (Andre et al., 1993). Here, the study provided an implemented system which was able to allocate domain information to output media in a systematic way. It represents interesting solutions to the four criteria addressed as addressed below.

1.2.3.1. Consistency

The presentation of *consistent* information was ensured by allowing the text editor and graphics editor to exchange information about domain objects. For example, to represent a coffee percolator switch being 'on', the word "On" was given a *semantic-state* which was sent to both the text editor and the graphics editor. Thus, the graphics editor produced an image which was consistent with the text label of the text editor, since both had the same semantic information. The use of an underlying conceptual domain description is an essential starting point for any interface design. A similar underlying conceptual description was used by Cornell et al. (1993) to allow multiple methods of representation, or *views*, of the same information.

Secondly, Andre et al. argued that there must also be consistency over time. Thus, if a certain domain concept is presented in one way, then the user would expect it to be displayed in the same way in the future. Implicit in this process is planning. The underlying system required some recording mechanism of what media allocations had been previously made . As Andre et al. pointed out:

"Content planning is strongly influenced by previously selected mode combinations.". (Andre et al., 1993: pp. 98).

In summary, any changes that are made in the choice of medium must be justified in terms of consistency. Potential caveats are changes forced by the unavailability of output resources which may be in use or inoperable. In this case, it may be necessary to violate consistency.

1.2.3.2. Completeness

The WIP system ensured that data which was divided across media was not degraded in any way, i.e. it remained complete.

1.2.3.3. Temporal Coherence

It was ensured the use of different parallel media was synchronised. Thus, textual descriptions were kept in step with graphics, and speech did not continue over images that it did not refer to.

1.2.3.4. Redundancy

Andre et al. suggest that multimedia interfaces provide the opportunity to mimic the information redundancy which is present in the real world. This can either be within a human sense, e.g. the WIP system uses textual annotations on pictures, or across senses, e.g. combining sound and light in an alarm situation. In the latter case, the two media offer redundant information but allow for one to be attended to while another may be undetected, e.g. if the operator were wearing protective ear-phones. However, it must be remembered that providing too much redundancy can be distracting, and may use media which could be used for representing non-redundant information.

In summary, a number of factors affect how media should be allocated in an interface, based on task and data characteristics. However, the literature shows that only cursory attention has been given to the relationship between different media and the cognitive facility of the user. It is this which the thesis will attempt to redress.

1.2.4. User Goal Descriptions

The previous sections have emphasised the importance of a consideration of the user goal descriptions in interface design. To address this, there needs to be some way of articulating the goals of the user to the system. Although the studies of Casner (1991), Marks (1991) and Fiener and McKeown (1993) provided some degree of goal description, they did not capture the inherent complexity of goal structures. However, this complexity has been discussed in other, more general, HCI literature.

The accepted method for task descriptions stems from Card et al. (1983) with their Hierarchical Task Analysis (HTA). This defines goals as being decomposable into sub-goals and actions; the actions must then be performed in sequence. The decomposition is recursive so must be stopped with some *termination* condition. This idea was made more relevant to user interface activity by Payne and Green (1986) with their *Task Action Grammar* description. Here, low level interface tasks, such as key-presses, are described in terms of grammatical constructions based on a *corpus* of allowable actions in the interface. This method, along with TAKD[4] (Diaper, 1993), TOM[5] (Diaper and Addison,1991), and KATS[6] (Johnson, 1989), all provide some form of description of the user's goals in terms of two important factors:

- The operations they wish to perform in the domain;
- The domain knowledge these operations require.

However, the question of how to feed such descriptions directly into the design of an interface is an area of active concern. One example of these methods is Diaper's TOM methodology. This has been used to describe the activities performed in a word processor, and in an air traffic control system (Diaper and Addison, 1991). The resulting conclusions suggested certain configurations of controls and presentation media.

Further work has been undertaken by Maybury (1993) in the MACPLAN system and to a lesser extent, Andre and Rist in WIP (1993). These systems conceptualise tasks in terms of 'communication act theory' and its derivative, 'rhetorical structure theory'. The system describes tasks in terms of speech acts (Searle, 1969), which represent the verbal intents, or aims of the user. Whilst these intents are limited to non-interactive presentations it is clear that they represent a valid way of conceptually formulating descriptions of the user's tasks. The process is comparable to Payne and Green's (1986) description of goal decomposition.

They represent an interesting meshing of traditional task analysis methods, linguistic theory, and human-human communication, whilst still stressing the goal driven nature of multimedia interface design. Though they are intended as generators of multimedia training aids, rather than representations which are used for problem solving, they are still of interest.

There are three levels of communicative-acts which can be seen to correspond to the goal, sub-goal and task/action description suggested by Johnson (1989), in Figure 1.3. This relationship is described in detail below:

---

[4] Task Analysis for Knowledge Descriptions.
[5] Task Oriented Models.
[6] Knowledge for Active Task Structures.

Goal<-> Main or Communicative act, e.g. *enable, motivate*;
Sub-Goal<->Rhetorical or inter-relational act, e.g. *inform, request, warn, promise, elucidate*;
Physical actions<->Surface or locutionary acts, e.g. *assert, ask , recommend.*

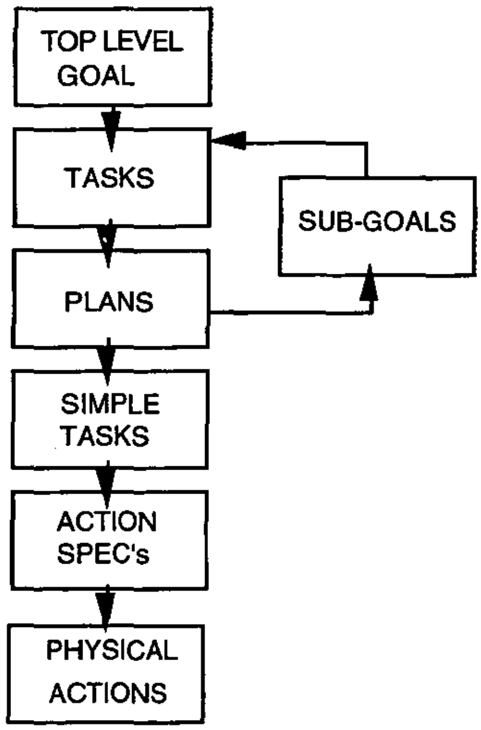


Figure 1.3: A goal/task hierarchy (Payne and Green, 1986)

The WIP system of Andre and Rist (1993) decomposses the sub-goals further which requires the system to process them further. Only when some physical action/locutionary act is possible are the *surface acts* rendered on the output hardware, e.g. 'ask' would be rendered as a dialogue box. In total, the result of processing the goal description is a *plan of action* for the presentation system to follow. This plan is made from a number of *frames* , which describe the intent of the communication, and surface methods that will be used to render it. For example (over),

Name: Describe Orientation;
Header: (Describe P A (Orientation ?orientation) G);
Effect: (BMP P A (Has Orientation ?orientation ?x));
Applicability: (Bel `P (Has Orientation ?orientation` ?x));
Main Acts: (S-Depict P A (Orientation ?orientation) ?p-orientation ?pic)).

In English, the *header* is translated as " The goal of presenter P is to describe the *?orientation* variable to recipient A", e.g. show that a certain dial must be turned. The frame outlines the *effect* of such a description, and the *applicability* ensures the presenter (P) believes this communication to be true (Orientation?=TRUE). Finally, the rendering of the goal is shown by the surface act *S-Depict*. This portrays the target object which must be *oriented*, using a picture, `?pic`.

In summary, Andre and Rist's system, described above, provides a complete description of what the presentation is to achieve. The system then produces an illustration that is matched to this target. In an analogous way, if a user's goal can be described, and the interface designed to allow the user to meet this goal, than a purpose-driven interface can be generated. However, whilst this may be possible, none of the approaches makes any reference to the cognitive facility of the user, and the effect this will have on effective task performance.

1.3. A Modular Approach

A second branch of the multimedia literature describes a multimedia system as a collection of communicating modules which deal with the various ontological concepts of the interface generation process. An example of such a design is shown in Figure 1.4 (over).

Here, the design of the system is seen as the processing of a task, which is described in some language, with some parsing system. The parsing process is informed by knowledge of system constraints and a description of the output media available.

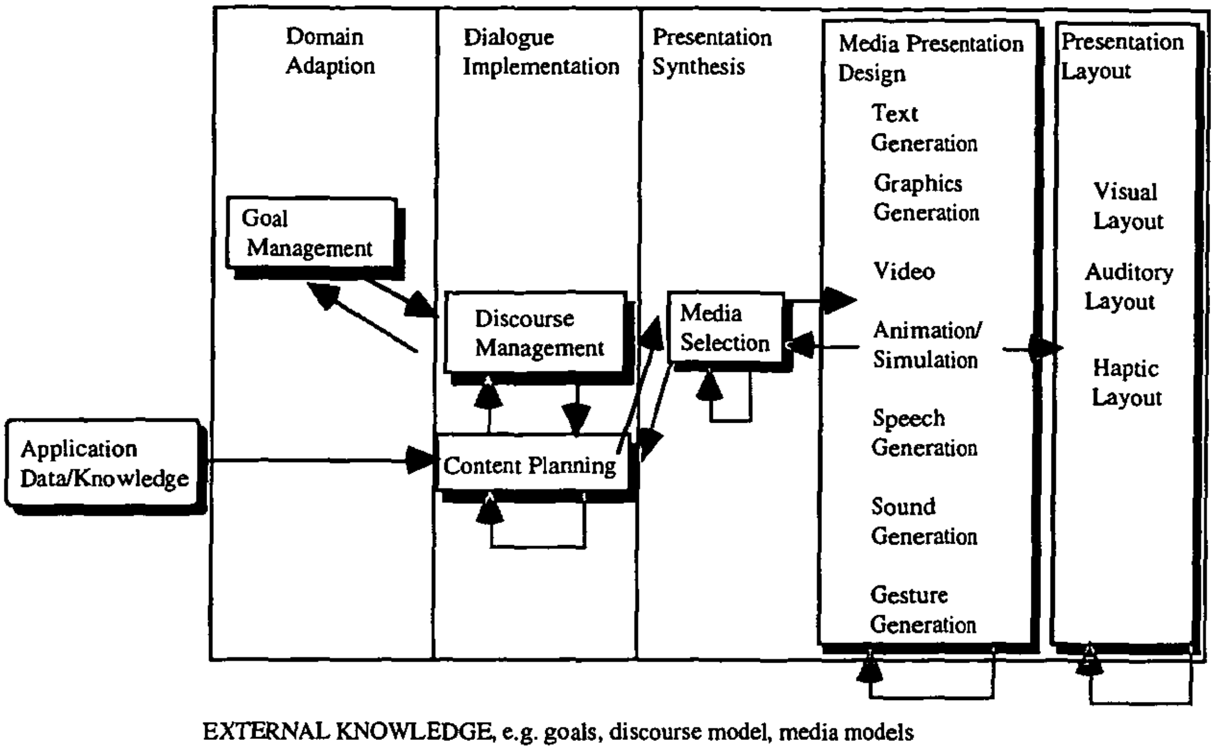


Figure 1.4 : A possible multimedia system architecture, after Roth and Hefley (1993)

1.3.1. The WIP Architecture

The modular approach was developed further by Andre et al. (1993) whose WIP system introduced user preferences to the system. This work borrows heavily from the field of User Interface Management Systems (Pfaff, 1985; Edmonds and McDaid, 1990) and provides an injection of user characteristics into the design process at an early stage, e.g. user ability (novice/expert). The idea of supporting interface design with user characteristics was addressed by Coll et al.'s (1993) categorisation outline in Section 1.2.2.

1.3.2. The PROMISE Architecture

The on-line processing of system constraints and medium choice is considered in the PROMISE system (Alty and Rijkaert, 1993). Central to this system, is the notion of an acyclic *tree* of media options which is pruned automatically according to resource conflicts which occur in real-time, e.g. two data objects competing for the same output medium. By allowing multiple representations of the same data, the multimedia presenter makes the user-dependent/resource availability dependent allocation of media possible. The architecture is shown in Figure 1.5.

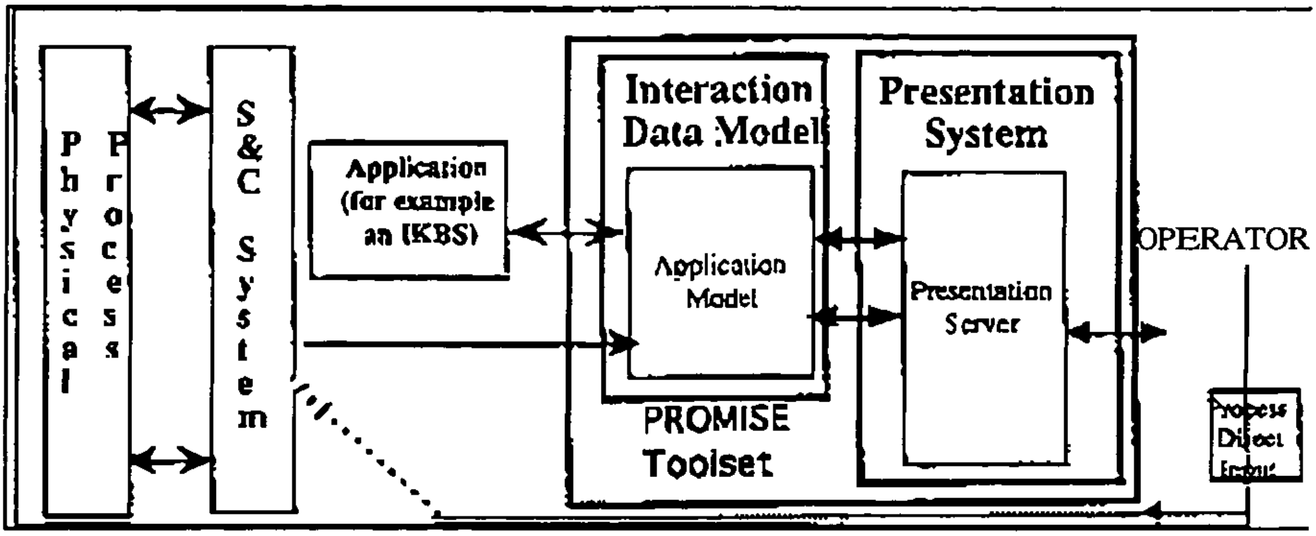


Figure 1.5: The PROMISE architecture Alty and Rijkaert (1993)

In summary, the questions posed by these modular descriptions are:

- How to articulate the task?
- How to select appropriate media?
- How to partition the interface from the application *back-end*?

Implicit in these questions are the cognitive capabilities of the user, and it is this which is not directly addressed by the literature.

1.4. Implemented Systems

The third salient area of research concentrates on the creation of functional prototype systems. Most effort has been devoted to systems which generate multimedia presentations, i.e. instructive dialogue design. Such systems are generally not interactive, but they do demonstrate how interfaces could be constructed from a description of data, user goals, and user characteristics.

1.4.1. A Multimedia Tool-Set

The PROMISE system of Alty and Rijkaert (1993) has already been briefly discussed. As an implementation, the system was designed to act as an aid to operators in process control environments, e.g. chemical plants, power stations. These environments are data rich and operators require many levels of data abstraction, from highly specific to highly aggregated representations. The system provided a set of tools to allow the construction of multimedia interfaces for any chosen application, but was mainly concerned with information output. Multimedia output was chosen to allow the output information from these applications to be spread across the senses, increasing the information *bandwidth* available or providing redundancy. The selection of appropriate media was seen as an essential issue, and a number of factors were identified which would affected this choice:

- Environmental constraints, e.g. noisy vs. quiet environments;
- Interface designer preferences;
- Operator preferences, e.g. 'I like graphs better than tables';
- Resource availability;
- General HCI/Multimedia interface design heuristics.

The interface was designed around a number of *application objects* which have predefined functionality incorporating input and output data. Each *interaction object* was allocated media according to the factors highlighted above. To represent this choice, an *interaction object tree* was specified which used logical functions to describe possible media allocations. An example is shown in Figure 1.6.

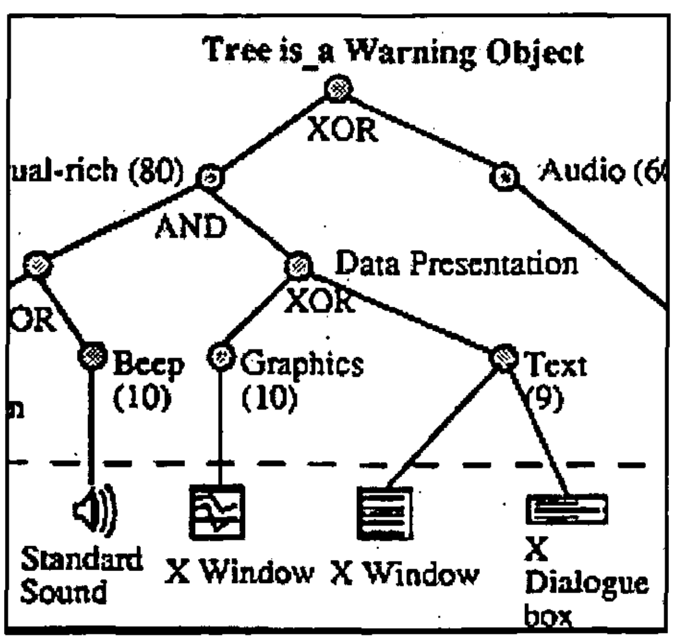


Figure 1.6: Interaction object tree (Alty and Rijkaert, 1993)

At the root of the tree was the interaction object itself. At each branch, the designer selects the type of medium which can be used to represent this object. This choice is also constrained by the M4I database[7], and the output hardware available to the system. An XOR term in a branch denotes that either branch would be a suitable representation of the object. Thus, if in the operation of the system, one of the resources were to become unavailable, then a free resource could be used instead. Consequently, although the design of the interface is not automatic, the interface is automatically represented in a number of predefined ways which depend on the run-time behaviour of the system. Also, user preferences can be incorporated into the structure at an XOR node, weighting it towards the use of one medium over another.

The PROMISE system showed how designer and user preferences can be incorporated into the interface design, along with other environmental constraints. However, the arrangement of media choices is out of the hands of the user, being decided by the designer. The designer must therefore consider all possible goals which may be attempted by the operator using the interface if all necessary representations are to be built into the *interaction object tree*. In summary, there is no way to articulate the goals of the operator to the system.

1.4.2. Automated Instructive Representations

The WIP system (Andre and Rist, 1993, Andre et al., 1993) was designed to generate goal-orientated instructive diagrams with graphics and text. As with the PROMISE system, the emphasis is on the designer of the presentation. Thus, the defined goals described what the presentation was to achieve. However, this method of goal description can just as easily allow the user to influence the representational form. The goals are described in terms of *plans*, a series of speech acts, as described in Section 1.2, which were held in the system. The original goal presented is an *intentional speech act*, and it was matched to a number of sub-goal *templates*. These *templates* are then used to either derive further sub-goals, or pass the presentation object to the graphics and/or text generator modules.

The output of the planner module is a directed acyclic graph (DAG) as shown in Figure 1.7, the leafs of the graph representing the rendered image, and the root the preliminary goal. This is pruned to ensure that there is as much reuse of objects as possible (e.g. a background could be used both to show the relative position of an object, and operations that can be performed on that object).

---

[7] A multimedia information database. This contained *a priori* mappings of data-types to output media.

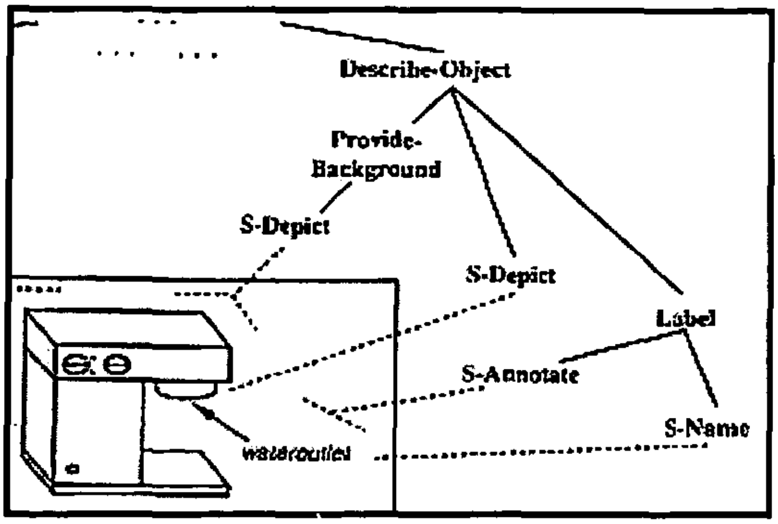


Figure 1.7: DAG for coffee percolator (Andre and Rist, 1993)

The system offers two main advantages:

- The design of the presentation is goal-oriented bringing the designer/viewer into the presentation loop;
- The presentation is adapted by designer/viewer preferences as well as interface design heuristics.

The system demonstrates it is possible to develop some form of goal description which can be parsed to drive the presentation process automatically. However, the initial goal *posted* by the designer is the only user-definable part of the planning process, since further subdivisions are based on rules already in the systems knowledge-base. Unfortunately, the system still does manage to produce varied presentations based on the speed at which sub-goals branches are identified, since any one goal may lead to a number of different possible sub-goals, in an analogous way to the OR node of the PROMISE system.

1.4.3. A Context Sensitive Presentation System

The AIMI system (Burger and Marshall, 1993) is designed to generate interactive multimedia applications with the following characteristics:

- The *intent* of the user is tracked allowing unsolicited information to be forwarded by the system;
- A representational language is used to define data objects;

•A context system keeps track of objects in the current *domain of discourse*. (after Grosz, 1986) allowing resolution of ambiguous user requests, e.g. put *that* there.

Like the work of Arens et al. (1988), formal models are used to define the data which was to be represented , the activities to be carried out using the data, and the different presentation media available. These models are described using the KL-ONE knowledge description language (In IS-A hierarchies, Brachman and Schmolze (1985)). In this representation, a series of *categories* are defined which have *role restrictions* placed upon them. An example is shown in Figure 1.8. These restrictions describe the members of the category in more detail, thus limiting the membership, e.g. a ship is limited by size.

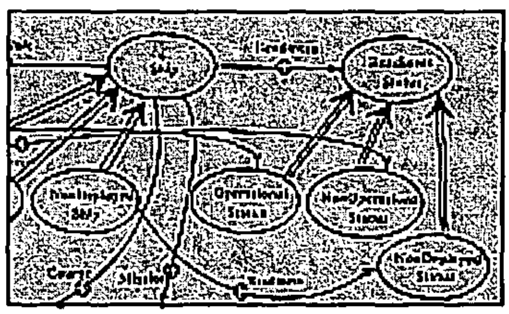


Figure 1.8:A KL-ONE Fragment (Arens et al. 1988)

The selection of media is still dependent upon heuristics supplied by the designer, but the added refinement is that *intent tracking* is used to ensure that information is presented in a consistent medium. For example, if a list of aircraft is presented and the user clicks on the button which gives a textual description, from then on, all descriptions will be given as text. This addresses the notions of consistency and coherence which were described in section 1.2.3. This allows the user to influence the presentation dynamically and soon develops an environment which is a result of previous actions by the user. Although, there is no explicit description of the goals the user brings to the system, this work does show how users may exert some influence on the presentation style, at *run-time*.

1.5. Summary

This chapter has examined the state of the art in multimedia interface design methodologies. By studying conceptual, architectural and implementational issues, a number of salient points have emerged:

- There are many factors which affect the choice of media in an interface, e.g. user goals, data characteristics, available output resources, user preferences and environmental factors, e.g. noise levels;

- A *first-cut* choice of interface media should allow only those media which can represent the domain data types;

- Interface design should be driven by the user's goals, but few systems have managed to produce a comprehensive and realistic description of these goals;

- Designers still rely heavily on task-independent media allocation design heuristics in the selection of appropriate media;

What all of these methodologies lack is a fundamental description of how different media represent information, tied to a description of how humans process this information within a task context. A preliminary step in this process would be the definition of a unifying dimension, which would allow different media to be compared This could then be used to in an allocation methodology by assessing their fitness to tasks by their position on this dimension. This may be based on their representational mechanisms, their effect on the mental structure of the user, or the support they give to particular tasks. In an attempt to define such a unifying dimension, the next chapter describes the investigation of prominent terminologies in the multimedia and HCI literature

Chapter 2

Developing a Terminology

2.1. Introduction

The description of methodologies for multimedia interface design did not make explicit definitions of what the terms *media* and *multimedia* mean. This question is far from pedantic since inconsistent definitions can lead to confusion. A number of authors have identified this as a major stumbling block in multimedia research (Arens, 1991; Stenning, 1994). To move towards a thorough definition of these terms requires a study, not just of user interface design, but also of social science, psychology, and ecology, i.e. all those areas where the effective communication of information is an active area of research.

To begin with, this chapter examines a number of different views of multimedia communication to find common ground between these disciplines. Then the literature which is explicitly devoted to a discussion of terminology is analysed and compared. Finally, a number of definitions are proposed which will remain in use throughout the thesis.

2.2. General Communication Theory

A number of authors have suggested a description of communication with the emphasis on at least one of the following:

- The technology used to communicate, e.g. monitors, loudspeakers;
- The producer or source of the information.

2.2.1. Technology Centred

The pioneering work of Shannon and Weaver (1949) described the process of communication in terms of the information conveyed (in data-*bits*). Their definition, which is based on probabilities, states that the maximum information that can be gained from a pair of complementary events is if both are equally likely, i.e. there is ambiguity. This information communicated is described as a stream of *bits*, but Shannon and Weaver make no attempt to describe the meaningfulness of the information.

The work of Bretz (1983) focused on broadcast media but still provides an interesting discussion of the technological aspects of communication. He defined a taxonomy of communication media as:

"the technology which carries a message", (Bretz, 1983: pp.10).

Bretz outlined a conceptual description of broadcast systems as shown in Figure 2.1. He also went on to describe a mixture of different types of data which included television signals in colour or b/w, audio information, telegraphy, and Morse code. Further media were defined as the physical carrier of the information which exist in the *environment*, e.g. air and wires.

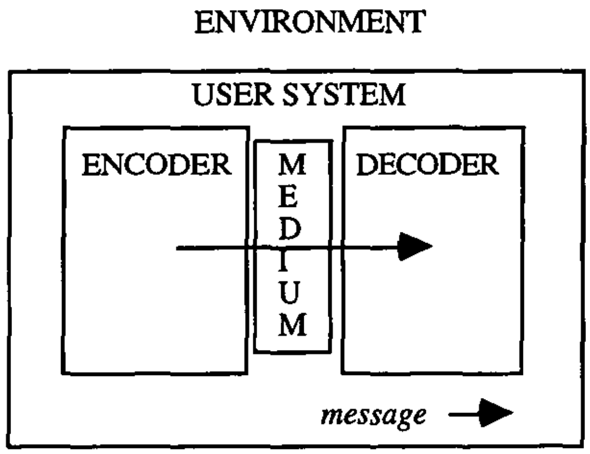


Figure 2.1: Bretz's block diagram of a communication system (Bretz, 1983)

Bretz's media are physical systems which are *disturbed* in order to transmit information. As with the Shannonian description, there is no attempt to define exactly what is being transmitted, in terms of semantic content.

Nigay and Coutaz (1993), draw a distinction between the technology which transmits a message, and the message itself. They term the Shannonian like description of bits-transferred, as a *mode* , which has no additional semantic message-related information. The opposite to this is a *medium* , which has data, plus semantic information. For example, the transmission of a binary encoding of a voice pattern as 'voice-mail', leaves the addition of semantic information to the user, this would be called a *mode*. Conversely, if the system attempted to form some semantic analysis of the message, in order to perform some post-processing, e.g. ordering messages by content, then they would call this, a *medium*. It is this separation of semantics from technology which has lead to the prominence of technological imperatives in the literature, mainly because they are easier to define.

In summary, what the technological view gives is a description of a communication system in terms of how data is transmitted. An analogy in interface design is a description of novel output media such as video, in terms of the video card, drivers, software tool kit, and example screen images. This will describe how video data is physically transmitted to the user, but will give no indication of any other characteristics, such as:

- What type of semantic information the medium is best at transmitting;
- The effect of the medium on the user.

It is this information the user interface designer needs if they are to make an informed decision about whether or not media should be used in particular tasks.

2.2.2. Producer Centred

Interesting work on the communication methods used by animals was collated by Krebbs and Davies (1987). In their discussion, they describe communication in terms of a specific *intent* or *goal* of the producer to alter the behaviour of the receiver in some way. For example, aggressive behaviour would be intended to force another animal away or to relinquish food. They state,

"communication is a matter of the manipulation of reactors by actors.", (Krebbs and Davies, 1987: pp. 76).

Here, the encoder or actor is the centre of study. Arens et al. (1993) develop this point by classifying the producers intentions in terms of 'a set of ideas or concepts to enforce upon the receiver'. Examples are,

- Affect the perceiver's knowledge: **teach, inform, confuse**;
- Affect the perceiver's opinion of topic: **switch, reinforce**;
- Involving the perceiver in conversation: **involve, repel**;
- Affect the perceiver's emotional state: **anger, cheer**;
- Affect the perceiver's goals: **warnings, orders**.

The categories are based on 'speech acts' (Searle, 1969) which characterised communication in terms of a hierarchy of *speech actions*. These actions provide a context for the communication. For example, a *request* sets up a communication situation where an answer is sought. Maybury (1993) also emphasised how a producer (in this case, a computer) has an implicit context for communication. This view is unusual in the user interface design literature, since most interactions are seen as constrained by the task and interface rather than by abstract intentions. Maybury states that these intentions will affect how the user processes the information trasnmitted by the interface . For example, a warning context should place the user in a state of alertness with their attention focused on the warning indicator. However, the warning may be implicit with the task domain description, i.e. a value moving above a critical level, so the *speech act* can also be implicit.

Thus it seems that the context of the communication situation is an emergent property of the task domain. However, since the interface *is* the domain (as far as the user in concerned) the interface designer must be aware which contexts are required and allow the interface media to produce this context, e.g. flashing light for a warning.

Work by Tubbs and Moss (1991) looked further into the phenomenon of actor centred communication by classifying factors which effected the force of a communicated message upon a reactor's ideas/concepts. Although some of these factors are a product of the environment, there are three factors which are relevant to the user interface.

- **Physical dominance:** e.g. loud voice, large size;
- **Status:** e.g. peer vs. boss;
- **Power** e.g. ability to influence life of reactor outside the communication domain.

These characteristics are interesting when considering the computer as a *producer* attempting to influence the user. Firstly, the *physical dominance* of the machine may be inherent in the hardware of the computer (large processing box and monitor). *Status* would ideally be a peer relationship, but generally the human would feel inferior due to a comparatively (by human standards) opaque and uncommunicative interface. *Power* will be dependent upon what information the machine holds, as will the ability of the machine to alter things outside the communication domain, e.g. switching off an important piece of machinery.

The final consideration in a producer centred view is whether human-computer communication should be modelled on human-human interaction. Since most people are adept at influencing those with whom they communicate and in making their communication context clear, what can be learnt? Chapanis (1981) and Hunt (1989) both use human communication as a basis for their studies of human-computer communication. However, there are important differences between the human-human and human-computer paradigms. As Hutchins (1986) points out:

"..the conversational metaphor does not quite fit the reality of most human-computer interaction. Typical conversations on "conversational interfaces" are very stilted ...using a severely constrained vocabulary and language syntax...of course the users interaction consists of typing not speaking.", (Hutchins, 1986: pp. 32).

Whilst this situation may be remedied by technological advances in voice recognition and theoretical advances in language parsing, Hutchins is not convinced. He suggests that user interface technologies can do more than simply model human-human communication, since they can provide characteristics that are useful to users, but are not human. For example, the option to communicate through some autonomous or intermediate agent which can perform complex activities for the user (Maes, 1996). This can occur in parallel with more conventional user controlled activity. Hutchins calls this 'collaborative manipulation', which suggests a co-operative process involving direct and indirect interaction. In the case of the system agents, the emphasis is on the interface's ability to interact with the world in a useful way. This places the communication focus onto the system to report the progress and results of this process. In this way, it is a return to the early batch command systems or UNIX-like command-line interfaces. where the producer of information is the opaque process within the computer system. This seems an unfortunate regression in interface design.

Other work stresses that since human communication uses a number of different message carriers, e.g. voice, gesture, body-movement, intonation, then interfaces should exploit similar physical characteristics. Assuming that this is a good idea, to implement this would require a knowledge of how humans allocate information to these different carriers. To the author's knowledge, there is little fundamental research in this area other than rough percentages of how much of a 'message' is carried by each mechanism. Without a *rationale* for allocating information to carriers, either redundantly or uniquely, such systems will remain unproven.

In summary, a producer-centred view of communication offers many opportunities for innovation, but little evidence of useful systems. What is clear is, however, is if the computer represents information in such a way, it can have an effect on the user's goals and beliefs about the system. Research must focus on how computers can offer new effects which are commensurate with their infinitely malleable nature. As Hutchins concludes,

"Seeking to imitate human behaviour with computers that are to have roles in task performance may be setting the wrong standard of performance. because computers can manifest behaviours that are not possible in any other medium, we should use our imaginations in the design process.", (Hutchins, 1986: pp. 27).

2.3. Terminology

The literature surrounding communication theory, and more specifically multimedia communication theory, disagrees on a number of terms. In this section, four descriptive taxonomies from the multimedia literature will be described and compared. In general, the classifications describe presentation media in a mechanistic way by defining what they are able to represent. The chapter will conclude with a statement of the terminology to be used for the remainder of the thesis, and a discussion of reasons for these definitions. These definitions are central to the level of descriptive abstraction placed on a discussion of output media. Consequently, they will determine the level of detail of task and user descriptions with relation to output media.

2.3.1. A Modality Classification

Bernsen (1994) avoids the notion of medium, and instead uses the term *modality*. He defines this as a "representational form" . Thus, he avoids the standard psychology definition (e.g. Ornstein and Carstensen, 1991) of a "sensory channel". He classifies modalities along four dimensions which are chosen since that they produce "profound implications for a certain modalities capacity to represent information.". Clearly Bernsen is very much concerned with the *type* of information a *modality* can carry, to the extent he does not consider any aspects of the task or cognitive capacities of the user. The chosen dimensions are:

- Linguistic vs. non-linguistic;
- Analogue vs. non-analogue;

- Static vs. dynamic;
- Arbitrary vs. non-arbitrary.

An excerpt from the classification is shown in Figure 2.2.

Modality	li	¬li	an	¬an	ar	¬ar	sta	dyn	graph	sound	touch
Static analogue written language	X		X			X	X		X		
Diagrammatic pictures		X	X			X	X			X	
Animated diagram pictures		X	X			X			X	X	
Real sound		X	X			X		X			X
Arbitrary touch		X		X	X			X			X
Touch structures		X		X		X		X			X

Figure 2.2: A selection from Bernsen's modality taxonomy (Bernsen, 1994)

As well as the definition of generic modalities using these dimensions, the physical mechanisms of representation is also considered. No specific definition is given of this, apart from the triple `<graphics, sound, touch>`. The *rationale*behind this distinction is not made explicit, but one assumes the triple represents technologies which can stimulate the three human senses, i.e. the psychology definition of modality.

This taxonomy excludes the term *medium*. However, in the sense of a *medium* meaning, "The means by which something is communicated" (Allen, 1990), the *medium* and the *modality* definitions are similar. This is because a *modality* also communicates information using some "representational form"; Bernsen therefore only implies additional semantic/syntactic constructs. As such, the words, *medium* and *modality* are not important, rather the meaning attached to them, i.e. the level of descriptive abstraction.

The taxonomy would rate "representational forms" which were both in the visual mode, e.g. written text, diagrams, as different modalities. Thus, using Bemsen's definition, a multi-modality system could exist in one sensory field such as vision.

It is suggested in the thesis that regarding a *modality* as a representation system, implies it is semantically and syntactically distinct. Thus, combining a number of modalities in parallel, which are all in the visual *mode* , would provide a number of rich, concurrent *media*. This is also true for combining *modalities* in different sensory modes, e.g. visual and aural. It is in this way that Bernsen's modality definition is best understood.

Other definitions run contrary to this. McMillan (1989), only distinguishes between representation forms which stimulate different senses, as being different media. Consequently, a multimedia interface *must* stimulate more than one sense. Clearly, this definition does not take into account the representational distinctness of different modalities, regardless of what senses they affect.

In summary, Bernsen's description is lacking for the purposes of this thesis since it does not adequately address the representational *mechanisms* of each modality, i.e. exactly how they encode information. This discussion is essential if an effective *rationale* for media allocation is to be proposed. Difficulties with Bernsen's work are further compounded by a lack of coherence in his definitions, for example:

"The presence of focus, and lack of specificity jointly generate the characteristic limited expressive power of linguistic representations.", (Bernsen, 1993: pp. 3).

On the contrary, intuitively, linguistic representations can be seen as highly expressive. This is an example of Bernsen's taxonomy being lacking of an adequate depth of definition. He never addresses why his *modalities* exhibit the qualities he identifies, e.g. linguistic and arbitrary, at a fundamental level of encoding mechanism, or in terms of the viewers cognitive processing. Without this level of investigation the taxonomy is useful only for stimulating discussion and identifying esoteric representations, e.g. dynamic hieroglyphics.

2.3.2. A Communication Environment

Arens et al. (1993) take a producer-centred, human-to-human metaphor for their discussion of medium and modality. The research constructs a highly detailed view of communication, which incorporates descriptions of the producer, the receiver, the data, and the media. Of interest here is the description of a medium which agrees with Bernsen's *modality* definition.

"Medium: a single mechanism by which to *express* information."(Author's italics)

Examples given of media are spoken and written language, diagrams, sketches, and graphs. The key word here is *express* , which suggests a representational *system* similar to that alluded to by Bernsen. However, unlike Bernsen's definition, this term is broken down into a recursive, multi-attribute description as follows:

- **Exhibit**: carried information item, e.g. an integer;
- **Substrate**: a background for the carried item, e.g. photograph background for a graph;
- **Channel**: one representational dimension of a carried item, e.g. size, position;
- **Carried item**: the actual information representation, e.g. point.

This mechanistic approach then focuses on how the information is encoded in the *carrier* item. Thus, this definition is further subdivided into:

- **Carrier dimension**: how many *channels* does the carried item have?
- **Internal semantic dimension**: how complex is the semantic system of the carrier channels? Do they all have the same semantics?
- **Temporal endurance**: is the information permanent? e.g. sound is not permanent;
- **Granularity**: how densely is information encoded?
- **Default Detectability** : how obvious is the information mapping?. e.g. perceptual or abstract;
- **Baggage**: is there any information which may be misconstrued? i.e. ambiguity;
- **Medium Type**: which sensory mode is used? e.g. aural, visual, haptic[1];

The term '*internal semantic system*' describes the semantic mapping which allows the carrying item to be decoded. An example is shown in Figure 2.3 using a multi-dimensional icon (Spence, 1989; Spence and Parr, 1990).

For example, in Figure 2.3 the number of windows (black squares) is the *carried item* for the number of bedrooms in the portrayed domain object. Consequently, the internal semantic system of the carrier is simple :

Number of bedrooms: *domain* **-> number of black squares:** *representation.*

---

[1] Touch.

Clearly, a more complex semantic system would be needed to describe a photograph or natural language. The icon also shows the recursive nature of the definitions, since windows are *carrying items*, as is the whole icon, but whilst the latter has multiple *channels*, the former has one.

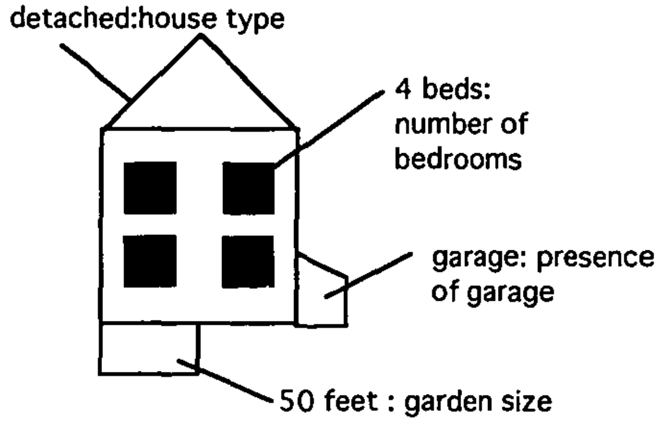


Figure 2.3: A multi-dimensional icon (Spence et al.)

The description adopted by Arens et al. allows for a comprehensive discussion of the representational qualities of media, but at a low level. In this way, it provides the missing aspects of Bernsen's taxonomy. It is rigorous enough in its approach to capture sufficient information about a representation system in order to correctly match it to a given data description, if not a task description. However, two drawbacks are evident.

• Firstly, the framework is geared towards pictorial or graphical systems which have physical correspondence with what they represent. In these systems the channels, substrates, and internal semantic systems are more easily identified. It is not clear how more abstract systems would be described.

• Secondly, the discussion is concerned with presentation graphics and therefore does not discuss interactive interfaces. A corollary of this is that the effect of representations on task activity is not addressed, this effectively rules out the notion of the interface supporting problem solving. This relegates the framework to a descriptive tool rather than a useful comprehensive methodology for user-centred design.

2.3.3. An Interface Design Space

Work by Frolich (1991) describes a *design space* for human computer interfaces. The method is not specifically constructed for multimedia interfaces, but it could be used to describe such interfaces. As Frolich says:

"to systematise the description of interfaces, but at a level of the whole interface rather than the input interface alone.", (Frolich, 1991: pp. 54).

An output interface design space is shown in Figure 2.4.

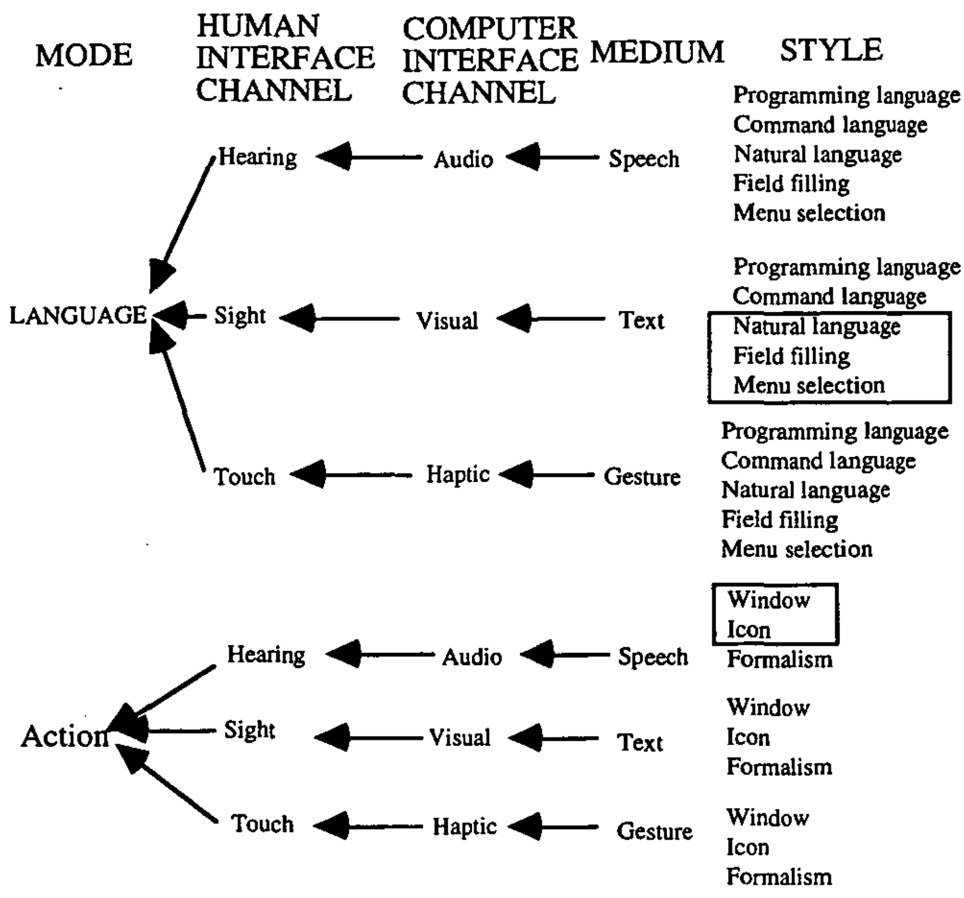


Figure 2.4: Frolich's output interface design space (1991)

The following definitions are made:

- **Mode**: "states across which different user actions can have the same effect.", e.g. using the language or action mode. A file could be deleted by entering the language based command **delete file <name>**, or by using the action of '*selecting* the file icon, *dragging* it to the *waste-basket* and then *dropping* it'.

- **Channel**: "an interface across which there is a transformation of energy", i.e. the hardware of transmission addressed by Bretz in Section 2.2.1

- **Medium**: "representational system for the exchange of information"

- **Style**: "recognised class of activity for supporting interface activity"

Frolich suggests that the combination of a *mode* and a *channel* creates a *medium*. For example, output using the language *mode*, and the voice-audio *channels* , defines speech as the *medium*.

The second stage is a decomposition of representations within a *mode*. For example, if spoken language is the mode, then this supports a number of possible styles or activities (in the interface). Frolich is careful to differentiate between language and action as modes within the interface. However, this distinction does not hold when these modes are *fused*, as may be the case in a multimedia interface, e.g. the verbal command, 'put *that there*'. This incorporates both a deictic[2] gesture and a verbal utterance, i.e. action and language (Hutchins, 1986).

Frolich's representational systems are also tied to the three sensory modes giving a course grain distinction between media in a similar way to McMillan (1989). This does not allow for intramodal media as defined by Bernsen's and Arens et al.'s frameworks. Clearly, media such as hieroglyphics, diagrams and written language stimulate the same sense but are very different in how they encode information; Frolich's description does not account for this. Also, no reference is made to the cognitive abilities of the user, or the task domain.

However, the framework does offer an comprehensive description of the interaction process, and demonstrates further definitions of *mode, channel,* and *medium.*

2.3.4. Meaning and Media

The direct encoding of meaning by a computer-based medium is addressed by Nigay and Coutaz (1993) and Salber (1994), in a descriptive framework for *interactive* multimedia interfaces. Their work is focuses on entire systems, rather than the interface alone. Due to this wider scope, the framework does not address the interface

---

[2] Pointing.

in sufficient detail. However, unlike Bernsen and Arens et al., a semantic related distinction is made between media, modality, and mode in the following way:

- **Medium**: "A Technological communication medium which holds no abstract concept of meaning.";

- **Modality**: "A type of communication used to convey or accept information.";

- **Mode**: "A a state that determines the way information is interpreted to extract meaning".

Hence a distinction is drawn between a *multimedia* interface and a *multimodal* interface, from the system's point of view. The former has no separate encoding of the semantics of the data, e.g. 'voice-mail'. The latter has some description of meaning, in addition to the raw data. However, they make the distinction that from the users' point of view the voice-mail system is *multimodal* since the user adds additional meaning to the raw message.

Using this concept, it is possible to *fuse* modalities together and observe the new meaning that can be interpreted from them. This is a central issue of multimedia/modality interfaces and the new types of interaction they can promote. Clearly, only by understanding the effect of combining meaningful information streams can the interface designer construct multimedia interfaces which support goal orientated behaviour. Though of some interest, this is outside the scope of this thesis.

Salber's work takes these concepts and, like Frolich (1991), constructs a design space for *modalities* and *media*. The 'Multiple Sensory Motor Framework' describes computer systems along the following dimensions:

- **Number of channels /modalities/media**;

- **Amount of meaning in channels**: medium vs. modality;

- **Dimensionality of channel**: same definition as Arens et al, i.e. 1-many

- **Fusion vs. Fission**: combination of channels vs. separation of channels;

- **Concurrence vs. Granularity**: parallel channels vs. sequential channels;

By distinguishing between the terms *medium* and *modality* the work further confuses the terminology, but it does bring in to focus the perspective to be adopted when discussing a multimedia interface, i.e. *human-centred* vs. *system-centred.* This differentiation should be made when discussing the communicative possibilities of an interface, and the meaning associated with the representational media used.

However, the description is lacking in its definition of what actually constitutes a *modality* or a *medium*, and whether they can share the same sensory mode. Therefore, as with Bernsen and Arens et al., without this detail the description is only of use as a general descriptive framework for multimedia systems rather than a methodology for media allocation.

Finally, a more interface-focused discussion from these authors is offered in a recent paper (Coutaz et. al., 1995), whose objective is stated as:

"Characterising and assessing aspects of multimodal interaction", (Coutaz et. al., 1995: pp. 115).

Four properties of multimodal interfaces are described, in an attempt to formalise the description of multiple representation modes in the interface for the solution of tasks. The framework is based on a primitive description of states, goals, modalities, and interactions:

- **state (s')**: "a vector of observables" of the system;

- **goal state (s)**: a state an agent wishes to reach (an agent can be a user or a system);

- **modality (m)**: an interaction methods the agent can use to reach a goal. *Expressiveness* of a modality defined by triple `Reach` (s, m, s'), which states a modality, m, allows the user to move from state s' to s.

- **interaction trajectory**: s_1->s_2...S_n;

A number of operations are also named and defined in a first-order predicate logic derivative.

- **Equivalence**: Two media with the same expressiveness, i.e. reach $(s, m_1, s') == reach(s, m_2, s')$;

- **Assignment**: Modality assigned to a trajectory if no other can allow `reach (s, m, s');`

- **Redundancy**: use two *equivalent* modalities at the same time;

- **Complimentary**: more than one modality is required to reach required state. `Reach (s, M, s')`, e.g. gesture and text, "put*that* there".

Within these definitions, two important notions are implicit. Firstly, the task centredness of the framework ensures it can be used by interface designers. Secondly, the importance of the representation in the solution of tasks, as defined by the *reach* triple, is central to the theory. This agrees with the main tenet of this thesis which stresses the importance of representation on interaction. Finally, the user is addressed by defining preferences on the combination and selection of modality. However, where the framework falls short is in the describing of *how* or *why* media support some tasks better than others.

2.4. Modern Interfaces and Multimedia

Given the discussion of general and computer-based terminologies we can now turn to common manifestations of interfaces. To begin with, we will study those interfaces in various application domains which purport to be multimedia. In doing so, the commonalties between these different definitions will become clear.

Firstly, it is important to remember that two domains are present in any user interface which must both be represented; the task domain, and the interface domain. The former is self-explanatory. An example of the latter is the functionality associated with windows, e.g. minimise boxes, scroll bars.

2.4.1. Multimedia Application Domains and Interfaces

As a starting point, we will study those interfaces which use multiple representational forms. Although we have already looked at research on terminological issues, it is interesting to examine definitions within the popular press and computer-based literature. Here, we see the application of multimedia to real domains rather than theoretical studies. The discussion will be divided into:

- Education and Computer Assisted Learning;
- Computer Supported Co-operative Work;
- Information sources, e.g. museum catalogues, libraries;
- Entertainment, e.g. computer games.

These will now be described in more detail.

2.4.1.1. Education and Computer Assisted Learning

The development of interactive teaching resources forms a large part of the educational multimedia literature. Though sparse in any empirical evaluation, there are a number of interesting applications and attendant definitions of what a multimedia application should achieve.

Morin (1992) states: "multimedia consists in combining the three different text, image, and sound media on the same platform", (Morin 1992: pp. 191).

He also stresses the difference between *producers* and the *consumers* of multimedia software. This is also addressed by Barnard (1992) who stresses the importance of interactivity in educational multimedia. He suggests students should be allowed to be producers and create their own films/presentations.

Hietala and Nummenmaa (1992) describe a multimodal children's database as "using several communication *methods*"[my italics], (Hietala and Nummenmaa, 1992: pp.1).

In summary, educational multimedia focuses on the increased learning potential and 'attention-getting' of multiple representational forms. The division of media is generally restricted to text, graphics, still/moving video because these are the media which are common in standard teaching text books.

2.4.1.2. Computer Supported Co-operative Work (CSCW)

The advent of inexpensive video and sound delivery on the 'desktop' has generated a large amount of research into how this technology can support group work. Of particular emphasis is the bringing together of geographically disparate groups of people into a virtual office or conference room. Definitions are again vague. For example, Miah (1994) offers the following distinction:

"Most often the terms 'multimedia conferencing' and 'video conferencing' are used interchangeably... Multimedia conferencing describes the ability to hold a conference ...using a variety of different communication sources, such as video, audio and data, whereas video conferencing only refers to the ability to hold a conference with audio and video sources.", (Miah, 1994).

Miah's gross distinction between video, audio, and data is typical of the literature it originates from the high profile of technologies in this field, rather than theories or examples of beneficial use. Other related work by Gale (1990), Gaver et al. (1991), Fish et al. (1992) and Sellen (1992), all use similarly coarse grain definitions.

2.4.1.3. Browsable Information Sources

Information science has been quick to embrace new interface technologies, but with little idea of how information should be best represented. As with the CSCW literature, the concept of medium is coarsely defined. The use of the video and audio capabilities of the Internet has provided a *test-bed* for a number of information sources, e.g. on-line museums (Mannoni, 1996). The majority of *sites* provide text interspersed with still images; though video can be down loaded if there is sufficient time,.

2.4.1.4. Entertainment

The expansion of the PC market has led to the term *multimedia* becoming common *parlance*. The term generally defines the hardware capabilities of system, e.g. sound card and video-capture card.

The brief survey of popular literature and present research suggests that technology is still in the ascendance. Consequently, the terms *media*, and *multimedia* are confined to describing hardware issues, with little consideration for the finer distinction which an interface designer requires. The lack of this distinction has compounded the lack of discussion of when media should be used and why.

2.5. Why are Multiple Media Needed?

From this discussion, we identify the need for a finer description of media which captures the multi-variant forms of representation which exist in common user interfaces. In defining media more sensitively, we are able to call on the psychology literature which represents investigations on how different representational forms are interpreted, e.g. (Cleveland and McGill, 1986; Stenning, 1991).

As an example, Figure 2.5 shows an interface from the Apple Macintosh, and outlines the different representation systems which are used to convey information about the state of the underlying domain. How should such an interface be described?

Since there are a number of distinct output media (representation systems) present we must call this interface *multimedia*, or more precisely, *multi-representational.*

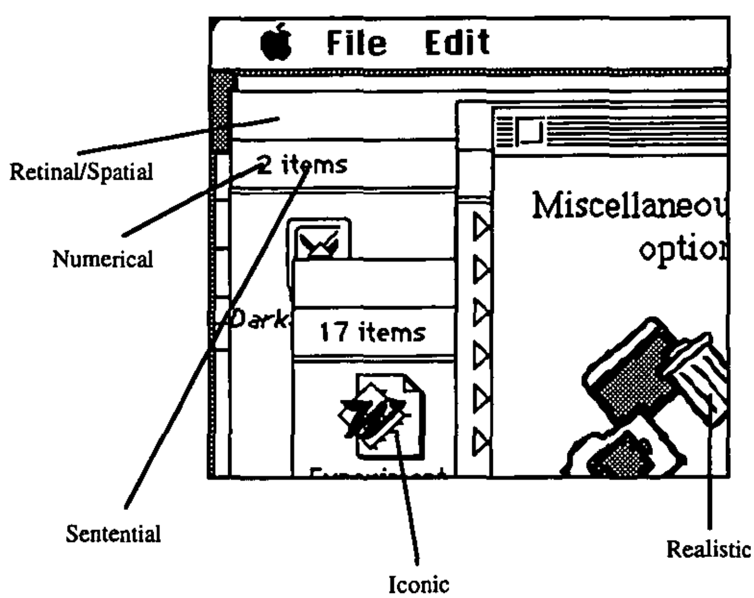


Figure 2.5: Decomposition of Macintosh GUI into component media

Of course, this amalgamation of many different forms of representation is not new. Since maps became mass produced in the 18th century, cartographers have developed a complex nomenclature which encompasses a range of representation systems. e.g. icons, size/shape/colour mappings, text. (Simutis and Barsam 1983; McCleary, 1983)

So what is new about modern computer interfaces? Firstly, they offer the ability to manipulate representations in real-time. Secondly, the designer can allocate representations that are not possible with static print media. Examples are temporal text (Wong, 1996), interactive 3D representations, and interactive full-motion video. These will be discussed in Chapter 4.

In summary, any interface which uses multiple representational forms must be considered a multimedia interface.

2.5.1. A Proposed Terminology

In summary, the following terms are defined:

medium: representational system with its own distinct syntax, semantics and pragmatics, e.g. graph, table, static video;

expressiveness: an emergent property of a medium which measures how much information it can carry, e.g. a single axis is not very expressive, natural language is very expressive;

multimedia: communicating simultaneously using a number of different media;

modality: a human sense, e.g. aural, visual;

multimodal: an interface which stimulates more than one human sense, e.g. visual and aural.

2.6. Summary

The confusion of terminology in the multimedia literature has been highlighted. To remedy this, a number of definitions were proposed for *medium*, *multimedia*, and *modality*. From this discussion, the need for a finer description of interface media is identified. Only by providing this can the multi-variant forms of representation that could be allocated in user interfaces be fully understood in terms of how they represent information. This must be the first stage in any consideration of the relationship of these media, in particular task contexts, to the cognitive characteristics of users. Clearly, there is a need for a unifying dimension over which media (that represent in different ways), could be compared within a cognitive context. Unfortunately, no such definition of sufficient detail has been found in the literature.

The next chapter begins the progression towards this unifying dimension by studying the relationship between common interface media and non-computer based media, such as natural language. In doing so it attempts to draw out the more fundamental representational issues, and address *all* of the media which may be deployed in the user interface.

Chapter 3

An Introduction to Representation Systems

3.1. Introduction: From Pictographs to Natural Language

Given the importance of addressing fundamental aspects of interface media, it is seen as essential to highlight what the representational bases of these media are. Moreover, this thesis proposes that the different ways in which representation systems encode information has important consequences for how to provide effective support for tasks. This is particularly true of the cognitive representation of the task domain that such media may induce. Thus, this consideration is an important step towards a dimension which links cognition, interface media, and tasks.

One approach to this, is to examine how non computer-based media represent information. This study can be used to identify situations where the representation mechanisms of non computer-based media are successful (or not) and relate these to user interface media. Thus, this chapter provides a grounding in the common ways that information is encoded in representations. This discussion is essential before any consideration of representations in detail can be made.

3.2. The Development of Writing Systems[1]

Many modern media have their genesis in early writing systems, so the discussion begins with writing system development. This will show the variety of ways in which information can be encoded, within an historical context. The importance of this historical perspective is also advocated by linguists (Sampson, 1985) as an important aid to studying modern spoken and written languages.

Since something analogous to 'natural selection' is evident in the development of these systems, with the poor systems being replaced by better developments, the discussion also shows the advantages and disadvantages of each.

[1] This discussion is equally applicable to non-verbal, auditory representation systems.

3.2.1. Different Forms of Representation

Sampson (1985) describes the different forms of writing which can be used to convey information. He stresses the relationship between the spoken sounds of language and its written form. In terms of this discussion, this distinction is unnecessary since both spoken and written languages are systematic communication systems. However, it must be stressed that spoken language, although obeying some of the rules of written language, is not the same. Hunt (1989) investigates this notion, and identifies a number of differences:

- Spoken language relies on gesture and intonation to carry meaning. This means utterances, whilst being easily understood, are difficult to translate into written form;
- Spoken language contains many colloquialisms and is therefore more culture-dependent.

To emphasise the many different forms of writing systems, Figure 3.1 shows a taxonomy based on Sampson's work. In this taxonomy the important distinction is between pictographic (concrete representations), and glottographic (abstract representation) systems. Here, is where the relationship between spoken and written language becomes clear. Pictographs bear a physical resemblance to what they represent, hence their concrete nature. Whereas glottographic systems represent (*graphic*), the sounds (*glotto*) of the words of the equivalent spoken language. The second split between abstract-phonetic and pictorial-phonetic is due to what Sampson highlights as the 'double articulation' of glottographic systems. This distinguishes between the ideas[2] of words that can be broken up and represented by pictures (the first articulation), and the sounds of words which can be broken up and represented by phonemes (second articulation). The relationship between pictorial phonetic systems and pictographic systems (shown by the dotted line in Figure 3.1) is based on the first articulation. Generally, however, pictographic systems represent discrete entities/concepts (such systems may be called *semasiographic*, making only meaning visible) whereas pictorial phonetic systems represent syntactically correct phrases (making meaning *and* abstract language structure visible), making the two systems separable. In fact for this reason, Sampson even suggests that pictograms should not even be considered as writing systems. Sampson states:

[2] Morphology refers to the decomposition of words into smaller units. The presence or absence of these components can alter the meaning of a word, e.g. *-ed*, will place an action in the past.

"..the units of the first articulation of language tend to be relatively apparent to native speakers of the language...while the units of the phonological second articulation, particularly phonological units smaller than symbols, are not obvious.", (Sampson, 1985, pp. 36).

An example of the second articulation is the *abstract phonetic* system of the international phonetic association where individual symbols represent segments of sound, e.g. 'The cat' would be represented as:

ðə kæt

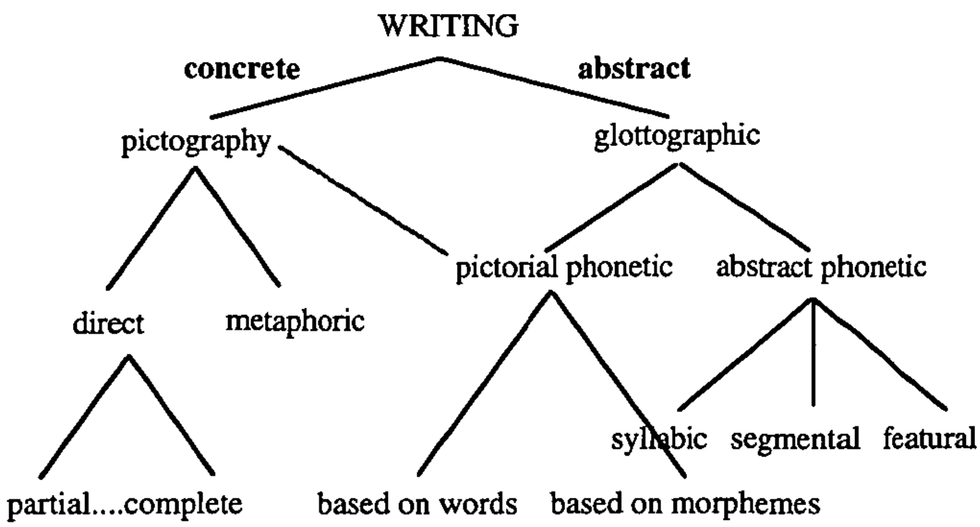
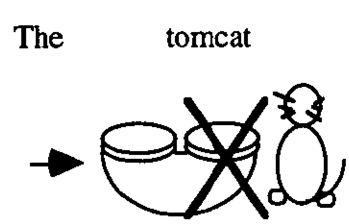


Figure 3.1: Forms of Writing, after Sampson (1985), pp. 32

Sampson also decribes how written form can also be categorised as either *motivated* or *arbitrary*. The former stresses the connection between the token or *graph* and what it represents. Thus, the forms in pictorial language are *motivated* because they need to have a recognisable, physical similarity with what they represent. This is also true in pictorial phonetic systems, where pictures similarly represent the morphemes or words of a language. For example, the pictorial phonetic-morphological *graph*,



relies on the graph for 'cat' being recognised as a cat in the represented world by its physical traits, e.g. tail, whiskers, etc., the arrow being recognised for 'the', and the 'tom-tom' drum (plus cross) being recognised for 'a single tom'.

The opposite of *motivated* form is *arbitrary* form. In this case, the graphs bear no relation to what they represent. This is true of all modern writing systems, with the exception of *imitatives* and *onomatopoeia* [3].

3.2.2. A History Lesson

Writing systems, by their intransient nature, provide evidence of ancient methods of conveying meaning. It is assumed that they developed after spoken language and that initially there was little connection between the surface representation of either system, i.e. the sounds of any spoken language bore no resemblance to the visual stimulus of the written language. At this stage, we distinguish between language, i.e. a systematic written representation, and the representation of concepts directly using physical objects. Although both carry information and can constitute a systematic language, we must assume that in the human computer interface, representations will always be synthesised or mediated in some way. Due to this, consideration is only given to the use of mediating physical form, rather than the physical object being the message. The same distinction is made by Pierce (Tran.[4]. 1994) who distinguishes between 'natural' and 'non-natural' representations. The 'natural' forms are physical manifestations which convey meaning, such as smoke meaning fire. The 'un-natural' are human-created, intermediaries, such as a flashing light meaning fire.

A highly condensed view of writing system development was shown in Figure 3.1. In general, writing systems have become more general purpose in their representation, moving from a physical resemblance of the represented concept (motivated), to a representation of the spoken word for the concept (unmotivated).

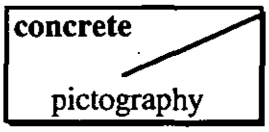
The following descriptions are referred to relevant parts of Figure 3.1.

---

[3] The exceptions are:
- onomatopoeia, where the word form reflects the sound *associated* with what is named, e.g. cuckoo.
- imitative language, where the word form copies non-verbal sounds, e.g. oomph, boom.

[4] New translation.

3.2.2.1. Pictographic Languages



The first representations were no more than sketches of what they represented; as the term 'depiction' suggests. The advantage of this system is its ease of comprehension, as long as the sketches are clear. The disadvantage is the difficulty of communicating abstract ideas, rather than physical objects. Thus, the pictograms are motivated *signs*, rather than arbitrary *symbols*; they have no other meaning beyond their physical form.

Gaur (1984) describes how American Indians recorded the history of their tribe in pictorial form (called *kekewin*) on long strips of wood called *Mide Scrolls.* The scrolls would depict important events in the tribe's life, for example Gaur describes:

"a drawing of a head and a body of a man covered with red spots records the fact that many people died of small pox; while three columns of ten parallel lines each drawn in black means that thirty dakota were killed in the course of a particular year.", (Gaur, 1984).

Another example of an ancient pictographic system is the Indus script of the Punjab region of India. These were quite stylised representations incorporating mystical symbols. The meaning of the symbols was not obviously related to their form and therefore required interpretation.

As well as depiction, the structuring of pictographic sequences also carried some meaning. Tverskey (1995) describes how these systems represented the notions of time and space (as shown in Figure 3.2). To convey temporal information or action, pictures were juxtaposed in a horizontal series, thus giving some form of systematic structural arrangement. However, unlike modern languages, the left-to-right inscription carries a clearly defined temporal meaning, whilst the left-to-right sequencing of text is merely a convention, and carries no additional meaning (as shown by Chinese and Arabic systems using different directions). The second notable aspect of these systems is that they use discrete entities to encode discrete concepts. These can be seen as precursors to words.

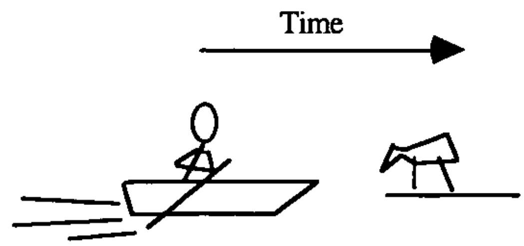
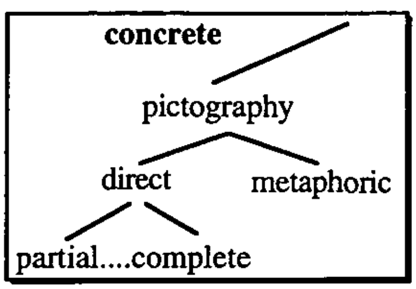


Figure 3.2: Pictography: "The hunter crossed the river to the Deer"

3.2.2.2. Metaphoric Languages



The next stage of writing development allowed the representation of concepts which had no obvious visual representative, or having some salient feature which precluded the need to draw the whole object.

In the first instance, a standardised vocabulary of pictures had to be developed within a writing community. These can be called *mnemonics*, since they require a translation from the *semi-arbitrary* graph into a concept. Examples of these are seen in the writings of North American Indians. Along side the pictorial writing already described (*kekewin*), there developed a second system called *kekinowin*. This was used by a select member of the tribe to record spells or incantations. In this case, the pictures represent concepts and ideas rather than physical objects, in a similar way to the mystical symbols described in the previous section. This method allowed the communication of concepts such as love, hate, war, and anger, through arbitrary symbols (see Figure 3.3). This is in contrast to the *signs* of pictorial language, since they convey something *beyond* their form. Metaphor is still very much a part of modern language for the same reason, to describe concepts which are difficult to convey literally.

The second development was the result of the need for faster writing. Less complex symbols used fewer marks but could represent concepts either by salient features (synecdoche) or in a simplified form (also shown in Figure 3.3). These methods became more widespread as writing systems developed within communities, allowing the meaning of symbols to become standardised.

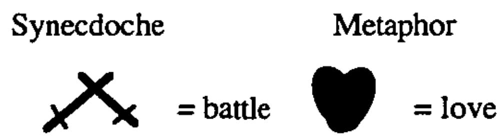


Figure 3.3: Synecdoche and Metaphor

The development of more complex structural conventions was restricted to the representation of numbers, where representations were placed in left-right, or up-down, order. In the Egyptian number system, position also represented value, the further to the left the symbol, the higher its value. Unlike modern number systems, the same symbol was not used in each of these positions. An example of the Egyptian number system is shown in Figure 3.4.

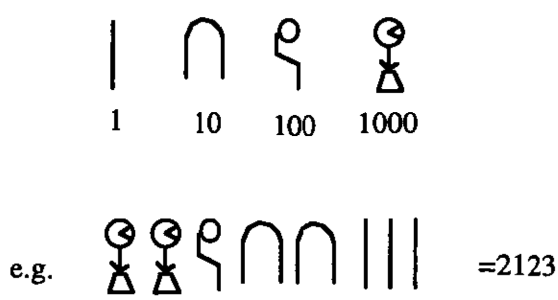
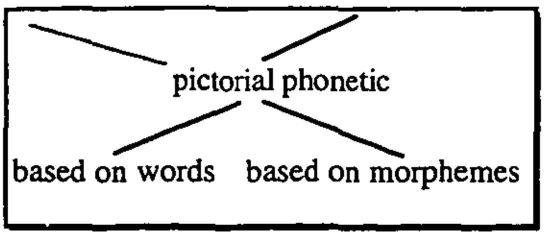


Figure 3.4: Egyptian Number System

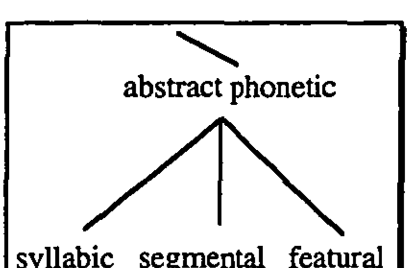
The form chosen for each symbol was not entirely arbitrary since it was related a physical object which was common in the domains where these systems were used. In this way, the number symbols provided information on the application domain of the symbols, e.g. 'heads of wheat' showed the use of number systems in the recording of crop yield.

3.2.2.3. Pictorial Phonetic Languages



The majority of pictorial phonetic systems devoted more of their resources to representing the words of spoken language, rather than the syllables. The oldest recorded Western language of Cuneiform, which originated in Sumeria around 3000BC, is an example of a pictorial phonetic language. This represented both thewords and morphemes of spoken language using pictures. It was also used in a similar way to pictography to represent ideas directly. Another example of pictorial phonetic systems is Egyptian hieroglyphs. In this system, the sounds of words were represented by picture meanings. This demonstrates the drawing together of spoken and written language, and overcomes the main drawback of pictorial systems, since it allows any spoken word to have a written equivalent (if an object existed to represent that sound). With the increase in the number of written records, the standardisation of symbols became widespread, particularly in the Egyptian nation. As testament to this, Egyptian symbols evolved through common usage into colloquial vocabularies called the *heitatic* and *demotic*. However, there was still no complex way of arranging symbols in a way which was distinct from the order in which they were spoken.

3.2.2.4. Abstract Phonetic Languages



The final stage in writing development was to expurgate the pictorial reliance of the pictorial phonetic systems, whilst retaining their connection with spoken language. This meant the use of *arbitrary*, rather than *motivated* form, and an increase in the structural complexity of the language beyond its spoken equivalent.

The answer lay in the develop of a *typed syntax* , where the meanings of symbols became dependent on their memberships of type sets, e.g. nouns, verbs, etc. These sets and the relationships between them were defined in an additional syntax. This gave the system a further layer of expression beyond its physical manifestation, making it as expressive as the number of type sets. Thus, pictorial symbols are replaced by arbitrary marks whose meaning must be learnt at three levels; a syntactic level where the *type* of the marks are described, a lexical level where the meaning of individual words are described, and a sentential level where word context affects meaning. The construction of groups of marks (words) into further meaningful sequences (paragraphs, etc.) is also dictated by syntactic rules. Clearly, the simplicity of learning and use has been sacrificed for expressiveness.

3.2.3. Discussion

This study of the development of written representations of language require the interface designer to consider a range of issues when choosing representation systems. These relate to the strengths and weaknesses of different types of representations that have been alluded to.

3.2.3.1. Abstract or Concrete

The main distinction that can be drawn between early writing and modern languages, is that early systems required the representation of concepts in a concrete way. In other words, every concept required its own pictorial representation. This made the systems limited in their expressive power, particularly when more abstract concepts had to be represented. Since modern systems are phonetic, they can represent any concept for which there is a spoken equivalent. The drawback is that their form does not bear any physical resemblance to the concepts they represent. This is not a problem if the number of words in the language is small, such as the computer language *Logo* (DiGiano et al., 1993), but larger vocabularies make modern languages difficult to interpret. This is compounded by complex and sometimes contextual syntactic conventions, which group the visual tokens in fixed ways. Finally, the close link between the arbitrary representations and the sounds of the spoken words (either in morphological or syllabic form), makes understanding difficult if the spoken equivalent is not available, e.g. through deafness.

3.2.3.2. Complexity and Expressive Power

The most important aspect of this development is the convergence of spoken and written forms and the consequent increase in the complexity of these systems. As complexity increased, it became increasingly difficult to become fluent in languages, without being formally taught. However, the range of ideas which could be represented in these systems was much more than their pictorial or abstract-pictorial counterparts.

3.2.3.3. Syntax and Lexicon

Early systems had no need for any rules to define how words were placed in sequence since only time was important. Also, the domains described were so specific that the vocabulary was not large enough to warrant *type* groupings. As written communication became more prevalent, there was a need to categorise words and provide rules of arranging words in sentences so as to increase expressiveness.

3.2.3.4. The Cognitive Impact of Language

Tverskey (1995) investigated the development of the linguistic ability of children. She concluded that the development of a child's language use is analogous to the historical development of language. First comes the use of pictorial representations, then the pictorial phonetic, and finally the abstract pictorial. Syntactic development is also similar, with the understanding of lexical and sentential rules developing with the use of abstract phonetic systems.

3.3. Representation in the User Interface

To apply this study of language to the problem of media allocation, it is necessary to provide common interface examples of the different types of languages that have been discussed. In general, modern interfaces rely on pictorial representation. Excluding 'help'-documentation, what little use there is of abstract systems is limited to single or small groups of words, e.g. error messages.

The highly pictorial nature of modern user interface representations is true both of visual representations, and experimental, aural representations. Gaver (1993) stresses the need for the use of what he calls 'everyday sounds' in the user interface. These are simulations of common, real-world sounds which can be associated with actions *on* *objects.* Gaver (1989) describes an Apple Macintosh interface which incorporates these sounds into its normal operation. A selection of these sounds are shown in Table 3.1. Gaver uses an *aural equivalent* of pictograms by making the *sound-form* of an action, on an object, directly represent the meaning of the action, e.g. selection example in Table 3.1. As with visual systems, when there is no aural equivalent of the meaning, aural metaphor is used, e.g. in the copying of data example. In this case, a metaphoric mapping is made between an action in the interfaceand an action in the world which bears some semantic similarity. The sound of this secondary action is then aurally depicted, e.g. a pouring sound.

Action on Object	Everyday Sound (Auditory Icon)
Selection	Sound source, depends on object (wood, metal, etc.)
Opening	'Whooshing' sound
Dragging	Scraping sound
Drop-in	Sound of object landing
Copying	Pouring sound

Table 3.1: Some of Gaver's 'Sonic-Finder' sound mappings (Gaver, 1989)

Generally, the reason for the prevalence of pictorial systems is that, unlike natural language, computer-based representations do not need to be general purpose. The only generality is provided by industry-standard usability guidelines and surface representations, e.g. OpenLook (Sun, 1990), Macintosh (Apple, 1992). However, the conceptual nature of interaction widgets and the choice of data representations are still specific to the requirements of the application domain.

The lack of generality required by domains is mainly due to the high profile of the functionality of interfaces and application domains. This is mainly the result of the tool paradigm, which portrays the computer as a tool, with certain functionalities. For example, the majority of applications provide a 'tool-bar' at the top of the screen:

This describes concrete functional concepts such as 'new document', and 'print document'. If applications are so concretely defined, then there is no need for the interface to include media which can represent abstract concepts such as love and hate.

3.3.1. Extending Pictorial Systems

In the defence of pictorial representations in user interfaces, some authors advocate that pictorial systems can provide some degree of complexity by using a simple superposition of pictures. Tortora (1990) and Horton (1996) both describe *pictorial languages* which allow a syntax to be defined for the structuring of pictures. Horton suggests the simple structuring of icons based on objects, e.g. *file*, and actions on objects, e.g. *open*. For example,

`Object=`**File** `Action=`**Open** *Combine* to give `Action on Object=`**Open File**

However, this still is an entirely pictographic system and thus is severely limited in what it can represent.

Given that concrete representations are prevalent in user interfaces, what advantages do the more primitive representations offer? The most obvious reason for using pictographic systems is the that they are easy to learn, given some knowledge of the world being represented. Secondly, simple syntactic conventions such as Horton's *combination operator* allow a consistency between icons which allows rapid learning, and generalisation of meaning to new icons. The key point is that interfaces meet specific needs, needs that were equally satisfied by early writing systems in that they were domain specific and easy to learn.

In summary, in historical terms the modern interface is comparable to languages at the early stages of writing development. It is ironic that these interfaces are considered modern.

3.4. Extending the Writing System Taxonomy

Whilst there is little use of abstract visual languages in user interfaces, there has been some research into the use of abstract sounds to convey meaning in the interface.

Work by Blattner (1993) and Brewster et al. (1993) attempt to use sequences of musical notes to represent actions. These *motifs* can be combined in a similar way to Horton's simple icon language to produce composite actions. Alty (1995) suggests that the use of rhythms, timbre (e.g. piano, marimba, drum) and melody, can also be used to carry information.

These musical representations use *arbitrary* rather than *motivated* form to convey information. However, they are neither pictorial phonetic since they do not use aural depiction, or abstract-phonetic, since they do not represent the phonetic components of words. They must be placed in a new category called *conceptual abstraction.* Since we have now considered aural depiction, we can generalise Sampson's writing taxonomy to all representations, both aural and visual. This is shown in Figure 3.5. For clarity, some of the media already mentioned is positioned within the taxonomy.

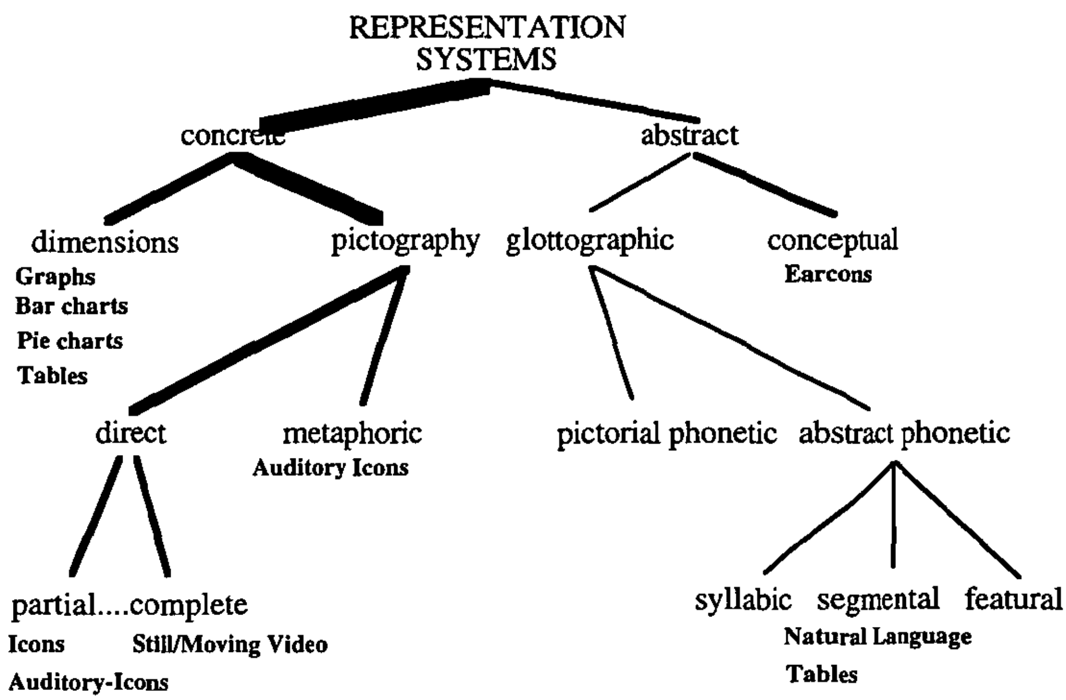


Figure 3.5: A General classification of representation systems

The taxonomy makes clear the wide variety of encoding mechanisms that have been discussed and which will reoccur throughout this thesis. The bold lines show where the majority of user interface media lie[5]. This makes clear the types of representations which have still not been addressed in modern user interfaces. It is important that these representations are given proper consideration by the interface designer since they provide poweful encoding mechanisms. The discussion of

[5] This will be returned to in Chapter 12.

encoding methods provides the first stage of a fuller consideration of *all* media with respect to their allocation in the interface.

3.5. The Importance of Expressiveness

The discussion of the various representation systems in this chapter show how the concreteness or similarity language sentences bare to the objects they represent determines the range of concepts they can convey, i.e. their abstractive ability. For example, pictographic writing systems have very little abstraction since they *must* bear a resemblance to the thing they convey. Thus, although they are easier to interpret, the amount of information they can convey is very limited. On the other hand a natural language statement is by nature abstractive, and consequently is able to convey an infinite variety of situations for which there is a spoken equivalent.

The abstractive abilities of media can now be related to the dimension identified in the previous chapter:

> ***expressiveness:*** *an emergent property of a medium which measures how much information it can carry*

Again, the notion of expressiveness has arisen as a product of the representational mechanisms of different media. This chapter has attempted to show this dependency more clearly. The next chapter addresses the use of interface media as problem solving aids. Of particular importance to effective problem solving is the provision of a sufficient amount of relevant domain information, as defined by a medium's *expressiveness.*

3.6. Summary

This chapter has shown the relationship between modern computer-based media and the development of non computer-based languages. This study of encoding mechanisms is central to the understanding of when particular media should be used in the interface, for the following reasons. Firstly, the representation provides a biased view of the world, enhancing some aspects of the world, and suppressing others. Proper consideration must be given to the selection of media to ensure those aspects of the domain which are necessary to tasks, are represented in an effective way. Secondly, it is essential to consider the effect different representations have on the user's cognitive representation of the domain, since this may determine the effectiveness of task performance.

The importance of media *expressiveness* was drawn out, as a property of the encoding mechanisms of the language, i.e. are they concrete or abstract.

Both the issue of task support provided by interface media, and their effect on the user's cognitive structures are addressed in vague terms here in order to highlight their importance. These issues will be further developed in the remainder of the thesis.

Chapter 4

Supporting Problem Solving with Interface Media

4.1. Introduction: A Positive View of HCI

Norman (1986) describes user interfaces as bridging the gulf of *evaluation* and *execution* (see Figure 4.1.1). Dix et al. also offer an addition to this (Figure 4.1.2). This view is somewhat negative in that it sees the computer interface as a hurdle to be cleared, rather than an empowering vehicle for new types of synergistic interaction. A similar view is described by Payne (1992), who regards the interface as the bridge to new activities. Lieberman (1996) also agrees, stating

"the goal of computing should be to enable collaborative problem-solving between people and machines", (Lieberman , 1996: pp 39).

Woods (1991) describes the interface in similar terms, as a problem solving aid. He argues that HCI should be fundamentally concerned with the design of these aids. Consequently, the interface designer should focus on the design of the representational form, rather than the visual form. He argues that the latter is concerned with superficial, ergonomic aspects of the interface, rather than deeper aspects of representation which predominantly effect the effectiveness of task performance. Woods' somewhat utilitarian view of HCI in general, and the user interface specifically is an interesting one, particularly in the light of the great deal of literature which will exists on problem solving being supported by suitable representations. This encompasses studies in the artificial intelligence community concerned with representation and reasoning and the visualisation literature which also stresses the importance of representation on problem solving.

Thus, the thesis derives its definition of HCI from both the view of Woods and the view of Payne, in the following form:

The design of artefacts to provide new human problem solving strengths in computer meditated domains.

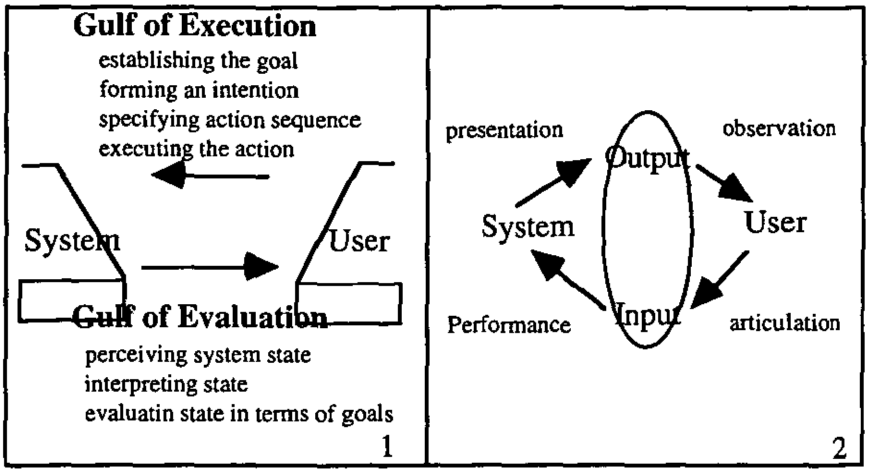


Figure 4.1: Views of HCI (1. Norman, 1986) and (2. Dix et al., 1993)

The representation of the artefact through the interface can then be considered as presenting the problem-solving space, and supporting navigation through this space in order to achieve tasks. In this way, the human and computer are yoked together in the common aim of problem solving. Clearly, the effect of the relationship between the representation of the problem and the user's cognitive apparatus is an important part of this. But how can new problem solving strengths be given to the user? To answer this question, human problem solving must be discussed with the emphasis on the effect *representation* has on the support of this activity.

4.1.1. Human Problem Solving

Dix et al. (1993) provide a broad discussion of human problem solving strategies. These are outlined as:

Behaviourist: skill-based, examining the problem from cues or stimuli in the environment and acting accordingly.

Gestalt: highlighting two aspects of problem solving.

1) Reproductive reasoning

2) Productive reasoning - using insight and restructuring of the problem information.

Problem Space Theory (Newell et al., 1958): uses a state space view of a domain. Problem solvers move from an initial state to a goal state, using *operators*. Heuristics may be used to select appropriate operators.

Use of Analogy (Gick and Holyoak, 1980; Gentner and Gentner, 1983): use of knowledge from structurally similar but semantically different domains where a solution is known.

Unfortunately, none of the literature which advocates these approaches addresses the effect external representation of the problem domain has on problem solving success. If support is to be provided for media allocation then a consideration of this is essential. Thus, it is necessary to consider other research domains where problem solving is paramount and is seen as a function of the problem representation. Though this discussion diverges form the user interface design literature, it does provide an important body of discussion which can be applied to user-interface design.

4.1.2. The Consideration of Domains

Before addressing the affect of representation on problem solving, it is necessary to discuss the notion of a domain. The common concept through the discussion has been the notion of a domain, some body of information which must be represented. A representation is chosen to allow the tasks within this domain to be made tractable.

However, a closer examination of computer-based application domains, as described in Chapter 3, is that there are two domains being represented. There are the widgets (interaction devices) in the interface which represent and control aspects of the *interface domain*, and the representations dedicated to the *task domain* itself. Relevant aspects of the less obvious former domain are:

- Providision of control over the interfaces view of the domain, e.g. scroll bars;
- They must adequately represent interface data, not domain data, e.g. a window icon.

This distinction between these two domains arises from the need to provide a common application functionality between domains. This allows third-party vendors to supply products which will all 'look-and-feel' the same. Both Apple (Apple, 1992) and SUN (Sun, 1990) identify this as an important quality of user interfaces, since it provides consistency between applications which leads to ease of learning and ease of use use.

Since both domains are mediated, two types of information must be conveyed:

- Representations of process and process state, e.g. a button for **printing** is depressed;
- Representation of data, e.g. 3D rendering.

As an example, Figure 4.2 shows a screen shot from Microsoft's Word application. The task domain contains:

```
Documents (text files + mark ups)
Processes carried out on documents
```

The interface contains:

```
Windows with associated functionality, e.g. scroll
bars
Desktop functionality
Other applications which may run concurrently
```

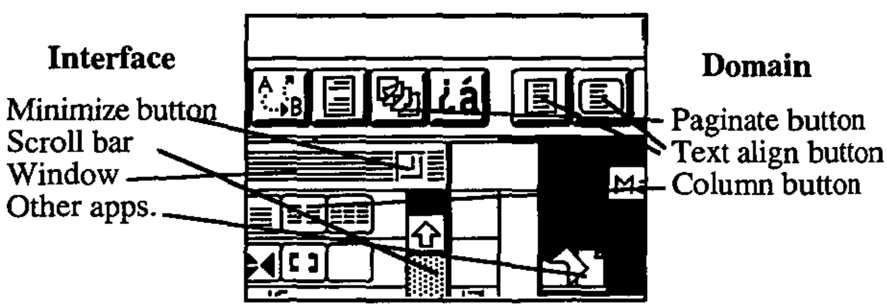


Figure 4.2: The Interface and task domain representations in Microsoft Word

This distinction is important since both interface and task domains require a decision to be made about their representation.

The domain concept itself requires further elucidation. Firstly, it must be realised that the domain is not only mediated by the interface; it is also mediated by the computer system itself (see Figure 4.3). Normally, an interface designer assumes the domain has already been represented in the computer system, but lessons can still be learnt form this initial conceptual encoding. In the first instance, the domain can be described as a collection of nebulous tasks, entities, and operators. The first translation formalises this description in terms of entities, relationships, and activities. The activities can be broken down into goals, tasks, and actions. Finally, the interface portrays this conceptual framework using static and dynamic representations and dialogue boxes. In doing so, some heuristics associated with the domain may be lost, perhaps those things which make the task easier to accomplish. Consequently, this representation may have an important effect on how tasks can be accomplished. This lesson must be learnt at the interface design level, since this interface representation may similarly restrict tasks.

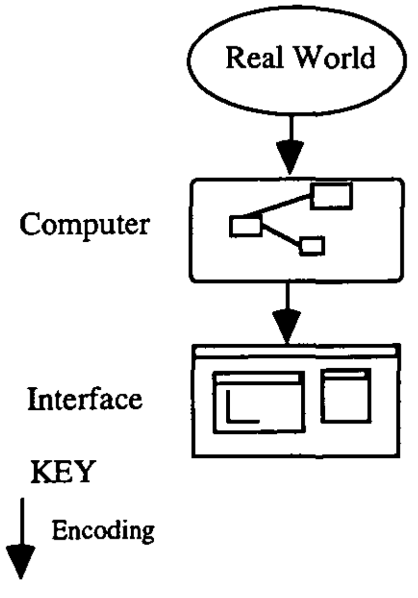


Figure 4.3: Domain representations

4.2. Problem Solving, Reasoning, and Representation

This section discusses literature where the effect of representation is seen as paramount to effective problem solving. The first examples come form the AI literature. These studies are concerned with the reasoning of computer systems over knowledge bases. The knowledge can either be domain specific, as in expert systems (Alty and Coombs, 1984), or general, as in humans (Newell, 1982). In both cases some representation must exist for the *knowledge-base*. As an example, Alty and Coombs state:

"..knowledge can usually be represented in terms of facts about the world (i.e. classifications) and relationships between objects, procedures or rules for manipulating facts, and information about how to apply these rules or procedures.", (Alty and Coombs, 1984: pp. 19)

This is one way in which knowledge can be organised, but it does not make the distinction clear between facts and knowledge. Facts can be declarative or procedural and are reasoned over to produce further facts. It is the act of reasoning over facts that makes the facts become knowledge. To clarify this, consider the set of declarative facts:

> The cat is black.
> The cat is lying down.
> The cat is purring.

All three facts express a *state* of the concept *cat.* Compare these with the following statements.

> The cat is happy because it is purring;
> The cat would be difficult to see in the dark because it is black;
> I will trip over the cat because it is lying down.

The difference is that the latter statement expresses beliefs which have been inferred from the facts by using other facts, e.g. *things on the floor cause tripping, dark objects reflect little light,* etc. After some time, the inferences made may themselves become facts, albeit of a procedural kind, e.g. if A then B. Thus, when facts can be reasoned over then they become knowledge. This point is stressed by Newell (1982) who argues that knowledge is the combination of *data* (facts) and *access* (reasoning) to support useful activity.

However, if HCI is to be seen as support of problem solving then the types of reasoning methods that are available must be considered. Classical logic is the basis for the majority of artificial intelligence work since it provides a wide variety of reasoning processes. For this discussion it is only necessary to describe these briefly.

- **Abduction**: Using facts from a semantically dissimilar but structurally similar domain;
- **Deduction:** taking initial premises and using axiomatic rules to provide a proof;
- **Induction**: nferring general concepts from initial premises.

These reasoning techniques are simply another way of describing problem solving over factual domains. In the case of expert systems, this problem solving will take place over a knowledge-base in computer memory which has been elicited from a human expert. In order to answer queries made over this knowledge base, the system engages in logical reasoning such as deduction and induction with the facts.

Given the *fact, knowledge* distinction, how are these facts represented, accessed and processed? To answer this question two aspects of the facts must be considered:

- How the facts are represented;
- How this representation allows ease of reasoning to solve problems.

Both points are addressed by Levesque (1986) who stresses the importance of representation on reasoning. He argues that the representation of facts decides whether the goal of a reasoning task is achievable or, to use the logic term, *tractable*. This is due to the number of cases in the knowledge-base which must be considered in order to give a valid solution. Some representations, Levesque argues, reduce the number of *cases* which need to be considered and make the reasoning task tractable[1].

In the field of artificial intelligence, the common representation of facts is first-order predicate logic. This system incorporates propositional descriptions of concepts, along with axioms which allow movement from one predicate form to another. In doing so, the logical equivalence of initial premises can be identified as well as the proof of inferences made over premises. It is the latter quality that can be called reasoning, since it takes facts (premises) and axioms and induces useful knowledge from them. An example is shown in the earlier cat description, "I know that I will trip over the cat because it is lying down.".

Given this discussion, it is now possible to make the novel analogy between the choice of representation for reasoning by machine, and the choice of representation in the human-computer interface for reasoning by users. The *knowledge engineer* must ensure facts are in an appropriate form for a computer to reason over. In the same way, the interface designer must provide interface representations that support the domain reasoning required by the user to perform the domain tasks. Of course, this assumes the domain is already described in some tangible way that is beyond the influence of the interface designer.

In making this connection, the body of literature on machine reasoning that has been alluded to can be used to provide a more fundamental description of why certain representations, should be used to convey a knowledge base, or in the user's case, a task domain. In essence, the AI literature argues for the choice of a representation

[1] This point will be returned to in Chapter 7.

that will ensure useful reasoning that is not outside the computational bounds of machines. Similarly, user interface designers must ensure that interface designers provide interfaces that allow tractable reasoning over application domains.

A second area of research where representation is seen as essential to problem solving is scientific visualisation. This will now be discussed in order to draw out the importance of representation on effective problem solving, and to see evidence of effective media allocation over different task domains.

4.3. Media in Visualisation Techniques

The use of visual images to represent information has been an essential part of all branches of science and engineering for many years. The reasons for this are manifold, but the essence was captured by Hamming, who stated:

"The purpose of computing is insight, not numbers", (Hamming)

In other words, data is useless unless something meaningful to a particular task can be extracted from it. This is analogous to the fact/knowledge distinction made in the previous section. Scientific visualisation provides visual representations which allow this to happen, particularly in the use of very large data sets. Like the artificial intelligence work described earlier, the main tenet is the choice of representation affects the effectiveness of reasoning. However, unlike the artificial intelligence literature, the majority of scientific visualisations rely on inferences which are not logically defined.

A lack of logical definition is due to the inherent structure of images which allow them to implicitly state relationships which non-pictorial, abstract descriptions would have to state explicitly. For example, a drawing of a ball on a plane can be interpreted as 'there exists a ball', but also carries the additional information :

- The ball's size;
- The ball's position;
- The ball's colour/texture.

Whereas, what would seem to be an equivalent logical definition, *ball(x)*, can *only* be interpreted as stating the balls existence, the remaining facts would need to be explicitly stated.

The complete definition of spatial relationships which is apparent in imagistic representation also allows a particular kind of reasoning to take place. This is called "perceptual inference" by Barwise and Etchmendy (1991) in order to capture the low-level, subconscious nature of the process. For example, a picture of three ball objects implicitly carries a *spatial transitivity inference* from A to C, which requires a four line proof in predicate calculus (as shown below).

ball(A), ball(B), ball(C)

left_of (A, B) (1)
left_of(B, C) (2)

Using transitivity axiom with
(1) and (2)
'A->B, B->C==A->C'

left_of (A, C) (3)

Thus, given the reasons for visual representations allowing more tractable problem solving have been described, it is no surprise that the visualisation literature sees visual representation as essential in problem solving. What has been learnt from common visualisation representation is now investigated with particular emphasis on how they support problem solving.

4.3.1. Maps

The use of maps became popular in the 18th century, and there soon developed a complex nomenclature. Due to the variety of application domains, each had its own motivations and referents and the advent of a *standard* nomenclature was slow in coming.

In detail, maps rely on a degree of spatial similarity with the represented domain. However, due to limitations on space, a degree of *symbolisation* is needed. The fundamental similarity with the domain is the laying out of objects in *Euclidean* X/Y space. However, the mapping process is not as simple as copying the mapped domain. A translation is required from a 3D world to a 2D planar map which entails some distortion of the 3D space. A common example is the variety of projections which are used for world maps.

The move from 3D to 2D is the first level of the map's abstraction away from the domain. Following this, the density of objects which must be mapped requires a further level of encoding. As Thorndyke and Goldin (1983) pointed out this encoding also varies in its abstractive level from the domain, being:

- formalistic: e.g. forest=green tree symbols;
- abstract: e.g. city = black circle;
- formalistic/abstract mix: e.g. *large* city=*large* black circle.

The use of maps in problem solving has a long history. A classic example was cited by Earnshaw and Wiseman (1993, pp. 9). A Victorian doctor, whilst investigating a cholera outbreak, had the position of the infected houses plotted onto a map of London (near Piccadilly (Figure 4.4)). By doing so, the lack of cases in the area of a workhouse that had its own water supply was noticed. This caused further investigation of the local drinking water which showed that all the other cases had drank from an infected street pump (indicated by the circle in Figure 4.4).

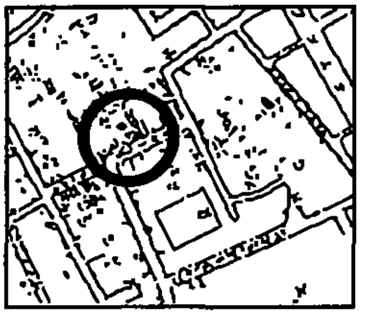


Figure 4.4: Dr Snow's problem solving with maps (c. Gilbert, 1958)

The generation, manipulation, and investigation of maps has grown into a research field in its own right, that of Geographic Information Systems (GIS). The primary motivation of this field is to allow geographic information to be more readily used in cartographic tasks.

In summary, geographic representations need to provide data in a form which makes the cartographic tasks easier. In more general terms, the representation is chosen to make the problem solving (reasoning) over the domain more tractable.

4.3.2. Graphs/Tables

Graphs and charts are a popular representation in numerical task domains due to their expected immediacy and simplicity. Even before the advent of CRT screens, graphs were being used to present statistics to the public. A notable example was described by Vardon (1957):

"An example of how people may be 'bamboozled' by a graph was given in 'the battle of the graphs' between Mr. Eden and Mr. Mayhew at the recent General Election. Mr Mayhew complained that 'a spectacular, terrifying rise' in the cost of living was exhibited by extending the ordinate and compressing the abscissa of the graph. ", (Vardon, 1957)

Generally, the nomenclature of graphs is limited to points and lines. However, Bertin (1983) and Tufte (1983) propose a wide variety of additional encoding such as circles to allow the encoding of a Z-dimension.

The choice between graphs and tabular representation became a point of much dispute during the late nineteen-seventies and early eighties. The advent of affordable information systems allowed managers to incorporate decision support systems into their office hardware. Research centred on the support of management decision making (problem solving) in deciding which representation was best for the decision-based tasks (Dickson et al., 1986; Remus, 1984). The conclusion was that although the right representation depended on a number of factors , the task was most important. What was required was a representation which was most suitable to the requirements of the task, allowing it to be achieved quickly and effectively. For example, an experiment by Jarvenpaa (1989) identified two task types, *acquisition* and *evaluation*; and two display types, organised by *attribute* and by *alternatives*. His results showed that succes in the *acquisition* tasks did not depend on either representation, whilst *evaluation* tasks were performed more successfully with the *attribute* representation. Jarvenpaa called this task-representation match ,*congruence*.

Thus, at the heart of the graph vs. table literature is the importance of matching the representation to the requirements of the task. Ideally this choice would allow difficult tasks to become more intuitive, thus demonstrating the importance of representation to problem solving.

4.3.3. Two-Dimensional Animation

Two dimensional animation is used to portray temporal domains and flow patterns. However, its use in real-time (rather than recorded) form has only recently become widespread. Both Barwise and Echtmendy (1991) and Stenning and Oberlander (1995) identify animation as an important tool in the solution of syllogistic logic problems since it allow subjects to see the transitions from one logic *model* to another, e.g. from ***A and not B*** to ***A and B***, by animating a Venn Diagram-like representation. Practical investigations of animation by Holan et al. (1984) and Bergan (1995) in the representation of complex and dynamic domains (process control plants), show animation to be useful for transferring knowledge of transitive domain states, and for highlighting patterns (loops, series) in domain behaviour. The use of animation in computer language execution visualisation (Cox and Roman, 1992; Douglas et al., 1995) allows similarly dynamic domains to be displayed. Problem solving in this field is related either to understanding the programme's conceptual algorithm or to error checking. Finally, a very recent work examines the representations of work flow between client and service (Buckhart, and Fucso, 1996).

The research of two-dimensional animation visualisations has provided a number of signposts for the use of this medium. Firstly, if the domain is dynamic or transient, animation may be useful in increasing comprehension. Secondly, if the domain is static, but has a large number of possible states (such as syllogisms), then animation is ideal. Finally, the encoding of each frame of the animation still requires investigation, as does the portrayal of transitions between states. Thus, animation is seen as an important problem solving tool in highly temporal tasks.

4.3.4. Digitised Images

The ease of storage of bit-mapped images provides an excellent resource for 'mug shot' analysis by the police and security forces and for cataloguing images. Also, the emergence of virtual museums on the world-wide-web is made possible by displaying digitised stills of exhibits and artefacts (Mannoni, 1996).

Here, the problem solving is of a less well-defined kind, but a general point can be drawn out. The analysis of mug-shots must be carried out at a facial level ,which necessitates the use of images. This would not be possible in any other medium, e.g. natural language. Thus, digitised images provide the correct kind of representation for the task. The same is true of the virtual exhibits which must fulfil the role of real-exhibits, and therefore must be inherently visual in nature.

4.3.5. Three-Dimensional Graphics

The use of filled and wire-frame graphics has allowed architects to test designs, military personnel to view targets, and biochemists to model molecular structures. Though less realistic than full-motion video, the representation does allow basic aspects of designs to be viewed and reasoned with. (See Figure 4.5). The image is chosen specifically to meet the needs of design tasks, such as the positioning of components or the planning of routes. Only by providing this highly realistic representation can these tasks be made possible. The alternative is the studying of tabular data which may prove intractable for large and complex images.

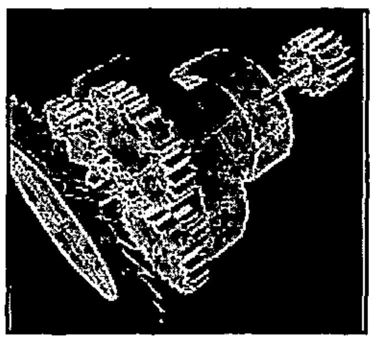


Figure 4.5: Three dimensional rendered image

4.3.6. Full-motion Digitised Images

The editing and production of film can now be entirely based on digitised images. This allows the introduction of virtual characters and scenery. Some of these techniques have filtered into user interface design. For example, Gibbs et al. (1993) incorporate digitised actors as virtual guides into applications such as museum databases. A second use is shown in process control environments. In a number of Japanese chemical plants, machinery may be monitored and controlled using superimposed controls on digital, real-time images of remote sites (Tani et al., 1992). An example is shown in Figure 4.6. This notion is defined as *telepresence* by Alty and Rijkaert (1995) allowing operators to feel they are 'with' the piece of equipment, even though they are geographically distant.

Educational applications are exemplified by teaching aids such as 'Cam-motion' (Bresnahem et al., 1994). This system allows students to digitally record natural phenomena such as balls bouncing or gymnasts tumbling. These recordings can then be analysed using superimposed mathematical tools, which allow quantitative studies to be made

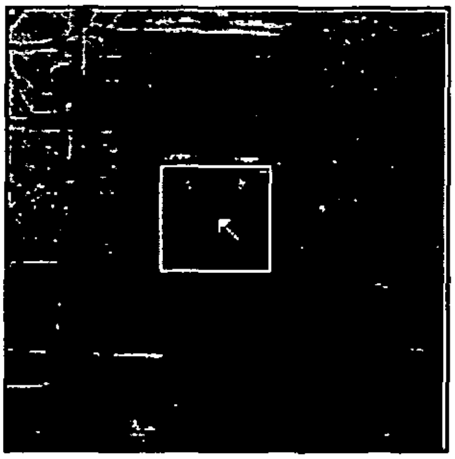


Figure 4.6: Example of *Telepresence* with remote machinery (Tani et al., 1992)

4.3.7. Three-Dimensional Animation

Real-time walk-through of buildings allow different furnishings and office equipment layouts to be tested. Also, the animation of landscapes using multiple satellite images is now possible (Keller and Keller, 1993; Earnshaw and Wiseman, 1994) and allows geological phenomena to be studied remotely. An example is shown in Figure 4.7. Clearly, these tasks are only made possible with a realistic representations.

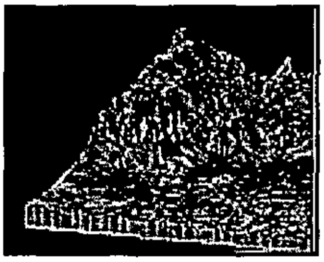


Figure 4.7: Frame from planetscape 'fly-through'

4.3.8. How Visualisation Techniques Must Develop

In a recent panel at ACM SIGCHI'96 (Gershon et al., 1996), key exponents of visualisation argued over the state of the art. Particular attention was paid to the visualisation of hyper-spaces, e.g. the world-wide-web. Their conclusions were:

- There is a need for a definition of task semantics in order to inform the choice of visualisation. For example, in the visualisation of hyperspace the focus should be on the meaning of information, rather than the *structure* of the network representation, i.e. visualising nodes and links.

- More systematic evaluation of different visualisations is required.

- Definitions of media strengths and weaknesses are needed, particularly for new media such as still and moving video

The visualisation literature is suffering from the same malady as the multimedia literature, namely the focus on new visualisation technologies and algorithms (Greenberg et al., 1996; Lamping and Rau, 1996) rather than justification for their use in terms of tasks. Secondly, the lack of formal descriptions of available representations is also in evidence. Finally, no consideration is given to role cognitive structures play in the task solutions, and how these will be affected by different representations.

4.4. The Importance of Abstraction in Problem Solving

The methods that have been discussed, all rely on an appropriate match between the representation and the task. This match must take place at a number of levels, which begin with the fundamental domain data-types that must be represented, through to a consideration of the task-relevant aspects of the data. Often, the latter case will override the former. An example of the this was shown in the use of digitised images, where the choice of an image rendering was matched to the realism required by the face-matching task. This is in spite of the fact that the domain information (e.g. pixel intensities) could have been equally well represented by another medium, e.g. a graph, but would have not allowed the goals to be achieved.

A key issue in this higher-level matching process is the notion of *abstraction*. This describes the amount of encapsulation of domain information into some higher form which captures commonalties between its constituents. An example is provided by Cox and Roman (1992) who are concerned with visualising the execution of computer programmes. The motivation behind this is stated as:

"With the advent of distribution and concurrence, there is a growing need to recognise and visualise events that are the result of independent sites....many investigators are turning to using program visualisation techniques to represent the data they are collecting.". (Cox and Roman, 1992: pp. 415).

Thus, the highly complex and distributed execution of program can only be understood by abstracting over different aspects of the program's behaviour and structure. To demonstrate this, Cox and Roman identify a range of levels of abstraction that visualisation algorithms can offer are described. These move with increasing abstraction from a direct representation of program elements to a complex narrative describing program execution.

- Direct Representation: map program elements directly to a image;
- Structural Representation: conceal/encapsulate program information;
- Synthesised Representation: display derived information which is not actually in the program;
- Analytical Representation: focus on correctness and completeness properties of the program;
- Explanatory Representation: an "aesthetically pleasing" narrative description of program behaviour.

Increasing Abstraction

Each of the levels is directed at a particular type of task, which requires a specific level of domain abstraction. More formally, if the domain is considered as a state-space through which the user navigates, then an increase in abstraction offers an increasing in the coalescence of these states. In cases where a large number of different states must be considered by a problem solver, this reducing of the state space is essential if tasks are to be effectively accomplished, i.e. the user has knowledge of every state.

Clearly, the dominant principle must be that the level of abstraction matches the requirements of the task. This is accomplished by considering a range of abstractions as in Cox and Roman's work, and matching the appropriate level to the task descriptions. To demonstrate the importance of this matching process in user interface design, a number of examples follow. These show the use of *higher* levels of abstraction in order to reduce the operational state space and support effective task performance. The abstraction is provided by varying means in these examples, including the external representation of the problem domain and a problem solver's mental representation. The consideration of lower levels of abstraction, i.e. direct representation, is prominent in the visualisation literature (Plaisant et al., 1996) and has already been addressed in the discussion of visualisation techniques in the previous section, e.g. maps, graphs, video.

4.4.1. The Importance of Higher Levels of Abstraction in Problem Solving Aids

To further demonstrate the importance of a higher level of abstraction in problem solving, two key domains are identified. What they have in common is that they require the problem solver to be aware of many states of a domain before a solution can be achieved. The first task is the solution of logic problems, the second is the control of complex processes. Both examples will be discussed in terms of a *state-space* view of problem solving, as described earlier.

4.4.1.1. A Graphical Abstraction of Syllogisms

The solution of syllogistic logic problems, using graphical rather than sentential representations, is addressed by Stenning (1991, 1995). He describes in detail the graphical representation and algorithm of 'Euler's Circles'. This algorithm aids the solution of problems, such as:

If all A are B and all B are C, what is a valid conclusion relating A and C?

Premise 1 (all A are B) *Premise 2* (all B are C)

These problems concern individuals, which are defined as taking on certain properties. Thus, within the premise statements (as above), the individuals are defined in groups or sets, each set being characterised by the properties of its members. For example, the premise, 'All A are B', describes a set of individuals who are both A and B. Of course, A and B are place-holders for any property such as 'tall' or 'green'. Thus, the premise 'All **A** are **B**' could be rewritten as "All individuals who are **tall** are **green**", or "All individuals who are **green** are **Martians**"

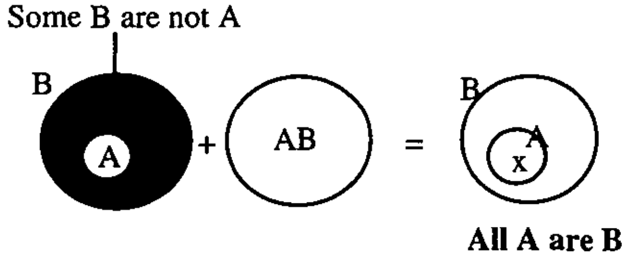
The solution of syllogisms is therefore a case of identifying (sets of) individuals which exist (are identified explicitly by each premise) and which satisfy both premises, e.g. those individuals which are **tall, green,** and **Martian.** A key consideration in this solution is the *explicitness* stipulation. This rules out those individuals which are only *implied* by a premise, e.g. 'All A are B' *explicitly* identifies individuals who are A and B, but also *implies* individuals who are B but are not A. To differentiate between explicit and implicit individuals, the term *maximal model* is used to refer to the statement of all possible individuals, and the term *minimal type* refers to the individual description which definitely exists. So in the previous example, the *maximal model* refers to All A are B **and** Some B are not A.

This dichotomy is essential to ensure the consequent existence of solution individuals is *explicit*, not just possible. Given this introduction, the algorithm can now be described.

Firstly, the algorithm deals with the number of premise combinations which must be considered to arrive at a solution. As has been shown already, the premise, 'All B are C', also *implies* the situations (domain states), 'Some B are C' and 'some C are not B'. Thus, these combinations include:

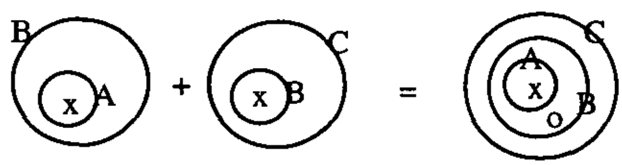
- Those situations (states) *identified* directly by the premise, e.g. 'All B are C'
- Those individuals/states implied by the premise, e.g. 'Some B are C'.

It is this distinction which is the heart of Euler's algorithm. To provide a complete discussion of solutions, it is necessary to consider the combinations of each of these states for the second premise, with each of similarly qualified states of the first premise. This is clearly too large a burden on the problem solver's working memory. What Euler's Circles do, is use a diagrammatic representation to *abstract* over the possible states for each premise by initially representing the *maximal models* for both premises. As described earlier, this portrays all of the other *implied* states, and signifies those domain types which must exist, as it is directly referred to by the premise, i.e. the *minimal* type. This minimal type is denoted by an 'x' marking. For example, in the diagram below (from Stenning, 1995) shows the maximal model and minimal type for the premise 'All A are B'. Clearly, this model captures all the implied states, i.e. 'some B are not A', 'All are B' (as shown below).



All A are B

The *abstraction* provided by the diagrammatic representation also allows the next stage of the algorithm, the combination, or *registration*, of *maximal* models. This involves the visual superposition of the pivot variable circle (the variable which appears in both premises) of the two premises, B in the example. This is followed by the alignment of the remaining circles. If any areas (individual descriptions) marked with 'x's are intersected by other areas without 'x's, then the 'x' is removed. This represents the change from individuals definitely existing, in either premise (as shown by 'x' in the premise diagram), to individuals only possibly existing in the solution (as shown by an empty area in the solution diagram). Any remaining areas after the combination of premises which are marked with an 'x', are termed *critical*, and it is from these that valid conclusions are drawn, i.e. conclusions based on the *definite* individuals which are identified by both premises. Thus in a second example from Stenning and Oberlander (1995), the *critical* area (marked with an 'x') shows those individuals which are described by '**All A are B and C**'. (As with Stenning's example, for clarity, the removal of *minimal type* ('x'), is shown by a 'o' symbol in the final diagram).



All A are B   All B are C   All A are C

In summary, the diagrammatic representation allowed :

- The representation of sets by analogy, i.e. an imaginary point inside or outside a circle is isomorphic to an individual's membership or non-membership of a set.
- The modelling of set types, i.e. maximal/minimal descriptions, and registration and expunging of minimal types. This is made possible by the isomorphism

In doing so, the representation abstracted across the premise domain, reducing the number of premise pair/states combinations that needed to be considered to form valid conclusions. More generally, the number of states that the problem solver was required to consider was reduced by representing the maximal model of both premises at the starting point of the solution. Implicit in these two diagram was the coalescing of the explicit and implied premises, which caused a reduction in the number of states which were then used in the combination stage of the algorithm. Thus, a high level of abstraction was the key to effective reasoning over the task domain.

4.4.1.2. Abstractions for Controlling Complex Processes

A large body of literature is devoted to the study of the effective control of complex domains (Sanderson et al., 1989; Verhage, 1989; Vicente, 1991). In these domains, the operator's tasks require monitoring and controlling of the system state, via.

multiple inputs and outputs to ensure it is within its optimal operational limits. Sanderson et al. (1989) provide a simplified description of this activity which is depicted in Figure 4.8. If one considers there may be over 4000 state variables, the difficulty of the operator's task becomes apparent. Clearly, some form of abstraction over the domain space is required.

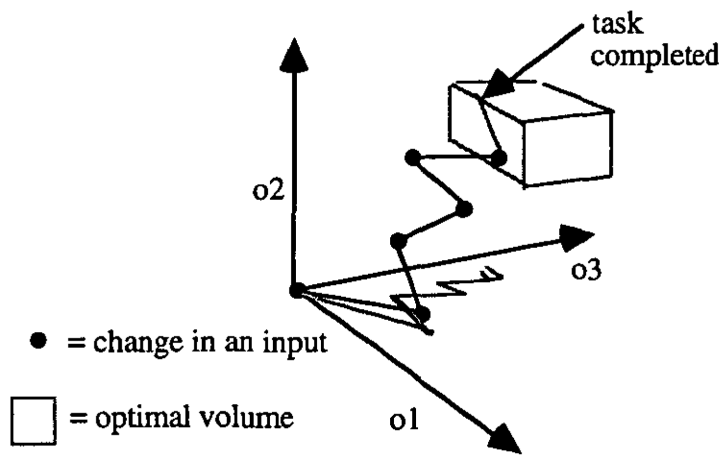


Figure 4.8: Sanderson et al.'s State Space Navigation (1989)

In agreement with this, Moray (1987) argues that during extensive experience with the system behaviour, operators decompose the prohibitively large domain state-space into small sub-systems, or *homomorphs*. These are self contained systems which have little effect on each other.

For example, a system has a behaviour defined by the variables a, b, c, d, and e. An inexperienced operator would treat the system as a five-dimensional state-space (Stage (1) of Figure 4.9). However, with increased experience, the operator notices that a and b are in a virtually independent relationship, A. In other words, two dimensions have been coalesced into one, resulting in the removal of a single dimension of the state-space (Stage (2) of Figure 4.9). With each discovery of independent systems (A, C, etc.) the state space is further reduced until it reaches manageable proportions, e.g. A, C, e. (Stage (3) of Figure 4.9).

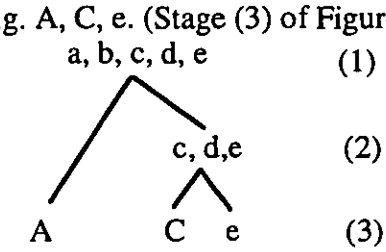


Figure 4.9: Decomposition of large state-spaces into *homomorphs*, after Moray (1987).

Unfortunately, this process may leave operators stranded, unable to return from their *part* description to the behaviour of the whole. To provide suitable displays for this behaviour, Moray suggests that control panels should be designed around these sub-systems, to allow their reconstitution into a less abstract system description.

In this example, operators have performed their own abstraction over domain behaviour, rather than relying on an abstractive representation like Euler's Circles (Stenning, 1991, 1995). However, this is only possible with extensive experience with the domain. Williams (1996), suggests that this disparity in performance between expert and novice could be reduced by letting the representation *do the abstraction for the operator*. A supportive result for this idea was provided by Grossen and Carnine (1992), who brought novice logic students up to the standard of experts by using the Euler's Circle representation described earlier.

4.4.2. Abstraction and Interface Representations

In general, these approaches identify the importance of providing the right level of abstraction over a problem domain. If the domain is seen as a state space through which the user navigates, abstraction represents the reduction of the state space to manageable proportions.

In the studies described a number of ways to achieve abstraction have been identified:

- *By post-processing of the domain data prior to representation.* This was shown in Cox and Roman's (1992) study, with encapsulated and derived domain variables, along with generalised non-domain narrative information. These abstracted forms can then be represented by the interface.

- *By using extensive experience with the domain to form a mental abstraction.* This was shown in Moray (1987), where, over time, operators performed mental coalescing of state variables. The interface representation should be designed to treat each stage of the decomposition as a domain concept, and represent it accordingly.

- *Using a representation which gives abstraction over a domain.* This is alluded to by Grossen and Carnine (1992), Stenning and Oberlander (1995), and Williams (1996). This ability to provide abstraction has already been identified in the thesis, as the *expressiveness* of a medium. The precise definition will be addressed in Part 2 of the thesis.

It is the third of the above solutions which is of most interest for the following reasons. Firstly, users often to not have the extensive experience with complex application domains which would allow them to form the mental abstractions suggested by Moray (1987). Secondly, the post-processing of domain information still requires the choice of representation to be made. Thus, by using the representational qualities of candidate interface media to provide abstraction, both of the above points are addressed. The choice of representation is driven by the level of abstraction required by the task, and the user must only come to terms with the encoding mechanisms of the medium, rather than the full state space of the domain.

The importance of providing the correct amount of abstraction of the task domain will be addressed again, in relation to the cognitive structures such representations may induce.

4.5. Cognition and Representation

The discussion so far has highlighted the following points:

- Users can be described as reasoning/problem solving over interfaces, as computers reason over knowledge bases;
- The representation of a domain must be congruent with tasks;
- A good match between representation and task will simplify the task solution;
- Thorough evaluation of representation between tasks is required.

Moreover, since it is assumed that the user holds some internal representation of the computer system's external representation, then both these representations must be considered along the the conceptual description of the domain. To clarify this, Figure 4.10 describes the range of domain representations which are coexistent in a period of human-computer interaction. It should be noted that these representations do not cover the real-world manifestation of the domain (there does not necessarily have to be one, e.g. 'Space Invaders'), since this is assumed to be already conceptualised in the computer.

The role of the user's cognitive representation of the domain in problem solving can now be seen. This is defined as:

The user's internalisation of the interface's externalisation of the domain.

Thus, the user engages in *explicit problem solving* with the interface, but the intended interaction with the domain is *mediated.* Moreover, it is this mediation which determines the ease of the user's problem solving, as a function of the user's conception of the domain as induced by the interface.

Consequently, it is essential that interface representations are chosen which induce the necessary cognitive representations to solve problems in the task domain. This is a user-centred way of describing the equivalent conclusions drawn from the earlier description of the visualisation and artificial intelligence literature, i.e. rthe right epresentation supports problem solving. By advocating this view of interface design, the thesis is taking a novel approach which will allow the support of problem solving to be couched in terms of mental representations .

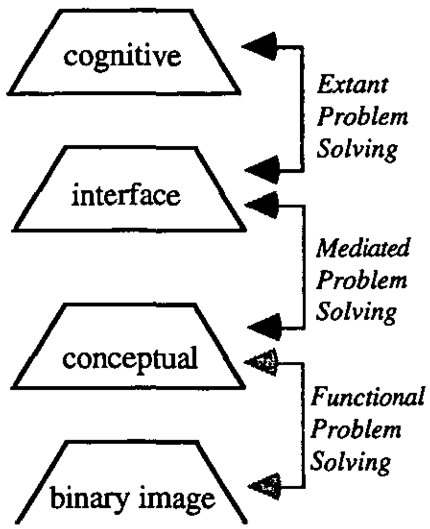


Figure 4.10: Explicit, mediated, and functional problem solving

4.6. Summary

The design of the user interface is described as the augmenting of the problem solving capabilities of users, in particular task domains. This novel view provides an opportunity for drawing on the literature which is concerned with problem solving (literature which is not normally associated with HCI research). The field of artificial intelligence offers important evidence of the effect of representation on reasoning, whilst the scientific visualisation literature demonstrates the utility of a wide variety of visual media in problem solving.

The notion that representation affects reasoning, leads to a consideration of the domains that are represented. In a user interface, it is noted that there are in fact two domains; the interface and the task. These domains can be viewed at a variety of levels, from the conceptual level through to the implementational level. There are also multiple representations extant at any given time.

A key aspect of successful problem solving was shown to be representing the domain at the *correct level of abstraction* that tasks required, e.g. direct domain representation vs. encapsulated domain representation. Since the majority of visualisation literature shows the utility of direct representation, the discussion focused on the importance of higher levels of abstraction. A number of ways to achieve this were described:

- *By post-processing of the domain data prior to representation.*;
- *By using extensive experience with the domain to form a mental abstraction*;
- *By using a representation which gives abstraction over a domain.*

The last case was seen as the best way to provide different levels of abstraction, since the match with the abstraction required by the task can be used to drive the allocation of media.

In terms of the user, the most important representation of the task domain is their own internalisation (the user's model). Consequently, the notion of effective reasoning being contingent on representation implies that the user's internalisation will be contingent on the interface representation. Thus, any bad design in the interface, i.e. the chosen representations hinder the required reasoning, will result in a similar deficiency in the user's model. Thus, interface design must focus on the cognitive effect of different representational media. This has not been considered in detail in the HCI literature, but it is clearly an essential consideration for any successful allocation of media in the user interface. The next chapter provides an introduction to this discussion, by describing the relevant mental models literature.

Chapter 5

Introduction to Mental Model Theory

5.1. Introduction: Mental Models and Expressiveness

Thus far, two of the three aspects that are crucial to a thorough study of human computer interaction have been identified, representation and reasoning. The third aspect is the user, the *interpreter* of the interface and *active* participant in the domain. If a cognitive-centred theory is to be advocated, then the importance of the user's mental model must be related both to the interface, and the reasoning required to effectively solve tasks in the domain. It has been suggested that because the expressiveness of media determines what aspects of the domain are represented (and at what level of abstraction), then this will have an effect on the user's mental model of the domain, and the effectiveness of their subsequent performance. The most novel aspect of this discussion, as regards multimedia interface design, is the utility of studying the user's mental model. This will now be discussed.

5.1.1. Mental Models and Task Performance

The relationship between task performance and mental models is suggested by Norman (1983):

"In interacting with the environment, with others, and with artefacts of technology, people form internal mental models of themselves and the things with which they are interacting. These models provide predictive and explanatory powers for understanding the interaction.", (Norman, 1986: pp. 7, 4)

Norman identifies the user's model as instrumental in dealing with artefacts. By applying this argument to computer-based artefacts, the model of an artefact held by an inexperienced user would be induced by interactions with the artefact. Therefore the following aspects will influence the user's mental model of an artefact;

- The artefact manifestation in the interface, i.e. output media deployed and the data allocated to them;

- The interaction method used, e.g. command line, mouse WIMPS, voice operated;
- The physical components of the system hardware, e.g. size, shape.

Since we are concerned with output media, by taking Norman's view along the first of the above criteria we can speculate on the following. If the only objective measure of the correct mental model is task success, and this model depends to a large extent on the application manifestation (i.e. the interface), then by investigating the different models media induce, we can better understand how these support problem solving. This knowledge can then be used to provide cognitive-centred support for user interface design. Few in the HCI literature have made any reference to this. A notable exception is Mayes (1992b), who asks:

"Will the user's mental model be improved by a more realistic or dynamic representation", (Mayes, 1992b, pp. 19).

Mayes makes the connection between representation and mental model. Moreover, implicit in this question is the relationship between the user's mental model and task performance. Given this relationship, the mental model literature must be investigated fully to observe any findings directly or indirectly related to computer-based artefacts.

5.2. What is Meant by 'Mental Model'?

If one is to talk of mental models, then the meaning of this term must be made clear. The language in this branch of cognitive psychology belies a range of definitions which cover both the organisation and the contents of mental structures.

Firstly, the fundamental implementation of the model is controversial. A distinction between imagistic and propositional representations has permeated the literature on mental representations since the Nineteen-sixties. It is not necessary to debate this issue here, since this interest lies in content, not implementation. Thus, what is important is the content and organisation of the model and how this is a function of an external representational form.

Secondly, the proposed location of the model in human memory varies between theories. Generally, this is a result of the type of behaviour that is being modelled. For example, Bainbridge (1992) argues that mental models are goal-structures which are held in long-term memory, along with situation-knowledge of how to apply them to a system when it is in different states. This study is characterised by the *vigilance* nature of tasks, with users having long periods of inaction to form models. This type of behaviour is synonymous with the depth of mental encoding associated with long term exposure (Craik and Tulving, 1975). Conversely, studies by Johnson-Laird (1983) and Bransford and Johnson (1972) have focused on short-term interpretation of sentences, without consideration of long-term learning. Consequently, their studies describe mental models as transient residents of episodic memory. Clearly, any study of mental models must make its intentions clear. This implicitly imposes caveats on the generality of the claims made about these models.

To begin with, it is necessary to highlight the common definitions that exist in the literature and draw out what is relevant to this study. For clarity, the definitions are divided between their proposed propositional or imagistic implementations.

5.2.1. Propositional

The following model definitions assume that models are encoded using abstract propositions. For example, the position of two objects relative to one another might be encoded as 'rel_pos (A, B)'. Proponents of this view (Jones, 1970; Pylyshyn, 1973; Anderson and Bower, 1973) argue that this representation is economical, since it stores general concepts, not specific objects. For example, the proposition, 'fork (x)'. Will represent all forks, whereas a picture of a fork will be tied to one particular kind of fork.

- **Mental Model**

  - Virtual System(Caroll and Thomas, 1982; Borgman, 1986; Staggers and Norcio, 1993): A structure which encompasses the knowledge a user has about a domain;
  - Functional Model (Williams et al., 1983; Young, 1983): A description of the knowledge a user has about the physical behaviour of a device;
  - Spatial Model: (Clark, 1972; Mani and Johnson-Laird, 1982): A propositional description of spatial relationships between objects in a domain;

- **Analogical Model** (Gentner and Gentner, 1983; Holyoak, 1987): A structural description of one domain using knowledge of a structurally similar one;

• **Cognitive Goal Hierarchy** (Bainbridge, 1992): A hierarchy of goals which must be achieved in a domain. These are couched in knowledge of domain relationships.;

• **Declarative Structure:** (Brachman, 1979): A semantic network storing knowledge about objects and relationships between objects in the domain;

• **Procedural Structure:** (Card et al., 1983, Bainbridge, 1992): A task hierarchy defining goals, tasks, and actions, that must be carried out in the domain;

• **Translational Grammars:** (Chomsky, 1965; Jones, 1970): A propositional description of surface (syntactic) and deep (semantic) structure created during sentence interpretation;

• **Complex System Decomposition:** (Moray, 1987): A break down of complex system relationships into discrete subsystems. This process is the results of long-term exposure to the domain behaviour.

5.2.2. Imagistic

The notion of a 'minds eye' which moves over internal images as the real eye does external images is the central concept in this theory. Important studies by Paivio (1969, 1975, 1978) and Mezler and Shepard (1974), suggested that internal images can also be manipulated and translated:

• **Cognitive Map:** (Medyckyj-Scott and Blades, 1993): An imagistic description of routes and landmarks in a domain. This can be used for navigation or description;

• **Images:** (Paivio 1969, Mezler and Shepard, 1974): A direct, photograph-like, representation of objects/actions that can be translated and matched to other images.

5.2.3. Other Theories (including mix of Propositional/Imagistic)

Other theories of internal representation of mental models are either a mixture of the imagistic/propositional theories or address cognitive structures at a lower level.

• **Primitive Mental Structures:** (McGonigle and Chalmers, 1986): A basic mental structure for ordering and categorising domain concepts such as mental axes for ordering objects on one dimension, e.g. *spatial paralogics*;

• **Binding Structure:** (Stenning and Levy, 1988): An associative network mapping attributes to objects which must possess those attributes;

• **Discourse Model:** (Grosz, 1986; Burger and Marshall, 1993): A model of topics discourse segments, and a focus of discourse which is used in tracking conversational semantics. The focus of discourse contains concepts which have recently been mentioned, allowing the resolution of anaphora;

• **Distributed Declarative Structures:** (O'Malley and Draper, 1993): The knowledge shared between user and environment, i.e. users only hold meta-knowledge on where to find information in the environment.

5.2.4. A Consensus on a Mental Model Definition

All of the above work has the same notion of an accessible cognitive structure which is used to inform human activity with the outside world. Given this discussion, the term 'model' is seen as misleading, since it suggests a mental structure which is isomorphic with an artefact. In some cases this is clearly not true. For example, those models which define preference and goal structures may be conceptually removed from the artefact and can be applied to any interaction. To use the term 'model' implies a specific type of representation, which is closely related to the interaction in a particular domain.

Thus, the author takes a view of mental models similar to Williams et al. (1983). A mental model is defined as a conceptual description of a domain which is interacted with, and is split between declarative and procedural knowledge. This specific description of the user's mental model has the following advantages:

• It is a structure which is amenable to experimentation since declarative and procedural knowledge can be identified using different types of experimental data. Verbalisation transcriptions can provide knowledge of the conceptual understanding of a domain, and performance data can indicate the level of procedural knowledge.

• The functional view also comes with a clear vocabulary, which describes concepts, relationships, constraints, and procedures. This gives the description of models a well-defined framework and allows comparisons between different functional models to be made in terms of these characteristics. Green and Benyon (1992) also identified similar reasons for using an extension of entity-relationship notation to describe the way tasks are physically solved within artefacts. The ERMIA[1] notation was described as providing a common, understandable language between cognitive psychologists and other non-experts.

However, if this definition is to be useful, the next stage of the discussion must investigate the effect of this structure on the success of task activity, and the co-existent relationship with the interface manifestation.

Given this definition, the position of this model in the context of the user's problem solving behaviour with the interface must be described. Figure 5.1 shows this in the context of a problem solving domain. All interaction with the problem domain depends on this mental world. Domain complexity is an important factor since subjects will be required to develop expertise over a period of time. This will be reflected in a parallel development of their mental model. The mental models literature can now be examined to show the importance of an adequate mental model for effective problem solving.

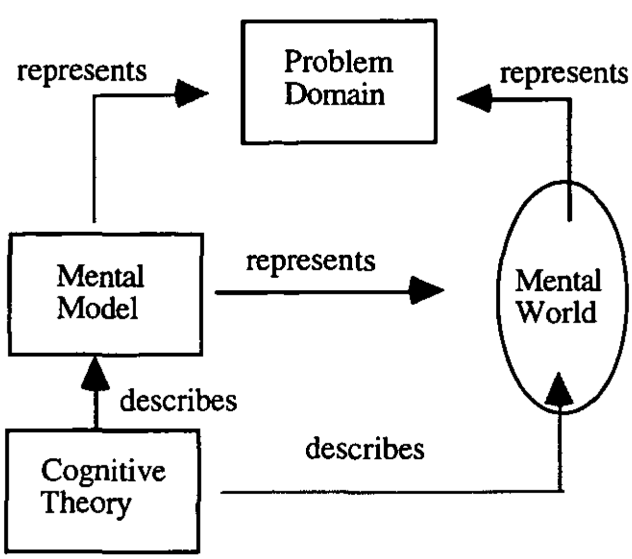


Figure 5.1: The Role of a mental model in problem solving (after Palmer, 1978)

[1] Enitiy Relationship Modelling of Interactive Artefacts

5.3. The Role of Mental Models in Problem Solving

As has been defined, user interface design is regarded as the design of problem-solving aids at the interface. Thus, the role of the user's functional mental model must be examined in this light. Firstly, whatever definition is chosen for a mental structure, the following principles are outlined in the literature with respect to problem solving:

- Models occur spontaneously during interaction (Norman, 1986);

- Different users can have a different model of the same artefact, whilst still both engaging in successful interaction. (e.g. Payne's (1992) study of ATM[2] interaction);

- Mental structures are dynamic (with the possible exception of *primitives* (McGonigle and Chalmers, 1986)) and develop with increased exposure to a domain. A corollary of this is the difference between novice and expert mental structures of the same domain. Rasmussen (1983) and Moray & Pajak (1986) suggests experts pursue robust strategies due to their highly generalised knowledge of the domain. Novices on the other hand work in a lock-step fashion responding to each system state independently with little or no overall strategy. Their structures tend be direct representations of particular 'system cue-operator response' mappings. Rasmussen calls this type of behaviour *rule-based, behaviour* the expert's is *knowledge based behaviour*;

- Mental Structures can be pragmatic or lazy, only encoding the minimum of information. They may also contain incorrect or superstitious knowledge (Norman, 1986);

- Different parts of structures may be inconsistent. (de Kleer and Brown identify this in discussion of naive models of physical phenomena, 1983).

These points make clear that mental models do not evolve in a systematic way. They may be incomplete and erroneous. However, user's still manage to perform adequately with them. The literature provides important examples of how such models have allowed effective problem solving.

---

2 Automatic Teller Machine.

de Kleer and Brown (1983) analysed the verbalisations of teenage physics students, who were explaining their knowledge of natural phenomena. The subjects were shown to have models which were limited in content, but enabled them to coherently explain behaviours which were consistent with the phenomena. In other words, subjects were applying their mental models to the problem of articulating knowledge of natural phenomena.

Greeno (1983) investigates the importance of domain representation on the effectiveness of problem solving. He defines domain *concepts* as:

"..the cognitive objects that the system (human) can reason about in a relatively direct way, and that are included continuously in the mental representation.", (Greeno, 1983: pp. 227, 3).

Greeno suggests these concepts need not be created when reasoning begins, but once created they remain in use. He calls the collection of all concepts which are use through a problem solution a *conceptual ontology*. He argues that given a static domain, the external representation may induce different conceptual entities of the domain in the problem solver's reasoning process, by representing the domain in different ways. Since, effective problem solving depends on the right conceptual entities, then the choice of representation is essential. Greeno uses the example of a geometry problem (see Figure 5.2 over).

As shown in the figure, the domain contains crossed lines and measurements of the angles between them (w, x, y, and z). The two solution representations draw out different parts of this domain. The sentential representation identifies the numerical relationship between angles, e.g. w+x=180 degrees. On the other hand, the diagrammatic solution emphasises the spatial relationships between angles, e.g. the straight lines and their angles are directly represented. Greeno suggests that the domain concepts present in the representation are the conceptual entities that will be used by the problem solver in their understanding of the domain, and any subsequent problems, e.g. prove x=y.

Greeno's work presents a cognitive theory which shows the importance of domain representations in problem solving. Though he does not mention the term 'mental model' explicitly, his notion of a *conceptual ontology* is very similar. The key is that he makes clear how the *ontology* determines the effectiveness of the reasoning that takes place using the domain concepts, and how this *ontology* is dependent on the representation.

Staggers and Norcio (1993), describe an experiment which investigates user's with naive and expert mental models. The study showed that users with more "developed mental models" were:

- Likely to make fewer errors;
- Quicker problem solvers;
- Aware of alternative problem solving strategies.

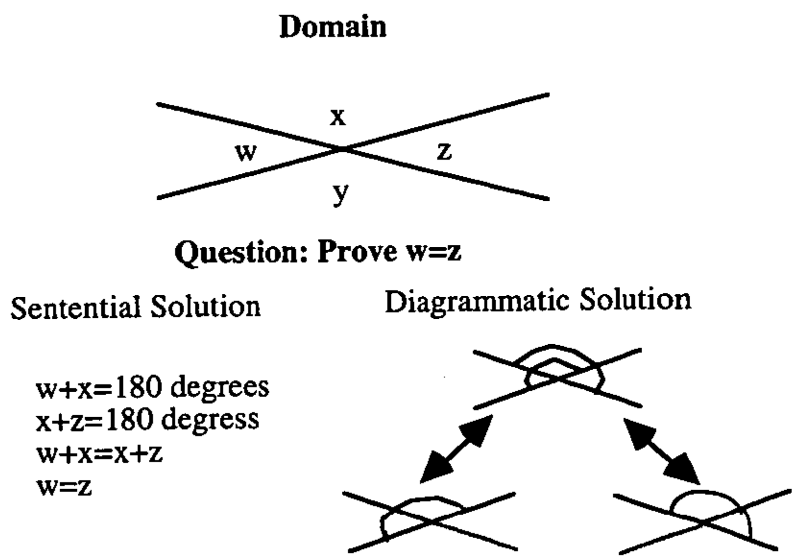


Figure 5.2: The importance of correct representation of geometry problems (Greeno, 1993)

A number of studies exist in the fields of cognitive engineering and information panel design which investigates the mental models experts use to control complex processes. For example, Ringelband et al. (1990), investigated the development of mental models of subjects using a complex computer simulation. Their experiment assumed that subjects developed a model which contained a simplified version of the system's structure and behaviour. Similarly simple models were programmed into robot operators, and the human and robot participants compared. The results showed comparable behaviour between the human and automated processes, suggesting subjects only require a simplified model to perform adequately in complex domains.

Like Ringelband, other authors (Bainbridge (1992); Moray (1987); Sanderson et al. (1989); Vicente (1991)) describe the mental models required for dealing with complex tasks. Their studies are directed at informing the design of control displays to meet the necessary information needs for these models. In this way, they are catering for those users who have already developed a model of a system, rather than those who are inexperienced. All these studies implicitly state the importance of the right mental model for successful task completion.

In summary, the evidence presented shows that mental structures direct effective problem solving. Thus, it is essential to have the necessary mental structure for a given domain. Moreover, this structure is a function of the chosen representation of the domain. The model can be a direct or generalised representation of the domain, but must match the requirements of task if performance is to be effective.

5.4. Summary

This chapter has identified the central position of the user's mental model in the successful solution of tasks. Secondly, the importance of the interface representation of the task domain in the construction of this model has been highlighted. This relationship between effective task solution, the interface representation of the domain and the user's mental model of the domain, means that the correct choice of interface media should induce the mental model necessary to effectively solve domain tasks.

Effective media allocation, therefore, can be described as choosing media based on a knowledge of mental models. In particular, the user's model of the domain which will allow the task to be most easily accomplished. Unfortunately, there are few studies which address this approach. Thus, to move towards the goal of effective media deployment, this thesis aims to identify the types of model that will result from the specific representational form of different output media, with a view to matching these to the task requirements.

To conclude part one of this thesis, the position of the user's mental model can be positioned within its interaction context (Figure 5.3). Moreover, at this stage of the discussion the following points must be made:

- The way media represent information dictates their effectiveness in supporting effective interface activity (problem solving);

- An important part of effective task activity is representing the task domain at a suitable level of abstraction;

• The user's mental model of a domain is instrumental in effective activity in that domain.

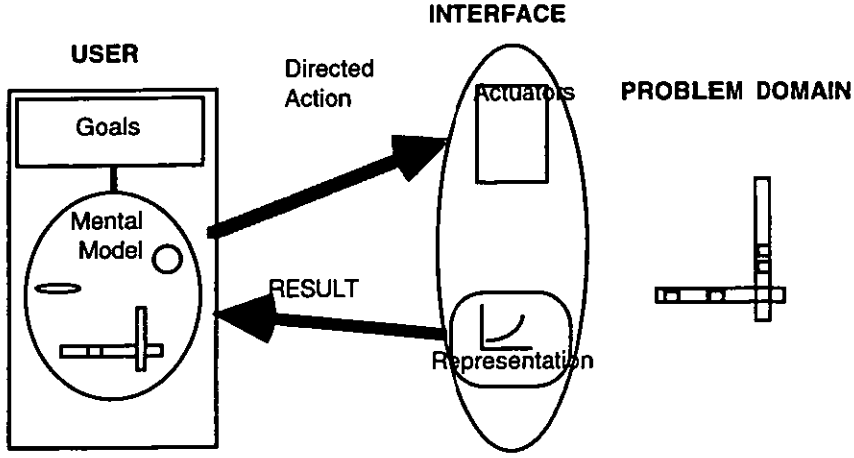


Figure 5.3: Mental models and problem solving

A unifying dimension is required to links these aspects together to describe and allocate media in terms of task, representations and mental models. The next part of the thesis will describe such an approach, based on the *expressiveness* of interface media and its relationship to the induced mental models of task domains.

Part 2

Mental Models and Representational Expressiveness

Chapter 6

The Representational Basis of Expressiveness

6.1. Introduction: Finding a Common Ground

The first part of the thesis described candidate methodologies for multimedia user interface design. Key terminological contentions were suggested and definitions for *medium, modality, multimedia interfaces and multimodal interfaces* were proposed. Unfortunately, the theoretical discussion of different interface media has three major drawbacks. Firstly, there is no uniform discussion which relates the representation, the task, and the user's cognitive capabilities. Secondly, it fails to address the wide variety of non-graphical media which may be used by the interface designer. Finally, it fails to make clear exactly how different media encode domain information.

The lack of proper consideration of the user's cognitive processes is seen as the major drawback in all these methods. Consequently, the notion of a user's mental model was introduced, along with its relationship to task performance. Thus, mental models of domains are defined as being dependent on the interface representation of domains, and concomitantly instrumental in the success task performance.

Given this introduction, the main challenge can be highlighted, how to subjugate all three parts of the interaction triptych, i.e. the task/domain, the representation, and the user's mental model, to the goal of media allocation. This problem can be reduced to two parts:

• A consideration of how different interface media encode domain information. This will allow the intrinsic representational strengths and weaknesses of different media to be made explicit. This is essential to the goal of effective media allocation.

• S discussion of the different types of mental model that will be induced by different media. This will allow media to be allocated based on the type of mental model they will induce, and the task support this model will give.

To recap, the goal of the thesis is to provide a unifying concept which relates these two parts under a common terminology. This concept has already been alluded to in Part 1 of the thesis, that of *expressiveness.*

6.2. Reasoning Over Representations

Consideration has already been given to how representations must support problem solving at the interface, which in the present context begs the question "how does the encoding mechanism effect the type of problem solving that it would most effectively support?". The AI literature that was highlighted in Chapter 4 allowed the following analogy to be made. The reasoning of users over interface representations is comparable to the artificial reasoning of knowledge-based systems over factual domains. This allows the literature of the latter to be brought to bear on the former. This discussion can now be continued within the context of a detailed discussion of representational form. Firstly, Newell (1982) states:

"It is clear to us all that representation...are the data structures that hold the problem and will be processed into the form that makes the solution available" (Newell, 1982: pp 88)

Implicit in this definition is the choice of a suitable encoding mechanism which will make the problem solution possible or easier. Barwise and Etchemendy (1991) and Stenning (1991) discuss the type of reasoning different representations can support. They argue for placing concrete[1] representations (diagrams) and abstract systems (predicate logic) on the same theoretical footing. This is done in order to emphasise the formal qualities of *both* systems in supporting reasoning. To emphasise the formal aspects of concrete representations in relation to abstract systems, Barwise and

[1] As described in the discussion of written languages in Chapter 3.

Etchemendy offer a range of dimensions over which to compare abstract and concrete encoding systems. These attempt to capture general aspects of the encoding mechanisms[2] which are present in some degree in either representation system:

- **Closure**: the number of implicit constraints imposed by the encoding form, e.g. for concrete systems two instances cannot occupy the same point in space. Concrete systems tend to have more implicit spatial constraints due to their physical nature. However, abstract systems, due to the complexity of their non-physical, abstract encoding mechanism, (more on this later) have a larger number of explicit syntactic constraints, e.g. *sentences must have verbs* , etc.

- **Conjunctivity**: the ability of an encoding form to convey more than one possibilitity. Concrete systems can only convey one possibility due to their physical nature, therefore are at a low level on this dimension. On the other hand, abstract systems do not have any such constraints and are therefore able to represent more than one state of affairs in an utterance;

- **Homomorphism**: the close physical relationship between problem domain and representation. Concrete systems tend to have a close physical relationship to what they represent, for abstract systems this relationship may be entirely arbitrary.

- **Perceptual Inference**: the degree to which an encoding systems supports spatial inference. For example, *symmetrical* reasoning - If the square is the right of the circle, then the circle is to the left of the square. This general property is not evident in natural language, due to its spatial encoding dimension having no analogue in the represented world. A possible exception may be the specific encoding of aspects of the represented world in the layout of text,

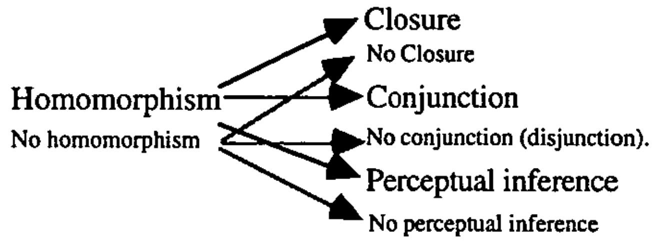
  e.g. It went to the right,
  .left the to went then and

  However, this type of additional encoding is unusual.

These four dimensions provide descriptive continua which can grade all representation systems in terms of the type of reasoning that they can support, e.g. simple logic operations such as symmetry and transitivity. However, the dimensions

[2] This discussion is equally applicable to aural representations.

are confusing since they are not orthogonal. A clearer way to represent them would be:



This makes clear that the basis of the type of reasoning different representations support is the *homomorphic* dimension. This determines the conjunction, perceptual inference, and closure dimensions. It also becomes clear that the dimensions are more concerned with concrete, rather than abstract systems, since it is a high degree of *homomorphism* which enables high ratings on the other dimensions.

What Barwise and Etchemendy do not make clear is that high ratings on these dimensions are not intrinsically ideal. These ratings must be assessed in terms of the effectiveness of the reasoning they support in a given task. Moreover, if such dimensions are to be defined they must be unbiased in their definition allowing all representations to be described, not just those at the high end of each dimension.

The use of a uniform continuum is essential if all media are to be compared and allocated in the task domain. Barwise and Etchemendy go some way to demonstrate this possibility but do not identify a suitably powerful dimension which allows the encoding mechanisms of candidate media to be described in equal detail.

The discussion not turns to a search for a similar dimension, without the drawbacks that have been identified. As mentioned at the end of Chapter 5, the *expressiveness* of media is seen as a candidate for this role.

6.3. A Linguistic View of Media

A number of authors have suggested that interface media can be described in the same terms as natural language (Mackinlay, 1986; Haberbeck, 1991; Alty, 1991). The reasons for this choice are generally based on the separation of concepts that these linguistic terms provides. Thus, a morphological/syntactical study provides a framework for discussing the form of media and how this form is legally constructed. The semantic/pragmatic study provides the discussion of how this form has meaning allocated to it. Already, in Chapter 3, the notions of form and encoding of meaning have been addressed, but this discussion was not within an apposite framework, relevant to media allocation. A consequence of this is that the property of *expressiveness* has remained vague. Thus, the following discussion will utilise the descriptions and evidence available in the linguistics literature in order to elucidate this property. A clear definition of *expressiveness* is essential if it is to be the unifying term for the proposed media allocation strategy.

6.3.1. Commonalties and Differences

Before findings from both the linguistics and the interface design literature can be investigated in order to illuminate the notion of expressiveness, a comparison must be made between the use of natural language and computer-based media.

6.3.1.1. An Open World vs. a Closed World

As stated at the beginning of Chapter 1, throughout the thesis we have seen interface media as 'windows' on application worlds. This metaphor is equally applicable to natural language since it too mediates the worlds that language users are describing. However, there is an important difference between these two cases, the *boundedness* of the world they describe.

Clearly, the world described by user interfaces is a highly *bounded* one which contains a finite number of objects which can behave in a finite number of ways (alone or in relation to each other)[3]. Natural language, on the other hand, is used to describe *boundless* worlds where any domain information can be used by the communicator as the content of their communication, or by the recipient in their interpretation. Thus, the fundamental issue is the effect this boundedness has on the interpretation[4] of representations of the domain, i.e. the interface media.

Evidence from the linguistics literature has attempted to define how language users are able to interpret what statements refer to, in unbounded worlds. The majority of this exists in the field of pragmatics which highlights the factors outside the communication which may affect its interpretation. Clearly, language users must

---

[3] This may not be the case for distributed domains formed by the connection of multiple machines. This paradigm may allow domains to expand indefinitely, thus allowing reference to be made over an effectively unbounded domain, as with natural language. Whilst this possibility is overlooked here, it is returned to again in conclusion of the thesis.

[4] At this stage of the discussion, interpretation can be described as knowing what the information carrying aspects of a medium are, and what they refer to in the domain.

employ a highly complex strategy in order to interpret statements in an open, *boundless* world. Conversely, the situation is much less complex in a *bounded* world. Here the number of concepts that can be referred to can be enumerated, reducing the possibilities that may resolve any ambiguous communication. To demonstrate the importance of this property in the discussion of computer application domains, it is useful to show the importance of boundedness in artificial language understanding research.

The complexity of natural language understanding in complex domains has been addressed by AI researchers. The objective of this research is to develop a machine which can understand natural language. The central problem in this research is the boundless of the referent domain. To counter this, artificial language systems have been applied to closed domains, e.g. 'block-world' (Winograd, 1972). Only by using such well bounded domains can anything approaching natural language interpretation be manifested. A similar application of using bounded domains, is the concept of *discourse structures* as a central part of the modelling of language discourse (Grotz, 1986; Burger and Marshall, 1993). In these dynamic structures, concepts that have been recently referred to[5] are held, to allow the resolution of future anaphora (e.g. this, that, it, etc.) which may occur within a statement or across the dialogue, and refer back to these concepts. Thus, the discourse structure is presented as a model of how such anaphors are resolved in human discourse, and attempts to approximate the possibilities of reference using a bounded structure.

The boundedness of computer domains means that a large part of the linguistics literature is insufficient for the purposes of investigating how media are interpreted.

6.3.1.2. Encoding and Interpretation

A second issue affecting the consideration of linguistics literature which relates to the boundedness of represented domains is the difference between *encoding* and *interpretation*. Given the fact that linguistics predominantly studies natural language, it is important that the notions of encoding and interpretation are not confused. The focus of Chapter 3 was to make apparent the many different encoding mechanisms that can be used in any representational system. Specifically, encoding is a physical trait of a representation which exists independently of the domain it describes. For example, the following encoding mechanisms were described in Chapter 3:

[5] Burger and Marshall's system, AIMI, resolved diectic anaphors, where objects were pointed to as part of natural language discourse.

Concrete:

- Pictorial;
  - Direct, e.g. visual icons;
  - Metaphoric, e.g. Gaver's (1989) auditory icons.
- Dimensional, e.g. graphs, pie charts.

Abstract:

- Conceptual, e.g. *earcons* (Blattner, 1991), sign-language, semantic networks;
- Pictorial phonetic, e.g. hieroglyphs;
- Abstract phonetic, e.g. natural language.

Later in this Chapter, the notion of encoding will be discussed in more detail with respect to *expressiveness*. For the moment, the importance rests on the difference between the mechanistic operation of encoding and the more nebulas notion of interpretation. The latter is considered as the action of the recipient in relating the physical form of the encoded message to the domain concept it describes. This process involves two stages:

- The understanding of the encoding mechanism;
- The interpretation of the domain concept.

Clearly, the boundedness of the domain addressed earlier will have little effect on the first case since the encoding mechanism is independent of the domain. However, boundedness may seriously affects the second for the reasons described in the previous section. Thus, the highly bounded nature of computer domains means that the predominant action in the user's understanding of the domain representation will be their dealing with the encoding mechanism. This consideration must be paramount when studying the linguistics literature since the reverse is true, with the majority of user's understanding being related to the complexity of the interpretation stage since they are assumed to already understand the physical encoding mechanisms of the language.

6.3.2. Introducing Terms: Morphology, Syntax, Semantics, and Pragmatics

Given the importance of considering that user interfaces are concerned with the representation of bounded domains, and the consequent focus on *encoding* rather than *interpretation*, the relevant linguistics literature can now be discussed.

6.3.2.1. A Language Definition

Tortora (1991) provides a simple description of a visual grammar which can be generalised to the discussion of any representation system. Each term of Tortora's definition will be discussed in further detail below.

A language (L) is defined by the triple:

```
L=L(ID, G0, B) where,

G0 = Underlying grammar (picture grammar in this case)
ID = Lexicon
B = Semantic rules
```

6.3.2.2. Generating Forms to Convey Meaning

6.3.2.2.1. Morphology and Lexicon (ID)

L=L(ID, G0, B) where, L = A language
ID = Lexicon
G0 = Syntax B = Semantic rules

The form that instances of a medium can take, which adequately support problem solving tasks, is described by the relation (1). This is the author's extension of Tortora's definition incorporating aspects of the discussion from Chapter 1.

ID=ID(F), where (0)
F=F(R, P, T, M) (1)

F = Form of a medium;
R = Presentation resources and characteristics of medium;
P = Perceptual characteristics of the viewer;
T = Expressiveness requirements of the task;
M= Set of morphemes and morphological rules to create lexemes.

Firstly, (0) states that the lexemes of a language are a product of its morphology, i.e. those physical constituents which make up its lexemes. For example, a functional graph language, L_g, may have a morphology which contains only geometric shapes. Rules within the morphology are then used to bring this constituents together to produce lexemes of L_g, e.g., axes, marks etc.

The display resources (and their characteristics) available (R) are an important constraint on a representation system's morphology. For example, a system with no bitmap facility will be unable to convey an icon based representation system. Moreover, the characteristics of the resources must also be considered. For example, a computer system with limited single tone sound capabilities will be unable to auralise two different warnings simultaneously.

The perceptual characteristics of the viewer (P) greatly effect the form of a representation system. This was demonstrated by Cleveland and McGill (1986), who identified dimensions of graphical representation systems which were best suited to human perceptual mechanisms. Their results showed the position, length, angle, and slope dimesions, allowed accurate encodings of quantitative data, i.e. subjects were able to distinguish more easily between consecutive representations of domain values. Conversely, dimensions which were less easy to interpret were shape, colour, density, area, and volume. Jones (1995) adds the consideration of cognitive bias towards imagistic or verbal processing to this discussion.

The final characteristic which affects form is the task. Casner (1991) describes a system which given a description of a task in terms of logical operations, automatically finds suitable graphical representations to encode the task data in a way which allows tasks to be carried out. Casner's system makes the task central to the choice of the basic components of the full representation, e.g. vertical and horizontal axes or lines of variable length, to allow domain values to be compared by analogous spatial distances.

Given the first three criteria, the morphology (M) which generates the lexemes of a language can now be related to the discussion of encoding and interpretation from the previous section. The question must be asked, how does F affect the encoding mechanism in user interface media? This is a different question to the more common study of how the encoding mechanism of a representation system constrains what it can encode. Clearly, the latter is dependent on the encoding mechanism, but does not address how this encoding arises from the constituents of the language.

6.3.2.2.1.1. The Perceivable Changes Criterion

The first encoding-related statement that can be made about the morphology M is its physicality, that is, its form must be perceivable. In this case, form describes all the physically perceivable aspects of the members of M, i.e. size, position, shape, etc. Thus, the encoding mechanism must be based on changes in this physical form representing changes in what is represented. A corollary of this is that for different encodings to represent different concepts, the changes between them must be perceivable. Symmetrically, if the referent[6] remains the same, the encoding mechanism should ensure no change in the physical form. In others words, morphologies used in encodings must not contain any ambiguities. This term will be called the perceivable differences criterion (PDC).

For example consider the language `L`, which has the following morphology and morphological rules (M):

> `M={A, B, AB, conc(A, *)}`
> where the morphological rule, function `conc(A, *)` concatenates A and *, where * is either A or B..

Here, *all* of the physical manifestations of L are entirely described in F. It is these which will constitute the language L's lexicon.

L is then used to encode the domain `D` which contains concepts, X, and Y.

There are a number of different ways the lexemes defined by `F` could be allocated to `X` and `Y`, but they must all obey the physicality of F which means they must all obey the PDC.

```
X->A                 (1)
Y->B                 (2)
X->AB                (3)
X and Y->conc(A, B)  (4)
```

(-> denotes an encoding)

[6] That which is referred to in the domain.

Of these four encodings of L, only (1) and (2) are valid, since they obey the PDC. Conversely, (3) and (4) violate the PDC since they use the same physical form (lexeme) to encode different domain concepts.

Whilst studies of this property in graphical representation systems (Bertin, 1983; Cleveland and McGill, 1986; Arens et al., 1993) have shown the importance of this notion in the encoding of quantitative data, its applicability to natural language may seem uncertain. Consider the lexemes (products of the morphology) 'bob' and 'man', which both refer to domain concept 'Bob'. Clearly, in this example, although the referent remains the same, i.e. *The man Bob,* there are perceptual changes in the lexemes. The question is how does this agree with the original definition that for the same referent, encodings must be the same?

The key is that the referent *has* actually changed since the natural language encoding mechanism is representing sounds, not domain concepts[7] directly. i.e. it is a glottographic system, as described in Chapter 3. However, in the further encoding of domain concepts the PDC need not apply, i.e. 'man' and 'Bob' can refer to the same concept. This is because natural language relies on non-physical, abstract mappings in its encoding system, and is therefore, at a domain referent level, not tied by the physically-based PDC criterion. This fundamental property will be addressed in the next section.

Thus, in abstract systems the importance of physical changes being perceivable is relevant to differences in sound, not necessarily in referent. Clearly, whether representing sounds or concepts, the notion of perceivable change is essential in any language system. Thus, the products of M, i.e. the language tokens or lexemes, must be encoded in such that any two lexemes representing different domain concepts must be differentiable.

6.3.2.2.1.2. Encoding Task Domains

Despite the bounded nature of computer application domains, there may still be a wide variety of concepts to be encoded. Thus, it follows from the previous discussion of perceivable differences, that the members of F will restrict the number of different physical forms that are *available* to encode information. For example, returning to the simple language, L, above:

---

[7] Sampson (1985) calls this the first articulation (concepts) as opposed to the second articulation (sounds).

L: M={A, B, AB, conc(A, *)}

Clearly, M can provide the distinct lexemes {A, B, AB, AA}, but not BA.

If a domain D, has six distinct concepts that required encoding, then the perceivable differences criterion (PDC) of the previous section, dictates that each concept should have a different encoding, say {A, B, AA, BB, AB, BA}. However, L does not provide this facility since BB and BA cannot be represented. Other authors have noted this constraint of lexicon on encoding. For example, Mackinlay (1986) describes this as a language being too *inexpressive* to allow the encoding of domain data.

Clearly, like the PDC, the encoding process has been again been restricted by the physicality of the morphology. However, as mentioned ,some encoding systems overcome this physicality by using a non-physical abstract encoding, e.g. natural language. This will be discussed in next section.

6.3.2.2.2. Syntax (GO)

L=L(ID, G0, B) where, L = A language ID = Lexicon
G0 = Syntax
B = Semantic rules

Syntax describes how groups of lexemes generated from a language's morphology are formally and consistently structured. In other words, the grammar of a language describes all those sentences which make up the language. The majority of concrete findings from linguistic research are in the area of syntax or grammar. This is for a number of reasons:

- A grammar is fixed across all domains which a language describes;
- A grammar can be described in terms of production rules;
- Simple grammars can be described in very few *production rules.*

Syntactic grammars have been proposed in the HCI literature for icons (Tortora, 1990; Horton, 1996), graphs and tables (Mackinlay, 1986), ancient writing systems (Kapolka, 1991) and interaction languages (Payne and Green, 1986). The reason for defining the grammars of these different languages is to provide consistency across representations. For example, Horton (1986) identifies why this is important in icon sets:

- It reduces the effort required to design, draw, and revise icons, since a set has a predefined collection of morphemes from which lexemes are created;
- It ensures and enforces fundamental consistency;
- It lets users, given a knowledge of the syntax, to predict what icons will look like;
- It allows the definition of icons to become extensible, making learning of new icons easier.

An example of a product of an icon syntax is shown in Figure 6.1.

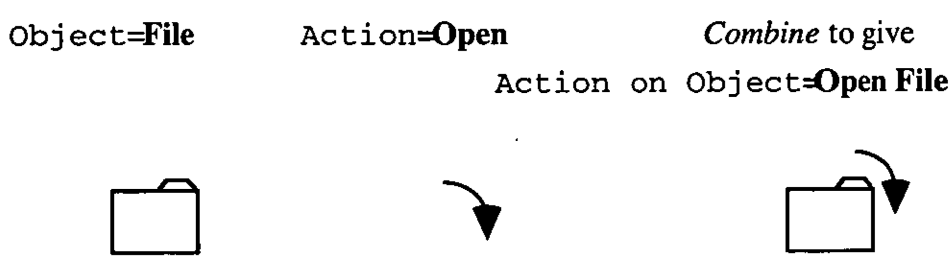


Figure 6.1: Combining Icon Lexemes using a Syntax

However, Horton does not address how the structure of the language affects how quickly it is learned. If subjects are to be able to generalise to new icons then they must first understand the language itself.

6.3.2.2.2.1. Structuring Lexemes

The syntax of a language (G0) brings the notion of structure to the language since it describes the language's well-formed statements. Thus, unlike the declarative nature of the lexicon, the syntax *actively* takes the members of the lexicon and lays them out in different, but *consistent* , ways. However, although the syntax does not directly[8] provide any mapping to a referent that may be encoded by these lexemes, it can be regarded as the first stage in the encoding process. Moreover, as with the morphology of a language, this syntax places a number of restrictions on how domain concepts are encoded.

---

[8] In natural language syntax, the position of a word may indirectly indicate some aspect of the domain referent, e.g. the position of a noun may indicate whether it is the subject or object in the domain.

Since we are concerned with bounded domains, an example of a syntax is taken from Mackinlay's (1986) study of graphical interface media. In this example, the syntax defines a horizontal axis language (HorzPos) which dictates the placing of discrete '+' tokens at a constant height above a horizontal axes, but anywhere along its length. This is shown below.

```
HorzPos(s)<=>                                   (1)
     s=h∪m ∧ <o,1> is a member of m=>[          (2)
     o='+' ∧                                    (3)
     Ymax (h) ≤ Ypos (1)=const ∧                (4)
     Xmin (h) ≤ Xpos (1) ≤ Xmax (h)]            (5)
```

Where <o, 1> defines a mark (o), at a horizontal position (l). m is the set of all lexemes, i.e. {+}, and h is an axis.
The functions Ymax, Xmax and Xmin, return the respective maximum or minimum Cartesian co-ordinate of their argument. Xpos and Ypos return the x or y co-ordinates of their argument.

From this semi-formal definition, a number of constraints on the legal utterances of this language can be enumerated:

- Lexemes must be located at a constant height above the axis, not above or below (defined in line 4 of the example);
- Lexemes must not be on either side of the axis (defined in line 5 of the example).

Mackinlay defines this syntax in predicate form. Thus, a legal instance of the HorzPos language can be generated or an example instance can be tested to show if it is well-formed. The definition makes clear the following issues about syntax:

- The syntax *generatively* defines well formed instances of a language. These instances take lexemes and arrange them in consistent ways;
- The syntax is entirely independent of any domain that may be encoded.

Mackinlay's grammar describes a system which places discrete lexemes in physical space (i.e. one-dimensional Cartesian space). Thus, the morphology of an utterance can be described as one or more'+'s, each with a varying horizontal displacement. Implicit within this description is that the PDC holds, since it is intended that a concept in the domain is encoded by a unique lexeme (located '+'). Thus, the syntax has compounded the constraints of the physical morphology, i.e. a single '+' can only have *one* horizontal co-ordinate.

A more usual definition of a syntax in linguistic literature is as follows (based on Tortora, 1991):

```
A syntax grammar GO for a simple numerical language can be
described as:

   GO=GO(P, NT, T, Ty)

   NT = a set of non-terminals, e.g. sum;
   T = a set of terminals or lexemes, e.g. {-, +, 0, 1};
   P = a set of production rules using the above;
     e.g. sum::sum+sum     (1)
          sum::sum-sum     (2)
          sum::{0, 1}           (3)
```

Thus, `production-rules` give options (shown by the disjunctive '|') for how the non-terminals are decomposed. In the example, the `non-terminal` `sum` can either be represented as further `sums`, or as a `terminal` (0 or 1).

As in Mackinlay's work, this definition leads to the generation or analysis of legal sentences. For example, a sentence of the language, '0+1-0' can be described in this syntax using a *parse tree*. (Shown in Figure 6.2 with the applicable production rules labelled)

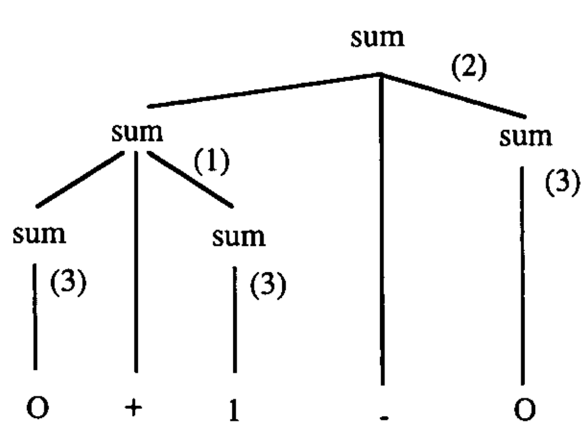


Figure 6.2: Parse tree for simple grammar product '0+1+0'

A sentence that cannot be represented in a way which obeys the production rules, is not a sentence of the language. Natural language incorporates a conceptually similar, but much more complex syntax. In this case, production rules recursively decompose sentences into phrases (well formed sentences in their own right which are of many different types, e.g. *verb phrase* and *noun phrase*) and constituents (collections of words that are not well formed). As with the above example, the final stage, are the terminals (lexemes) of the language itself.

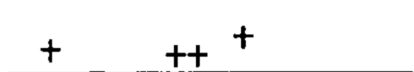
6.3.2.2.2.2. The Effect of Grammars on Available Encodees

Two grammars have been described, one which defined a graphical representation, and one which defined a linguistic representation. As with the morphology and lexicon of a medium, it must be considered how the nature of a grammar affects the encoding mechanism, or more correctly, how does the syntactical description constrain the number of possible encodings.

Firstly, grammars describe all the physical forms of the language by describing what are valid collections of lexemes. Thus, they constrain possible encodings in a similar way to the morphology of the language described in the previous section. For example, the simple linguistic grammar above would not allow the following morphological form:

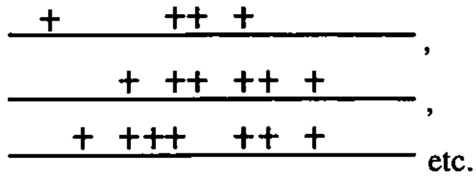
0----1+0 (1)

And also, the graphical grammar would not allow the morphological form:



Thus, it would appear there must be a finite set of sentences that are available for encoding. However, the example given in the morphological section did not take into account the generative nature of an additional syntax. At first glance, the recursive nature of the syntax implies the description of an *infinite* number of legal sentences,

e.g. 0+1, 0, 1, 0+1+10+1, etc.



The only constraint on all these sentences is that if they encode different concepts, then they must be physically distinguishable in some way. Clearly this is possible for the linguistic example since different orderings of tokens will always be differentiable, if studied for long enough. However, this is not the case for the graphical syntax, since the form may cause confusion if points are clustered too closely together. Mackinlay (1986) describes this with the term *effectiveness*. In his discussion, a medium which is *ineffective* can encode the domain data, but does not allow this data to be used in the task due to perceptual confusion. This is an essential consideration, since a medium which is syntactically suitable can become useless if it is unreadable.

However, something more important is at work here which is related to the morphological variation (i.e. different mark positions above the axis) on which the syntactic structure is based. In the linguistic system, the variation is mainly in the order in which the symbols are concatenated[9]. In contrast, the graphical system bases its variation on the horizontal position of the lexeme in one-dimensional *Cartesian* space. The key difference is that the ordering of the linguistic system in *space* is incidental, symbols could just as easily be ordered in time, i.e. spoken . Conversely, the graphical system is inherently spatial in its representation, so is tied to the resolution of the Cartesian space. Thus, this places an upper limit on the number of legal sentences that can be represented.

In this discussion, the following essential point has been made. The number of available encodees that a syntax provides over a lexicon is dependent upon the morphological variations on which the syntactic structure is based. If this variation is a physical property, e.g. a specific spatial position or temporal position, then the number of encodees will be limited. This property is true of all physically encoding systems, e.g. icons, auditory icons (Gaver, 1993), pie charts, etc. Conversely, if the variation is not tied to a physical quality, e.g. the ordering of (a finite set of) *glottographs* in natural language, then an infinite number of encodees can be generated. The importance of this difference will be highlighted in the next section.

---

[9] Physical differences also exist between the symbols, e.g. 0 and 1.

6.3.2.2.2.3. An Abstract Syntax: Subverting Physicality

Thus far, natural language has been treated as a system which represents concepts directly, as with non-linguistic systems such as Mackinlay's HorzPos language . However, as noted earlier, the physical form of natural language actually represents sounds of the spoken language, rather than domain concepts. The reasons for this have been outlined in Chapter 3, fundamentally relating to the possibility of representing any concept for which there is a spoken word. However, since there are no other media common in user interfaces which have this glottographic quality, why should this study be of interest? The key lies in how natural language uses a particular type of syntax to provide encodee structures which are not evident in the language's physical form. This notion can be generalised to other media where tokens, though not glottographic, do not directly encode by their physicality.

As a clarifying example, consider a sentence of the linguistic grammar described earlier:

1+0-0+1

In addition to the ordering relation that was described, further relations can exist between the tokens in the language. These are defined in its syntax and contain the lexemes of the language. These relations, like the earlier production rules, allow a defined set of consistent physical forms to be available for encoding. However, they are more subtle than simply arbitrarily differentiating between the order of lexemes to provide these forms. For example, there could be two distinct types, (typea, typeb) of which the *same* lexeme structures could be members, but would allow to have a different meaning.

typea: (contains) 1+0-0+1;
typeb: (contains) 1+0-0+1.

Thus, these types allow two identical physical forms to be available for encoding in two different ways, by the use of two different types. The fact that a language user would recognise the duplicity of meaning in the lexeme '1+0-0+1' means it does not violate the PDC identified earlier since it has two perceivable different meanings. A natural language example is a word that is both *verb* and *noun* , e.g. cave. In natural language this property is taken further by extending the range of types to which lexemes can belong. The necessity of this process in systems where the morphological variations on which the syntactic structure is based are not physical, is described by Stenning (1995):

"For there to be more than a single uniform semantic interpretation of concatenation, there must be an abstract syntax which provides the diversity of abstract relations to be interpreted", (Stenning (1995): pp. 16).

Thus, Stenning is pointing out that the arbitrary physical variations of token order ("concatenation") is insufficient to provide the wealth of encodees ("semantic interpretation") that are provided by natural language. Stenning also notes that this property is not limited to natural language. Other languages can use (simplified) versions of this process, e.g. semantic networks (Brachman, 1979), where links and nodes symbols belong to types. In this case, the position of the same link in relation to two different nodes allows the arrow meaning to be bifurcated.

Thus, by using an abstract syntax to give a formal description of a wide variety of different types and type relationships, languages which have limited physical encoding mechanisms (i.e. they generally don't use size, position, colour etc.) are able to subvert this deficiency. In doing so, they produce encodees which are not constrained by their physicality (as with graphical systems) and can therefore be extended indefinitely[10]. It is interesting to note that by using physical and abstract methods together, encodings could provide an even wider range of encodees, e.g. combining a two-dimensional spatial syntax with a typed syntax. These would have the advantages of both the physical graphical systems and the type-based linguistic systems. A limited example of this is the semantic network outlined above.

6.3.2.2.2.4. The Effect of Syntactical Complexity on Ease of Use

As mentioned earlier, languages can have a complex or simple syntax. The more complex the syntax, the more varied the forms of representation instances but the harder the language will be to learn. This is compounded by the fact that often the learning of a syntax is based on experience with a small subset of the syntax's products, i.e. the representations. This may lead to erroneous conclusions about what is valid in the syntax.

Weir (1991) shows how mistakes such as these are common in household artefacts, resulting from the highly *moded* nature of their interfaces. These mistakes can be

[10] The question of whether humans could differentiate an infinite number of types and type relationships is an interesting one, but will not be addressed here.

discussed in terms of the interaction *language*. For example, a user may know that the utterance of pressing the '>>' button on a VCR will fast-forward the tape and may attempt to generalise this to all interactions. However, the *moded* nature of this interface means that if the tape is playing, the utterance will have a different meaning, i.e. it will instead advance the tape slowly. An additional utterance is now required (pressing the 'STOP' button) to achieve the same effect. Here, the interaction-language users have attempted to generalise their grammatical knowledge to all utterances when exposure to utterances in all modes of VCR operation were actually required.

In contrast to such misunderstandings, long term exposure to representation systems allows more instances of the grammar to be interpreted, making mistakes of this kind unlikely. A clear example of this is the human ability to articulate grammatically correct sentences subconsciously.

The lack of exposure to interface media make it essential that media syntax is simple enough for users to understand with only minimal exposure to representational instances. However, as with the morphology of the language, care must be taken that the language is also able to provide as many distinct forms as the task domain requires.

Finally, the syntactic differences will be manifested in users' mental model of the world the representation is conveying. This will in turn affect task performance in a way which may be irrespective of the intended match between the syntax of the representation and the description of domain values (as advocated by Mackinlay, 1986).

Clearly, there is an important difference between the way linguistic and graphical syntaxes generate encodees. It is this difference which provides the basis of the expressiveness of a language. However, since it was suggested earlier that the level of language expressiveness is related to the represented domain, then further discussion must take place in the description of second stage of the encoding process, the semantic description. This will be addressed in the next section.

6.3.2.3. Encoding Meaning in Forms

The discussion of computer-based media as languages began with the following description of how the understanding of a language can be divided into two parts:

• The understanding of the encoding mechanism;
• The interpretation of the domain concept.

It was suggested, that the bounded nature of computer domains means that the predominant action in the user's understanding of the interface representation will be their dealing with the encoding mechanism. This has two parts, understanding how a language is able to represent differentiable forms independently of a domain, and understanding how a language encodes meaning in these forms.

The previous section described how languages can generate differentiable forms. The next section describes how these forms can have meaning inputed to them. The key notion identified was that, by using an abstract syntax to give a formal description of a wide variety of different types and type relationships, languages which have limited physical encoding mechanisms (i.e. they generally don't use size, position, colour etc.) are able to subvert this deficiency. In doing so, they produce encodees which are not constrained by their physicality (as with graphical systems) and can therefore be extended indefinitely. This variety of possibleof forms was described for different systems as the basis of the *expressiveness* criterion of languages which will be extended here.

6.3.2.3.1. Semantics (B)

L=L(ID, G0, B) where,
L = A language ID = Lexicon G0 = Syntax
B = Semantic rules

The semantics literature covers a wide range of issues, a number of which are not relevant to this study. The predominant issue for computer-based media is the encoding mechanism, so it is this that the discussion will focus on. Thus, the first question must be: 'Given the multiplicity of forms that a lexicon and syntax provide, how is meaning from the domain encoded? The syntactical section has indicated that it will be fruitful to structure this discussion around physical and abstract languages, with the intention of further illuminating the *expressiveness* characteristic.

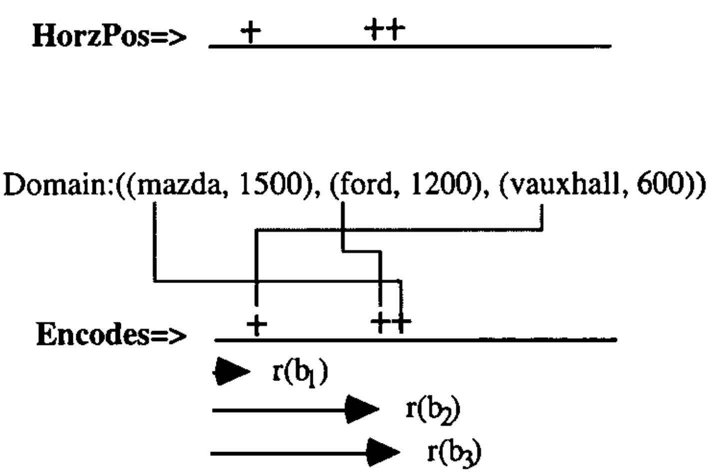
6.3.2.3.1.1. The Encoding Process in Physical Systems

Mackinlay (1986) provides a semantic definition which complements his syntactic description described earlier. Now a domain is considered along with the lexicon already identified. The domain is set of tuples, of the form r{A,B}. Mackinlay gives the example that A is the name of a car, and B is its price. A semantic predicate is now defined called `encodes`. This takes a relation (a_i, b_i) and encodes it using a located '+' on a horizontal axis.

```
Encodes(oi, ai' HorzPos)=>                          (1)
bi=scale*(Position (oi, h) + offset) ^              (2)
Encodes(Position(oi, h), r(ai, bi), HorzPos)  (3)
```

Where `scale` and `offset` are constants that equate the domain value b_i with the axis position of the mark o_i for the domain value a_i.

This predicate will be true for those marks ('+') on the axis (h) which have been positioned by the syntax relation `HorzPos` and that encode a domain relation $r(a_i, b_i)$. Thus, the set of possible encodees that have been provided by Horzpos have now been allocated meaning by the `encodes` relation, as below.



As described earlier, Mackinlay's example, in using the a concrete physical encoding dimension, is representative of any similar graphical language, e.g. bar charts, pie charts, maps. Thus, the *reading* of the language rests on perceiving physical quantities in the representation, e.g. horizontal position, and being aware of how these map to the domain. The central issue in such systems is that if the encoding is carried out in this way, domain values must map *uniquely* to marks of the language (This was described by the perceivable differences criterion (PDC) in the previous section).

Thus, a car with a price of 1500 must have a position on the axis h that is perceivably different from a car with a price of 1200. Given the nature of the encoding method, it is inherently impossible for this not to be the case, since no two marks can be encoded in the same Cartesian position and not violate the PDC[11].

In conclusion, if encoding mechanisms rely on the measured variation of perceptual properties of representing languages, then each dimension of a lexeme of a language may have only represent *one* domain value.

6.3.2.3.1.2. The Encoding Process in Linguistic Systems

We have already described how linguistic systems provide a rich variety of information carriers by utilising an abstract syntax. Unlike the physical systems described above, this syntax is not dependent upon measured variation of perceptual properties, rather it relies on a non-physical, abstract system to provide differentiable encodee structures.

The next stage that must then be considered is how this abstract system allows encoding of domain concepts. This can be divided into two consecutive parts, the consideration of abstract type categories which have already been described, and the way that these types refer to domain concepts. These will now be discussed.

6.3.2.3.1.2.1. The First Encoding Step: Types

The notion of abstract types was discussed in the syntax section, but these are also important to relating encodees to domains. To recap, abstract type categories allow sentences of the language to be grouped under common terms or *types* which are related to each other in the abstract syntax. For example, natural language groups its sound representations into categories such as *verb*, *nominal* ,*determiner*, *pronoun*, etc. The abstract syntax then provides further rules which dictate how these categories are related, e.g. a noun-phrase may be described as the relationship between a DETERMINER (e.g. 'the') and a NOMINAL (e.g. 'chair'). Other non-glottographic systems use a similar technique, e.g. semantic network described earlier (Brachman, 1979).

However, unlike the syntactic descriptions of physical systems such as graphs, the syntactic definition also has a semantic aspect. In other words, the abstract syntax

---

[11] Assuming, as mentioned in the syntax section, that the Cartesian resolution is not so high that different positions are perceived as the same position.

does provide the beginnings of a link with a represented domain, albeit in very generalised terms. For example, consider the *verb* type in natural language. Part of its formal grammatical description is:

"a word used to indicate an action, state, or occurrence..", (Allen, 1990)

Thus, examples of verbs are *demand, rain, sit* and *fall.*

The key is that in their definition, these types are describing general concepts , e.g. action, state, occurrence, of a particular kind of domain. Clearly, for natural language, the domain is a temporal, causal world, thus these types must be relevant to it. For another linguistic system which describes an inherently different world, a different set of type definitions would be required. For example, a non-temporal world would not require descriptions of action or state, since actions resulting in changes of the world state would be impossible.

Clearly, the abstract definition of the verb-type dictates those lexemes (glottographs, in the case of natural language) which will be members of the type. Since lexemes in linguistic systems are of arbitrary physical form, and bear no physical relation to what they represent (for natural language this assumes we are concerned with the domain concept the glottograph refers to), then the types *begin* to give these lexemes some relation to the domain that will be represented.

In conclusion, types in linguistic systems can go some way to encoding domain meaning by their syntactic definitions.

6.3.2.3.1.2.2. The Second Encoding Step: Type Reference and a Lexical Dictionary

So far, the arbitrary symbols of systems which utilise an abstract syntax have had their encoding described in a limited degree by abstract types. The next step in the encoding process is identified by Stenning (1994), who differentiates between two types of semantic encoding:

- **Token Reference**: Tokens of language refer to objects in the domain uniquely. Thus, no two tokens can refer to the same concept.
- **Type Reference**: Tokens of a language refer to sets of things. In these systems it is possible for two *different* tokens to refer to the same object.

Stenning highlights graphical systems as an example of the former, whilst linguistic systems are seen as the latter. For example, in natural language three lexemes could equally refer to the domain concept 'The male, taxi-driver, Bob'.

"The man" (1) and "The taxi-driver" (2) and "Bob" (3)

The abstract syntax tells us that they are all noun-phrases and therefore describe *objects* in the world, thus giving a first-cut semantic description. The next stage of the encoding process is to provide a more specific description of the object. This can be done by the use of a lexical *dictionary* which provides general definitions of individual lexeme encodees, not multiple lexeme encodees (sentences). Further explicitness in the domain can only be provided by considering the context of the utterance. This is outside the encoding mechanism and is only considered here for completeness. As the first two lexeme sequences stand, the only way a further specificity can be obtained is by placing the statement in a wider context, e.g.

"*Bob is standing by the window.* The man/taxi-driver *is by the window*"

This allows the referent of these statements to be made apparent, but is clearly not a product of the encoding process itself, rather it is an act of interpretation based on wider knowledge. In fact, the encoding system provides no further specificity, due to its type-referential nature. In other words, it is generally impossible for the encoding system to be specific in and of itself. The only exception is the use of *proper nouns*, as in case (3) of the verb-phrase examples . A proper-noun is a semantic type which describes unique names for objects due to their uniqueness in the represented world. Thus, 'Bob' is used to refer to the unique entity 'Bob' and therefore, in effect, becomes *token* referent, i.e. it refers to objects in the domain uniquely

In summary, is the non-physicality of the encoding mechanism of linguistic systems which causes their inability to be specific, as manifested in the notion of type reference described.

6.3.2.3.1.2.3. Expressiveness

To summarise the discussion of semantics thus far, the following points have been made:

- The encoding mechanism of physical and linguistic systems have important repercussions for what they can encode. More specifically:

    - Physical systems can only encode a single domain concept in each perceptual encoding dimension. They are restricted by their physical nature.
    - Abstract encoding systems generally encode *sets* of domain concepts for each of their *typed* lexemes. They are inherently unable to offer the specificity of physical systems (except proper nouns).
- The encoding mechanisms operate independently of the represented domain.

There is an implicit notion in this discussion, linguistic systems, by virtue of their encoding mechanisms can encode more than one domain concept in a single lexeme or sentence(s). For example, the natural language statement 'Disk' encodes any disks that may be in the domain. It does this by being type-referent, and being only as specific as this type-reference allows. Conversely, physical systems require a physical similarity to what they represent, and since what they represent has a single form, then so must the representation; these systems cannot be anything but specific in their encoding. Thus, the icon:


,

can only encode one disk. Any additional encoding of other disks within this icon, can only result from an arbitrary mapping of this icon to an abstract type set, say *collective-icon* [12]. Clearly, this would go beyond the *physical* encoding mechanism of the language.

The discussion has now reached the stage to recall the definition of expressiveness which was given in Chapter 2.

***expressiveness:** an emergent property of a medium which measures how much information it can carry.*

The importance of this property was alluded to in Chapter 4, with respect to the amount of abstraction over problem domains that tasks require to be effectively achieved. For example, there may be some situations were it is possible for each domain concept to be browsed individually. As concepts can be represented individually, individual lexemes of the representation need only carry specific information, e.g. '+'s on a graph. It may also be that the number of specific concepts is large. For example, the multi-dimensional icons of (Spence, 1989; Spence and

12 This could be in the form of a manual entry, e.g. "The disk icon shows a disk is presently located in the floppy drive.".

Parr, 1990) may encode many different aspects of domain in a single representation instance, but each aspect is still encoded specifically (See Figure 6.3).

Conversely, it may not be possible to view all concepts, thus some abstraction will be required. In this case, individual lexemes will encode more information per dimension, e.g. 'All Greeks are philosophers'. In other words, the former requires specificity, and the latter generality. The key is that both of these representations can be defined in terms of levels of *expressiveness*,.

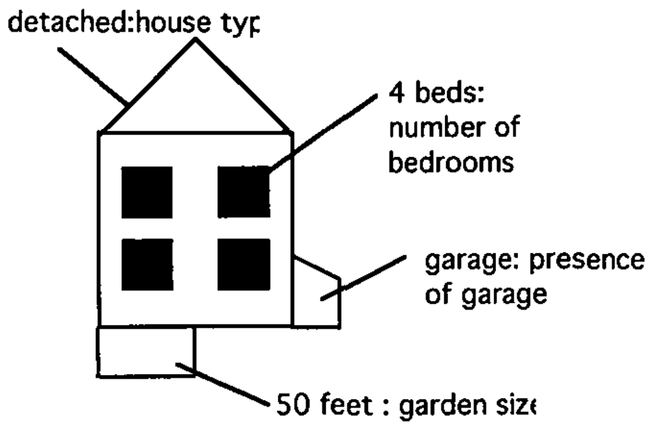


Figure 6.3: A multi-dimensional icon (Spence et al.)

In conclusion, expressiveness can be described as a property of the encoding mechanism of a language, i.e. a product of its lexicon, syntax, and semantics. Since all interface media can be described in these terms, then expressiveness can be offered as a unifying dimension over which to compare different media. At this stage, the discussion of expressiveness has been built from its representational genesis, the next chapter describes media in terms of this characteristic, with particular reference to performing tasks and mental models.

6.3.2.4. Interpretation

The discussion of interface media in linguistic terms addressed the first of the following issues.

- The understanding of the encoding mechanism; (1)
- The interpretation of the domain concept. (2)

This was due, mainly, to the highly bounded nature of computer application domains. However, for completeness, the second stage of interpretation was addressed briefly in the semantics section. This discussion, which is predominantly based on natural language understanding, goes beyond the notion of an encoding mechanism. However, it does offer important insights into the cognition of language understanding which will be addressed with respect to computer-based media and expressiveness in Chapter 8.

6.3.2.4.1. Intensional vs. Extensionsal

This study of interpretation can be divided into two parts, intensionality, and extensionality or *sense* and *reference*. The first relates the meanings of tokens to each other. e.g. synonyms, hyponyms, and syntactic types. It is this aspect that has been touched on in the previous section. The second, relates tokens to the domain they are describing, e.g. "the sun is shining", carries additional information to those with the real-world domain knowledge of cloudless, blue skies accompanying sunshine. The discussion will focus on the latter cases since it is the representation of the domain by the interface which is of paramount importance.

Stevenson (1993) provides a more concise description of the relation between a representation system's lexemes and the world it describes. She states:

"..to know the meaning of a sentence is to know what the world would have to be like to make the sentence true" (Stevenson, 1993: pp. 83).

For example, to identify if user knows the meaning of the dialogue message, "Do you really want to Quit?" would require the user articulating states of the domain where this message would be applicable. Such as, "When I've pressed the QUIT button".

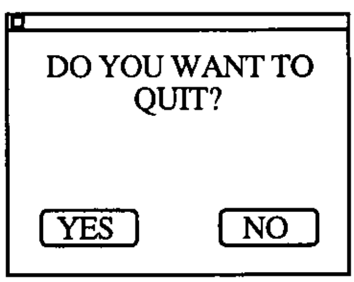
This is one way of describing semantics, but this definition does not capture other aspects which may come to bear on sentence meaning, such as the listener's general knowledge or the speaker's intentions, e.g. a user may have an unnatural fear of technology which would influence their interpretation of the statement "Do you really want to quit?". These additional factors will be described in the pragmatics section.

6.3.2.4.2. Model Theoretic vs. Truth-Theoretic

A further distinction in the discussion is required between the approaches of truth-theoretic semantics and those of model-theoretic semantics. The former concentrates on logical descriptions and inferences in order to evaluate the meaning of a statement. In this process, to know the meaning of a statement is to be aware of a binary division of situations in the referent world; those which make the statement true and those which make it false. In doing so, the method focuses on what the statement 'says' and therefore relies more heavily on the syntax of the language. Unfortunately, this theory does not identify which of those domain states that make the statement true is the actual state referred to. Since the thesis is moving towards a media allocation strategy based on mental models of domains induced by different media, the model-based approach is the most suitable.

The model-theoretic approach was originated by Tarski (1931), and deals with models which are implied by statements. The meaning of the statements are then described in terms of the model by highlighting those entities which make the statement true. Furthermore, by increasing the number of constraints on these models a more specific meaning can be established. Thus, the statement "Do you want to Quit?" implies a world where one state can be left to move to an implicit one. If the interpreter knows of no such *model*, then this implication will not be made and no useful meaning will be interpreted.

Thus, this view is particularly relevant to a mental model-based view of media allocation. The mental model reflects the interpretation (meaning extraction) of the representations used in the interface. In line with the model-theoretic view, this mental model should represent the domain model which has implied from the representation instance. Thus, returning to the example, the mental model of the dialogue interface representation:



Should describe the present world-state and the next state depending which answer is given.

6.3.2.4.3. Contextualising Interpretation

Unfortunately, both truth-theoretic and model-theoretic approaches are unable to identify the actual referent in the domain since they can only propose candidate states which will make the statement true. This truth will be evaluated either in the *open* *world* of truth-theoretic semantics, or in the closed *world* of model-theoretic semantics[13]. Thus, what is required is a notion of the context of the utterance, something that goes beyond what is 'said'. This should provide the additional information to make the actual referent apparent.

Higgenbotham (1988) argues that the model theoretic view of semantics does not allow for different language users who identify different possible worlds in which a statement is true, but still arrive at the same interpretation. He suggests that this shows that model-theory is "too coarse for linguistic processes" (Higgenbotham , 1988: pp. 45), since such discrepancies cannot be explained by speakers being aware of increased constraints on models, as Tarski suggested.

Higgenbotham also suggests that the reliance on logic to provide theoretical descriptions of admissible models is far too specific, since language users rely on much less formalistic descriptions. For example, a speaker knows that *apple is true of x, if and only if x is an apple.* On the contrary, a rigorous logical definition could rightly state, *apple is true of x, if and only if x is an apple and Florence is near Pisa.* Clearly, it would be unusual for a language user to place this further constraint on the interpretation of this statement. Thus, Higgenbotham draws the same conclusion for formal definitions of syntax and semantics, stating:

"..the rules of syntax, are more finely discriminated than their output, so the principles of semantics are more finely discriminated than the classes of models that they determine.", (Higgenbotham, pp. 46, 16).

What Higgenbotham is describing is a pragmatic view of semantics, i.e. one which deals with real, rather than theoretical, semantic interpretation. The relationship this notion bears to user interface design is important.

Barwise (1983) and Johnson-Laird (1983) propose that utterances are understood within the context in which they are spoken. Both theories rely on knowledge of the world which is combined with the sentence to provide a context for its interpretation. For example, the statement 'The car' would exist within its context, e.g. In a description of my new car. As addressed earlier, Kamp (1981), Grotz(1977), and Heim (1983) suggest this context is provided by a *discourse model* which holds all those entities introduced into the discourse. This allows the meaning of indefinite phrases such as 'it' and 'that' to be resolved within this context. Finally, the work of

[13] The 'closed world assumption' also allows the description of what is **not** in a domain.

Bransford et al. (1972), Bransford and Johnson (1972) show how subjects who received a pictorial description of the context of a prose passage, rather than a linguistic one, were more able to recall aspects of the passage when question afterwards.

A final link between sense and reference is provided by Kintsche's (1977) model of comprehension. This incorporated a three-level model moving from syntactic, to truth theoretic (prepositional), to contextual knowledge (a discourse model). This work stressed the mutual relations between these three levels. Debate is still active as to which level should take precedence. For example, Higgenbotham (1988) argues that there are some primitive semantic elements which can be interpreted as distinct from a context. These 'disquotational facts' rare unaffected by context. For example, if a language states 'that Snow is white' is TRUE, then this will always induce the interpretation *snow is white*. Higgenbotham argues a language user is aware of a wide variety of these disquotational facts.

So whilst there is a large amount of literature on the description of semantic encodings the majority of this work is restricted to the open-world of natural language. The smaller amount of literature on concrete representations glosses over semantics, since the semantic mappings are too straightforward. The simplicity of representations mean that only one interpretation is possible, which rules out any discussion of the semantics defined by the system being too "finely discriminating" (Higgenbotham). However, a wide range of representations, including some which may not have this simplified quality, must be considered .

6.3.2.5. Pragmatics

Pragmatics describe characteristics of language in use. These may include the bringing of other 'world knowledge' outside the communication domain into play, or a consideration of the intentions, goals, and beliefs of the speaker and listener. In terms of physical representations, a pragmatic definition could be the limit on the maximum number of objects which can be displayed without causing confusion in which lexeme referred to which domain concept.

Since the thesis is addressing bounded computer domains, the consideration of communication pragmatics need only be limited. A handful of studies have addressed how these pragmatics may affect the choice of output media. Briefly these include:

- Marks (1991) used a study of pragmatics to affect the output of network diagrams. For example, interest in inputs and outputs would mean a representation with inputs at the top of the screen and outputs at the bottom.
- Tubbs and Moss (1991) discussed the effect of power and status roles on communication. These may affect how output messages are interpreted, e.g. "Do you wish to make an alias?" may be affected by the whether the user is expert enough to know the usefulness of an *alias*.

These studies are described in more detail in Chapters 1 and 2. Though, the pragmatics of computer-based media are an area of active research (Douglas et al., 1995), as yet, there have been few concrete findings. A consequence of this is that the applicability of linguistic descriptions of pragmatics to such media has not been proven. However, in an attempt to remedy this situation, this term will be addressed further in Chapter 8 in terms of the pragmatics of physically encoding languages.

6.3.3. Examples of the Linguistic Descriptive Framework

The following examples describe the representation of a multi-variable, two traffic light road. They are intended to demonstrate the aspects of a linguistic view of computer-based media that have been discussed in this chapter[14]. The example media will be seen again in Part 3 of the thesis, since they are the chosen experimental media. However, at present a brief description of the experimental domain will be given in order to ground the discussion of the relationship between lexicon, syntax, and semantics. As mentioned earlier, linguistic-like pragmatics are not overtly relevant to computer-based media. Neither is the discussion which addresses any task-specific characteristics of the representations since the focus is on the task independent encoding mechanisms of the media.

6.3.3.1. The Domain Description

The domain is a real-time simulation of traffic flow. Two roads are positioned at right-angles to each other, each with two sets of traffic lights located one-third and two-thirds of the way along their length. The system has a number of state variables, which include, queue length on each light (and a total), delay on each road (and a total), and average flow rate into each light and each road. In the examples, only one road will be shown.

[14] These examples do not address expressiveness as this is discussed fully in the next chapter.

6.3.3.2. Animation: Using Physical Languages

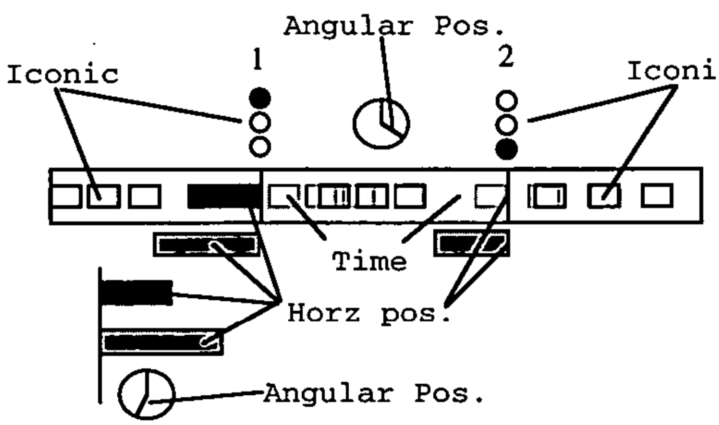
The animation incorporates a range of simple graphical systems which encode both by position (horizontal and radial) and by pictorial similarity to the referents (i.e. traffic lights and the flow of objects). These two aspects of the system will now be described graphically.

Lexicon

The diagram below shows the result of the combination of the morphological structures of the medium, i.e. its lexicon. The syntax of the language will vary one of the physical dimensions of each lexeme in order to define a set of encodees.

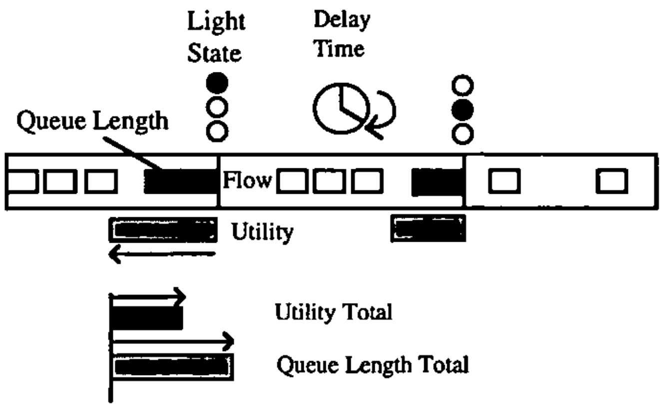
Syntax

In the diagram below an example, legal, representation is shown. This is accompanied by descriptions of each physical encoding used in the medium, e.g. Horz. pos.=Horizontal position encoding. It is the syntax which generates all of the possible encodees to which meaning in the domain can be imputed. Thus, at this stage no semantics are associated with the well-formed *sentences* of the language.



Semantics

The final stage is to give the different encodees generated by the syntax a meaning in the domain. Given the real-time nature of the domain, the specific meaning, i.e. variable value, that is imputed to the physical variation of each lexeme will change over time.

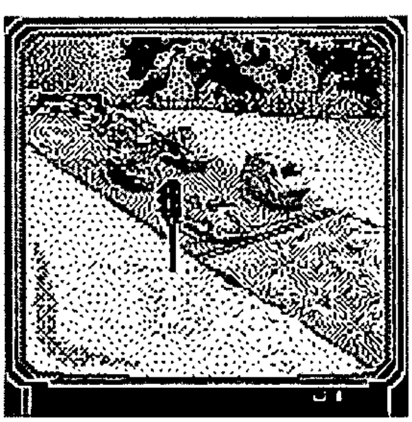
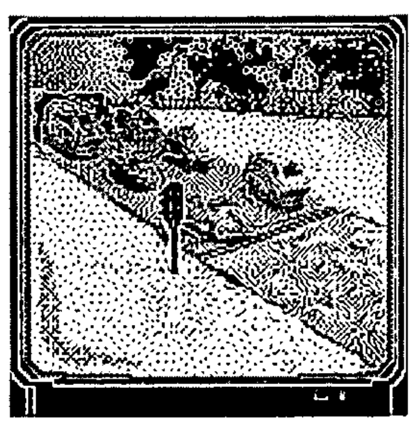


6.3.3.3. Static Video: Using Simple Type Encoding

This medium uses syntactic types to classify its lexemes. In a similar way to the natural language types described earlier, they are closely related to the domain, e.g. '20<Queue Length<30', refers to a specific range of values of a domain variable.

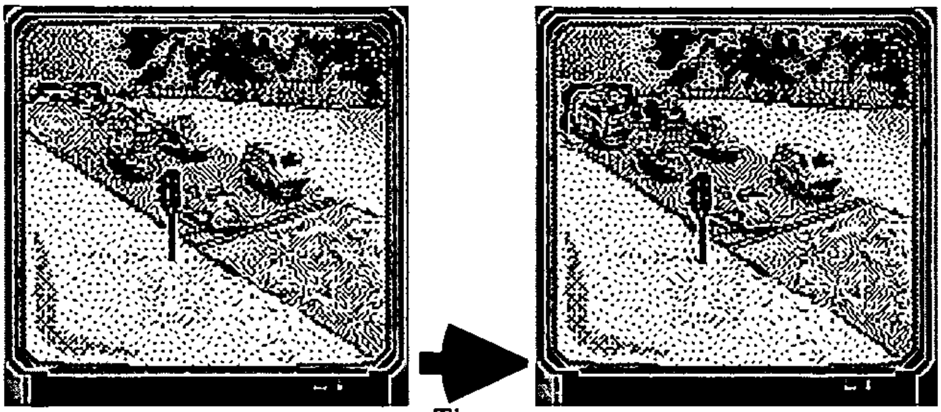
Lexicon

As with the animation the diagram below shows the result of the medium's morphology. In this case, the lexemes are images of different states in the domain, e.g. a traffic light with no cars queued, a traffic light with five cars queued, etc.


,

etc.

Syntax

The syntactic rules of this medium are based on limited type definitions. The types define ranges of domain values, and members of the lexicon are allocated to each type. This allows the temporal sequences of domain values, as represented by their members, to be encoded by the products of the syntax (as shown over)



Time

Semantics

The syntactic types identified allow the images to carry a range of meanings in the domain. For example, the image below encodes a range of queue length values between 21 and 29 cars. This is made possible by the physical encoding system of the medium being replaced by a type-based encoding system. This allows a lexeme to have a meaning as complex or as simple as the type definition dictates. The drawback of such a system, is that the user of the medium must be aware of the different types that different lexemes belong to. Thus, they are harder to understand.

20<Queue Length<30

6.3.3.4. Table: Using a Specific Linguistic Representation

This medium encodes domain values using an arbitary symbols which do not directly encode by their physical form, i.e. numbers. However, the *uniqueness* of each alphanumerical symbol allows them to be regarded as *specific* representations of domain values, in a similar way to the use of *proper nouns* in natural language. Thus, although the form of the symbols is arbitrary, they only have one meaning in the domain. Thus, once learned, the symbols effectively become physical encodees, i.e. their referent value is obvious from their physical from.

Lexicon

The lexicon defines the unique alphanumeric lexemes which though arbitrary in form, refer in a *unique*, physical way.

0...9   a..z   A..Z   ☐

Syntax

The syntax defines that there are two columns with four values in each rectangular area, save the smaller areas where there is one value. Additionally, a four value history is shown in each rectangle.

			Total Value	
Value1	value1a(t-3)	value1b(t-3)		
	value1a(t-2)	value1b(t-2)		
	value1a(t-1)	value1b(t-1)		
	value1a(t)	value1b(t)		
Value2	value2a(t-3)	value2b(t-3)	vt2(t-3)	v2t
	value2a(t-2)	value2b(t-2)	vt2(t-2)	
	value2a(t-1)	value2b(t-1)	vt2(t-1)	
	value2a(t)	value2b(t)	vt2(t)	v2t
Value3	value3a(t-3)	value3b(t-3)	vt3(t-3)	v3t
	value3a(t-2)	value3b(t-2)	vt3(t-2)	
	value3a(t-1)	value3b(t-1)	vt3(t-1)	
	value3a(t)	value3b(t)	vt3(t)	v3t
Value4	value4a(t-3)	value4b(t-3)	vt4(t-3)	v4t
	value4a(t-2)	value4b(t-2)	vt4(t-2)	
	value4a(t-1)	value4b(t-1)	vt4(t-1)	
	value4a(t)	value4b(t)	vt4(t)	v4t

Semantics

The semantics of the table define the representation of domain values with *unique* numerical lexemes structures, e.g. 7500. Along with this, lexemes are grouped in meaningful arrangement in order to show the progress of domain values over time, and the relationship of variables of a similar type, e.g. *Flow* values.

			Totals	
Flow	10000	7500		
	10000	10000		
	7500	10000		
	7500	0		
Queue	128	43	171	80
	129	42	171	
	130	40	171	
	131	38	169	0
Delay	300		300	0
	300		300	
	300		300	
	320		320	150
Utility	6	1	7	0
	6	1	7	
	6	5	11	
	6	5	11	7

6.4. Summary

This chapter showed how the linguistics literature, though mainly concerned with natural language, can be used to discuss the range of interface output media. The understanding of media was broken down into two stages:

- The understanding of the encoding mechanism;
- The interpretation of the domain concept.

However, a number of caveats were placed on the applicability of this literature. Firstly, the bounded nature of computer-based domains means that the focus of the study can move away from the study of interpretation and focus on the understanding of the encoding mechanism.

Interface media were then discussed in terms of the linguistic categories of:

- *The morphology of a language*: This is restricted by a number of factors. These factors are less apparent for natural language since it has evolved within relatively fixed constraints, i.e. human vocal chord, paper quality, etc. However, for computer-based media, the resource limitations can be highly restrictive, e.g. processing speed, display refresh rate;

- *The syntax of a language*: This determines the number of encodees that may carry meaning. Languages that use variations in their physical form to carry meaning are limited by the number of variations in the form that can be perceived by a reader. Natural language, on the other hand, overcomes the limitations of a finite set of physical tokens through the use of a complex syntax which groups tokens into *type* categories;

- *The semantics of a language* This describes how meaning in the referent world is imputed to encodees. In physical systems encoding is on a one-one basis, where each physical dimension can carry one meaning. Linguistic systems rely on reference by type, and therefore encode sets of domain objects. The encoding mechanisms of the former preclude any generality in representation and the mechanism of the latter precludes specificity.

• *The pragmatics of a language*: This defines the additional effect on a recipient that a language has when in use. Pragmatics arise from the highly contextual nature of natural language use which allows a large body of additional information which is not directly represented in the utterance to be brought to bear in its interpretation.

The difference between the two types of encoding mechanism was then used to illuminate the notion of expressiveness addressed in Chapters 2 and 4. *Expressiveness* can be described as a property of a language's encoding mechanism, i.e. a product of its lexicon, syntax, and semantics. Since all interface media can be described in these terms, expressiveness can be offered as a unifying dimensions over which to compare different media.

Given the description of the representational basis of expressiveness, a detailed description of its usefulness with respect to media allocation will be discussed in the next chapter. This includes its relationship to mental models and the suitable representation of task domains.

Chapter 7

Defining Expressiveness

7.1. Introduction: Linking Mental Models, Interfaces and Domains

Throughout the thesis, the aim has been to link the three parts of the interaction triptych-the user's mental model, the interface representations, and the tasks which must be achieved in a domain. At the end of part one, the notion of *expressiveness* was proposed as a possible unifying dimension along which interface media could be compared. This property was defined in Chapter 2 as:

expressiveness*: an emergent property of a medium which measures how much information it can carry.*

The importance of this property was alluded to in the two further chapters. These addressed the relationship between problem solving and mental models. Firstly, the importance of the right level of *abstraction* in a problem solver's view of a domain was seen as instrumental in the successful completion of tasks. In relation to this, effective task performance was also related to a suitable mental model of the domain. The bridge that links these two needs is the interface representation. The discussion showed that this can provide the necessary *abstraction* in the user's view of the task domain and is a key determinant of the user's mental model of the domain.

Chapter 6 described the representational basis of *expressiveness*. This was argued in terms of the difference between the encoding mechanisms of different media, and how this determines the number of domain concepts that can be represented by a number of encoding dimensions. The thesis is now in a position to provide an in-depth discussion of the uses of *expressiveness* along with its relationship to domain abstraction, mental models, and the ultimate goal of media allocation.

7.2. Expressiveness

In Chapter 6, expressiveness was described as a result of the encoding mechanisms of languages. The notion of levels of expressiveness was also touched on as a way to compare media. However, a more formal definition of expressiveness is required to allow any medium to be assessed in terms of its expressiveness. In the immediate discussion, the importance of the relationship between expressiveness and mental models will not be addressed. This will be developed in the next chapter once the definition of expressiveness has been expanded.

Before defining expressiveness in this concrete way, it is helpful to investigate similar notions in the HCI literature. These studies may offer important indicators on how expressiveness can be more strictly defined.

Mackinlay (1986) describes expressiveness as :

" a set of facts is *expressible* in a language if there is a sentence in a language that encodes every fact in the set", (Mackinlay, 1986: pp. 118)

He stresses that this property is the result of the syntactic and semantic definitions of a language (Examples of Mackinlay's syntactic and semantic definitions, `HorzPos` and `Encodes`, are described in detail in Chapter 6). Thus, he defines the predicate expressible:

```
Expressible (facts, lang)<=>
     if there exists an s[lang(s) ^ for all f[
     f ∈ facts=>Encodes (s, f, lang) ^
     f is not ∈ of facts=>¬ Encodes (s, f, lang)]]
```

where `lang` is a syntactic predicate, e.g. `HorzPos` (as defined in Chapter 6), and `s` is a sentence. However, on closer inspection, Mackinlay's definition should be:

```
Expressible (facts, lang) defines
     for all f there exists an s where ∈lang ^ f∈facts
     Encodes (s, f)
```

This predicate is then used by Mackinlay to define a number of *theorems* which prove that certain domain types cannot represented by the `HorzPos` syntax, e.g. one-to-many relationships as shown below:

$r(a_i, b_j) \wedge r(a_i, b_k) \wedge b_j$ does not equal $b_k \Rightarrow$
$\neg$ Expressible (r, HorzPos).

Like the thesis' proposed notion of expressiveness, Mackinlay states his definition is a product of the encoding mechanisms (syntactic and semantic) of the language. However, the aim of Mackinlay's expressiveness criterion is different from this study. In his case, a language is deemed expressive enough if it can encode a certain type of domain relationship (e.g. HorzPos is *expressive* enough to encode binary relations, $r(a_i, b_j)$). Whilst this is an essential requirement for media selection, as described in Chapter 1, it is only a *first-cut* solution. Moreover, the latter notion of expressiveness assumes that the medium *is* of a suitable syntactic nature to encode domain values. Thus, the focus is then on how the medium can support effective problem solving over the domain by using the right level of (abstractive) expressiveness from amongst the possible choices. Mackinlay's notion of a suitably expressive medium corresponds to the lower end of the expressiveness continuum, i.e. direct representation of domain facts through perceivable dimensions. It is only at this level of expressiveness that the mapping of individual domain relations can be seen and analysed using the `expressible` predicate. Conversely, at the higher levels of expressiveness the mapping between domain elements or encodees may be less obvious.

Finally, Mackinlay assumes that the encoding of the fact-set, `facts`, will use a physical encoding system which cannot abstract over members of facts within a single encoding (e.g. '+'). What is necessary is an extension to the theoretical notion of expressiveness in order to describe the abstract, type-referent media described in Chapter 6.

Coutaz et. al. (1995) define interface expressiveness as:

"the capacity to allow an *agent* to reach state s" from a state s' "

As described in Chapter 2, an *agent* is a user, and the domain is seen as a multi-dimensional state space through which the user navigates. Thus, starting from an initial state in this space (s'), users interact with the domain, through a particular interface, in order to reach the goal state (s"). A medium (m) is thus defined as having an expressiveness for this trajectory, by the triple `reach (s', m, s'')`.

This approach is relevant for the following reasons. Firstly, the view of the domain as a state space through which the user navigates has already been advocated (in Chapter 4) as providing a more tangible description of interaction. This then allows media to be seen as aids to goal directed navigation, viz. problem solving, and effective media allocation to be the provision of suitable *aids*. The second important aspect is the description of expressiveness as a match of the medium to the task, as defined by the `reach` triple.

However, the description fails to offer explanations as to why a medium `m` would satisfy a reach triple, `reach (s', m, s'')`. The thesis postulates that it is the provision of the right amount of abstraction by a representation over the domain state-space, i.e. its expressiveness , which allows the reach triple to hold. Clearly, effective media allocation can then be seen as providing this match.

As a final point, Stenning and Oberlander (1995) attempts to define a tripartite expressiveness continuum shown in Figure 7.1. Like Coutaz et al., Stenning and Oberlander define the represented domain as a state space. The *MARS* level is equivalent to analogous representations and the *LARS* level is equivalent to an intermediate expressiveness. In the latter case, Stenning stresses:

"..abstraction is only permitted over models (of the domain) which differ with regard to one object's value on exactly one dimension...abstraction is limited, in that little flexibility is allowed in picking out regions of the space of possible models (domain states)" (Author's brackets), (Stenning and Oberlander, 1995: pp. 104).

Finally, the *UARS* level is equivalent to the unlimited expressiveness afforded by the use of a *typed* syntax, as described in Chapter 6.

expressiveness

↑

UARS (Unlimited Abstraction Representation System) e.g.
Natural language
"can express arbitrary positions of the state space"

LARS (Limited Abstraction Representation System) e.g.
Euler's Circles
"can express several models, but not in an arbitrary dissection of the state space"

MARS (Minimal Abstraction Representation System) e.g.
Graph, table
"can only express one model of the domain"

Figure 7.1: Three levels of expressiveness (Stenning and Oberlander, 1995)

Unlike the other authors, Stenning offers reasons for the expressiveness of a range of different media which are based on their encoding mechanisms.

7.2.1. An Expressive Definition

The full definition of expressiveness is based on parts of all the approaches outlined above. From Mackinlay's (1986) work, the use of a semi-formal definition is seen as clarifying the concepts described. However, the work of Coutaz et al. (1995) and Stenning and Oberlander (1995) provides the descriptive framework which weds expressiveness to tasks. Added to this mixture is the importance of the right level of abstraction in problem solving, as identified in Chapter 4.

Firstly, media must be described in terms of their encoding mechanism:

$$\{m_i \in M \mid \mathrm{lex}(m_i) \wedge \mathrm{syn}(m_i) \wedge \mathrm{encd}(m_i) \neq \emptyset\}$$

- M is the set of available media;
- `lex(m)` defines the set of lexemes of a medium;
- `syn(m)` defines the set of all well-formed encodees;
- `encd(m)` defines the set of encoding dimensions of m, e.g. position, size, types.

Secondly, domains (D) and tasks (T) must be defined in terms of states, and state space trajectories, respectively:

$$D \Leftrightarrow S \cup T$$

$$d_i \Leftrightarrow (s_i \ldots s_{i+n}) \cup (t_{ij} \ldots t_{jz}) \wedge$$
$$t_{ij} \in T \wedge s_i \in S \wedge$$
$$t_{ij} \Rightarrow \mathrm{traject}\ (S_i, S_j)$$

where multiple task trajectories (solutions) are defined between an initial state, s_i, and a goal state, s_g:

$$\mathrm{traject}\ (S_i, S_g) \Leftrightarrow [s_i..s_e..s_k..s_g] \vee [s_i..s_b..s_y..s_g\ ]\ \text{etc.}$$

Finally, the expressiveness of a medium m is defined as:

$$\mathrm{expr}\ (M) \Rightarrow \mathrm{abstract}\ (M,\ \mathrm{encd}(M),\ D,\ Si,\ Sg)$$

where paths can be encoded in total, or in parts:

```
abstract (M, encd(M), D, Si, Sg)=>
   si∈d ∧ sg∈d ∧ sx ∈d∧ d=enc(m) ∧
   encode (encd(m), traject (si, sg)) ∨[
   encode (encd(m), traject (si, sx)) ∧ encode (encd(m),
   traject (sx, sg))]
```

Where encode `(encd(m), traject (s1, s2))` defines the encoding of all state paths between states s_1 and s_2, with the set of encoding dimensions defined by `encd(m)`.

A number of properties of expressiveness are apparent from this definition.

7.2.2. Expressiveness is Domain Dependent

The predicate:

```
expr (M) => abstract (M, encd(M), D, Si, Sg)
```

shows that the expressiveness of the media in M is partly defined as, the ability to encode all, or part of a trajectory defined by `traject` (s_i, s_g), within its set of encoding dimensions, `encd(M)`. Of course, if the path s_i->s_g covers a large number of states, then some abstraction will be required. However, if the path covers a small number of states, then no abstraction will be required, e.g. s_1->s_2. However, since encd(M). is also the result of `M`'s `lex(M)` and `syn(M)` predicates, then the expressiveness is also dependent on the encoding mechanism of members of `M`.

A main motivation for developing the expressiveness criterion was that it would provide a common ground over which media could be compared. Now it seems this common ground has been removed since the quality is dependent on domains and tasks. However, the following points must be stressed:

- If media are being allocated, domains will be fixed;
- If media are being allocated to tasks, then the tasks will be fixed.

Thus, within the task/domain context absolute values of expressiveness can be defined.

7.2.3. An Expressiveness Continuum is Defined

The predicate,

```
expr (M) => abstract (M, encd(M), D, Si, Sg),
```

shows that a medium has an expressiveness which varies as a function of a number of factors. Given this variation, there must be a range of expressiveness both within one medium, and across media. However, given D will be fixed, it is the variation in `encd(M)` which is of interest, i.e. variation of expressiveness across different media. Of course, we have already seen this, albeit indirectly, in our discussion of abstract and analogous systems in Chapter 6. An expressiveness continuum can now be described in more detail.

Generally, representations allow the representation of one or more of the abstractions that are possible over a domain. The higher the expressiveness, the more of these abstractions can be represented.

The discussion assumes a given a domain d and task t_i where $d_i = t_i \neq \emptyset$. It is also assumed that d_i is large enough to require some abstraction in order to solve t_i effectively, i.e. there are enough states to warrant the coalescing of states by abstraction.

7.2.3.1. Minimal Expressiveness

It is relatively straightforward to *define* representations that have minimal expressiveness, but often this quality is subverted in some way in actual interface media. For example, representation languages may incorporate a number of different encoding dimensions to encode multiple domain *values*. Clearly, this extension is limited to languages which have multiple encoding dimensions, i.e. generally physical languages, or abstract languages with additional physical encoding dimensions[1].

However, if these values are simply different attributes of an atomic domain concept, (e.g. multi-dimensional house icon showing number of beds, size of garden, etc., (Spence and Parr, 1989)) then expressiveness is still minimal since only one domain concept is represented by a sentence of the language, e.g. a house. An increase of expressiveness comes when different encoding dimensions represent conceptually

---

[1] An example, described in Chapter 6, is a semantic network (Brachman, 1979). This incorporates an abstract syntax into a two-dimensional spatial framework, i.e. *typed* nodes and links.

unrelated domain concepts. An example of this is the use of *object displays* which represent multi-variant domains (Jacob and Egeth, 1976; Carswell and Wickens, 1987; Buttiegieg, 1989; Coury et al., 1989). These are graphical representations of geometric shapes (in Jacob and Egeth's case, faces), whose dimensions are used to encode multiple domain values. For example, Figure 7.2 shows an enocoding of three distinct domain values (i1, i2, o) in the heights of the three vertices of the shape:

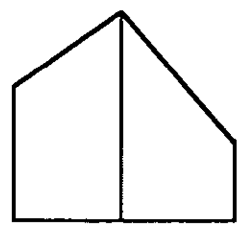


Figure 7.2: Object Display from Buttigieg (1989)

In this case, multiple domain concepts are encoded in a sentence of the graphical language. A similar example is an apposed axes graph, where the horizontal and vertical positions of a point may encode two unrelated dimensions. However, the increase in expressiveness is limited by the nature of each encoding dimension, i.e. they can only represent one domain concept. Moreover, the number of physical encoding dimensions that can be used simultaneously is also constrained by the number of different dimensions the human perceptual system can differentiate between.

In summary, low expressiveness representations are able to encode a small subset of all the abstractions over domain states that are possible. The size of these abstractions can range from one to e, where e is the number of encoding dimensions. The number of different abstractions will be the total number of permutations of states represented by these e dimensions, e.g. a medium with two dimensions (e=2) will define abstractions that contain two states. A domain with 4 states will therefore allow $P_2{}^4$ abstractions through this medium.

7.2.3.2. Higher Expressiveness

As described in Chapter 6, the physical nature of non-abstract encoding mechanisms (e.g. graphics) enforces limitations on:

- The number of physical dimensions which can be used to increase expressiveness;
- The granularity of the dimensions which determine how many *differentiable encodees* [2], the syntax of a language can generate;
- The effectiveness of different encoding mechanisms in the representation of certain types of domain information.

The question is, how can these limitations be overcome in order to increase expressiveness? The answer has been described in Chapter 6 as the use of abstract *types*. The definitions (and relationships) between these *types* are defined in the representation's syntax in a non-physical meta-language, e.g. a semantic network uses natural language to describe types and type-relations. A result of these types is that domain entities are generally not referred to specifically[3], rather they are referred to as sets, e.g. ball, taxi-driver, to which referents belong. In this case, the specification of the actual referent can only be provided by considering the lexical definition of the token and the context of the message, e.g. 'there is only one ball in the domain, so the sentence must be referring to that'.

The abstract syntax makes no reference to the physical manifestation of a sentence. For example, the natural language type definition, 'noun-phrase=naming an object', and noun-phrase relation, 'noun-phrase= determiner+nominal', does not stipulate whether its glottographic members are spoken loudly or softly, slowly or quickly.

7.2.3.2.1. How Many Encoding dimensions?

If an abstract syntax is used, without any additional physical encoding beyond concatenation (e.g. natural language), then the language only has *one* encoding dimension, i.e. the available encodees can only vary as a result of the syntax, and nothing else. However, the difference between this and a uni-dimensional physical encoding like Mackinlay's single axis is that the former dimension does not represent concepts directly, e.g. by horizontal displacement, rather it encodes by :

- Grammatical classification which is relevant to a particular type of domain, e.g. an*action* is only relevant to a dynamic world;
- Referring to objects by type, i.e. by an object's membership of a set;

---

[2] Well formed sentences of a language which can carry meaning.

[3] As mentioned in Chapter 6, *specificity* can be provided in natural language by the use of proper nouns, e.g. London.

• Using a lexical *dictionary* to define the general meaning of words. This can be in the form of a hidden meta-language or a visible *key legend.*

Other systems may use physical encoding in addition to an abstract syntax. An example described in Chapter 6 is the semantic network. In this system, expressiveness is provided by an abstract syntax tied to a perceptual encoding. In this case, encodees can vary both in their type context, and their physical context. An example of this is shown at the end of this chapter. Other examples would be a natural language system which utilised encoded information in its physical form , e.g. font size, highlighting, position, etc. An implemented example of this is 'temporal text' (Wong, 1996). Here additional domain information is encoded in the dynamic position of text. For example, the word 'speed' would be shown as moving quickly across the screen. Less formal examples are the use of annotations on text such as arrows and caveats.

7.2.3.2.2. Varying Expressiveness

Like the low expressiveness systems, the expressiveness of abstract systems can vary. If the system has only an abstract type encoding mechanism, an increase or decrease in the number of types and type relationships that are present in their syntax will extend expressiveness. This is due to new encodees that such an extensions will make possible.

For example, given a domain with states `(0, 1, 2, a, b, c)`, a language exists with the following type `set:lexeme` pairs (`T`) and *dictionary*entries (`D`):

```
T: {inequality: x, right: g, number: k);
D: {x: greater than, g: less than, k: an integer}.
```

This can represent abstractions of the domain, e.g. the sentence 'k ' will encode all the integer values in the domain. However, if it is necessary to abstract across the whole domain, this will require an increase in expressiveness. This can be provided by the additional type

```
T∪{symbol: alpha};
D∪{alpha: number or letter}.
```

This addition to the language now allows it to express all domain members with the sentence, 'alpha'.

Variations of the expressiveness of abstract systems which use additional physical encodings have the option of using more physical dimensions in addition to the extension of the abstract type set. However, extensions of this type are limited due to reasons outlined in the previous section.

In summary, higher expressiveness representations are able to encode a large subset of all the abstractions over domain states that are possible. The size of these abstractions (in states) will be larger for the typed based expressiveness, rather than the use of physical encoding mechanisms (as described in the previous section).

7.2.3.3. Maximal Expressiveness

The limit of expressiveness will be the definition of abstract types and physical encodings which allow every possible abstraction of the domain to be described. For example, the domain (s1, s2, s3.....s_n), the maximum number of different abstractions is equal to the number of state combinations, i.e. s_1, $s_1 \wedge s_2$, $s_1 \wedge s_2 \ldots \wedge s_n$, $s_1 \wedge s_3$, $s_1 \wedge s_3 \ldots \wedge s_n$, etc.

7.2.3.4. Summary of Expressiveness Levels

The description of expressiveness has shown that it is a measure of the *number of different domain state abstractions that can be encoded.* This notion is visualised in Figure 7.3. Thus, expressiveness moves from the representation of single states, through to the representations of a subset of the abstractions of domain states, and finally to all abstractions of domain states. Exactly which of these abstractions is possible, and implicitly the number of states in these abstractions, is also a function of expressiveness.

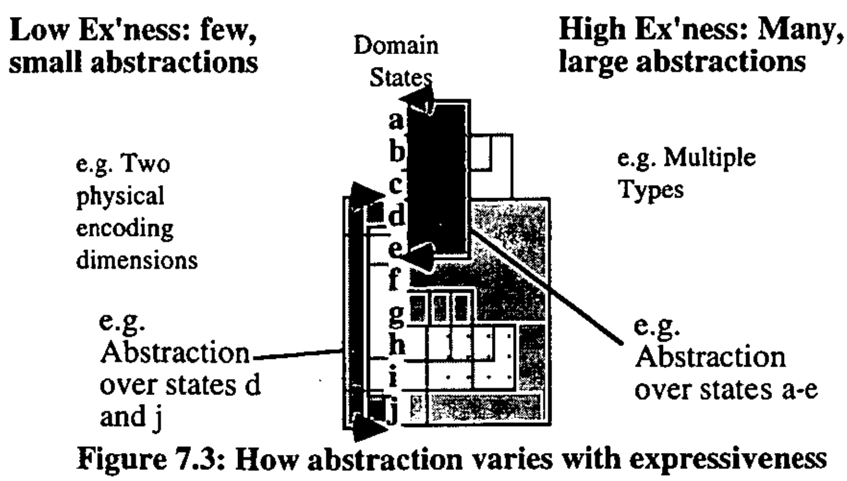


Figure 7.3: How abstraction varies with expressiveness

Increased expressiveness can be either the result of a physically based syntax, an abstract syntax, or a combination of the two. However, extensions to a physically based syntax, by the addition of encoding dimensions, is limited by their perceptual nature.

7.2.4. Why is Media Expressiveness Important?

A number of reasons for identifying expressiveness (the abstractive ability), of media have already been identified in the thesis,

- Expressiveness is a common dimension over which to compare media with a view to deploying them in the user interface;
- An important part of problem solving is viewing the problem domain at the right level of abstraction, the expressiveness of a medium defines the latter property;
- The Expressiveness of a medium has an important affect on a user's mental model of the domain .

It has been suggested in Chapter 4 that representation affects reasoning. This is particularly true in artificial reasoning. Thus, the analogy was made between computers reasoning over knowledge-bases, and users reasoning over interface representations of domains. Given the definition of expressiveness, these studies may offer important indications of when low or high levels of expressiveness are appropriate.

7.2.4.1. Expressiveness and Tractability[4]

If domains are regarded as state spaces through which users navigate, in a way similar to Newell's 'problem spaces' (1958) and Coutaz et al.'s state vector (1995), then two types of domain[5] can be defined:

- Static domains populated by many similar concepts or states, e.g. word processor environment;

- Dynamic domains populated by sample values of many different concepts. The vector of all sampled values is a state, e.g. real-time traffic simulation.

---

[4] In the following discussion, it is assumed that domains are complex enough to allow differences in expressiveness between media to be apparent.

[5] The distinction between these two types of domain is unimportant when one is considering how each domain state will be encoded, but is made here for completeness.

In both cases, the user is described as starting at some initial state, s' ,of the domain and moving to some predefined goal state, s". This is achieved by affecting some action on the problem domain thus causing the domain to change its state. Newell calls these actions *operators*, whilst Coutaz et al. call them *interactions*.

Given this description, the importance of abstraction becomes clear. It allows the effective state space size to be reduced. For example, the number of similar domain concepts which must be examined can be reduced by aggregating them on some common characteristic. A broader issue can now be described, *computational tractability*.

Computational tractability is a mathematical term which describes the number of cases (the same definition as the Euler Circle algorithm in Chapter 4) that must be considered to solve a problem. Though originating in logic, tractability is a useful index of problem complexity. Moreover, tractability is defined by the representation chosen for the problem domain. Two approaches deal with this notion, work by Levesque (1986, 1988) and Brachman and Levesque (1984). These will now be described in more detail.

7.2.4.1.1. Vividness and Reasoning

Levesque (1988) suggests that it is important to consider the *computational complexity of logic* in artificial knowledge-bases. He argues that the incorrect representation of facts can make certain kinds of reasoning, over this knowledge, intractable. Thus, Levesque argues for a restricted form of predicate logic which will allow tractable reasoning to take place. Sentences of this restricted representational language are termed *vivid*. As Levesque (1986) describes:

"vivid information...is the kind of information we think of as the ultimate answers to questions as opposed to the information requiring further calculation and reasoning.", (Levesque, 1986: pp. 97).

More formally, for information to be in *vivid* form it must have the following properties:

1. *A collection of ground, function-free atomic sentences of predicate calculus*: Sentences with predicates that state *fundamental* relations/facts about a domain;
2. *Inequalities between all constant names*: every constant has a unique name;

3. *Sentences are universally qualified*: A predicate p(x) is defined for **all** x.
4. *Closed-world assumptions are expressed*: Those statements made are the only ones that exist in the defined domain, e.g. `sister (bob, jan)` is interpreted as "There is a no more appropriate `sister` of`bob`, than `jan`."

Levesque argues that the key factor of this type of information is that it is not incomplete in the same way as full predicate calculus. By *incompleteness* he means that things can be left unsaid about the world. Consequently, information can be implied from what a representation does not say. If *valid* (i.e. consistent and complete) solutions are to be presented for reasoning over this kind of information, then all *cases* must be considered, both *said* and *unsaid*. For example, the first-order predicate calculus statement:

¬ `In (block, box)`   (1)

States the case that, `block` is not `In box`, *but it also does not state where it is.* Thus, the statement only describes part of the situation. Levesque shows that the implication of this type of knowledge on *tractable* reasoning is dramatic:

"The more that is left unsaid, the more possibilities are allowed by what is said...the problem is that the cases do not simply add up, they multiply: with n-independent binary choices, there are 2^n cases to consider." (Levesque, 1988: pp. 371-72).

For example, if (1) is extended to the first line of:

```
¬ In (block1, box) ∨ ¬ In (block2, box)    (2, a)
¬ In (block1, box) ∧   In (block2, box)    (b)
  In (block1, box) ∧ ¬ In (block2, box)    (c)
  In (block1, box) ∧   In (block2, box)    (d)
```

For (2) above, n=2, therefore 2^2=4 cases must be considered with each case now having two terms, i.e. *(block1 not inside the box and block2 not inside the box (a))* and *(block1 not inside the box and block2 inside the box (b))*and *(block1 inside the box and the block2 not inside the box (c))*and *(block1 inside the box and the block2 inside the box (d))*.

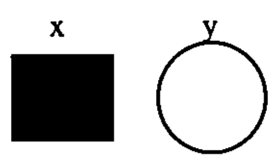
Levesque argues that vivid knowledge overcomes this problem by sacrificing some of its expressiveness, i.e. the ability to represent incomplete knowledge. Thus, if (2) were represented vividly, the original disjunction and negation functions would not be valid, only the atomic statements of the initial case:

```
¬ In (block1, box)
¬ In (block2, box)
```

In addition, rule (4) of the vividness definition (Closed-world assumptions are expressed) would remove the need to consider any additional cases, i.e. where is `block1`, if it is not in `box`? Thus, the vivid knowledge is a series of fundamental statements about a domain.

7.2.4.1.1.1. Knowledge Representation and Domain Representation

Vivid knowledge is effectively a database of domain facts (or n-ary relations). Moreover, Levesque argues that the syntactic structure of this database reflects the structure of the domain. Thus, objects in the domain which are related by some characteristic, e.g. position, will be similarly related by an appropriate predicate in the vivid knowledge-base. For example, a domain described by the diagram,



will have a vivid knowledge representation which is an *analogue* of the relationships between the objects, i.e. the database of facts,

```
{square(x), circle(y), left_of(x,y), right_of(y,x), etc. },
```

on which reasoning could be carried out, e.g. where is the square in relation to the circle? Levesque also suggests that mental structures could exhibit similar properties and allows humans to "reason about the world by operating directly on the (vivid) symbolic structures." (Levesque, 1988: pp. 373), rather than an abstract representation which is not an analogue of the domain. This notion will be addressed in the next chapter.

7.2.4.1.2. Vividness , Expressiveness and Tasks

The notion of vividness has a clear relationship with the notion of expressiveness and its relationship to tasks. It is important to compare the goal of Levesque's study to the goal of expressiveness.

Levesque was interested in making reasoning tractable. This was done by reducing the number of cases of a domain whose consideration was necessary in a solution. Moreover, it made the represented knowledge of a domain complete, i.e. expressing all the fundamental relationships in a domain. In the same way, the right level of expressiveness is seen as a way to represent the domain information in a way which allows tasks to be effectively performed.

However, it is important to note that Levesque is describing the *conceptual* , not the physical representation of knowledge using vivid terms. Conversely, the thesis is concerned with the physical encoding of a domain that has *already* been conceptually defined (for a deeper discussion of this , see Chapter 3). Therefore, whilst the latter still supports reasoning, it does so by representing knowledge in different physical forms, not as different conceptual forms, e.g. atomic sentences, function-laden sentences, existentially qualified sentences, etc.

Given this consideration, vividness can also be seen *as the property of an analogous physical representation* , e.g. a picture, which has a low level of expressiveness. Conversely, incomplete knowledge can be seen as the property of an expressive representation, i.e. one which is not specific in its representation.

Finally, in terms of matching representations to tasks, we can recast the vividness property. Vivid representation is essential, if the fundamental aspects/relationships of a domain are to be known. However, if reasoning requires that there are too many of these to consider (a kind of intractability), then a more expressive representation will be situation.

In conclusion, Levesque's *vividness* illuminates the importance of matching representations to tasks. A task which requires the consideration of fundamental aspects of a domain, but need not consider a large number, should be supported by a vivid or low expressiveness representation. However, if many of these aspects must be considered, then a more expressive representation must be chosen. It also makes a connection between representation of knowledge and effective mental processing. This will be addressed further in the next Chapter.

7.2.4.1.3. The 'Computational Cliff'

The importance of representation on tractable reasoning is also addressed by Brachman and Levesque (1984) who investigated a small reasoning problem in isolation. Their interest lay in how the form of conceptual representations dictates the tractability of reasoning about the domain. They state:

"We address the fundamental problem in the nature of the service to be provided by knowledge representation systems: the greater the expressiveness of the language for representing knowledge, the harder it becomes to compute the needed inference", (Brachman and Levesque, 1984: pp. 34).

As with Levesque's own work described in the previous section, the *expressiveness* of a language is a measure of the number of cases that can be implied from a given sentence.

In Brachman and Levesque's study, two different conceptual representations were chosen to describe the concept of a frame , one simple and one more complex. The notion of a *frame* is defined as an object-like description. Typically, a frame is described by its parent frame and a number of attribute slots which hold atomic values (can be empty or an equality) separated by colons. For example, the frame PERSON describes a person with attributes (number of children, profession of male children, profession of female children):

```
[PERSON                                  (1)
        child (>=1):
        son:          LAWYER
        daughter:     DOCTOR]
```

More complex frames can include sub-frames, for example the PERSON frame can be extended to include a description of a CHILD.

```
[PERSON                                  (2)
        child (>=1):
                [GIRL
                        hobby:      FOOTBALL
                        age         24]
```

These structures can be compacted into a natural language phrase, e.g. Example 2 could be rewritten as, a person who has at least one-child who is a girl whose hobbies are football and is 24 years old.

Brachman and Levesque identify *subsumption* as their example reasoning task that can be carried out over frame-based descriptions. This is defined as:

Type B subsumes type A, if by virtue of the form of A and B, every instance of B must be an instance of A.

For example, in frame 2, every PERSON subsumes GIRL, since every GIRL must be associated with a PERSON.

Brachman and Levesque describe two languages which describe frames. In these languages, concepts refer to a frame name (e.g. PERSON), roles refer to attribute slots (e.g. hobby), and atoms refer to undefined concepts, (e.g. football). Logical connectives are used to describe parts of frame relations, e.g. concepts (frame names or atoms, c), slots (r), restrictions:

AND: x is (AND c_1, c_2...c_n) iff is c_1 and x is c_2..and x is c_n. (1)
ALL: x is an (ALL r c) iff each r of x is a c; (2)
SOME: x is a (SOME r) iff c has at least one r; (3)
RESTR: y is a (RESTR r c) of x iff y is an r of x and y is a c. (4)

Frame examples of each connective are:

- (AND adult male)=> [adult
  [male
- (ALL child doctor)=> [
  child: doctor]
- (SOME child)=> child: >=1
- (RESTR child male)=> *new role*=son

Thus, example (2) can be written in language 2 as:

(AND person
    (SOME child)
    ((AND girl

(ALL hobby football)

(ALL age 24)))

The two languages shown below were compared in terms of the tractability of the proof of subsumption (SUBS?) between two sentences of either language, say d_1 and d_2. The resulting algorithm SUBS?(d_1, d_2) was shown to be tractable for language 1. However, the proof of SUBS?(d_1, d_2) in language 2, with its RESTR operation, is shown to be intractable. This is because the new types (role/filler pairs) defined by RESTR cannot be easily decomposed in order to prove whether they imply d_1. (For the full proof, see the paper).

Language 1: Simple

```
<concept>::=<atom>
          | (AND <concept1>...<conceptn>)
          | (ALL <role><concept>)
          | (SOME <role>)
<role>::=<atom>
```

Language 2: More Expressive

```
<concept>::=<atom>
          | (AND <concept1>...<conceptn>)
          | (ALL <role><concept>)
          | (SOME <role>)
<role>::=<atom>
          | (RESTR <role><concept>)
```

Brachman and Levesque found that determining subsumption with the simple language required a limited number of cases to be considered, i.e. the problem was tractable. However, when the complexity of the language was extended by a limited degree by adding the RESTR connective, reasoning became intractable. This singularity was defined as the 'computational cliff' as shown in Figure 7.4. It was noted that it is difficult to know whether the addition of an operation to a language which currently supports tractable reasoning will move over this 'cliff'. For example, Brachman and Levesque showed that the addition of an AT-LEAST(r, x) connective which replacedSOME (r), allowing the specification of a lower limit of role fillers, did *not* cause the proof of subsumption to become intractable. This results demonstrates how carefully the representation system must be chosen.

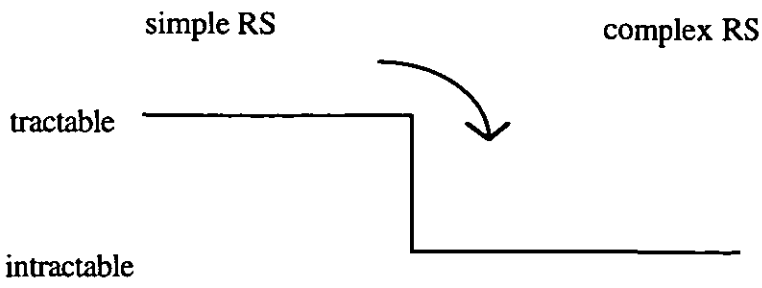


Figure 7.4: Brachman and Levesque's 'computational cliff' (Brachman and Levesque, 1984)

Brachman and Levesque's show how dependent tractable reasoning can be on the representations of knowledge. Though, as with the previous section, the representation is the domain that is reasoned over, rather than considering reasoning over a representation of the domain, such as an interface. This makes the applicability of the study less obvious. However, the representations chosen are both *type* based, and therefore are potentially (given a complex domain and task) highly expressive. Moreover, the addition of the additional type definition (RESTR), is an example of the single encoding dimension of types being made more expressive by the addition of a type to the three already defined. Whether this expressiveness is useful clearly depends on the reasoning that is performed over this new representation.

7.2.4.1.4. Reasoning and Representation

The studies from the artificial intelligence literature have highlighted the following:

- *Vividness* is a property of minimally expressive representations;
- Incomplete knowledge can make reasoning intractable due to the proliferation of *possible* cases.;
- An increase in the expressiveness of a representation may be detrimental to task performance.

7.3. Examples of Expressiveness

Expressiveness, and its relationship with mental models is at the centre of the framework for media allocation. Before describing this relationship, a number of examples of expressiveness will be described. These will demonstrate how the expressiveness of a medium is estimated both from perceptual pragmatics[6] and syntax. It is assumed that the chosen domains are conceptually suitable at the lowest level, i.e. syntactic, to the representing media.

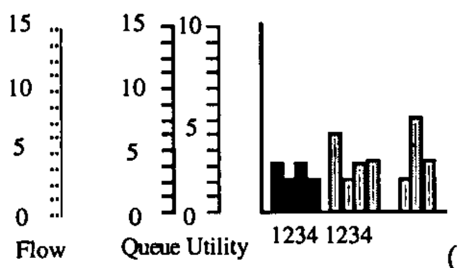
A number of media will be suggested at increasing levels of expressiveness and the reason for their abstractive abilities will be described.

7.3.1. Minimal Expressiveness

Example 1: Bar Chart

Domain=dynamic, four-dimensional space, e.g. traffic characteristics.
Medium=bar Chart
Morphology=bars, axes
Syntax=bars parallel to axes/scales, bars vary in height, not '-'ive
Semantics=one-one mapping of each domain value to *height* of each bar .

e.g.



(1)

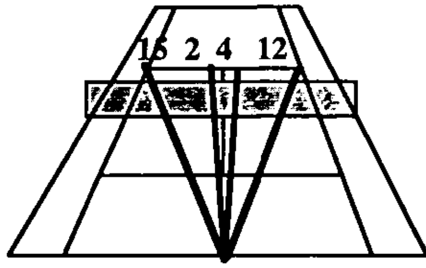
Example 2: Tennis Serve Distribution Graphic (IBM, 1996, over)

Domain=two dimensional space, e.g. numerical service distribution of each game.
Medium=graphical+numerical
Morphology=court, lines, numerals
Syntax=non-negative numerical at end points of lines.
Semantics=one-one mapping of each game service distribution to numerical values distribution. Bold lines show position of serve.

---

6 This notion will be described in the next chapter.

e.g.



(2)

Example 3: Word-processor Icons

Domain=enumerated space, e.g Word processor (AWORD) functions

Actions={new_file, open_old_file,save_file,print document, cut_selected_area, copy_selected_area, past_saved_area, undo_last_operation}

Medium=pictorial iconic

Morphology=stylised images with limited pixel resolution

Syntax=icons shown in a row, must be distinct and recognisable as their referent

Semantics=one-one mapping of each action to a picture signifying that action by synedoche or analogy.

e.g.


(3)

Expressiveness Explanation

(1) and (2) can only show **one** domain state, i.e. values of all variables, in each snapshot of the domain. (1) depends on a dimensional mapping, (2) on an abstract mapping, but both are isomorphic. (3) can only represent **one** type of action with each icon.

7.3.2. Medium Expressiveness (see over)

Domain=N-dimensional space, e.g. process control plant data

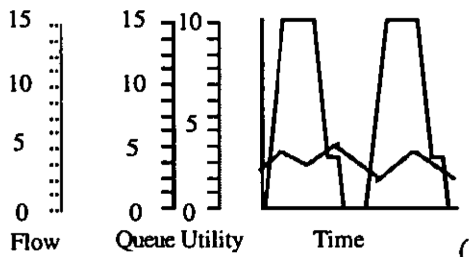
Medium=Apposed axes graph

Morphology=Axes, lines, numerical values, text

Syntax=Each value-pair plotted and joined by lines. Parallel to axes/scales, points vary in height, not '-'ive. Different coloured lines possible.

Semantics=Each domain value-pair is encoded by horizontal and vertical displacement of a point. Value-pairs joined by lines with colour mapped to variable.
Pragmatics=Observe relationships between variables by line patterns.

e.g.

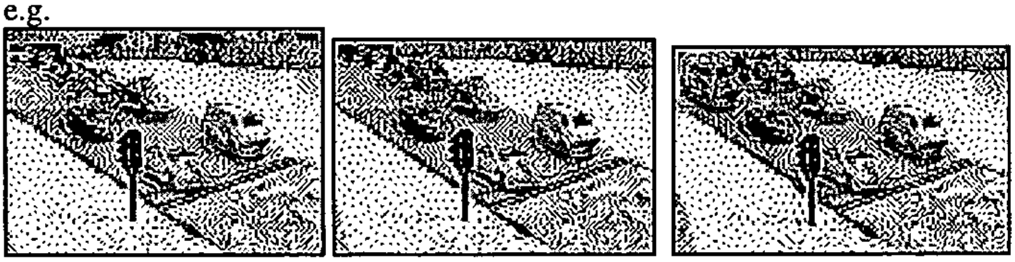


(1)

Example 2: Quantised Still Video

Domain=1-dimensional space, e.g. varying queue length.
Medium=realistic still video
Morphology=video stills
Syntax=single still.
Semantics=each still mapped to a range of domain values.

e.g.



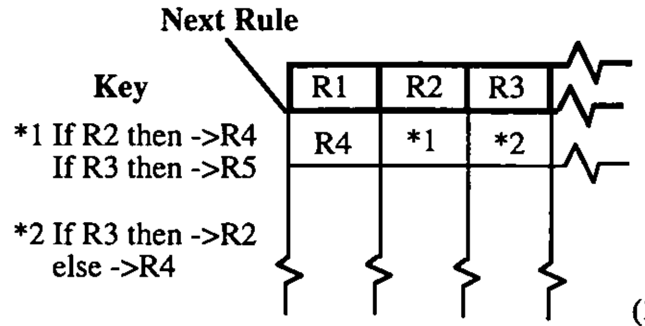
(2)

30<queue length<=50   50<queue length<=70   70<queue length<=90

Example 3: Table with Formula Legend (see over)

Domain=rule base, e.g. expert system
Medium=table
Morphology=text, boxes
Syntax=number or special symbol in each cell. Legend of special symbol meaning outside.
Semantics=each cell denotes the next rule to *fire*.

e.g.



(3)

Expressiveness Explanation

(1) is able to abstract due to its emergent perceptual properties (pragmatics[7]) and the use of the x axis to show time. (2) abstracts by the semantic classification of pictures into range *types*. This shows the expressiveness of a concrete pictorial system being extended by a more complex semantic mapping defined using *types* (e.g. 30<queue length<=50). (3) extends the minimal expressiveness of the table by using a limited key. This allows cell contents to describe more than one next state, as a function of the present state.

The three systems exhibit the two ways expressiveness can be extended. In (1) emergent perceptual properties allowed abstraction, whilst in (2) and (3) an increase in the complexity of the semantic mapping by the introduction of types.

7.3.3. High Expressiveness

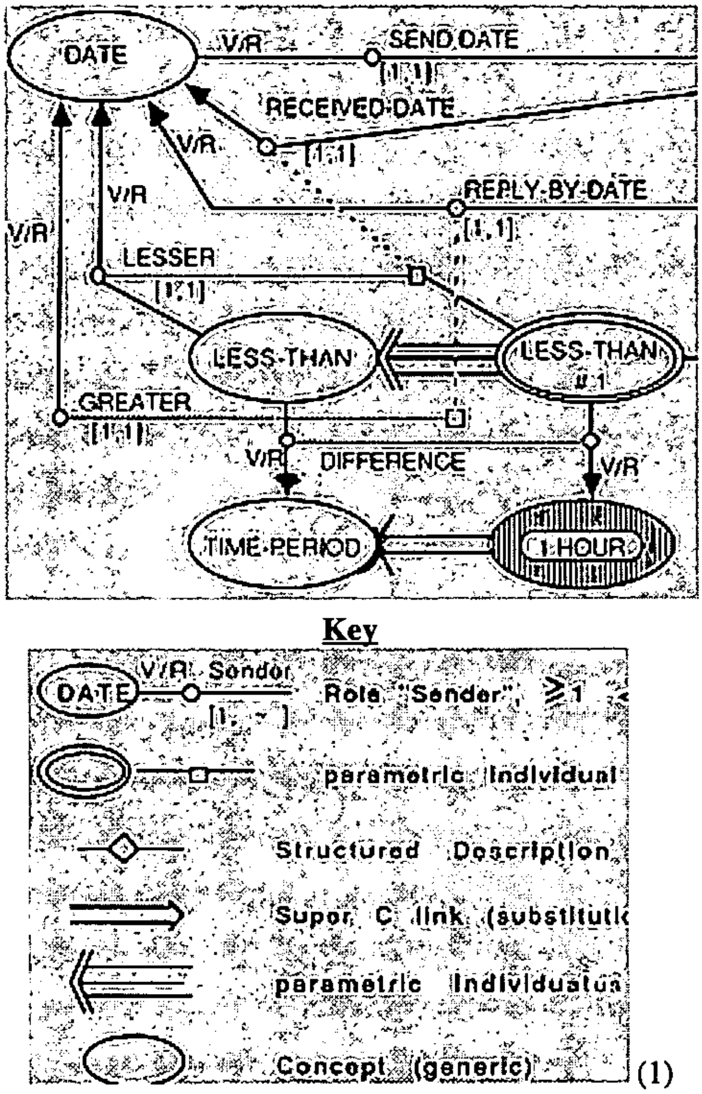
Example 1: Semantic Network

Domain: descriptive, e.g. human memory model
Medium=semantic Network (KL-ONE, Brachman and Schmolze(1985))
Morphology=lines, circles, arrows, text
Syntax=arcs linking nodes, both vary in form
Semantics=symbol meaning denoted by types, e.g. concepts (individual/generic) shown by ovals, relationships (role, description) and description (parameter, value restriction) by arcs. The same symbol has a different meaning in different contexts . Concept nodes can refer to *types* such as 'DATE' or a *numerical comparison* such as

[7] This is a result which emerged from the study of mental models and expressiveness which is described in the next chapter.

'LESS-THAN' depending on the surrounding context. For example, the concept node will take on a *numerical comparison* if it is 'individuated' by a double oval, in this case 'LESS-THAN 1'. Alternatively, an oval is interpreted as a *type* node if it is the 'sink' for a number of messages, as with the 'DATE' concept.

e.g.



(1)

Example 2: Natural Language

Domain= narrative, e.g. Title: "On a Ship"

Medium=natural Language (English)

Morphology=arcs and lines (letters)

Syntax=discrete 'packets' of letters, 'packets' of letter groups, additional separation marks.

Semantics=letter 'packets' are words which have relationships with other words (syntax) and relationships with the domain (syntax and dictionary definitions).

Syntactic structure defined by types, e.g. verb, noun, etc. These affect how words are joined but can also vary depending on word context (as in cat)

e.g.

As the **cat** ran, the captain
whipped the sailor with the **Cat**.
Meanwhile, a fight broke out. (2)

Expressiveness Explained

The expressiveness of both systems is defined by three mechanisms:

- The use of extensive type definitions which allow forms to have meaning beyond their form.
- The contextualising of types membership. Thus, in context c_1, w_1 belongs to type T_2; in context c_2, w_1 belongs to type T_2.
- In the case of (2), the type referent nature of the representation, e.g. man=all men.

7.4. Summary

Expressiveness has been identified as an important index of media selection. Through discussion, a tentative description can be proposed:

The number of abstractions of a domain state-space that a medium can represent

This definition has the following assumptions:

- Domains are state spaces. Tasks are the movement from an initial state to a target state;
- It is a quantitative measure;
- It is dependent on the domain.

A number of examples of media were described which had a range of expressiveness. From these, the following guide is proposed in determining the expressiveness of a representation (these assume a domain with enough states to make the difference in expressiveness obvious).

A medium is of low expressiveness if it:

- Encodes domain states by one or more physical dimension.
- Does not represent many abstractions of the domain.

Is intermediate expressiveness if it:

- Has a limited typed syntax.
- May also use physically encoding dimensions.
- Can represent many abstractions of the domain.

Has highest expressiveness if it:

- Encodes by a typed syntax and physical encoding;
- Can represent all abstractions of the domain.

The next stage of the expressiveness discussion is to examine its relationship with mental models . Specifically, the models induced by different levels of expressiveness, and the effect this has on task performance. This is described in the next chapter.

Chapter 8

Mental Models and Expressiveness

8.1. Introduction

In the previous chapter, the expressiveness of an interface medium was investigated and defined in detail. The property was seen as a product of the specific meaning encoding mechanisms of an interface medium. The definition is shown below:

The number of abstractions of a domain state-space that a medium can represent

This property was presented as a unifying principle which allowed a connection between tasks, media (representation systems) and mental models. The former connection was made after the consideration of the importance of the right level of domain abstraction to successful activity in this domain. Since expressiveness describes the amount of abstraction a medium can offer over domain states, then this property allows the task's abstraction requirements and the medium's expressiveness to be matched together. This satisfies the goal of media allocation in the interface, but without consideration of the cognitive impact of expressiveness.

The relationship of the expressiveness of interface media to the user's mental model of a domain must also be addressed. The thesis had already shown in Chapter 5 the importance of mental models to problem solving, but has not discussed the importance of external representation on this model. Given the definition of expressiveness as a key measure of this representation, this chapter can now investigate the mental models that will be induced by different levels of expressiveness. This discussion is essential since it highlights the cognitive consequences of different levels of expressiveness, in particular, the situation where the expressiveness matches the abstraction required by the task.

8.1.1. The Effect of External Representation of Domains on Mental Models

The importance of the correct mental model on task performance was described in Chapter 4. However, this implicitly assumes that some form of mediating representation is used to convey the task domain, e.g. a user interface. A large part of what the user knows about the domain comes through the interface so as far as the user is concerned the interface *is* the domain. This is true even of the 'superstitious knowledge' which was described by Norman in the previous section. Such knowledge is the result of users filling in the functional gap between cause and effect with guesses based on prior experience. Norman (1986) describes the repeated pressing of the 'clear' button on a pocket calculator (when one press will suffice) as an example of such knowledge. Users hold the superstitious belief that the memory will not be cleared otherwise. Norman points out that this is a function of an incorrect representation.

Clearly, the importance of the interface representation requires more research to be devoted to the effect different *types* of interfaces have on the mental models of domains in terms of expressiveness. Studies of this kind in the HCI literature are limited. Moreover, in this thesis the discussion is limited to output media, rather than interaction 'widgets'[1], which reduces candidate studies still further. However, it is important that some preliminary findings are described to guide research into this area. Firstly, the few studies within the HCI literature will be discussed, followed by a broader survey encompassing cognitive and engineering psychology.

8.1.1.1. Relevant Studies from the HCI Literature

The study most relevant to the consideration of the effect of interface representations on mental models is described by Faraday (1995). The aim of the study is the evaluation of visual training materials, e.g. operating instructions, maintenance manuals. The evaluation shows whether the pictorial/textual instructions convey all the knowledge required to perform the main goal. In other words, does the representation allow the task to be performed effectively? This approach can be described in three parts:

- The *a priori* task and domain knowledge analysis. This takes the form of a goal hierarchy of a main goal and sub-goals. Sub-goals are described in terms of preconditions, task actions, and post conditions. The task actions are carried out by *actions* on *objects* and the pre and post conditions are states of the domain. An example is shown in Figure 8.1 which describes the goal-'change toner cartridge' of a laser printer. In addition to these stages of the goal, other operators describe the objects and actions in more detail, e.g. the '*button*' object has

[1] Buttons, fields, scroll bars, etc.

descriptive (form and name) and spatial (position) attributes. The full description shows all of the domain knowledge which an operator must have to reach the goal;

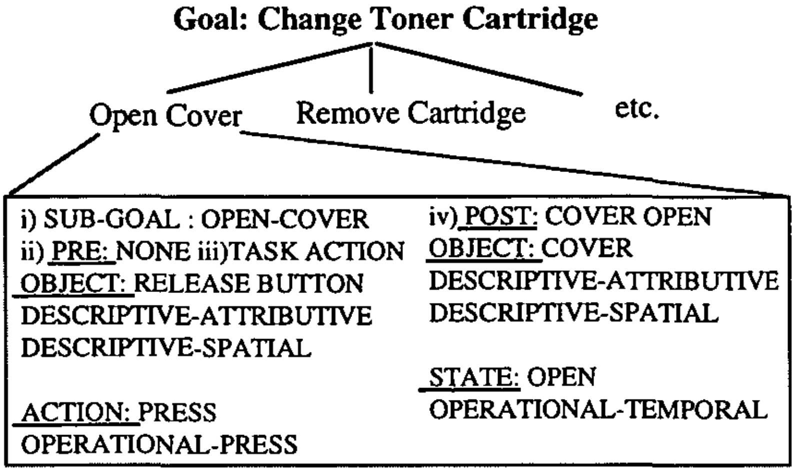


Figure 8.1: Task and domain knowledge description for 'Goal: Change Toner Cartridge'. (Faraday, 1995).

• The second stage represents a breakdown of the training material under evaluation. This involves the segmentation of the training material into its component parts, i.e. text, action symbols[2], and realistic images. Each part is then analysed in terms of the type of mental model it induces. A number of model types are suggested. These include a *relational* model showing an object with its properties and a *spatial* model showing the spatial relationship between a group of objects. This structure is described in a tree hierarchy as in Figure 8.2;

• The model description is then analysed to see if it meets the information requirements of the goal. For example, the representation evaluated in Figure 8.2 will not induce the mental model which will allow the task to be performed properly. This is due to the post-condition, 'COVER-OPEN' not being explicitly represented and therefore not inducing a spatial mental model of this state. To remedy this fault, an additional image is required after this one, showing the cover in its open position.

2 Arrows

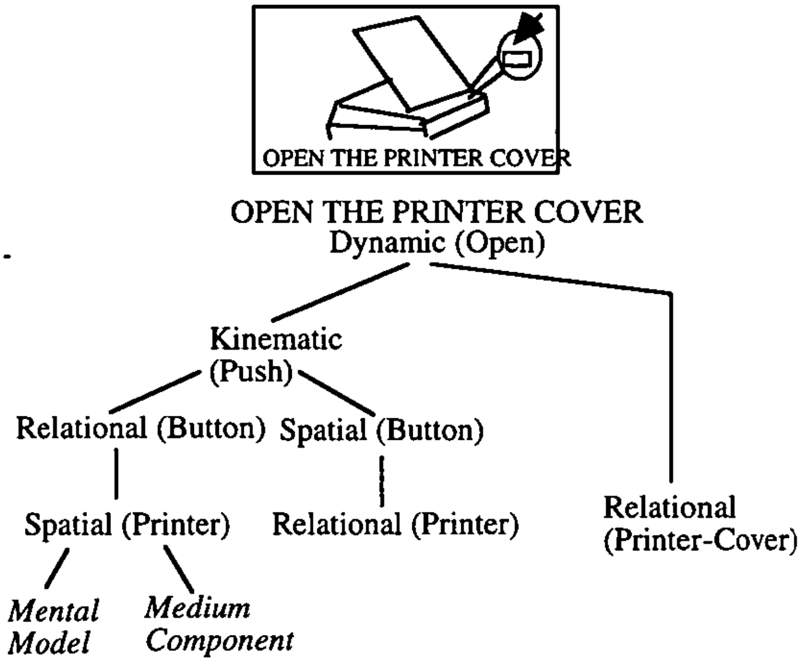


Figure 8.2: Faraday's (1995) mental models induced by training materials

This study makes a number of points which are relevant to the mental models discussion. Firstly, the correct mental model of the represented task is seen as essential to effective task performance. Secondly, the nature of the mental modal (i.e. relational, spatial, etc.) depends on the external representation of the domain concepts. Thus, the *correct* choice of representation causes the right model which in turns allows effective performance of the task.

Unfortunately, Faraday fails to offer any experimental evidence of the *presentation form->model* pairs he describes. Neither, does he address whether the encoding differences between different representations (e.g. linguistic, pictorial), affect the type of model that is induced. Also, there is no attempt made to generalise these results into the broader HCI context of user interface design. Finally, there is no description as to why a particular model type is useful to the task, thus the essential link between model and performance is not considered.

Mayes et al. (1988) investigated what they called the 'information flow' between visual interfaces[3] and users. This described how much of the information displayed in an interface was remembered by subjects. They found the subjects' memory of the visual information in an interface to be very poor. O'Malley and Draper (1992)

[3] MacWrite for the Apple Macintosh

argued this was due to users relying on interfaces as external memories. So rather than taking explicit information from the interface, they rely the interface to cue them into specific actions. Consequently, mental models are represented as 'knowledge-in-pieces' ; a distributed model between the user and system. This would explain why subjects retained so little explicit information. Whilst both studies admit the connection between external and internal representations, neither offer any other comparison interfaces which incorporate different representational media, e.g. command line interface vs. a desktop environment (icons, text).

Payne (1992) investigates the mental models induced by the artefacts with which users engage in problem solving, e.g. calculators, screw drivers. He also considers those artefacts which have some reflexive internal structure[4], i.e. computer systems. Payne argues that users construct mental structures which:

"explicitly represents the relationship between the artefact (interface) and some represented world (task domain)." (author's brackets)., (Payne, 1992: pp 115)

Payne stresses this model will be dependent upon how the artefact represents its internal state, but does not commit to a defined relation between the two.

Work by Borgman (1986) investigates how different forms of training material affect how well a user gains insight into common system operations. The domain investigated was database searching and the two methods chosen were:

- A tutorial describing searching terminology with worked examples using only text. ('Procedural group');
- An interactive Hypercard tutorial based on a card metaphor using text, images, and animation. (' Model group').

Whilst the study was concerned with selecting a suitable metaphor for training, the visual representation of this metaphor was an important, albeit unnamed, variable. The study concluded that a metaphoric, analogous representation which showed the querying and searching operations as the selection of appropriate card induced more complete knowledge. This can be regarded as affirmation of the use of a semantically similar and well known domain (index cards) being used as to induce a mental template which allows understanding of an unfamiliar domain (searching techniques).

[4] Self knowledge.

8.1.1.2. Relevant Studies from the non-HCI Literature

We have already identified the correct mental model of a domain to be essential in successful interaction, and made reference to the effect of representation on mental models[5]. Bergan (1995) makes a similar reference to this relationship in his study of process control panel design:

"Although abstraction in mental models cannot be equated with abstraction in representation..it is likely that an exploration of the relationship between the two can be useful.", (Bergan, 1995: pp 80, 1).

Though Bergan's conception of a mental model owes a great deal to research in cognitive task analysis (Grant and Mayes, 1991), being highly specific to process control tasks, he is at least aware of some connection between external and internal representation.

As mentioned in Chapter 5, the work of Greeno (1983) showed the importance of the correct domain representation on the construction of effective mental representations for reasoning. Greeno's notion of the conceptual *ontology* (mental model) is seen as a function of the domain concepts which are encoded in the representation. Thus, if the representation changes so as to represent different concepts, so the user's mental model will also change. Since the correct mental model is seen as instrumental to effective problem solving, the importance of choosing the correct representation becomes paramount.

Other studies investigate models based on textual descriptions (Egan and Grimes-Farrow, 1982; McGonigle and Chalmers, 1986; Trabasso and Riley, 1975; Sternberg, 1980). However, a closer examination of these studies reveals subtle distinctions within the textual medium. Generally, they differentiated between those textual descriptions which induced certain primitive mental structures and those that did not. The predicted structure was a spatial axis (identified as a spatial *paralogic* which subjects used to mentally order the syllogistic terms, e.g. A->B->C). For example:

All A are B, Some B are C. What is a valid conclusion?

The results showed the ordering of terms in the premises effected subject success in a way which suggested spatial comparisons were used, i.e. if the *pivot term* (B, in the

[5] This will be discussed more fully in Chapter 7.

example) did not occupy its central position in the three terms, then the subjects struggled. It was postulated this was because subjects were unable to easily order the premise variables on a one-dimensional axis. Although these studies were concerned with more fundamental mental activity, they are a clear example of how mental structures affect task performance.

Finally, the issue of the *vividness* of mental representations is addressed by Levesque (1988). Vividness was described in Chapter 7 as a type of knowledge representation which is analogous with the types and structures of the represented domain. Levesque states:

"This is a very powerful device since it allows us to reason directly on the symbolic structures...knowledge about a device is represented in vivid form, and the operation of the device is simulated by analogous operations on this representation.", (Levesque, 1988: pp. 373).

Thus, Levesque argues that the type of mental model that a user holds, i.e. a *vivid* model, has important implications for their interaction with the represented domain.

On the whole, the research into the affect of representation on mental models is limited, particularly in the comparison of different representational forms. However, the literature does demonstrate the usefulness of a cognitive approach in the study of interaction and in particular the importance of effective media selection.

8.1.2. Expressiveness and Mental Models

Given the importance of representation on problem solving, the more specific issue of media expressiveness can now be addressed. The discussion follows the levels of expressiveness that were discussed in the previous chapter, first dealing with the limited expressiveness of concrete media, followed by a discussion of abstract media. At this stage, the key point of expressiveness must be emphasised. As described in the previous chapter, expressiveness is a property of a medium, being used to represent a given domain. This allows the discussion of the mental models induced by different media to address general aspects of the model contents, within a given domain context.

8.1.2.1. Mental Models of Limited Expressiveness

Both Stenning and Oberlander (1995) and Palmer (1978) try to go beyond the discussion of representations in isolation from their human recipient. As Stenning says:

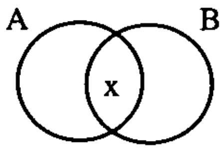
"We aim to explain differences between people's *facility* in reasoning with graphics, with language, and with calculi." (Author's italics), (Stenning and Oberlander, 1995: pp. 126)

Stenning had already provided some evidence of the relationship between cognitive process and external representation in his work with Levy (1988). Here they addressed (as in the 1995 paper), how problem solvers deal with logic syllogisms, with particular reference to the memorising of bindings between individuals and the properties identified in the syllogism premises, e.g. individuals who have the properties 'A and B and C' or '¬A and B and C', etc. The results suggested that bindings were memorised using a low-level, *connectionist* structure which was only able to hold *one* set of premise binding patterns at a time, e.g. individuals implied by the premise 'Some A are B' which gives the individual descriptions, All A are B, Some B are not A, Some are A and not B.

Thus, Stenning and Oberlander suggests that if the memory architecture is itself minimally expressive, i.e. it represents each domain state directly, then those external representations which are congruent with this will be more easily processed. Thus, ideal representations would be minimally expressive; anything higher would cause difficulty in processing, and subsequent reasoning. This is the reason Stenning gives for humans being more adept at using minimally expressive graphical representations. For example, returning to the Euler's Circle representation, abstraction was provided by the description of initial premises using *maximal models* [6] and the marking of *minimal types* (with 'x's, as below). These denote *types* which cannot be empty and dictate empty or *minimal types* after premise pairs are combined. This reduces the number of models that must be considered.

[6] For a fuller discussion of this algorithm see Chapter 4.

e.g. Maximal model of 'Some A are B'.



Given this description, Stenning and Levy's model suggests that

• Although abstracting over possible models implied by a premise, the maximal model is *interpreted* as a minimally expressive representation (due to its diagrammatic nature) and is therefore congruent with the minimally expressive architecture of working memory;

• The topological superposition process of sets during premise combination, i.e. combination of the pivot term with the others, is modelled in the memory architecture by the resetting process which sets up a new model (See Figure 8.3)

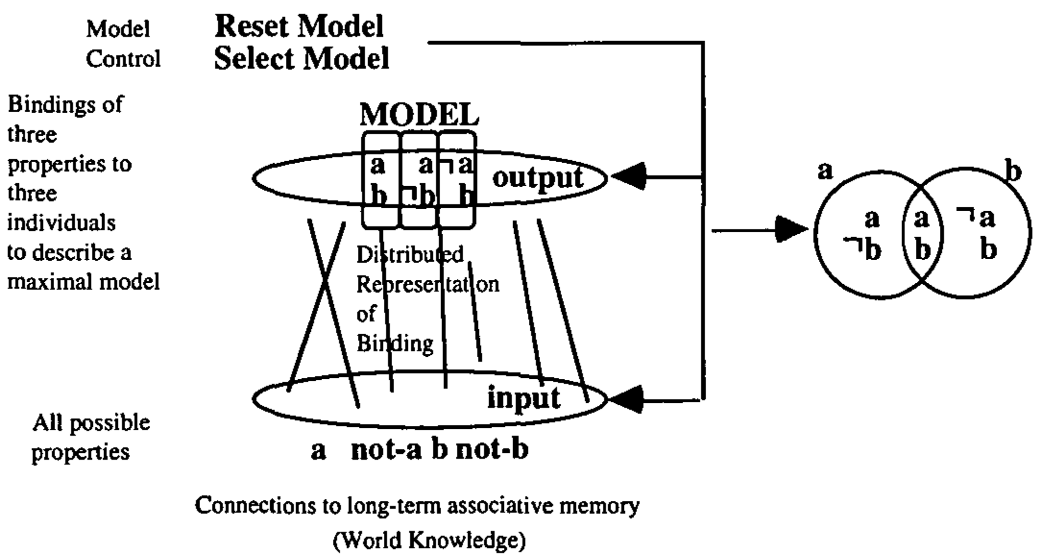


Figure 8.3: Stenning and Levy's Model of Working Memory for the maximal model of 'Some a are b' (Stenning and Levy, 1988)

In this work, Stenning and Levy provide a link between external expressiveness and human cognitive capabilities.

A second study by Shepard (1975) was concerned with imagistic mental representations and described them at two levels. The first level is relevant to the discussion of low levels of expressiveness, the second will be addressed in the next section. The first level was defined as:

- **First-order isomorphism**: " a concept of a mental representation in which the properties of the real-world objects are retained in the internal representation of those objects.". This can be concrete (image of a square) or abstract (four related points).

This mental representation of external images suggests a highly concrete, isomorphic mapping to the external entity. Of course, the external entity is itself minimally expressive (i.e. pictorial), as with the maximal models of the Eulers Circle algorithm. This is a view echoed by Levesque (1988) with his *vivid* mental representations described in the previous section. Thus, the tentative conclusion can be drawn that mental representations of minimally expressive media will themselves be similarly inexpressive.

However, they only address cognitive processes for limited expressiveness representations. What is required is a discussion of the full range of expressiveness that have been identified.

8.1.2.2. Mental Models of Higher Expressiveness

Earlier in this chapter, studies concerned with the use of mental models in problem solving showed a variety of representations being used. Given the thesis discussion of expressiveness, these can be described as covering a range of expressiveness, e.g. graphical/pictorial to natural language. Unfortunately, there is little evidence of the nature of the mental models induced by higher levels of expressiveness. However, some studies have been found which offer important pointers to this phenomenon.

8.1.2.2.1. Studies from Cognitive Psychology

As mentioned earlier, Shepard (1975) was concerned with imagistic mental representations and described them at two levels, the first level has already been addressed, the second level is of importance to the consideration of higher levels of expressiveness:

- **Second-Order Isomorphism**: Object defined by its relationship with other objects. e.g. *greener than x*, rather than *green.* In this way it is similar to a non-concrete, propositional representation.

The notion of relationships in mental representation introduces some degree of abstraction into the representation, since a move is made from the quantitative properties of the first-order model to the qualitative, relativistic property, of the second-order model. This has already been identified in Chapter 4 as an important aid to tasks which require abstraction over large domain state spaces. This point was also made by Rasmussen (1983) and Moray (1987) who argued that operators of complex systems formed an abstract model of the system over long periods of time. In this model, relationships replaced *stimulus-action* rules allowing operators to deal with unexpected events in the system. The resulting behaviour is described by Moray et al. (1986) as *open-loop* (based on present system state), rather than the more simplistic *closed-loop* behaviour (based on specific rules learned) exhibited by novices. This behaviour, though not directly attributable to expressive representations[7], shows how mental models can be more abstract than those suggested by Stenning and Levy.

8.1.2.2.2. Linguistics Studies

In linguistics, model-based interpretations of language offer a somewhat different view to those of the cognitive psychology literature. Johnson-Laird (1983) and Mani & Johnson-Laird (1987) attribute the interpretation of natural language to the establishing of *concrete* (minimally expressive) mental models.

In these models, any abstractive term in the natural language is encoded by a representative concrete instance in the model. For example, the abstract term 'man' will be encoded with a representative man, say 'Bob' who has the characteristics that the listener associates with a man. Once this model has been formed, any subsequent comprehension questions will be answered based on this model. Concreteness in mental structures was also suggested by Clark (1972) who investigated the knowledge gained from textual explanations as compared to the equivalent pictorial description. He suggested that at a deep cognitive level, pictures are represented in the same way as text, i.e. abstract propositional descriptions (in a similar way to Shepards second order isomorphisms). This is made possible by capturing the salient aspects of the picture in propositional form. This suggests that regardless of the level of expressiveness users would form an abstract representation of what they interpreted.

---

7 In fact Woods (1991), suggests that control panel designers are too tied to the idea of "one value-one indicator displays", i.e. minimally expressive representations.

A further study by Beggs and Paivio (1972) showed that subjects encoded concrete concepts in natural language sentences in a different way to abstract concepts. It was suggested the former used an isomorphic representation (minimally expressive) whilst the later retained an abstract encoding which was closer to the linguistic structure of the statement. Though the mental encodings do not change as a function of the expressiveness of the representation, this is still an interesting result. Clearly, different mental representations result from the type of domain concept that is represented. If the mental encoding supports reasoning, then this highlights the care which must be taken over the conceptual domain representation.

8.1.2.2.3. Comparing Approaches

What separates the two sides of the discussion is the choice of domain. Clark and Johnson-Laird's studies deal with tasks over small domain state-spaces, Stenning et. al., with tasks over large state spaces. Since the importance of higher levels of expressiveness in dealing with complex tasks which require consideration of many domain states has already been highlighted, then this will account for the limited models in Clark and Johnson-Laird's work when dealing with simple domains. In their case, the expressive representation of natural language does not induce an abstract model, since there is so little information to abstract over. This is a clear indication of the importance of the effect of domain complexity on mental representations induced by media, as described in Chapter 7. Consequently, different results may have been found if Johnson-Laird's subject had a more complex domain described in natural language, perhaps with many anaphora. The presence of these unresolved pronouns makes it difficult for a concrete model to be formed until the concepts already identified in the discussion have been matched to them. Stenning (1978) showed that until this is the case, mental models will remain abstractive.

Thus, the studies present a more convincing argument for mental representations of low levels of expressiveness. These should be analogues of the stimuli, representing isomorphic concepts and structures. Consequently, the task support provided by such structures will be limited to simple tasks which do not require abstraction over the domain.

As expressiveness increases, the evidence becomes sparser. Shepard's *second-order isomorphism* suggests that mental models will tend to be more abstract, representing relativistic rather than concrete quantities. These representations will be more suitable for tasks which require the consideration of many domain states since abstraction is inherent in the mental representation. However, the results of Clark and

Johnson-Laird suggest that this may not be the case if a very simple domain is represented.

A number of conclusions can be drawn from this limited discussion:

- Expressiveness has a key effect on mental representations;
  - Low expressiveness media will induce concrete models of the domain;
  - Increasing expressiveness will increase the abstraction of the model.
- The level of abstraction in the mental model must match the requirements of the task. This suggests the possibility of a mismatch between model and task, e.g. the model is too abstract for a simple task, or the model is too simple for a task which requires abstraction.

It is clear that experimental study of models induced by different media in different task domains would allow an investigation of these predictions. First however, it is necessary to examine an interesting anomaly in the relationship between the low expressiveness representations and the mental models that they induce.

8.1.2.3. Extending Expressiveness: Mental Models of Perceptually-based Systems

Cleveland and McGill (1986) showed different graphical systems effected subject performance. Taking performance as an index of the induced mental model, these presentations systems of the same expressiveness (i.e. low) induced different mental models. This seems to contradict the previous discussion. This assumed representations of similar expressiveness would induce similar mental models. If this is so, then the consequent performance of subjects should also be the same, this is clearly not shown by Cleveland and McGill's result. However, a similar contradiction is not evident in research on more expressive systems. Studies of the mental models induced by abstract media (mainly natural language) show that performance depends on word ordering, rather than the choice of abstract symbols and syntax, (Egan and Grimes-Farrow, 1982; Sternberg, 1980; Hayes and Simon, 1983). An explanation for this needs to be found within the expressiveness theory.

8.1.2.3.1. The Effect of Encoding Dimensions

The key lies in the method of encoding in the two systems. This is demonstrated in Figure 8.4 which shows the progress from an interface representation of a domain to the mental model that is induced. Two levels of expressiveness are described. The first is a low expressiveness interface which relies on the perceptual interpretation of physical encodings, or *perceiving*. The second, a higher expressiveness medium which relies on the abstract encoding of the domain using a *typed* syntax (as described in Chapter 6), or *reading*. A similar distinction is made between *perceiving* and *reading* in the effective design of graphical representations (Bertin, 1983) and the design of *object displays* [8](Jacob and Egeth, 1976; Carswell and Wickens, 1987; Buttiegieg, 1989; Coury et al., 1989). Here, good design allowed users to *perceive* relationships in the graphically represented data, thus leaving high-level cognitive resources free for other activities. It has already been postulated in the previous section that these two forms of representation will induce different mental models. The interest lies in the multiple models that the results of Cleveland and McGill's (1986) work would suggest for low expressiveness media (as shown by the multiple arrows in Figure 8.4).

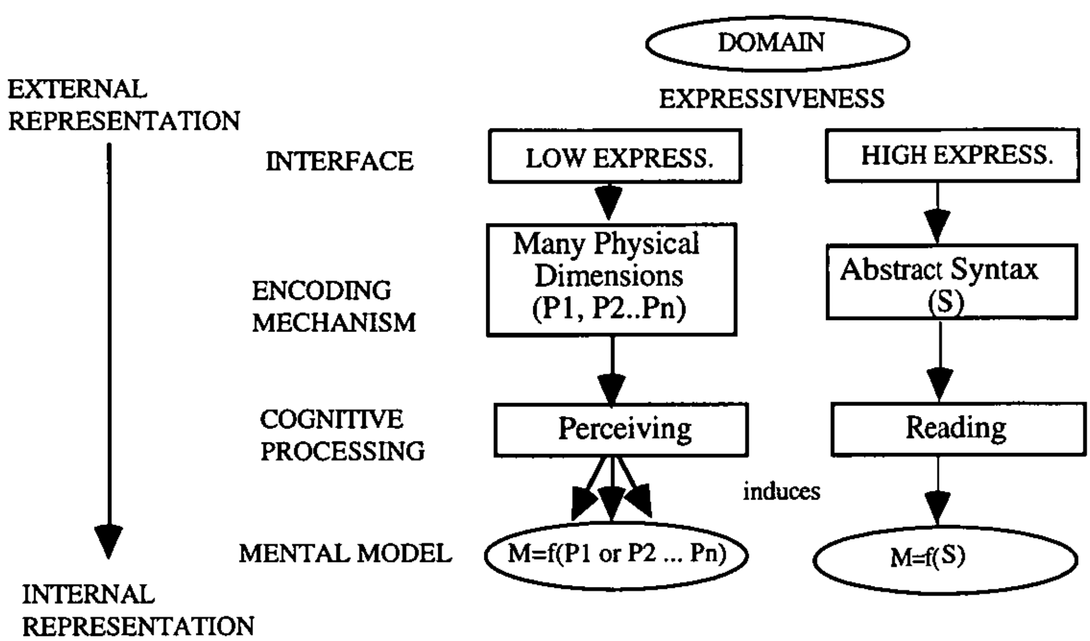


Figure 8.4: From External Representation to the Induced Mental Model

Firstly, in those systems which are *perceived*, there are a wide variety of perceptual encoding mechanisms, e.g. size, shape, colour, density, position, angle. Whilst these may individually only be able to represent one domain concept, i.e. minimally expressive, they will be cognitively processed in different ways. A consequence of this is the variety of mental models, and the observed differences in task performance that Cleveland and McGill (1986) noted. On the other hand, the paucity of physical encoding mechanisms in the abstract systems explains why there are few variations of

[8] Encoding of multiple domain values in closed geometric forms, or separable but proximate forms.

mental model types for experimentation using these systems. This is because models are the result of the same single encoding dimension, i.e. the abstract syntax.

8.1.2.3.2. Extending Expressiveness: Pragmatics

The difference between *perceiving* and *reading* which was drawn out in the previous discussion has important implications for the discussion of expressiveness. Thus far, it was considered that the expressiveness of a medium (resulting directly from the encoding mechanisms and the represented domain) could be identified *a priori,* that is before the medium was used. However, as with the *pragmatics* of natural language which where discussed in Chapter 6, important properties of language may not emerge until the language is in use.

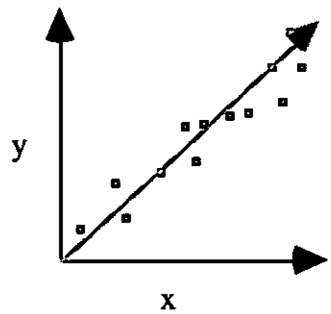
Thus, out of this discussion comes an important observation, the ability of some apparently low expressiveness media to abstract This is due to an abstractive ability which is not inherent in the formal makeup of the representation system, rather it is due to the emergent or global properties of its form. These properties become apparent when the system is in use, so the analogous linguistic term of *pragmatics* is adopted.

For example, a table must be considered at a *global* level. Its syntax would generate minimally expressive, individual entries, which (though linguistic) have a direct mapping to domain values. However, its pragmatics will be the collective effect of *all* of the entries in the table. In doing this, it can be seen that if cells are ordered on some attribute then it is possible to observe *trends* in the data by changes in the appearance of entry contents, e.g. 1-2-4 as shown below. This is an act of abstraction which may reduce the number of system states which need to be considered in problem solving, making the solution more tractable.

e.g. An increase of A with an increase in time

Time	1	2	3	4	5	6	7
A	1	1	1	2	2	4	4

A second example is the pragmatics of a graph (shown over). Although the physical encoding mechanisms of the graph (horizontal and vertical position of lexemes) suggest, *a priori* , that it has low expressiveness , a consideration of pragmatics shows that abstraction is possible. The grey line in the graph represents the relationship *read* by a viewer, in spite of the graph's fundamental encoding limitations. Since the trend is, by definition, an aggregation of the domain data, then more than one domain state is represented by a single instance of the representational system; thus increasing the expressiveness of the medium.



8.1.2.3.2.1. An Explanation of Pragmatics: Reading vs. Perceiving

Pragmatics arise when the perceptual encoding systems are *read* at a global level, rather than perceived at an instance level. Thus, their interpretation is based on abstract types such as *trends*, rather than perceptual characteristics of individuals, such as size. In terms of the discussion of increased expressiveness in Chapter 7, the expressiveness of the medium has been extended by the introduction of a typed syntax.

This increase in expressiveness can be shown on an updated version of Figure 8.4 (Figure 8.5). The movement from a perceptual to abstract system is shown by the left-right arrow. From this diagram, a consequence of this will be the loss of the dependency of the induced mental model on the particular physical encoding mechanism that is used at an instance level, e.g. spatial position.

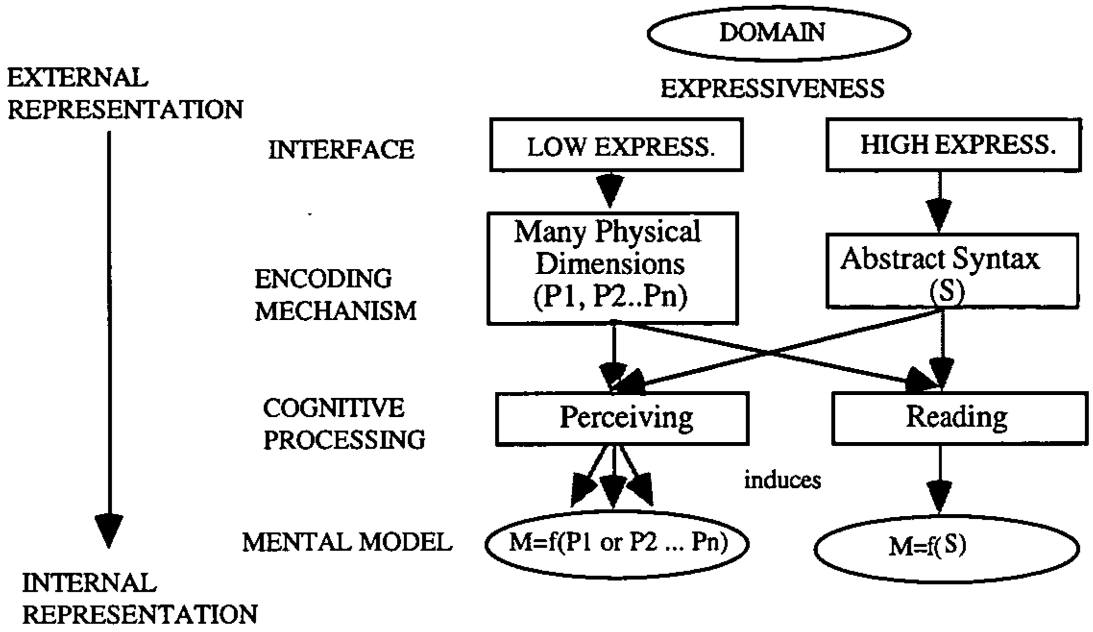


Figure 8.5: Moving between Expressiveness levels

As Figure 8.5 shows, a similar move can be made in the opposite direction (shown by the right-to-left arrow in Figure 8.5), from abstract to perceptual interpretation, with the same consequences for the mental model. An example is shown in Figure 8.6. Here, an expressive abstract/physical encoding (layed out page of text) is interpreted at a physical level (as light and dark areas) to give an overview of its layout. The mental model this will induce will now depend on which physical aspect of the representation was focused on, e.g. light and dark areas, colour, highlighting, etc.

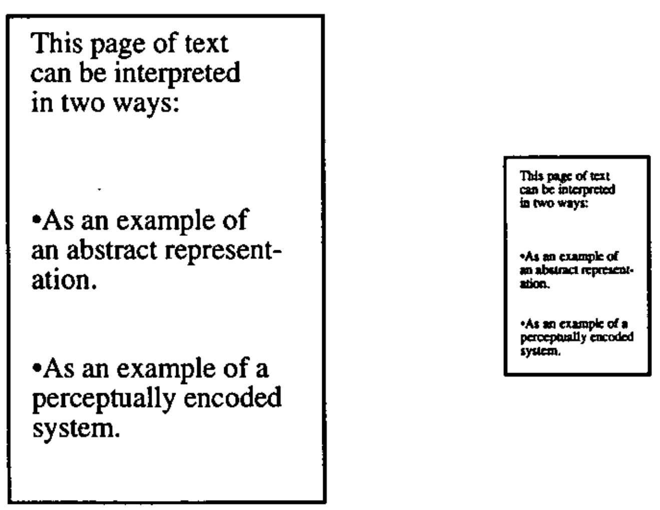


Figure 8.6: Abstract to perceptual dimension encoding

8.1.2.3.2.2. Extending Pragmatics

Perceptual abstraction can be achieved in varying degrees using the same medium. An example is the 'Perspective Wall' (Card et al., 1991) which is used to show general (context) and specific (focus) information about the distribution of objects. An example of documnents on a virtual desktop is shown in Figure 8.7.

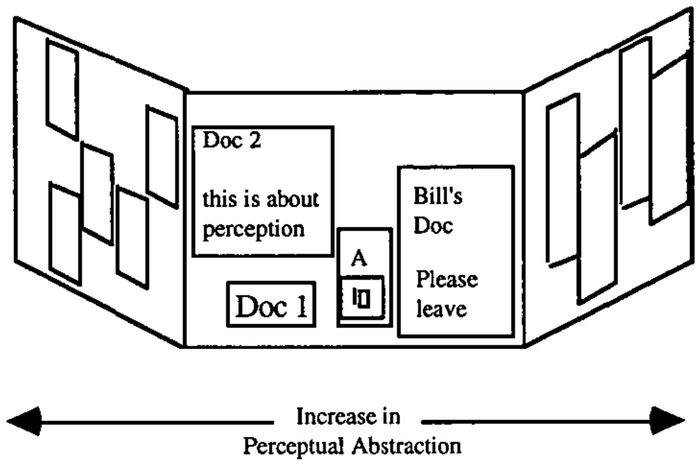


Figure 8.7: Varying perceptual pragmatics with the 'Perspective Wall' (Mackinlay et al., 1991)

Though all three panels allow an increase in expressiveness with perceptual abstraction, the outer panels increase this quality by condensing the horizontal displacement of the documents.

8.1.2.3.2.3. Caveat

Clearly perceptual pragmatics are only of use if the perceived effects have some abstract meaning in the domain, as in the case of the graph. An example where this is not the case is shown in the Hyperbolic Browser of Lamping and Rao (1996). Here, a hyper-link network is visualised in 'hyperbolic space' but the topology of the nodes (layout and distance between nodes) is only due to presentation constraints . Thus, any perceived clusters of nodes which result from a global view of the representation cannot be used to abstract over any meaningful aspect of the domain. Consequently, expressiveness is not extended. This is a further example of the dependency of expressiveness on the encoding mechanisms of a representation, as described in Chapters 6 and 7.

8.1.3. Complete Expressiveness Definition

It is necessary to integrate the discussion of pragmatics into the expressiveness theory. In a similar way, the addition of an abstract syntax or the use of multiple physical encoding dimensions, pragmatics offer an increase in expressiveness. However, there is an important difference between these two methods which is related to their physical characteristics, this is summarised in Figure 8.8.

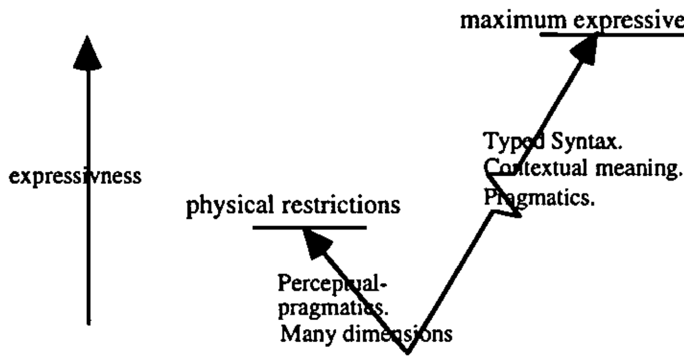


Figure 8.8: Ways to increase expressiveness

Whilst the typed syntax can be increased to any number of terms allowing expressiveness to increase accordingly; pragmatics (such as multiple physical dimensions) are limited by their perceptual nature. They are only able to abstract over sequential states of a domain as shown in the *trend* pragmatics of a graph. However, some clustering of points may be apparent if multiple data sets are encoded on the same axes, i.e. sequences of domain states which exhibit a functional relationship between two or more variables. This is particularly prevalent in the display of dynamic domains which were described in the previous section. However, the limited amount of *real-estate*[9] available to display these media defines the number of perceptual items which can be displayed without causing confusion. This constraint is addressed by Mackinlay (1986) with his *effectiveness* criteria and was addressed in Chapters 6 and 7.

In general, perceptual systems do not have the arbitrary nature of abstract sentential systems which, through the linking of clauses, can abstract over disparate parts of a domain.

[9] e.g. screen space for visual representations, stereo space for auditory representations.

For example, the natural language sentence,

The **sky was bluish** (1) and **roses were given in the usual way** (2) and **he lost his watch** (3),

, refers to *three* entirely entirely different areas of a domain state-space and inherently provides abstraction over these aspects by *type reference*. However, these systems do not have pragmatics which are related to their physical form, rather they come from cultural and intentional aspects of the communication process.

8.1.3.1. Caveat: Expressiveness, Mental Models, and Dynamic Domains

Time can be incorporated into the interface at two different levels, within the task domain or within the interface representations, e.g. animation. In either case, the effect this could have on the representation's expressiveness and the induced mental model must be considered.

The experiments with textual media were concerned with non-computer generated images. Consequently, they were static, representing a static domain. For example, Johnson Laird's (1983) experiments studied fragments of printed text. Computer technology offers the ability to represent dynamic as well as static domains. Static domains are where changes will be brought about in the domain through intervention by the user or by an event extraneous to the domain. Dynamic domains are where domain values change as the process executes. For example, a word-processor is a static domain, and a process control plant is a dynamic one. As mentioned in Chapter 4, typical user interface representations are of static domains and are generally achieved with low expressive media, since each domain state is extant for some time between user inputs and system responses. Consequently, there is generally no requirement for abstraction across these states. Conversely, dynamic domains can be represented in two types of display,

- **Expressive displays**, e.g. the use of natural language, "The temperature is rising";
- **Low expressive displays with history,** i.e. repeated representation of state variables. These displays can utilise the perceptual pragmatics defined earlier.

In both cases, some expressiveness is evident. Thus, mental representations of the dynamic domain should be more abstract, focusing on relationships between states. Within this representation there may be non abstractive information, such as users noting particular system states as important, e.g. 'Temperature over 50C is critical'. Bainbridge (1992) describes such knowledge in her description of operator's mental models in complex and dynamic domains.

Alternatively, static representations of dynamic domains will force the user/viewer to consider *states* of the domain individually. However, it is postulated that these will not constitute the majority of the user's mental model. Of course, relationships could be inferred due to representation changes as a result of interaction, but these will be rare in an overtly static domain.

In the representation of dynamic domains, static media induce state-based mental models whereas dynamic media induce relationship-based or behavioural mental models. It is in the latter case that there is a paucity of research in the HCI literature. Consequently, the discussion focuses more on dynamic media.

8.1.4. Examples of Mental Models for Different Levels of Expressiveness

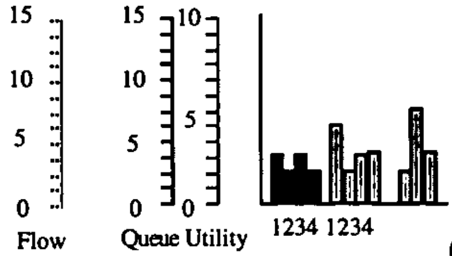
To demonstrate the relationship between the expressiveness of a medium and the mental model it induces, the examples from the previous chapter will now be described. The proposed type of mental model which will be induced by these representations are then discussed. In all cases, the models are defined within a specified domain.

8.1.4.1. Minimal Expressiveness

Example 1: Bar Chart (see over)

Domain=dynamic, four-dimensional space, e.g. traffic characteristics.
Medium=bar Chart
Morphology=bars, axes
Syntax=bars parallel to axes/scales, bars vary in height, not '-'ive
Semantics=one-one mapping of each domain value to *height* of each bar .

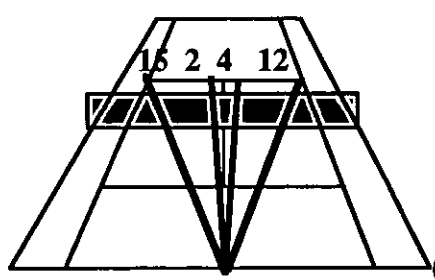
e.g.



(1)

Example 2: Tennis Serve Distribution Graphic (IBM, 1996)

Domain=two dimensional space, e.g. Numerical service distribution of each game.
Medium=graphical+numerals
Morphology=court, lines, numerical
Syntax=non-negative numerical at end points of lines.
Semantics=one-one mapping of each service game service distribution to numerical values distribution. Bold lines show position of serve.
e.g.



(2)

Example 3: Word-processor Icons (see over)

Domain=enumerated space, e.g Word processor (AWORD) functions
`Actions={new_file, open_old_file,save_file,print document, cut_selected_area, copy_selected_area, past_saved_area, undo_last_operation}`
Medium=pictorial Iconic
Morphology=stylised images with limited pixel resolution
Syntax=icons shown in a row, must be distinct and recognisable as referent
Semantics=one-one mapping of each action to a picture signifying that action by synendoche or analogy.

e.g.

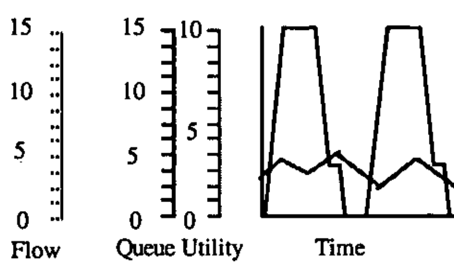
(3)

Predicted Mental Model

Levesque (1986) suggest concrete representation result in concrete knowledge. Consequently, some mental analogue of the domain representations will be the result of concrete representation. The simplistic nature of the model is ideal if the domain is simple (as in (2) and (3)). The drawback of such a mental models is that it is simplistic, providing no abstraction which may be essential if a complex task is to be effectively accomplished.

8.1.4.1.1. Medium Expressiveness

Domain=n-dimensional space, e.g. process control plant data
Medium=apposed axes graph
Morphology=axes, lines, numerical values, text
Syntax=each value-pair plotted and joined by lines. Parallel to axes/scales, points vary in height, not '-'ive. Different coloured lines possible.
Semantics=each domain value-pair is encoded by horizontal and vertical displacement of a point. Value-pairs joined by lines with colour mapped to variable.
Pragmatics=observe relationships between variables by line patterns.
e.g.

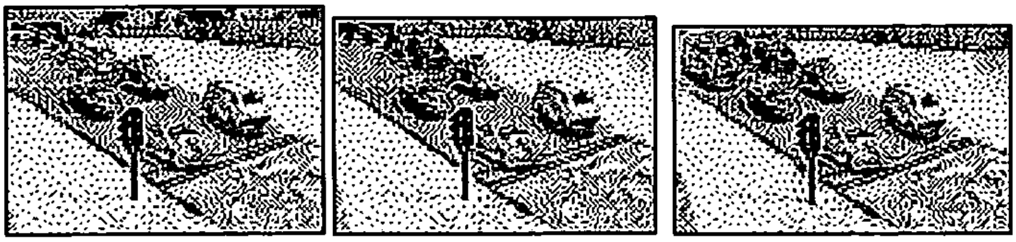


(1)

Example 2: Quantised Still Video (see over)

Domain=1-dimensional space, e.g.
Medium=realistic still video
Morphology=video stills
Syntax=single still.
Semantics=each still mapped to a range of domain values.

e.g.



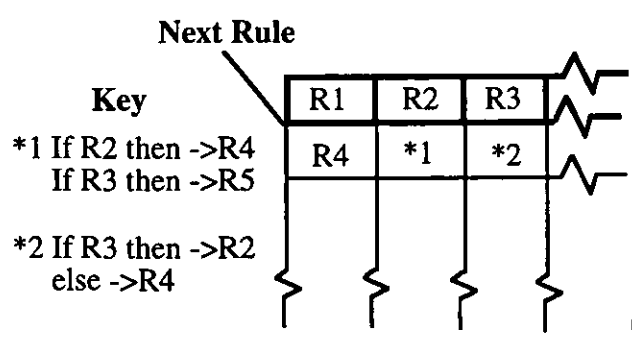
(2)

30<queue length<=50 50<queue length<=70 70<queue length<=90

Example 3: Table with Forumla Key

Domain=rule base, e.g. expert system
Medium=Table
Morphology=text, boxes
Syntax=number or special symbol in each cell. Key to special symbols outside.
Semantics=Each cell denotes the next rule to *fire*.

e.g.



(3)

Predicted Mental Model

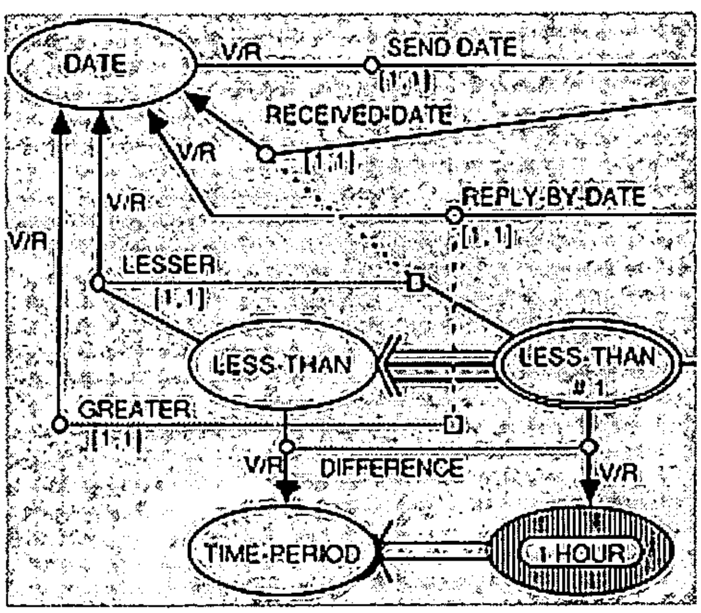
The increase in expressiveness will have a noticable effect on the subject's knowledge. Relationships between domain variables become apparent due to representational abstraction which results in a functional description of the domain, rather than a conceptual one. The move is akin to that described by Rassmussen (1983), where the mental model is no longer based on the concrete representation of signals in the environment, but is made of rules which *fire* as a function of environmental stimulus.

8.1.4.1.2. High Expressiveness

Example 1: Semantic Network

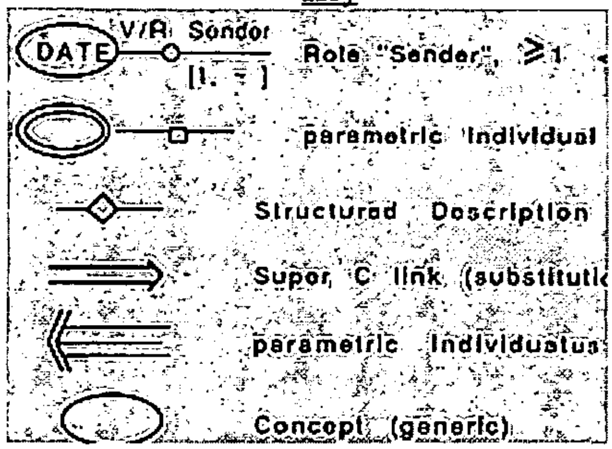
Domain=descriptive, e.g. human memory
Medium=semantic Network (KL-ONE, Brachman and Schmolze(1985))
Morphology=lines, circles, arrows, text
Syntax=arcs linking nodes, both vary in form
Semantics=symbol meaning denoted by types, e.g. concepts (individual/generic) shown by ovals, relationships (role, description) and description (parameter, value restriction) by arcs. The same symbol has different meaning in different contexts . Concept nodes can refer to *types* such as 'DATE' or a *numerical comparison* such as 'LESS-THAN' depending on the surrounding context. For example, the concept node will take on a *numerical comparison* if it is 'individuated' by a double oval, in this case 'LESS-THAN 1'. Alternatively, an oval is interpreted as a *type* node if it is the 'sink' for a number of messages, as with the 'DATE' concept.

e.g.



Key (over)

Key



(1)

Example 2: Natural Language

Domain= narrative, e.g. Title: "On a Ship"
Medium=natural Language (English)
Morphology=arcs and lines (letters)
Syntax=discrete 'packets' of letters, 'packets' of letter groups, additional separation marks.
Semantics=letter 'packets' are words which have relationships with other words (intensional) and relationships with the domain (extensional). Intensional meaning defined by types, e.g. verb, noun, etc. These affect how words are joined but can also vary depending on word context (as in cat)

e.g.

As the **cat** ran, the captain
whipped the sailor with the **Cat.**
Meanwhile, a fight broke out. (2)

Predicted Mental Model

The maximal level of expressiveness provides a complex view of a domain, the mental model will reflect this. Firstly, the basis of the model will be abstract concepts which capture entire aspects of the domain concepts and behaviour. This is suggested by Moray (1987) who describes expert operators breaking a complex system down into independent subsystems or *homomorphs*. As mentioned before, expert user's mental constructs are regarded in the same light as users who have been using an expressive representation. In the case of the expert operator, the abstraction has taken place internally in the model rather than externally in a representation.

The abstract nature of the model allows the domain to be understood if it behaves in an unusual way. This is Rasmussen's 'knowledge based' level of understanding.

8.2. The Match and Mismatch of Task Complexity, Mental Models and Expressiveness

The final consideration of this chapter must be the relationship between tasks, mental models and expressiveness. This is summarised in Figure 8.9. The ideal relationship is for the expressiveness of a medium to be at such a level that it induces a similarly abstract mental model which is what the task requires. Moreover, this match may be at a number of levels of abstraction, from concrete media and mental models to the highly abstractive (Areas 1 and 4 in Figure 8.9). If this match does not occur, then task performance will be effected. Two cases of this can be identified.

- The abstraction required by the task is too low for the abstraction induced by the chosen medium. In this case, the domain may be internalised at too high a level to provide the specificity required by the task. (Area 2 in Figure 8.9)

- The abstraction required by the task is too high for the abstraction induced by the chosen medium. In this case, the domain may be internalised at too low a level to provide the abstraction required by the task. (Area 3 in Figure 8.9)

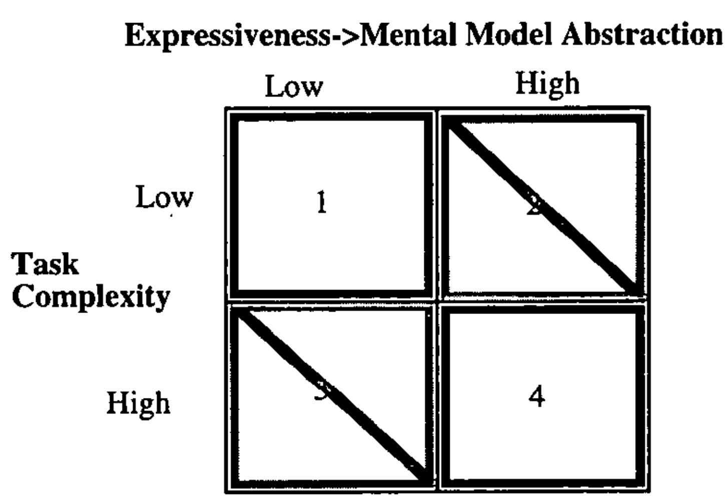


Figure 8.9: Matching mental models to task complexity over a domain

Thus, interface design can be described as ensuring the right level of abstraction is present in the user's mental model in order for them to solve tasks.

8.3. Summary

This chapter has investigated the relationship between the expressiveness of a medium, and the nature of the mental model it will induce. The following conclusions were drawn.

- Expressiveness has a key effect on mental representations;
  - Low expressiveness media will induce concrete models of the domain;
  - Increased expressiveness will increase the abstraction present in the model.
- The abstraction in the mental representation must match the abstraction required by the task, if effective task performance is to be achieved.

In addition to this discussion, the notion of perceptual pragmatics was added to the definition of expressiveness. This describes the abstraction that can be afforded by the emergent properties of the physically-based encoding mechanisms, when multiple instances of the mechanism are viewed at a global level. Thus, media which at first would seem to be minimally expressive are, due to their pragmatics, *able to provide more than an isomorphic description of a domain.*

Now these tentative conclusions have been drawn, it is suggested than an empirical investigation of the relationship between task performance, expressiveness, and mental models will allow validation of these theoretical conclusions. This process is described in Part 3 of this thesis.

Part 3

An Experiment to Investigate Expressiveness, Mental Models and Task Performance

Chapter 9

Developing an Experimental Method

9.1. Introduction

Part 2 of the thesis proposed that the mental models induced by media were the result of the match (or not) between the level of the medium's expressiveness and the level of task complexity. A method for empirically investigating this will now be discussed.

Since the investigation of the mental models induced by different media is unusual, the experimental design must be carefully considered. Thus, this chapter begins with a discussion of possible designs before describing the actual method used.

9.2. Investigating an Experimental Method

An experimental study is required to show the effect on mental models of matching expressiveness to the domain abstraction required for the effective performance of tasks. This can be described in terms of different levels of expressiveness inducing mental models of the domain, of varying abstraction. It is postulated that it is essential that this model matches the abstraction of the domain required by tasks. Thus, the quality of this match will be evident in the amount of correct knowledge subjects hold about conceptual aspects of the domain, and in their consequent task performance. Given that the expressiveness of a medium is a function of its encoding mechanisms and the domain it represents (See Chapter 7), it follows that to allow comparisons between media of varying expressiveness, tasks must be accomplished over a common domain. Therefore, the experimental design must solve the following problems.

- Whether to use a real or a laboratory experimental environment?
- How to learn from other relevant studies?
- How to elucidate the user's mental model?
- How to choose a common task domain?
- How to select representative media?

These will now be discussed in more detail throughout the rest of this chapter.

9.2.1. A Real or Laboratory Environment?

How valid is a laboratory environment in any investigation? A large body of experimental work exists in supporting the design of control panels for complex physical processes, e.g. power plants, chemical production. Sanderson et al. (1989), Verhage (1989), Vicente (1991) all use classical psychological laboratory environments to investigate control panel designs and problem solving strategies. Baker and Marshall (1988) provide a critique of this work and suggest that the control an experimenter has over a subject's environment is detrimental to the fidelity of the results. However, they also noted that the more realistic the environment, the less control the experimenter has which reduces the possibility of a highly focused study with extraneous variables cancelled out.

To counter these criticisms, Alty et al. (1992) and Bergan (1995) in their design of a prototype multimedia monitoring system, describe both empirical and *in situ* methods. This involved an extensive laboratory experiment along with the use of the system in real processing plants. As Bergan points out:

"It is ... important to strive for the highest possible experimental validity in laboratory studies, within the bounds of what is possible, so that the tentative results can inform *further examinations* in simulated settings in the best possible way.", (Author's italics. Bergan, 1995: pp 162).

The results obtained from these studies highlight what can be expected from the two types of environment. The empirical data was the backbone to the study's conclusions and guided the later evaluation of the system in real control rooms.

Unfortunately, the "further examinations" of real environments, identified as important by Bergan, are rarely conclusive, due to time and legal constraints. Consequently, in Alty et al. and Bergan's study, the evaluations in real plants were limited to operator interviews, thus little quantitative data was obtained. However, the analysis of the qualitative data did give an impression of the usability of the interface. Unfortunately, this was not sufficient to evaluate the different output media in the monitoring system.

Returning to pure user interface evaluation, similar problems are observed. Studies are caught between the classical but fundamentally limited results from text-editing environments (Card et al., 1983) and the realistic but anecdotal evidence of *in situ* evaluations. An example of the latter is the use of subjective *user-preferences* in the evaluation of a desktop video application (Fish et al., 1992). Moreover, a number of factors external to the investigation may also dictate the experimental method chosen.

- Is the study concerned with conceptual issues or the implementation of a prototype? The latter is more likely to require higher fidelity evaluation by prospective users;
- The financial resources, time, and person-hours available for the study. For example, Alty et al.'s work was conducted as part of a multi-million pound European Commission project.
- How much access is there to people in real environments? Increased access will allow more results to be obtained, and will also examine users over realistic time-periods, i.e. days.

In conclusion, the psychological paradigm has the advantages that it requires few resources (a machine and a quiet room) and allows the experimental design to cancel out (at least some) unwanted variables. Whilst this isolated environment may be unrealistic with regards to networked systems, this would have provided evidence beyond the focus of study. Thus whilst Lave (1988) argues that results are only valid in a wider ecological context, such speculations are the luxury of the research group, not the lone researcher. Moreover, since the study is investigating an inherently cognitive phenomenon, the cognitivist/psychological experimental paradigm seemed the most appropriate.

Whilst these choices were clearly pragmatic, the main goal remained. To provide the most useful data within the many constraints that were placed on the work.

9.2.2. What can be Learnt from Other Relevant Studies?

As the author has already described, there is a paucity of empirical studies in the HCI literature of the effects of different media on task support, within a common task domain. This gap is more obvious if the mental models induced by different media are cited as a further focus of study. This is surprising considering the growth of multimedia graphical user interfaces which incorporate a range of different representational media. The few studies which have been made have focused on the following areas:

- A quantitative description of the effect of using different media in problem solving (Alty et al., 1992);
- A qualitative description of the effect different media have on knowledge transfer from the task domain to the user's mental model (Mayes et al., 1988).

Unfortunately, none of these studies specifically addresses how media affect of the user's mental model, and how this in turn influences behaviour within a domain. For more comprehensive studies one must look at the number of non computer-based disciplines identified in previous chapters. Here, lie the majority of the investigations which link mental models, representing media, and domain comprehension. Although the majority of these studies have already been described, they are listed here for completeness.

- Mental models constructed during natural language interpretation (Bransford et al., 1972; Craik and Tulving, 1975; Johnson-Laird, 1983);
- Mental representation of spatial descriptions and routes. (Pick and Lockman, 1983; Thorndyke and Goldin, 1983);
- Mental representations constructed during solution of logic problems. (Jones, 1970; Sternberg, 1980; Egan and Grimes-Farrow, 1982; Stenning and Levy, 1988; Stenning, 1995);
- Mental visualisation of algorithms/processes (Douglas et al., 1995).

Almost all of these studies carried out experiments on non-specialists in a laboratory environment. The only exception is Thorndyke and Goldin, who evaluated the spatial skills of army personnel to inform the design of operational contour maps. Moreover, the majority of the studies were concerned with static media such as text and diagrams. This ignores the malleable and transient nature of computer-based representations. By studying more representative computer-based media, the results from these disparate fields can be placed in the context of a realistic user interface environment.

The final point is that the literature described does not address how changes in the task domain will affect the mental models induced by different media. Moreover, if this effect is to be investigated, the domain must provide a wide range of complexities. This will allow matches and mismatches between media, mental models, and task to be analysed. Thus, of particular interest, is an increase in the knowledge required to successfully perform tasks. This is alluded to by Alty et al. (1992) and Bergan (1995) who suggested a trade off exists between task complexity and the support afforded by media. Bergan concludes:

"When difficult tasks need to be performed by users, intelligent application of multimedia presentation can improve their understanding and efficiency in performing these tasks.", (Bergan, 1995: pp. 248, 1).

To summarise, the experimental method must provide the following:

- A rich task domain which is able to present a range of generic user interface tasks, i.e. activities such as deductive and inductive reasoning and consideration of time and space;
- A range of task complexities. The levels of complexity must be sufficient to ensure a noticeable difference in performance between media.
- A range of media which are representative of media found in common user interfaces must be selected, which offer a range of expressiveness;
- An investigation of new media such as video and animation.

9.2.3. A Method for Examining Mental Models

The aim of the study is to examine mental models. Unfortunately, they are inherently unobservable. Therefore a suitable method must be chosen which will allow the *implication* of this structure. Two methods are considered, methods which give quantitative results, and methods which give qualitative results:

9.2.3.1. Quantitative Sources

The use of quantitative performance data has the advantage that immediate numerical analysis is possible. This removes the possibility of errors introduced in post-processing or transcription. However, the data is limited by its recording method. For example, Alty et al. (1992) and Verhage (1989) both used *keystroke* logs, together with task start/stop times. Since this will only provide evidence of the number of actions and the task solution time, these methods can only postulate about what the subject understands about the domain as a function of their performance.

9.2.3.2. Qualitative Sources

If mental models are to be analysed, some introspection is required from subjects during task performance. Examples aof this approach are Bainbridge (1992) and Verhage (1989) However, qualitative data is prone to the following problems identified by Nisbett and Wilson (1977).

- Subjects may only verbalise what they believe the experimenter wants to hear;
- Subjects do not have access to subconscious skills/processes;
- Speaking aloud may interfere with current task activity.

However, this kind of data is the only way to access unobservable mental structures and strategies, so its disadvantages must be overruled. As a precaution, this data can be used along with other sources, e.g. *keystroke* logs, to allow corroboration between different data sources.

9.2.4. Choosing a Domain

9.2.4.1. Generalising the Results

Time and resource constraints limited the investigation of this thesis. Consequently, only one task domain could be implemented. To overcome this limitation, the choice of domain was seen as very important. It was therefore intended to have a wide variety of aspects which must be learnt by subjects in order to perform tasks in the domain effectively, and which are prevalent in common computer-based domains. It is postulated that the knowledge types will appear in the subject's mental model of the domain, in a form dictated by the medium, regardless of the domain portrayed.

Some types of knowledge prevalent in mental representations have already been discussed in the thesis, e.g. spatial and temporal knowledge (Talmy, 1983), relational knowledge (Williams et al., 1983), and abstract and concrete knowledge (Greeno, 1983). A final important category highlighted by Resnick (1996) is *centralised* and *decentralised* knowledge . For example, describing a flock of birds as a coherent, moving mass is an indication of *centralised* understanding. Conversely, considering the individual behaviour of each bird is a *decentralised* conception. These different conceptions of a domain should be apparent in the type of domain knowledge that subjects hold.

Similar categories were also identified by Rauterberg (1995) in his discussion of modelling systems using petri-nets.

- Temporal knowledge;
- Topological knowledge;
- Functional knowledge.

The description of mental models based on these knowledge categories is now described.

9.2.4.2. Describing Mental Models

Since we are interested in the mental model formed during experience with the domain, the categories we have identified can be used as mutually descriptive dimensions of the models. (Green (1991) uses a similar method to classify interfaces with his *'cognitive dimensions'*) For example, a mental model can have a high *functional* content, but with little *temporal* or *spatial* knowledge. For example, a state automata description, where the time interval between state changes is not known.

Verhage (1989) used a similar method in his analysis of the mental models of subjects controlling a real-time system. However, he only identified functional knowledge as the salient dimension of the user's model. This is done by analysing verbal protocol transcriptions to identify any statements made about system relationships. This data was then displayed in graphical form along with a comparison with the ideal relational model of the system. The comparison identified which relationships of the ideal model were identified correctly by the subject. In addition to this first-cut description of the user's knowledge, relationships were divided into two groups, *hard* and *soft*. *Soft* relationships were defined without constraints, whilst *hard* relationships held only within certain operational criteria, e.g. increasing the heater setting increases water temperature <only if the water level is constant or falling>. An example of the representation of a subject's model is shown in Figure 9.1. Relationships between input and output variables are shown by lines, and their direction (proportional, inversely proportional) are shown by + and - signs. *Hard* relationships are bracketed.

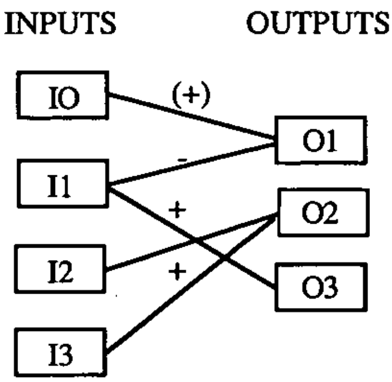


Figure 9.1: An example model from Verhage's (1989) experiment

Verhage's method has two main deficiencies. Firstly, it fails to break down the system description into useful knowledge categories such as those described earlier. Instead the model mostly relies on functional knowledge, possibly because this is what subjects are most likely to verbalise. This was shown by Verhage's results as subjects identified the majority of relationships *without contingencies,* i.e. *soft* rather than *hard.*

Secondly, the quantitative data is kept separate from the qualitative data, when in fact both sets of results are manifestations of the subject's mental model. The functional relationships (and constraints) represent the declarative knowledge that subjects have about system variables. However, this static structure includes knowledge of when relationships hold, and what they are, but it does not provide strategic information on to how to use the information in problem solving. These plans are not consciously accessible (Nisbitt and Wilson, 1977) so the only external evidence available is the quality of the subjects performance, in other words, the quantitative data. By utilising both sources, a fuller picture of the subject's auxiliary and strategic knowledge of the domain is gained. Both of these knowledge types will be affected by the medium used.

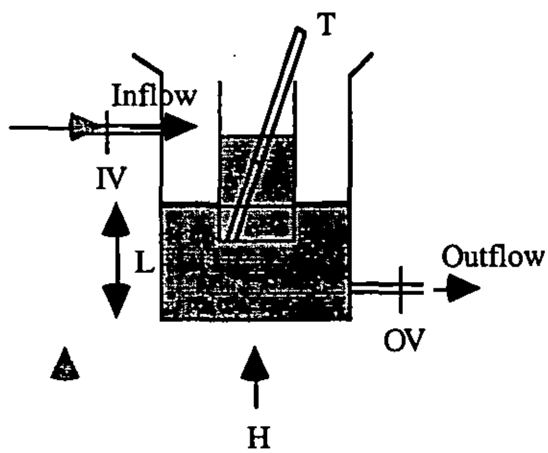
9.2.4.3. The Crossman Waterbath Exemplar

Work by Sanderson et al. (1989), Moray (1986), and Alty et al. (1992) investigate the problem solving activity of operators in complex domains by using a simplified process control task. This task is interesting in that it allowed a wide variety of user behaviour to investigated across a range of task complexities, within a simple system. For this reason, this experimental design is studied to draw out its key characteristics.

These studies are based on the 'Crossman water bath', a deceptively simple task which captures the salient aspects of complex systems.

- Multiple inputs and state variables;
- Simple and complex interactions between state variables;
- Time delay between input changes and system response;
- Continuous behaviour.

The waterbath is filled by an input valve, *(IV)*, which controls the flow of water into the bath (Inflow), from a constant flow, external source. It is then emptied by an output valve, *(OV)*. The water is also heated by a controllable burner, *(H)*, the water level is measured, *(L)*, and the temperature taken by a thermometer, *(T)*, isolated in a smaller waterbath within the main bath. These are all shown in the diagram below.



Subjects are required to reach predefined steady-state values of temperature(*T*), water level (*L*) and outflow (*Outflow*), individually or in combination. The three output variables vary in conceptual and control complexity.

Conceptual *(understanding)*:

- *L* and *T* are simple quantitative measures.
- Outflow is a derived measure of flow over time.

Control *(ability to control within spatial limits)*:

- *L* is directly related to *inflow* and *Outflow*.
- Temperature has a latency associated with it and changes depend on the volume of water being heated (a function of *L*)

- *Outflow* is a measure of pressure at the *OV* which is due to *L*.

The system also has a number of dependencies which, if known, can make control easier. For example, as mentioned, *Outflow* is dependent on *L*. There are also causal relations which are useful to know, such as the need to set *L* before *T*, as *T* depends on *L*, but not vice-versa. To describe these relationships the notion of *incompatibility* was introduced by Sanderson et al. (1989) to give a measure of problem difficulty. *Incompatibility* describes a problem where the target variables are inversely related, so that an increase in one, causes a decrease in the other, e.g. L and *Outflow*

In summary, Crossman's Waterbath exhibits variations in task complexity by:

- Varying the target ranges;
- Having conceptually complex and simple variables;
- Varying the compatibility of target variables.

Since this thesis is aiming to generalise results from the chosen task domain, the water bath was seen as being too specific to the process control domain. Consequently, a more engaging domain was chosen that would provide a sufficiently rich environment which captures the knowledge categories outlined in the previous section. However, the salient aspects of complexity were still present in the this domain as these were considered essential in causing inherent difficulties in system operation.

9.3. Experimental Design: A Traffic Flow Exemplar

The chosen domain was the control of road traffic which had already been used in some HCI studies to good effect (Lewis and Rieman, 1993; Resnick, 1996). Subjects were required to control traffic flow at a number of road junctions using four traffic lights (Numbered 1-4 in Figure 9.2 over). To simplify the system, cars moved at a constant speed (0, val1, val2, etc.) and had instantaneous acceleration and deceleration. This provided a step function for flow rate changes at any point in the system. The topology of the domain is shown in Figure 9.2. Subjects were required to control the phasing of the traffic lights (red and green time in seconds; amber time was fixed). The flow of traffic into either road was controlled by the computer and could either be constant or periodically vary, e.g. val1 for 5s., val2 for 5s., val1 for 5s., etc.

Important points to notice in this system are the dependence of the *downstream* lights 2 and 4 on the *feeder* lights (1 and 2) and the anti-phase relationship lights 2 and 4 at the cross-roads. The state variables of the system are based on realistic street planning criteria (Wohl, 1992).

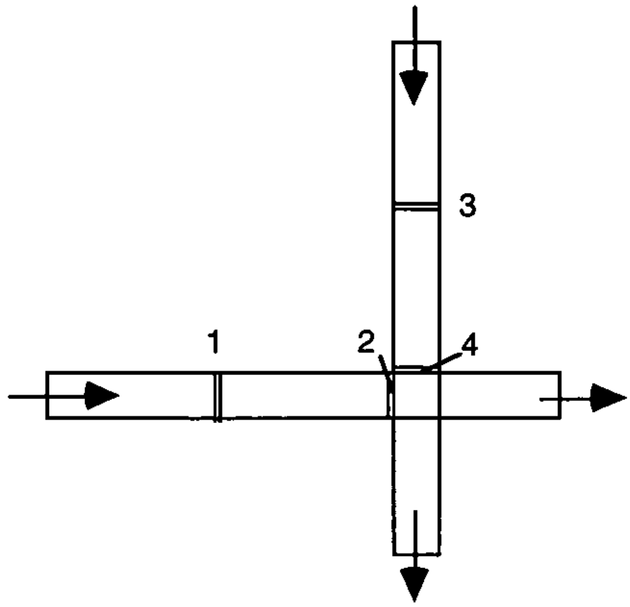


Figure 9.2: Road layout and flow directions for the traffic domain

The state variables of the traffic system are shown in Table 1. The three target variables on which tasks are based are the *summation* of the four *Utility* values (*Utility Total*), the four *Queue Length* values (*Queue Length Total*) and the two *Delay Time* values (*Delay Time Total*).

Like the water bath, the complexity of the problem domain can be varied. This is done by varying the number of target variables which must reach and exceed (positively or negatively) specified limits.

Easy: Exceed target level on one target variable;
Medium: Exceed target levels on two target variables;
Hard: Exceed target levels on three target variables.

Also, like the Waterbath, the system variables differ in *conceptual* and *control* complexity. (For simplicity, in the following discussion *individual* variables will be referred to, rather than the target variables. The definitions are applicable to either). These are shown in Table 9.1.

Variable	Description
Utility	*The extent to which lights are being used efficiently.* A positive integer ratio of the number of cars stored, to the number of cars released, in a red-green cycle. The ideal (and minimum) value is 1. The value is measured at the end of either light's red/green cycle;
Queue Length	The positive number of cars stored (queued) at a light. The value is measured on every system time step.;
Delay Time	The positive time (in seconds) it takes a car to travel along a road, including time queued. Measured at the end of either lights red/green cycle;
Flow Rate	The average number of cars passing through a light, per hour.

Table 9.1: State variable description from traffic flow exemplar

Conceptual:

- **Utility** is a derived ratio of cars queued and released on a light. The value always shows the highest part of the ratio and is never less than the ideal of unity. The new value is calculated at the end of the red/green cycle a light which introduces input/output latency.
- **Delay Time** for a road is a quantitative value defined by time to travel along clear stretches of road added to the time spent in queues. The value is calculated at the end of the red/green cycle of either light which introduces latency between input/output result.
- **Queue length** is a simple quantitative value.

Control:

- **Utility** is a measure of the change in the queue at a light over a red/green cycle.
- Any changes in the phasing time is only registered at the start of the next cycle of that colour, thus introducing input/output latency;
- **Delay Time** on a road is a function of the queue lengths on each light;
- **Queue Length** is a function of relative values of red and green time, and changes according to the rate of flow of traffic reaching a light.

Causal relations are also present. As mentioned, *Delay Time* is directly dependent on the value of *Queue Length*, and *Utility* is directly dependent on the change in *Queue Length*. Thus, since *Utility* is unaffected by the absolute value of *Queue Length*, *Delay Time* and *Queue Length* should be brought within their target bounds first.

As with the water bath, *incompatibility* exists between variables. The main incompatibility is between *Utility* and *Queue Length/Delay Time*, since *Utility* measures the ratio of the cars released to the cars stored. Thus, reducing *Utility* may increase queue length. However, reducing *Utility* may also mean reducing *Queue Length*, (lower *Utility* may mean more cars were released in the previous cycle, than stored, therefore causing a lowering in *Queue Length*).

9.3.1. Analysing the Traffic Domain in Terms of Mental Models

Since the goal of this study is to show how task performance is dependent on the right mental model, to ensure this model, the expressiveness of the medium and the complexity of the task must match. Thus, an analysis of the domain in terms of mental models is required. The system was formally defined in a similar way to Verhage's (1989) functional description of the water bath domain. Thus, along with simple (soft) functional descriptions, e.g. variable X increases as variable Y increases, additional constraints on these relationships are added (hard constraints). Thus, these constrained functional relationships described the system in its entirety, and represented the domain knowledge subjects would need to have in their mental model to effectively control the system. This is called the *ideal model* description and will allow subject models to be compared on a common basis, i.e. as a percentage of the ideal model. This model is shown in Appendix B.

9.3.1.1. Decomposition of Domain Knowledge

To allow different aspects of the subjects mental model to be elucidated, i.e. implied from their knowledge of different domain aspects, a comparison can be made between this model and the ideal system model and a description of optimal performance. The system aspects were grouped into declarative (conceptual) and procedural (operational) knowledge. This will now be discussed in detail.

9.3.1.1.1. Declarative System Knowledge

To allow for elucidation of the learning of different system aspects by subjects and to allow results of this domain to be generalised to other domains, aspects of the domain model were grouped according to the knowledge categories identified in the previous section. To recap, these were:

- Functional, cause and effect relationships (FUNC), e.g. If Queue Length increases, Utility increases.

- Spatial sequence knowledge (SEQ), e.g. cars leaving light 1 will reach light 2.
- Temporal sequence knowledge (TEMP), e.g. cars will arrive at light 2 10 seconds later.
- Decentralised behaviour knowledge (DEC), e.g. *Queue Length* will only increase if cars are joining the queue.

Thus, these represent the declarative knowledge that will be identified from subject verbalisations in order to speculate on the mental model that subjects hold about the domain. Table 9.2 shows example system characteristics belonging to each of these categories. The number of each type of these characteristics in the system model is also shown. Moreover, within these variables, it is possible to identify which categories a subject will identify more easily. Thus, by studying how many system relationships from each category have been identified, a measure is provided of the *depth of knowledge* subjects have obtained about the domain.

These are classified from easiest to hardest as SEQ, TEMP, FUNC, DEC for the following reasons:

- Spatial sequence knowledge (SEQ) and Temporal sequence knowledge (TEMP) will be inherent in any training description users are given about the domain since it is an inherently sequential, real-time process.
- Functional knowledge (FUNC) must be explicitly learnt. However, most relationships are straightforward.
- Decentralised Knowledge (DEC) must again be explicitly learnt, but is not always an obvious part of relationships, e.g. the *Queue Length* will only increase on a light when it is on red, **if there are cars joining the end of the queue**. It is the latter aspect which is not part of the simple description of *Queue Length.*

Along with the knowledge of the above aspects of the system, it will also be important for subjects to understand the conceptual nature of the state and target variables. Like the knowledge categories, these can be ordered in terms of difficulty of understanding. The level of knowledge subjects have about these variables will be a further measure of the depth of system knowledge in their mental model.

These are classified from easiest to hardest as Queue Length, Red/Green Times, Delay Time, Flow Rate, and Utility. The reasons for this are described in the domain description in the previous section.

This concludes the declarative knowledge measures that can be obtained from the experimental design. The procedural knowledge measures must now be considered.

Knowledge Category (**Based on relationships, constraints or both)**	**Example**	**Number in Ideal Model**
Functional (FUNC)	**R1 + D1** (change in light 1 red time will cause will change road 1 delay time	185
Spatial (SEQ)	***fina + F1*** (change in flow into **road-A** will change flow out of light 1)	84
Temporal (TEMP)	**t+10** (10 simulation steps after event at time t)	100
Decentralised (DEC)	**cR1>cG1** (cars stored in red time is greater than cars released in green time)	110

Table 9.2: Example members of knowledge categories

9.3.1.1.2. Procedural System Knowledge

As well as the declarative measures identified, it is also necessary to describe those *procedural* aspects of the domain which must be understood in order to purposefully control target variables. The knowledge of these aspects should be evident in subject performance, rather than in their verbalised knowledge. It should also vary according to a number of system characteristics which effect the complexity of the tasks subjects will attempt. Thus, the level of success in tasks of differing complexity acts as the main indicator of the depth of procedural knowledge a subject possesses, as a result of using a particular interface.

The main complexity characteristic is the gross variation in complexity at the three levels of task complexity. In the *easy* condition subjects will require knowledge of how to control each variable individually. However, in the *medium* and *hard* levels, the control of two or more variables will be required. Encapsulated within these three main levels, more subtle variations in complexity are present. These are a result of the *conceptual* and *control* difficulties identified in the previous section. In brief, *Utility Total* is considered the most conceptually complex target variable, followed by *Delay Time Total* and *Queue Length Total*. In terms of control, it is more difficult to bring *Utility Total* to a lower value, than it is *Queue Length Total* and *Delay Time Total*.

9.3.1.1.3. Task Complexity

Tasks involve the control of the target variables by moving these variables to levels that were higher or lower than initial values. When more than one target variable is part of the task, the direction of target limits from this initial value is varied (this variation allows the description of the different *task-types* defined below). For example, a task at the medium complexity level, could require *Queue Length Total* to be Raised above 50 cars and *Utility Total* to be reduced Below 5 or *Total Queue Length* to be reduced above 10 cars and *Utility Total* to be raised above 15. Clearly, given the variation in control complexity between different variables, the direction of the target level can also affect the task complexity.

Given these factors, Table 9.3 shows the assessment of complexity for the different task *types* (In the table, QT< denotes a target limit below the initial value, etc.). Within each main complexity level (the easiest tasks are shown as the smallest text, with increasing text size showing increasing complexity). These are a combination of the *control* and *conceptual* complexities. For example, the hardest task in the easy level will be one which requires a reduction in *Utility Total*, due to its *conceptual* complexity. One of the two harder tasks in the medium level is 'UT< and QT<', due to the conceptual complexity of *Utility Total*, combined with the control complexity of reducing the*Queue Length Total*.

EASY	MEDIUM	HARD
QT>	QT> and UT<	QT< and UT< and DT>
QT<	UT< and QT<	UT< and QT< and DT<
UT<	DT< and QT<	DT< and QT< and UT>
UT>	QT< and UT>	UT> and QT< and DT>
DT>	DT> and UT<	DT< and UT< and QT>
DT<		

Table 9.3: Task difficulty at each complexity level

It is here that the match between the mental model induced by a medium (of a certain expressiveness) and a task (of a certain complexity) is made clear.

9.3.2. Ensuring a Match between Expressiveness and Task Complexity

Since the thesis is investigating the mental models induced by different media of varying expressiveness, over a range of complexity levels, the least time consuming experimental design would allow a subject to solve questions at all levels of complexity, with one medium. This was the procedure advocated by Sanderson et al. (1989), Vicente (1991), and Alty et al. (1992). Of course, since subjects work through all three levels of complexity during the experimental period, they will be exposed to tasks which the medium is not suitable for, i.e. it has the wrong expressiveness. It is postulated that this *mismatch* will affect the results of the knowledge analysis. For example, too high a level of expressiveness will not allow the specificity in the mental model that is required by an easy task, whereas too low a level of expressiveness will not allow the abstraction required in the mental model by a complex task. The ideal situation is to match these two criteria in order to induce a mental model which will allow effective task performance. This is shown in Figure 9.3.1-3.3. (over)

Here, hypothesised learning curves are shown for three media of increasing expressiveness (e0, e1, e2). Each curve represents a subject's domain knowledge which is learnt over a period of time spent controlling the system. Along with this, a dotted line shows the level of knowledge (declarative and procedural) that is required for a particular task solution. The mismatch between task complexity and media expressiveness is shown by an arrow. Thus, matches and mismatches for e1, e2, and e3 vary according to the level of task complexity.

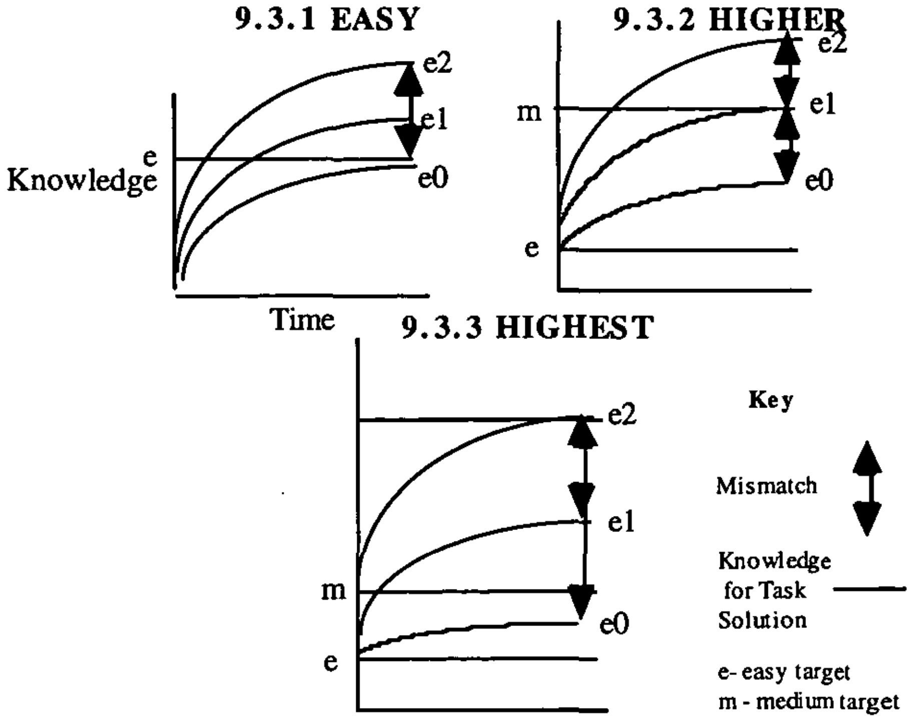


Figure 9.3: Effect of Expressiveness on learning in three task scenarios

9.3.3. Task Complexity in Terms of Expressiveness

Given the discussion of the task complexities which are present in the traffic domain and the dependency of effective task performance on the subject's mental model, the following predictions can be made. These predict the level of understanding of system aspects for subjects using media of differing expressiveness. For example, the ideal state of a high level of understanding, is the result of the match between the expressiveness of the task complexity producing a model which is sufficient to understand this level of complexity. Clearly, by using a consistent medium, the match between expressiveness and task complexity may vary at, and within, each complexity level, either in the declarative to procedural knowledge. However, the complexity of the declarative knowledge remains consistent across the three coarse complexity levels. Table 9.4 brings together the various complexity measures from the previous section (as before, complexity increases with text size).

EASY	MEDIUM	HARD
QT>	QT> and UT<	QT< and UT< and DT>
QT<	UT< and QT<	UT< and QT< and DT<
UT<	DT< and QT<	DT< and QT< and UT>
UT>	QT< and UT>	UT> and QT< and DT>
DT>	DT> and UT<	DT< and UT< and QT>
DT<		

SEQ, TEMP, FUNC, R/G, D, Flow Rate, U, DEC

Table 9.4: Complexity of Declarative and Procedural knowledge

For clarity, the following predictions are accompanied by the relationship between expressiveness and induced mental models that was identified in Chapter 8.

9.3.3.1. Low Expressiveness Medium Predictions

Low expressive media will induce more concrete models of a domain.

Declarative

- Good knowledge of Queue Length (Q) , sequential knowledge (SEQ), temporal knowledge (TEMP) and functional knowledge (FUNC). The understanding of these concepts indicates that the inexpressive medium has induced a mental model of sufficient simplicity to allow this. For a similar reason, these results will be better than a higher expressiveness medium due to the mismatch between low conceptual difficulty and high expressiveness. However, for the remaining more complex categories, the low expressiveness will be insufficient to induce a suitably abstract mental model to allow these concepts to be understood. Thus, knowledge of these will therefore be worse than using a medium of a higher expressiveness.

Procedural

- Good performance for EASY questions; better than more expressive media due to their mismatch with the task complexity. In this case, the good performance is a manifestation of a mental model which has been induced by the low expressiveness medium, and which is congruent with the needs of the task, i.e. little abstraction. However performance will be worse than the more expressiveness media for controlling the more complex Utility (U) and the lowering of other variables. In this case, the simplistic model of the domain induced by the inexpressive medium is insufficient to adequately support the task.

9.3.3.2. Higher Expressiveness Medium Predictions

Higher expressive media will induce more abstract models of a domain.

Declarative:

- Good knowledge of *Delay Time*(D), better than low expressiveness. However, the mismatch between expressiveness and conceptual complexity may cause easier concepts (SEQ, TEMP, FUNC, R/G) to be less well understood. Knowledge will be worse than higher expressiveness for highly complex concepts, Flow *Rate*, U, and DEC.

Procedural

- Good performance for medium questions; better than low expressiveness in all tasks. However, 'UT< and QT<' and 'QT< and UT<' task types may be performed less well than a more expressive medium.

9.3.3.3. Highest Expressiveness Medium

Higher expressive media will induce more abstract models of a domain.

Declarative:

- Good knowledge of the DEC category, but it may be too expressive for other less complex concepts.

Procedural

- Good performance for all task types, though task types 'QT< and UT< and DT>' and 'DT< and UT< and QT>' may be worse due to their complexity.

9.3.4. Choosing Representative Media and Investigating Expressiveness

9.3.4.1. Media in Other Studies

Given the discussion of the effect on mental models of varying expressiveness, a range of media of varying expressiveness is required. The first criterion for chosing of media must be generality. Contrary to this, other studies generated interfaces based on an analysis of the information requirements of the task (Mackinlay, 1986; Casner, 1991) or conceptual descriptions of key domain relationships (Vicente, 1991). These syntax-driven exercises produced highly task-specific media combinations which were unlikely to be usable in another domain. An example of one of Vicente's interfaces is shown in Figure 9.4 and shows graphical representations of multi-variant domain relationships. Thus, whilst incorporating common representations such as textual, positional, and retinal systems, the purpose of the study was that they should be used in a highly task-specific way.

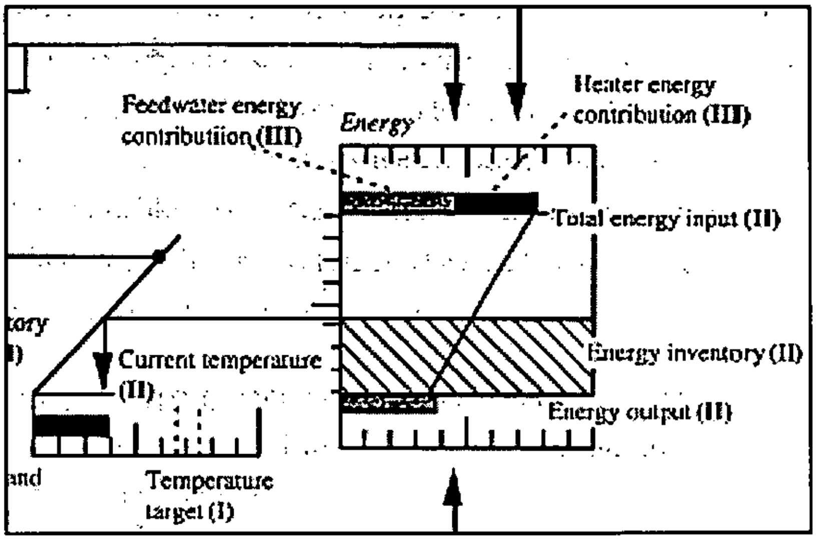


Figure 9.4: An example interface from Vicente (1991)

Alty et al's study (1992) represents the most relevant collection of output media to this study. This included a schematic animation, a table, graph, auditory media and combinations of these. The key difference between the proposed study and Alty et al., is that the latter was investigating media in combination, rather than individually. This makes it difficult to extrapolate whether the use of particular media was a success. However, the study does indicate a range of media that were considered generalisable to other application domains.

9.3.4.2. The Importance of Domain Constraints

As described in Chapters 6 and 7, the domain must be considered in the choice of output media. Mackinlay et al. saw this as the most important part of media allocation. This choice is mainly based on the congruence between syntactic and morphological qualities of the available representations and domain. For example, the traffic domain was highly numerical, with 17 output variables and 8 input variables. Since many variables are important for the control of the system, subjects must be allowed to decide whether certain variables are displayed or not. The importance of numerical values also meant that media which are not able to represent absolute values, such as video or dimensional/pictorial systems, must be ruled out.

9.3.4.3. The Chosen Media

Based on the consideration of expressiveness and the traffic domain characteristics, the media must offer the following:

- A range of expressiveness;
- The ability to represent multiple numerical variables without confusion;
- The ability to represent numerical variables accurately.

The five chosen media are described in Table 9.5 (and over), along with their theoretical expressiveness[1]. The latter is based on the discussions of the representation systems in Chapter 7. The number of variables which were displayed on each medium was controlled from the input panel. A *highest* expressiveness medium was not considered due to time constraints. This is alluded to in the conclusion of the thesis. Screenshots of the media and the input panel are shown in Appendix A.

Medium	**Expressiveness**	**Description**
Animation + Table (As described at the end of Chapter 6).	**Low**	A physical graphical encoding along with absolute numerical values in a separate table. The *flow* of cars is shown by the spacing of blocks, which move in real-time through the road system. Traffic light state is shown by coloured circles. *Delay Times* are encoded by pie segments and all other variables by horizontal bars. Target levels are shown as horizontal bars. This **is the only medium which directly represents the flow of cars *into* either road.**
Static Video + Table (As described at the end of Chapter 6).	**Higher** Due to representation of value ranges (See Chapter 7)	A realistic bitmap representation along with absolute numerical values in a separate table. *Flow rate*, *Queue Length* and traffic light state are shown in temporally sequential images, for each road. Target variables are shown by aerial views of the road for *queue length* and *utility*, and delay by a clock face.

[1] Due to time constraints on programming, all of the media were visual.

Dynamic Table (As described at the end of Chapter 6).	**Higher** Due to *perceptual pragmatics* (See Chapter 8)	A numerical representation with a five value history. All system variables are shown numerically. Variables are grouped by type, e.g. all *Flow Rate* values together. Target limits are also shown on either side of the target values. **The states of lights are not represented.**
Dynamic Graph	**Higher** Due to *perceptual pragmatics* (See Chapter 8)	A graphical, apposed axes representation of all system variables. All domain values of multiple time-steps are shown by continuous lines on a dynamic graph; 50 values can be represented. When points reach the right hand side of the graph the graph scrolls from right to left. Variable traces are colour-coded to match appropriate numerical scales. Target limits are shown by horizontal lines on the main graph. **The states of lights are not represented.**
Dynamic Bar Chart	**Low**	A graphical, apposed axes representation of all system variables. Values are shown by vertical bars which are updated on each simulation step. Variable bars are colour-coded to match the appropriate scale. Target limits were shown by horizontal lines. **The states of lights are not represented.**

Table 9.5: The experimental media

9.4. Summary

This chapter has described the development of an experimental design to investigate expressiveness, mental models and task performance. The investigation led to the consideration of dynamic domains which allow a wide range of task complexity to be modelled. The chosen traffic-flow domain was based on Crossman's Waterbath and exhibits the following characteristics:

- Real-time behaviour;
- Multiple state and target variables;
- A wide range of complex and simple variables and relationships.

Subjects are required to solve tasks in this domain by controlling target variables in order to exceed specified limits. This is done using a specific output medium. The knowledge subjects exhibited about the domain's structure and behaviour, is recorded. This is then taken as an indication of the mental model that had developed as a result of the relationship between the expressiveness of the medium, and complexity of the particular task. This model can be further elucidated by separating declarative and procedural knowledge. The former is assumed to be knowledge of domain concepts, the latter knowledge of how to control the domain. This dissection allows two results sources to be utilised. Thus, subject verbalisations can define declarative knowledge, and performance data can define procedural knowledge.

The notion of complexity is seen as central to the domain, thus a need for a wide range of complexities was identified. The main purpose of this was to allow as many indications of the match (or mismatch) between the mental model induced by different levels of expressiveness and the task complexity; as exhibited by the domain knowledge that the subjects exhibited. Thus, as well as three coarse levels of complexity (targets were to be reached on one, two or three target variables), complexity was also defined on declarative aspects (e.g. the conceptual complexity of variables) and procedural aspects (e.g. the difficulty in controlling certain variables) of the domain. The next chapter will describe the full experimental used method and the results obtained.

Chapter 10

Experimental Method and Results

10.1. Introduction

This previous chapter described the development of an experimental design to make an exploratory investigation of expressiveness, mental models and task performance. This chapter will now describe the actual method in detail, along with the results.

10.2. Experimental Description

10.2.1. Materials

The experiment was conducted using a 80486 PC with 16MB RAM and keyboard/mouse input. Auditory output was played through a small pair of loudspeakers. The traffic simulation and results analysis software were written in Microsoft's Visual C++ interface prototyping environment[1]. Subjects' verbalisations were recorded on audio cassettes using a microphone attached to a tape recorder.

10.2.2. Subjects

Due to time limitations, subjects were taken from the undergraduate and postgraduate population, with degrees in Computer Science, Psychology, Economics, and Manufacturing Engineering. They were aged 18 to 40. Of the 34 subjects, 26 were male and 8 were female.

10.2.3. Method

The experimental method was outlined in Williams (1996) and the design is shown in Table 10.1. Subjects were divided randomly, with the 8 female subjects divided equally across the five experimental conditions. The conditions were the five experimental media which corresponded to two levels of expressiveness. Each subject solved as many questions as possible within each of the three complexity

[1] A description of Visual C++ is available in Appendix B.

levels . The entire experimental period was split into two sessions (morning and afternoon) described in Table 10.2. The simulation notes described the layout of the road network and the meanings of system variables. The training period familiarised subjects with the input mechanism and in reading the output medium. Training questions allowed subjects to solve questions from a typed sheet based on a single road, two light, constant traffic input system. (The simulation notes and training questions are listed in Appendix A)

		Medium				
		Animation	S. Video	Table	Graph	Bar Chart
	Easy	s1, s8, s19, s21, s24, s30, s31	s5, s10, s14, s17, s22, s27, s25	s2, s7, s11, s20, s23, s28, s32	s4, s9, s13. s16, s25, s29, s33	s3, s8, s12, s15, s18, s26, s34
C' lexity	Medium	s1, s8, s19, s21, s24, s30, s31	s5, s10, s14, s17, s22, s27, s25	s2, s7, s11, s20, s23, s28, s32	s4, s9, s13. s16, s25, s29, s33	s3, s8, s12, s15, s18, s26, s34
	Hard	s1, s8, s19, s21, s24, s30, s31	s5, s10, s14, s17, s22, s27, s25	s2, s7, s11, s20, s23, s28, s32	s4, s9, s13. s16, s25, s29, s33	s3, s8, s12, s15, s18, s26, s34

Table 10.1: Experimental Design Matrix

As described in the previous chapter, there were two kinds of complexity variation in the task domain. The first kind involved the variation of *control* and *concpetual* complexity , this determined intra-level question types of varying complexity. The second kind was defined by the number of target variables that subjects were required to bring within specified limits, .i.e. easy, medium, and hard (as shown in Figure 10.1).

- **Easy**: One variable; QT, DT, UT (Six question types)
- **Medium**: Two variables; QT and DT, QT and UT, UT and DT (Six questions types)
- **Hard**: Three variables; QT and DT and UT (Five question types)

(The experimental questions are listed in Appendix A)

Session	Activity	Allocated Time (min.)
Morning	Read simulation notes	10
	Training period Training questions	45
	Easy Questions	20
	Debriefing	5
	Distracter Task	3
	Medium Questions	40
	Debriefing	10
Afternoon (After two hour break)	Hard Questions Debriefing	50 15

Table 10.2: Experimental Session Outline

10.2.4. Questioning Procedure

There was no fixed number of questions that subjects were required to answer . The experimenter moved subjects on to the next question if they were clearly having no success. This was to ensure that subjects had sufficient exposure to as many of the system aspects as possible during problem solving. Subjects were not permitted to abandon a question by themselves. If a task was completed, the domain was reset and the system waited for the subject to activate the next question.

10.2.5. Verbalisations and Debriefings

For the last 10, 15, and 25 minutes of the three complexity levels subjects were asked to verbalise their problem solving. At the end of each question set, the monitor was switched off and subjects answered debriefing questions asked by the experimenter. These responses were recorded. (Debriefing questions are shown in Appendix A)

10.3. Overview of Data Capture

Before discussing the results, it is necessary to recap on how results were captured. Two sources were available in the experimental design:

- *Performance logs* which recorded:
  - Time spent on a problem;

• Whether the problem was completed by the subject, or aborted by the experimenter.

  • Success, the difference between the final value and the target value, as a percentage of the target variable, e.g. if target=50, and final-value =200, then success=300% off target.

• *Verbalisations* which were recorded:
  • During the latter part of task-solving section of complexity level;
  • During the debriefing sessions.

The investigation of the subject's mental model was divided into two parts, corresponding to the performance and verbalisation sources.

10.3.1. Declarative Results

This was based on the analysis of the *verbalisation* data. The data was broken down into statements about system relationships (with constraints) and concepts. The statements were then grouped into nine categories; four described general aspects of system behaviour, and five were based on conceptual descriptions of system variables . These are shown below, along with the abbreviations that will be used throughout the discussion.

State Knowledge: Queue Length (Q), Red/Green Times (R/G), Delay Time (D), Flow Rate (Flow), and Utility (U).

General Knowledge:Functional, cause and effect relationships (FUNC); Spatial sequence knowledge (SEQ); Temporal sequence knowledge (TEMP); Decentralised behaviour knowledge (DEC).

The total number of members in each category (identified from the subject verbalisations) were compared, by category, to the *ideal* system model. This was recorded in the form of a percentage of the ideal model identified, e.g. 59% DEC, describes that a subject identified 59 percent of the DEC relationships in the ideal model.

10.3.2. Procedural Results

This was based on the *success* criterion of the performance log. The success ratings were measured as a percentage of the target value, *from* each target variable. For questions involving more than one target variable, this measure was averaged across the target variables. For example, at the *hard* complexity level, for question type t, if the success (QT)=90 and success(UT)=100 and success(DT)=110, then the recorded value will be 100. The minimum value of this measure is 0, which signifies that the target variable is within the target range, i.e. 0% of the target value, off target.

The target variables were QT (queue length total), DT (delay time total) and UT (utility total). As mentioned earlier, the target variables varied in their conceptual and control complexity. Thus, a number of different task types were made possible by varying the direction in which a variable was to be moved to reach the target threshold, and varying the combinations of target variables. For example, 'QT<' and 'QT>', were two different task types at the easy level; 'UT< QT>' or 'UT> QT<' were two different task types at the medium level; ‘UT< QT< DT<‘ and ‘UT< QT> DT<‘ were two different task types at the hard level. Thus, the question difficulty was not based on the particular target value but on the target variable and the direction in which it had to be guided. The different tasks types appear in Appendix A and will also be described in the procedural results analysis section.

10.3.3. Media Groupings

As described at the end of Chapter 9, media were grouped into two expressiveness categories (with abbreviations)

- **Low Expressiveness (LOW)**-Bar Chart, Animation;
- **Higher Expressiveness (HIGHER)**-Table, Static Video, Graph.

Consequently, in the analysis the results for members of these groups were averaged together.

10.4. Analysis of Media in Terms of Expressiveness

We now examine specific results for the five media, in terms of expressiveness. The predicted effects of the mental model induced by a medium of a certain expressiveness on the understanding of system relationships (declarative and procedural knowledge) are described, and then compared with the actual results obtained. The study of the declarative and procedural data is based on comparisons with the predictions made in Chapters 8 and 9 of how expressiveness will effect the type of mental model formed of the domain.

The analysis and discussion is divided into three parts. Firstly, the declarative aspects of the subject's mental model are investigated. This includes general comments on the data, along with specific comparisons with the expressiveness predictions made at the end of the previous chapter. This is based on verbalisation data taken a the end of the 'hard' complexity level, as it was assumed subjects would have obtained their maximum amount of declarative knowledge at this stage.

Secondly, the procedural data is addressed at *each* level of coarse task complexity (easy, medium and hard). Again, this is based on comparisons with expressiveness predications which relate both to this coarse complexity and the *intra*-level variations in conceptual and control complexity. Finally, a general discussion addresses the overall effect of expressiveness on the procedural knowledge across all levels of complexity.

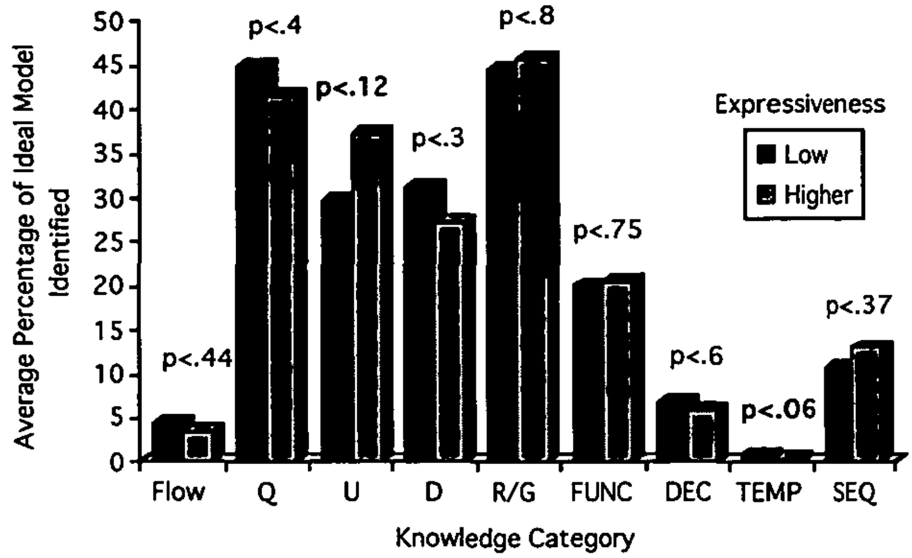
10.4.1. Declarative Knowledge Results[2]

For *each* knowledge category (Flow, Q, U, etc.) at the *hard* complexity level, a single factor ANOVA was conducted, with expressiveness as the *independent* variable. The *dependent* variable was the average percentage of the ideal model which was identified by subjects. (defined in Section 10.3.1). This results of this are shown in Figure 10.1. (over)

10.4.1.1. General Discussion of Declarative Knowledge

Excluding FLOW, the most striking characteristic of the results in Figure 10.1 is the difference between the last four aspects, and the previous four. This is primarily due to the method used to capture these results. The SEQ, TEMP, DEC aspects were based in part, or entirely, on the recording of the subjects' recognition of *constraints* [3]on system relationships. Generally, subjects did not provide such caveats, which accounts for the relatively low level of the correct knowledge identified for each category. Moreover, this shows that the low results may not show ignorance of constraints, rather they show poor verbalisation (this flaw in verbalisations was discussed in Chapter 9).

[2] Descriptions of the abbreviations used are given in **section 10.2.1**.
[3] For example, 'increasing the red time on a light, increased its queue, *but only if there are cars flowing into the light*'. The constraint is shown in italics.



Variable	Statistic
Flow	F(1, 32)=1.021
Queue Length (Q)	F(1, 32)=1.101
Utility (U)	F(1, 32)=1.987
Delay Time (D)	F(1, 32)=1.301
Red/Green Times (R/D)	F(1, 32)=0.0004
Functional (FUNC)	F(1, 32)=0.101
Decentralised (DEC)	F(1, 32)=0.140
Temporal (TEMP)	F(1, 32)=5.765
Sequential (SEQ)	F(1, 32)=1.355

Figure 10.1: Average knowledge at the end of the 'Hard' level of complexity for each knowledge category between expressiveness levels

To overcome these poor results, the *relative* amount of correct knowledge was analysed. However, the particularly low level of knowledge for TEMP warrants discussion. This variable measures the number of temporal constraints on relationships correctly identified by subjects. As with DEC, subjects were reluctant to verbalise any caveats on the knowledge they identified. A similar reluctance was described by Verhage (1989) so it is assumed that this is not unusual.

However, TEMP does show a significantly lower level of response than the other constraint-based measures (see Table 10.3), therefore this requires further investigation.

Comparison:	Mean Diff.:	Fisher PLSD:
TEMP vs. DEC	7.571	4.315 *
TEMP vs. SEQ	-9.286	4.315 *

* Significant at 95%

Table 10.3. Difference in difficulty between constraint-based knowledge categories

The representation of time was explicit in one medium of each of the two groups. In the animation of the low expressiveness group and the graph of the high expressiveness group (The interpretation of the graph's horizontal axis as showing time was assumed to be obvious as they were mostly studying mathematics related subjects[4]). Since this difference in the representation of time exists between media, analysis of the individual scores for each medium within the two expressiveness categories is shown in Table 10.4.

The animation is significantly higher than the other four media (p<.001) but all media show a negligible average amount of TEMP knowledge. In addition, the standard deviations (St. Dev.) shows that there was a narrow range of results on all media. Thus, it seems that either subjects did not verbalise their temporal knowledge, or such knowledge was not a large part of their mental model, even if time was explicitly represented in the medium. The latter aspect will be addressed in the Final section of this chapter.

Medium	Mean:	St. Dev.:	
Animation	1.571	1.618	LOW
Bar Chart	.286	.756	
Static Video	.286	.756	HIGHER
Table	.286	.488	
Graph	.286	.488	

Table 10.4. Average TEMP knowledge for media within low and high expressiveness groups

[4] One subject interpreted the horizontal axis as showing distance.

10.4.1.2. Specific Expressiveness Differences for Declarative Knowledge Categories

This section discusses empirical evidence of the predictions made about the relationship between task complexity and the mental models induced by different levels of expressiveness made in the previous chapter.

It was postulated that the declarative knowledge of each category gained by subjects by the end of the experiment would be differentiated by two levels of expressiveness. This is due to the support given by the induced mental model in the understanding of conceptual aspects of the domain. To recap, the following predictions were made in Chapter 9 about the level of correct declarative knowledge in a subjects' mental model that would be induced by the two levels of expressiveness, as a function of concept complexity. In this discussion. match and mismatch are used to signify when the model induced by the expressiveness of the medium is congruent with the concept complexity, thus allowing better understanding:

- **Low Expressiveness**: More correct knowledge of Q, SEQ, TEMP, and FUNC than more expressive media (match). Worse, than more expressive media for other categories (mismatch)

- **Higher Expressiveness:**: Better than low expressiveness media for correct knowledge of D, Flow, U, and DEC (match). However, a mismatch between high expressiveness and low concept complexity may cause easier concepts (SEQ, TEMP, FUNC, R/G) to be less well understood (mismatch).

The results (see Figure 10.1) concur with the predictions for Q and TEMP with only the TEMP level approaching significance ($p<.06$). However, the SEQ and FUNC results, though in the opposite direction, are not significant. Of the five remaining variables (Flow, U, D, R/G, DEC), R/G and U are in the predicted direction. However, only U approaches significance for better comprehension with the more expressive media ($p<.12$). Since, none of the other relations significantly violate the prediction, i.e. show better understanding for low expressiveness, the expressiveness difference is tentatively identified as a distinguishing factor.

But why the lack of significance for those categories which were in the incorrect direction, but where not significant (Flow, D, DEC, FUNC, and SEQ) ? A clue is the lower level of correct knowledge, for either expressiveness level, which is evident for all these categories. In terms of expressiveness, this suggests that a mismatch is occurring between category complexity and medium expressiveness. Whilst this may be true for the more complex categories (D, Flow, DEC), this does not explain the poor results for the minimal complexity, FUNC and SEQ, categories.

A probable explanation for FUNC is that the difference between the expressiveness levels was not sufficient to cause a similar difference in FUNC knowledge (Shown by $p<.76$). SEQ, on the other hand, shows more significant difference between expressiveness levels ($p<.37$), but in the wrong direction. This suggests that SEQ may be conceptually more difficult than was predicted, therefore requiring a higher level of expressiveness (as shown by the higher knowledge rating for the higher expressiveness media).

As may be expected, this effect is most pronounced for the two categories identified (in Chapter 9) as the most difficult to understand, i.e. Flow and DEC. However, U is also identified as one of these three categories, but its knowledge rating is significantly higher than Flow and DEC (see Table 10.5). Thus, given the result for U, it can be suggested that the other two aspects are more complex than U and would therefore require a medium with higher expressiveness to induce more correct knowledge about these concepts. This explains why an unclear result (relationship in the wrong direction, but not significantly so) was obtained for these variables.

A similar conclusion was alluded to by Alty et al. (1992), who suggested that if the complexity of a task was too great, any differences in performance due to the use of different media would be cancelled out. This can be best described as a *mismatch* between task complexity and medium expressiveness, as described in Chapter 9.

Comparison:	Mean Diff.:	Fisher PLSD:
Flow vs. U	-30.886	5.306*
DEC vs. U	25.543	4.315 *

* Significant at 95%

Table 10.5. Difference in difficulty between knowledge categories, irrespective of expressiveness

10.4.1.3. Summary of Declarative Results

The predictions made for the effect of expressiveness on the comprehension of the 9 knowledge categories has received some validation. Two important issues emerge from these results:

• The comprehension of temporal aspects of a domain may be difficult, even with an inherently temporal representation such as the animation. This was shown by the low comprehension levels for the TEMP category.

• Mismatches between expressiveness and task complexity are more likely to be manifested in declarative knowledge if the complexity of the task is too high for the expressiveness (inducing too simple a model), rather than vice-versa (inducing too complex a model). Examples of the former case were the SEQ, DEC and Flow categories, an example of the latter was the FUNC category.

The procedural knowledge of subjects will now be discussed. This is based on the automatic performance-log data which was recorded during computer-based activity and the comparative difficulties of different question types performed by subjects.

10.4.2. Procedural Knowledge Results

(In the discussion the following target variable abbreviations will be used, QT for Queue Length Total, UT for Utility Total, DT for Delay Time Total.)

This section is broken into three parts, for the discussion of results of each of the three coarse complexity levels. Again, each level is discussed in terms of the *intra*-level diffculties which arise from variations in individual question difficulty due to different *conceptual* and *control* complexities. Within each level, as with the declarative knowledge section, the discussion is divided between general and specific comments on the data. These are based on comparisons with the predicted effects of expressiveness on mental models made in Chapter 9.

The analysis of procedural data was based on an automatic log which recorded the success of subject activity for each question, at each of the three levels of complexity (easy, medium, hard). Easy questions involved controlling one of the three target variables to with specified limits; medium complexity involved two variables, and hard, three. The measure of interest was how close subjects were, on average, to the target thresholds when the question was either completed or aborted. Within these three levels, *intra*-level question complexity was affected by the choice target variables, vis-a-vis their *conceptual* and *control* complexity (as described in the previous chapter). Thus, they were varied by altering the direction of the threshold values from the initial target value, e.g. '*QT>*' means the QT target variable had to be increased to the target threshold.

For each complexity level and for each task *type*, a single factor ANOVA was conducted with expressiveness as the *independent* variable. The *dependent* variable was the average percentage from the solution at the time the last action was made. The results at each level of complexity are shown in Figure 10.2, 3, and 4. Unlike the declarative knowledge, the discussion is now based on each set of performance data at the three levels of complexity. This begins with the easy level results as shown in Figure 10.2.

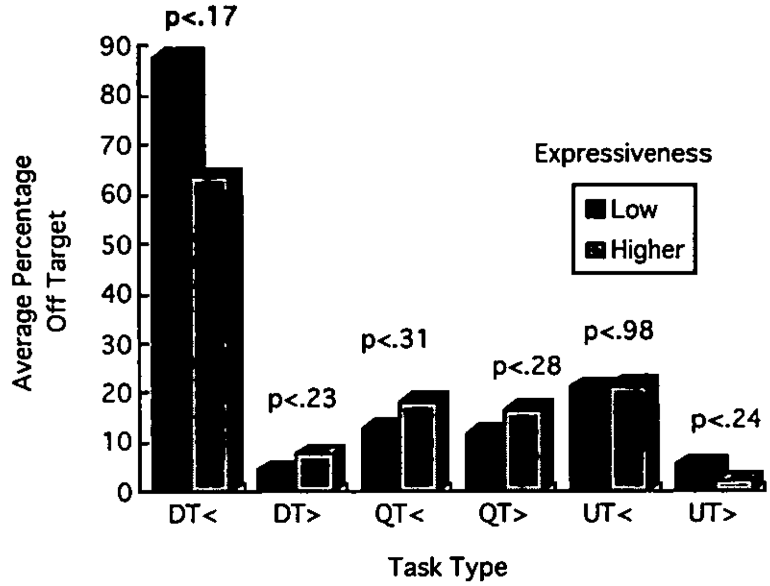
(In the following results, the higher the dependent variable, the *worse* the performance)

10.4.2.1. Procedural Results at Easy Complexity Level

At this complexity, tasks were based on the controlling of *one* of the target variables. Question complexity was varied by the conceptual and control complexity of the target variable, e.g. QT was conceptually simple but difficult to control. The discussion begins with general comments about the results.

10.4.2.1.1. General Discussion at Easy Complexity Level

Why do subjects perform significantly worse ($p<.001$) on the 'DT<' questions as opposed to the other categories? It had been predicted that subjects would find the 'UT<' questions the most difficult for both low and higher expressive media. The answer may lie in the extra comments made by subjects. Many subjects commented that it was very difficult to bring the system delay down (though generally, it was these subjects who were unaware of the strong correlation between DT and QT). This difficulty was due to the initial rapid build-up of traffic on all lights, which made it increasingly difficult for subjects to reduce DT, particularly when they were inexperienced with the domain, as at this early stage. Thus, if subjects did not react quickly to initially heavy traffic conditions, the system soon got into such a state, that only an experienced operator would be able to bring variables within an acceptable *lower* threshold. This is also shown in the *medium* task complexity results where the 'DT<QT<' condition caused the worst performance. Conversely, at the *hard* level of complexity, subjects were able to control delay better in one of three cases where delay had to be reduced from its initial value. Thus suggesting that by this stage their mental model was of a sufficient complexity to understand this relationship.



Variable	Statistic
DT<	F(1, 76)=1.849
DT>	F(1, 42)=1.448
QT<	F(1, 23)=1.3
QT>	F(1, 92)=1.33
UT<	F(1, 67)=0.0004
UT>	F(1, 52)=1.34

Figure 10.2. Performance data at easy level for low and medium expressiveness

10.4.2.1.2. Specific Expressiveness Predications at the Easy Complexity Level

The following predictions were made in Chapter 9.

- UT<, DT<, and QT< will be better controlled by the more expressive media (match).
- QT>, UT>, and DT> will be better controlled by the less expressive media (match) than the more expressiveness media (mismatch).

The results do not show significant effects in line with any of first three categories (UT<, DT<, and QT<). The only result approaching significance does validate the hypotheses, but the other two categories show a relation in the opposite direction to the hypothesis. The lack of significance of the latter two results (p<.9, p<.31) along with the DT< result do weakly indicate that the hypothesis is correct.

Of the second three categories, two exhibit a relationship in the specified direction (DT>, p<.23; QT>, p<.28) and one in the opposite direction (UT>, p>.24). This offers some evidence to support the predictions.

10.4.2.2. Procedural Results at Medium Complexity Level

At this complexity, tasks were based on the controlling of *two* of the target variables. Again, *intra*-level complxity was provided by the *conceptual* and *control* complexities for the target variables in each question type. The results are shown in Figure 10.3. (over)

10.4.2.2.1. General Comments at Medium Complexity Level

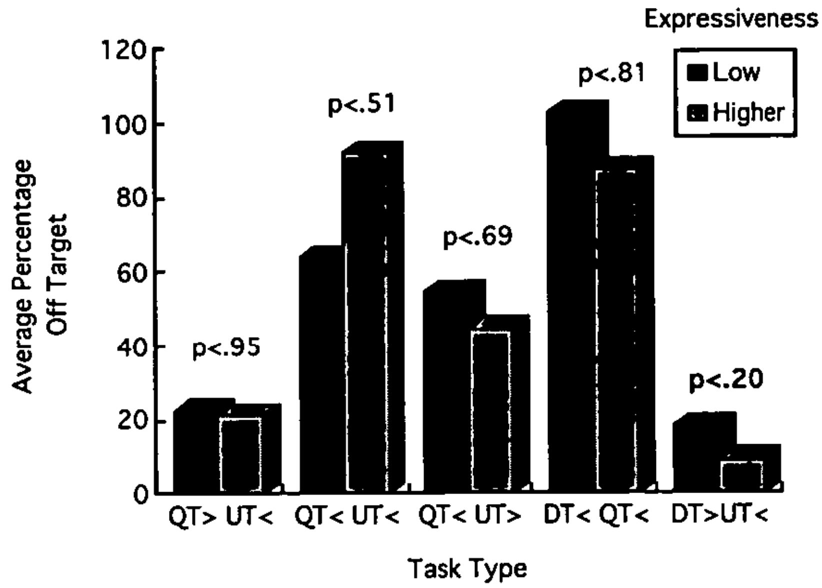
The most difficult condition for both expressiveness levels was 'DT< QT<', rather than any of the conditions involving the control of utility, the most complex of the three variables. This was for the reason identified in section 10.4.2.1.1; subjects did not react quickly enough at the beginning of the questions causing queues to build up to a level that was difficult to reduce. The majority of subjects do not seem to have understood this problem.

10.4.2.2.2. Specific Expressiveness predications at Medium complexity level

The following predictions were made in Chapter 9.

- Performance for all question types would be better for the more expressive media (match);
- The more expressive media would allow better performance for the 'QT<' and 'DT< QT<' (match) than the 'UT< QT<', 'QT< UT>', and 'DT> UT<' question types (mismatch).

The results agree with the first prediction in four out of the five cases, although only the 'DT< UT> condition approaches significance (p<.20). This lack of a clear result is due to taking performance data at the time of the subject's last action, not at the end of question. The results indicate a mismatch between task complexity and the model induced by the level of expressiveness. Thus, for more complex system control, a model induced by a higher level of expressiveness is required.



Variable	Statistic
QT> UT<	F(1, 67)=0.004
QT< UT>	F(1, 35)=0.462
QT< UT>	F(1, 46)=0.159
DT< QT<	F(1, 32)=0.063
DT> UT<	F(1, 20)=2.147

Figure 10.3. Performance data at medium level for low and medium expressiveness

To investigate the second prediction, the difference in performance between the five task types (for the medium expressiveness media) is shown in Table 10.6. (over) Of the three conditions identified as most difficult (Groups 2, 3 and 4 in Table 10.6), only 'UT< and QT<' (Group 2 in Table 10.6) is significantly more difficult than the easier 'QT> and UT<' (Group 1) condition. These results indicate that the complexity differences between the two groups of question types are generally not sufficient to be differentiated by expressiveness.

10.4.2.3. Procedural Results at Hard Complexity Level

At this complexity, tasks were based on the controlling of *three* of the target variables. The results are shown in Figure 10.4. (over)

Group	T. Type
1	QT> UT<
2	QT< UT<
3	QT< UT>
4	DT< QT<
5	DT< UT<

Comparison:	Mean Diff.:	Fisher PLSD:
Group 1 vs. 2	-71.283	46.35 *
Group 1 vs. 3	-22.903	43.034
Group 1 vs. 5	12.333	57.008
Group 2 vs. 4	4.583	52.959
Group 3 vs. 4	-43.796	50.084
Group 4 vs. 5	79.032	62.501 *

* Significant at 95%

Table 10.6: Comparison of 'QT< UT<' ,'QT< UT>' and 'DT> UT<' with other task types

10.4.2.3.1. General Comments at the Hard Complexity Level

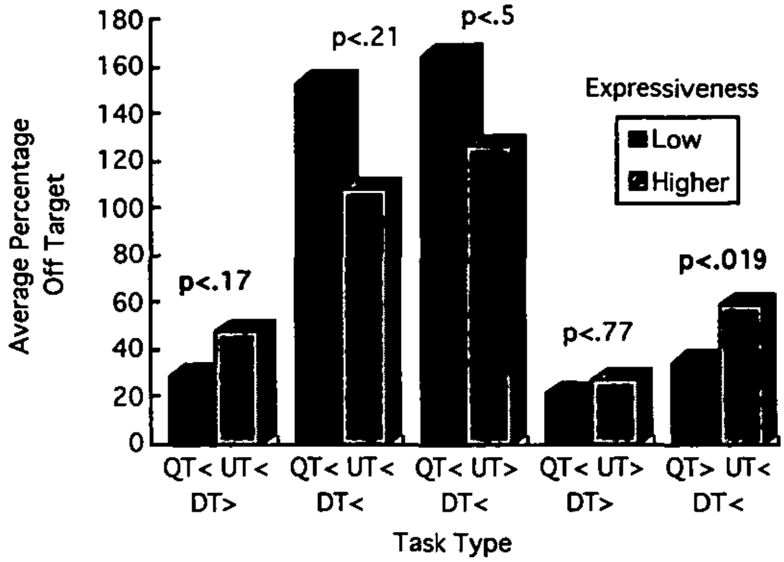
The trend of subjects struggling to bring DT and QT down was continued. The 'QT< UT< DT<' (Group 2 in Table 10.7) and 'QT< UT> DT<' (Group 3 in Table 10.7) conditions showed significantly poorer performance as shown in Table 10.7 over.

Group	T. Type
1	QT< UT< DT>
2	QT< UT< DT<
3	QT< UT> DT<
4	QT< UT> DT>
5	QT> UT< DT<

Comparison:	Mean Diff.:	Fisher PLSD:
Group 2 vs. 1	-86.961	39.898 *
Group 2 vs. 4	102.966	48.784 *
Group 2 vs. 5	83.362	59.418 *
Group 3 vs. 1	-91.483	50.968 *
Group 3 vs. 4	107.488	58.187 *
Group 3 vs. 5	87.884	67.353 *

* Significant at 95%

Table 10.7: Comparison of 'QT< UT< DT<', 'QT< UT> DT<' with other task types



Variable	Statistic
QT> UT< DT>	F(1, 44)=1.945
QT< UT< DT<	F(1, 64)=1.613
QT< UT> DT<	F(1, 24)=0.108
QT< UT> DT>	F(1, 23)=0.089
QT> UT< DT<	F(1, 13)=6.543

Figure 10.4: Performance data at hard level for low and medium expressiveness

10.4.2.3.2. Specific Expressiveness Predications at Hard Complexity Level

The following predictions were made in Chapter 9.

- As with the medium complexity level, the low expressiveness media should produce poorer performance on all task types (mismatch);
- Within the higher expressiveness level, the hardest conditions should be 'QT< UT< DT>' and ' QT> UT< DT<' (mismatch);
- Subjects will also struggle with the remaining conditions 'QT< UT< DT<', 'QT< UT> DT<' , and ' QT< UT> DT>' (mismatch).

The first prediction is not clearly shown in the results, since the two results ('QT< UT< DT>' and 'QT> UT< DT<') approaching significance (p<.17, p<.019) are in the opposite direction. This is weakly contradicted by the two most poorly performed conditions ('QT< UT< DT<' and'QT<UT>DT<') which agree with the prediction, though only one approaches significance ('QT< UT< DT<', p<.21). To investigate the prediction further, Table 10.8 shows the comparison of overall performance on all task types for the two expressiveness levels (the lower the value the better the performance). The difference is in the right direction which suggests expressiveness causes noticeable differences in performance at this level of complexity.

Expressiveness:	No. of Media	Mean:	Std. Dev.:	Std. Error:
Low	5	80.6	71.867	32.14
Higher	5	73.6	41.741	18.667

Table 10.8: Comparison of Expressiveness Levels (p.<.8)

The second prediction was shown in the 'QT< UT< DT>' category, but not in the ' QT> UT< DT<' category. However, this is still the third hardest category out of the five. The remaining conditions were easier (with the exception of the 'QT< UT< DT<' category).

10.4.3.2 Procedural Data Across Complexity Levels

Since subjects attempted questions at all three complexity levels, some learning effect must be present in the performance data. Of particular interest is the effect of a mismatch between the models induced by different levels of expressiveness, and the model required to effectively support a task of a certain complexity. Exactly when a mismatch occurs will depend on the expressiveness of the medium. For example, a subject using the higher expressiveness representation will face a mismatch in the easy, and hard tasks, i.e. the induced mental model will not best support these tasks. The question remains how this will be borne out in the results. It is postulated that where there is a mismatch performance will be worse than where there is a match. This will now be investigated.

Firstly, the change in performance for both levels of expressiveness, over the three complexity levels, is shown in Figure 10.5. A test of correlation shows a clear decrease in performance over the levels of complexity ($r^2=1.0$). This suggests that any advantages inherent in the match between higher expressiveness and higher complexity was not sufficient to counter the increase in task complexity. This was in spite of the one-and-a-half hours experience the subjects had with the system.

However, it is interesting to note the divergence in performance between the two complexity levels. It is here the mismatch principle may be apparent.

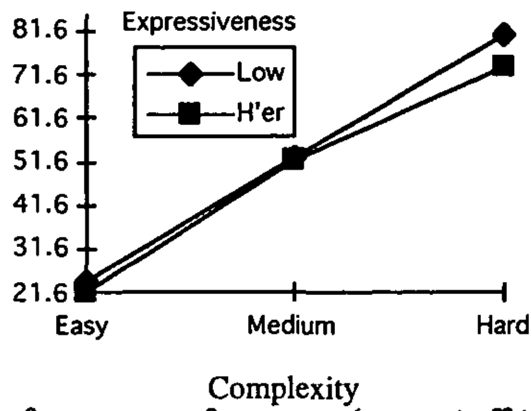


Figure 10.5: Graph of average performance (percent off target) over complexity levels for low and higher expressiveness

The divergence in performance between low and higher expressiveness increases significantly across complexity levels (correlation=.697). In other words, the more expressive media allow performance to degrade less, as complexity increases. This indicates a match between the higher expressiveness and the high task complexity. However, a similar match is not shown at the Easy level, with the lower expressiveness. This is in agreement with a result from the declarative knowledge analysis FUNC category which suggested that a match is more likely if the complexity of the task is *too high* for the expressiveness, rather than vice-versa.

10.4.2.4. Summary of Procedural Knowledge Results

The performance results were not conclusively in favour of the predicted effect of expressiveness on procedural knowledge, although they did indicate a small effect, particularly at the Hard Complexity level. Also, the following points are of interest.

- Generally performance got worse at all levels of expressiveness, in spite of any learning effect.

- Matches and mismatches between expressiveness and task complexity were alluded to in all levels of task complexity. This effect was more apparent at the Hard complexity level.

10.5. Possible Improvement of Results

The results were not totally in favour of expressiveness and its relationship with mental models and tasks, nor were they totally against it. Moreover, since there were fewer significant results against it, a tentative affirmation of the theory can be made.

However, further experimentation is clearly required. Given the importance of expressiveness, the flaws in the experiment which may have caused limited results must be addressed. This will be useful if a similar experiment were conducted with a wider range of expressiveness.

10.5.1. Experimental Design

Firstly, the aim to make the ideal system description as rich as possible may have, in the end, been detrimental to results. This was because the complexity of the model was such that many relationships and constraints were identified for each declarative knowledge category (TEMP, SEQ, etc.). Consequently, the number of references to these categories in subject verbalisations was normalised against what may have been, an unrealistic ideal total. This would in turn be reflected in generally low percentages for all categories. This was seen to be the case with no category result correctly identifying more than 50 percent of the equivalent category in the ideal model. Furthermore, the model itself is difficult to validate in terms of its completeness, i.e. does it capture ALL of the system aspects that a subject may identify? Although the analysis of subject verbalisations showed that were no statements which were outside the defined model behaviour, this may occur in a future experiment and must therefore be considered.

Secondly, it is difficult to define media which are neutral to the problem domain, i.e. are not task-dependent. This is particularly important when considering that the main hypothesis of the experiment was that the interface significantly affects internal representation. An important factor in this is that it is difficult to ensure that all media represent the same fundamental domain information. For example, in some media the state of a light had to be inferred from the other variables, in others it was directly represented. This was the result of the encoding mechanisms of the culprit media, i.e. it was difficult to show another variable in the already over-crowded graph, bar chart and table. This is an example of the encoding mechanisms of the media interfering with the common representation of whatever domain is chosen. Unfortunately, this can only be avoided by choosing a less simple domain.

10.5.2. Experimental Procedure

Firstly, the number and background of the subjects could have been more representative. Since the majority of participants had a mathematical background, the performance in the numerical based domain could have been inflated.

A second area of concern was the training period. To allow the subject to learn the basic control of the system, this predominantly used a two light rather than four light system. Although subjects did answer some training questions on the four-light system, their simplified view of the system may have been detrimental to their overall performance. The training period of 45 minutes may also have been too short which would have meant that the first set of questions (easy level) were treated as an extended training period and again performance was degraded.

The random question order at all levels meant that some subjects received more difficult questions which limited the number of questions they answered in the time available.

The logging algorithm only recorded the system state at the last action, as it was assumed this state would approximate to the final state. However, some subjects left the system running for long periods of time after their last action, which invalidated the logged result as an accurate measure of the system state. To counter this, for half of the subjects the experimenter entered a dummy input if the questions had to be aborted. This ensured that the true final state of the system was recorded.

Finally, the inherent problems in verbalisations must be stated again. There is no systematic way of knowing whether subjects are verbalising all that they know.

10.6. Summary of Results

The experiment aimed to show the effect expressiveness would have on the mental models of complex domains. Of interest, was the match between the mental models induced by different levels of expressiveness (as manifested in subject's levels of declarative and procedural knowledge) and the consequent support to tasks of varying complexity. The results showed this induced effect over two aspects of subject system knowledge. However, it must be stressed that this experiment was investigating a novel concept and thus it is no surprise that the results were not conclusive. However, the study, though preliminary, is essential in the development of the concepts described.

Firstly, for the declarative knowledge (Q, U, D, FLOW, R/G, FUNC, TEMP, SEQ, DEC) gained by the end of the final session the following results were obtained:

- Temporal knowledge (TEMP) was low for all categories at both levels of expressiveness. This indicates that temporal knowledge is difficult to induce, even with animation;
- Better comprehension was seen on all complex variables at the higher expressiveness level (only utility total (UT)) approaching significance ($F(1, 32)=1.987$, $p<.12$). This stresses the importance of a match the expressiveness of the medium to the task complexity.

Secondly, procedural knowledge was extracted from the automatic performance logs. These results took into account the coarse difficulty level as well intra-level task complexity (which was still an issue of induced models matching complexity). The latter was based on the comprehension and control difficulties inherent in the question variables.

- **Easy level**: None of the three discriminating categories (UT<, DT<, QT<) exhibited significant behaviour in the right direction. However, the DT< condition did approach significance ($F(1, 76)=1.849$, $p<.17$). The remaining three categories exhibited equality of the two expressiveness levels, as expected.

- **Medium level**: The prediction was that all question categories would be more difficult with the less expressive media. This was shown in four of the five categories (not 'QT< UT<') but only the 'DT> UT<' condition approached significant ($F(1, 20)=2.147$, $p<.2$)

- **Hard level** : The higher expressiveness level showed better performance on two of the expected five conditions ('QT< UT< DT<' and 'QT< UT< DT>'). Only the former case approached significance ($F(1, 64)=1.945$, $p<.21$). However, average error was higher for the low expressiveness level. The predicted intra-expressiveness level difficulty was shown in the 'QT< UT< DT>' category, but not in the 'QT>UT<DT<' category. However this still the third hardest category out of the five. The remaining conditions were easier (with the exception of the 'QT< UT<DT<' category).

A study of the overall results showed that subjects found questions increasingly difficult. However, those subjects with the more expressive media showed a significantly less rapid degradation in performance. The *divergence* of performance over complexity levels due to the level of expressiveness had a correlation of 0.697.

Given these preliminary indications, in the next chapter a framework for media allocation, based on expressiveness and its effect on mental models, will be described in full.

Part 4

Towards a Methodology for Multimedia User Interface Design

Chapter 11

A Media Allocation Framework

11.1. Introduction

The discussion in the thesis is now in a position to bring together the theoretical and experimental studies under a tentative media allocation framework. The former addressed how different media encode domain information. The discussion concluded that some media were able to encode more domain information in a single domain instance than others, thus they were deemed more expressive. The importance, or not, of this quality depends on the requirements of tasks. If tasks require the consideration of only a few domain states then the expressiveness of the medium need only be low. However, if many states must be considered, then high expressiveness allows a medium to encode a number of these states in an economical way and allow abstraction to take place.

The cognitive repercussions of the relationship between tasks and representations are an image of the domain which is a function of the expressiveness of the interface media. Since this image determines performance, then getting the right level of abstraction in the mental model is essential.

The experiment provided some evidence in separate but related parts. Firstly, the results suggest that media (categorised according to the way they encode information, in terms of expressiveness) will induce predictable types of mental model, i.e.

particular abstractions. There are two extremes to these types which are the result of low and high expressiveness respectively. The first type are those which are predominantly isomorphic in their representations and therefore able to convey concrete concepts. Concrete concepts are those which are not derived from other variables, such as the red light time and the green light time in the experimental domain. At the other extreme, models induced by high expressiveness are based on abstractions of the domain, e.g. relationships (with/without) constraints.

The second part of the evidence is the relationship between media and task complexity. Since the experiment provided a range of complexities both in the understanding and control of single or multiple variables, the success of different media at supporting different complexities became clear.

It is the relationship between these two aspects which is the goal of the study. Media were well matched to tasks, i.e. they allowed better performance, by virtue of their expressiveness inducing the correct type of model for task complexity. In different situations, this could mean a very simple model or a very complex model, the key being that the medium 'said' no more and no less than was necessary.

11.2. A Tentative Framework

From the outset, the thesis has aimed at an approach to defining a methodology for the allocation of output media. Thus, the culmination of the thesis is to use the results of the theoretical and experimental studies to this end. This can now be shown in Figure 11.1.

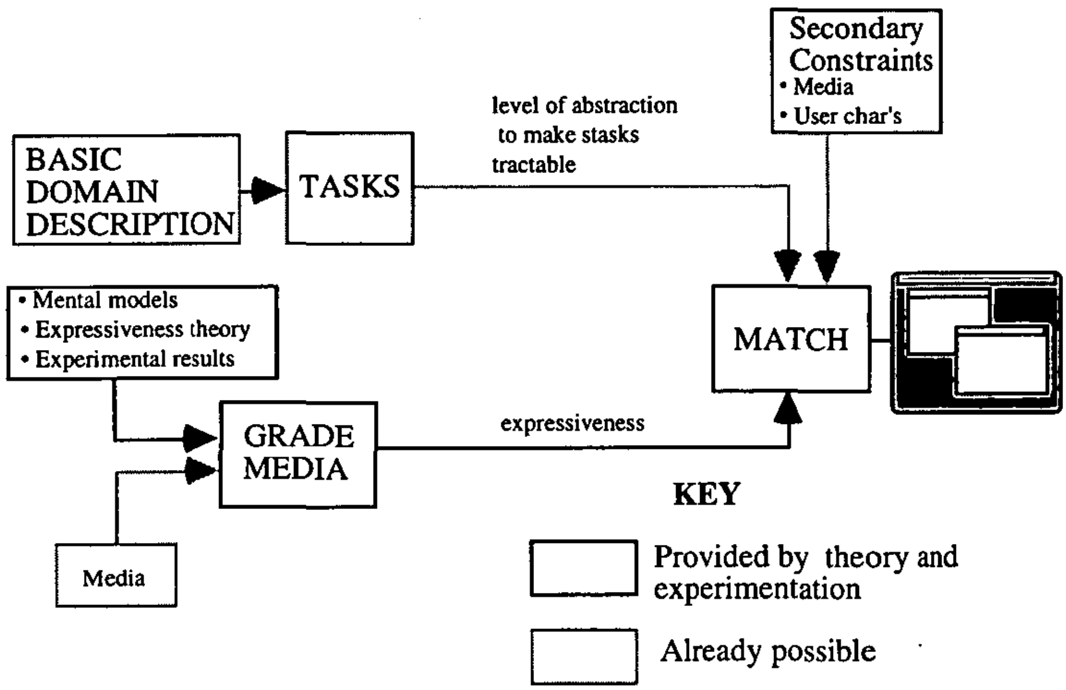


Figure 11.1: Using mental models and expressiveness to match media to tasks

The matching process indirectly relies on the cognitive consequences of allocating a particular medium for a particular task. It is assumed that the hard work has already been done by using the experimental and theoretical evidence to correctly grade the media at an accurate level of expressiveness. The matching process itself should then be straightforward. The main stages will now be described:

11.2.1. Domain Description

As already mentioned, it is assumed that the interface design process begins with a domain which is already described in some conceptual form. The description must capture all of the fundamental aspects of the domain. This includes a description of the prevalent data-types, objects and actions. Moreover, seemingly complex domains can be described in this way since their definitions are generative, e.g. formulaic descriptions of 'black-box' behaviour. The characteristics will provide a first-cut choice of media but may not be sufficient to solve the identified tasks. Only an astute choice of representation will allow this.

11.2.2. Describing Tasks

The tasks which to be carried out in the domain must also be described. Of particular importance is how many states of the domain must be visited by the user in order to solve the task. The abstraction required over the domain will be dependent upon this characteristic. For example, a task which requires 10000 states of the domain to be viewed will require more abstraction than a task which only required ten states to be examined.

11.2.3. Grading Media by Expressiveness

This state represents the application of the main part of the theoretical and experimental studies.

11.2.4. Matching Media to Tasks

The abstraction required by tasks and the expressiveness of the available media can now be made congruent. However, since the theory does not offer a mathematically precise description of how many domain states a given medium can represent, the matching process cannot be done automatically. The decision must be based on the different ways a medium affords expressiveness (pragmatic abstraction or non-physical abstraction, see Chapter 8) and the secondary environmental constraints. For example, if more expressiveness is required by a task, those subjects which are not trained in the use of the non-physical encodings should be provided with perceptual pragmatic methods. These are regarded as more intuitive in their use of emergent Gestalt properties, rather than the complex (hidden) syntax and semantics of non-physical encodings.

An example of this process has already been demonstrated in Williams et al. (1996) In this case, three video cassette recorder interfaces were described, and reasons given, based on expressiveness matching task complexity, of why one out-performed the others. The three interfaces were:

- **Conventional**: set start-time, end-time, day and channel using a key-pad;

- **VideoPlus** : uses a time-code scanned from a TV-schedule;

• **PC-Based, direct manipulation**: Display individual programmes in a TV-schedule format. Programmes can be selected and dragged and dropped into a video area which automatically programming the video.

However, this example was limited in that only one task was investigated. Thus, a range of task complexities were not examined with respect to expressiveness. However, a number of important issues can be highlighted in this study:

• The choice of output representation can dictate the type of input activity, e.g. low expressive media *allowed* direct manipulation since objects (programmes) could be interacted with individually;

• Expressiveness can be used in two ways. To critique interfaces or to design them.

There now follows more involved examples which provide a range of task complexities which require different levels of abstractions to be made tractable. They demonstrate the use of the experimental evidence that has been gathered, and the more esoteric facets of different media.

11.2.5. Worked Examples

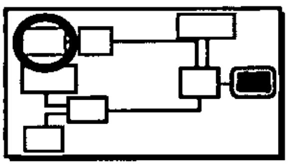
Two examples are described which address two types of domain; searching a large data-set, and providing a general purpose application environment. In the first case, the problem solving takes place in a directed way, in the second case, problem solving is part of the wider interaction. It is assumed that the tasks which must be accomplished in this domain have not been automated. This requires that the user must solve tasks manually

The discussion follows the stages in the Figure 11.1 at the beginning of this section. Stages one and two occur in parallel to stage three (a miniature of Figure 11.1 shows the position in the process at each stage). Stage three assumes that experimental evidence has been gained for the media in question and stage four assumes user differences along with available output media have been identified.

11.2.5.1. World-Wide-Web Visualisation for Searching

The visualisation of the hyper link network of the world wide web is a popular topic in the present visualisation literature (Gershon and Eick, 1995; Mukherjea and Foley, 1995; Lamping and Rao, 1996). The majority of approaches focus on the portrayal of the network topology (arrangement of nodes and links) rather than any semantic aspects of the domain. Thus, specific task considerations play a small part in the visualisation.

Stage 1: Domain Description



The fundamental aspects of the domain can be described in terms of nodes and links. Both can be decomposed using a frame-based language.

i.e. **node**-----

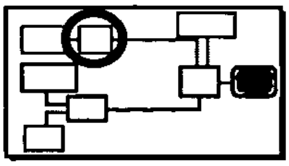
```
        Title-text string
        Creation date-text string
        Location-HTTP formatted address
        HTTP Version-Real
        Body-text body
        Semantic Description-

text body-----
        Text|Input widgets|Output Widgets|Images|Link

link-----
        Age-integer
        Source-HTTP formated address
        Destination-HTTP formatted address
```

Each domain *state* is regarded as a page which is defined by the above definitions.

Stage 2: Identification of the level of abstraction required



Sample WWW tasks are:

- Task 1: *Find all documents relating to a subject, e.g. multimedia.*

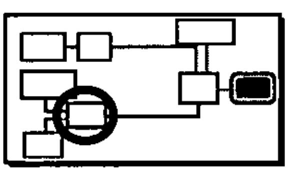
  This requires *extensive* abstraction over nodes (by subject) but also allowing subjects to move to the specific nodes thus requiring a specific representation.

- Task 2: *Show all the paths between two documents.*

  There will be many paths, but specific paths must be described.

Both tasks can be divided into two stages. Firstly there is an abstraction stage where a large number of domain states must be browsed before suitable candidates can be found. This is followed by the selection of possible candidates for viewing. This archetypal process has been the subject of much visualisation research, particularly with work on the *fish-eye lens* approach. In this literature (Furnas, 1989; Card et al., 1992; Greenberg et al., 1996), these two stages are called 'context' and 'focus'.

Stage 3: Grading Media in Terms of Expressiveness



Firstly, it is assumed that the approach advocated by the thesis has allowed media to be allocated an expressiveness rating. Secondly, the nature of this expressiveness (pragmatics vs. abstract syntax) must is also known. In this example, both of these types of encoding will be demonstrated.

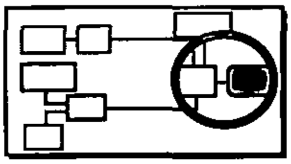
Thus, to begin with there are a range of media available.

- **Low Expressiveness**: Animation, bar chart.

• **Higher Expressiveness I** (using perceptual pragmatics): graph, table.

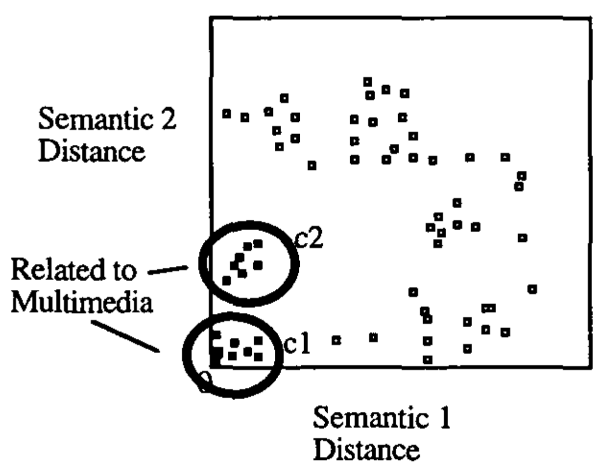
• **Higher Expressiveness II** (using abstract encoding): natural language, semantic networks.

Stage 4: Match appropriate expressiveness to domain tasks



• Task 1: *Find all documents relating to a subject, e.g. multimedia.*

For the first stage of task one, a medium with more than minimal expressiveness is required if the user is to perform the task in a reasonable time[1]. The graph provides a higher level of expressiveness due to its emergent perceptual pragmatics[2] when a number of individuals are shown together. This allows it to show interesting clusters in the data. If a graph is used in this way, the user will be able to see groups of candidate domain states. In the example, the graph plots the semantic-distance of each page on two criteria which define the target subject (based on the semantic category of the description). The perceptual abstraction of the graph allows two interesting clusters to be identified which are close to the target concept (c1 and c2).



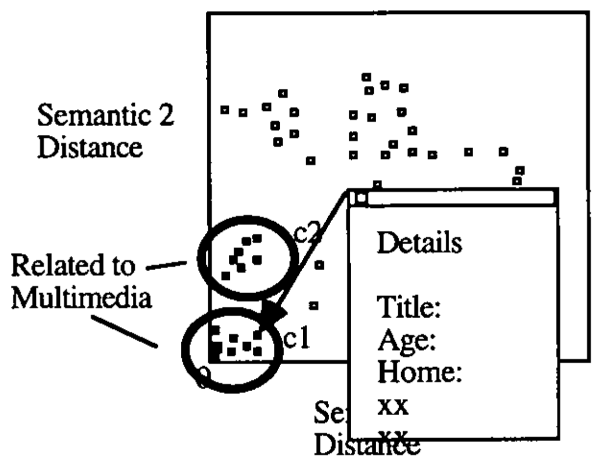
The second stage of the search process requires the description of individuals. In the case of the graph, since abstraction was provided by purely perceptual means, it can still be interacted with at an individual-level, i.e. through a data point. This is an example of the output-leading-input quality identified by Williams et al.'s VCR

[1] This will depend on the wider context, e.g. time constraints on the user.

[2] Defined in Chapter 8.

examples. This is also alluded to by Sutcliffe and Patel (1996). In both cases, the low expressiveness representations used had emergent perceptual pragmatics giving increased expressiveness.

Thus, in this example, the points of the graph can become 'hot spots' to view specific pages. The page details can now be specified in detail, so a low expressiveness medium must be used for this, e.g. text list using an unambiguous domain language, e.g. http://site.co.uk/root/home.html

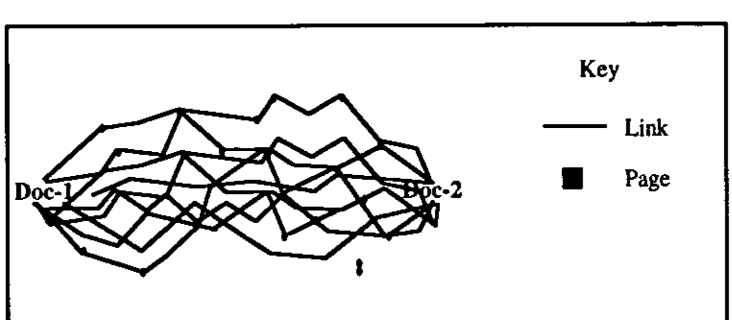


• Task 2: *Show all the paths between two documents.*

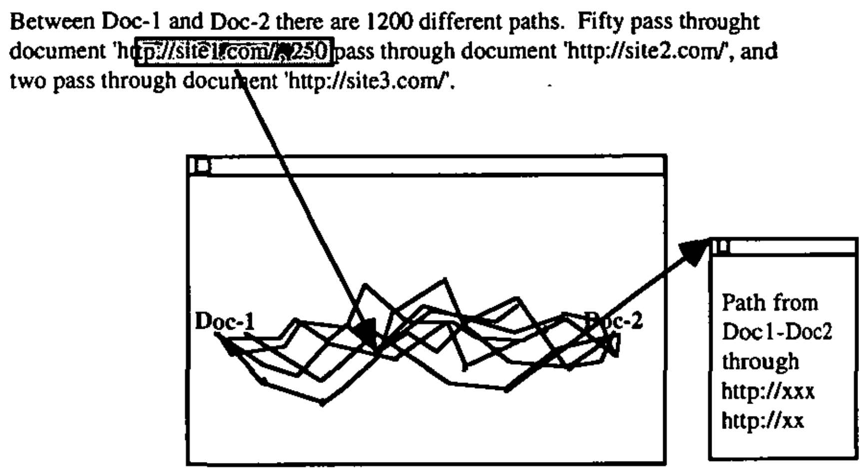
The first stage of the second task requires the description of a large number of paths. Natural language is chosen since it can describe large numbers symbolically rather than physically.

Between Doc-1 and Doc-2 there are 1200 different paths. Fifty pass through document 'http://site1.com/', 250 pass through document 'http://site2.com/', and two pass through document 'http://site3.com/'.

The representation of this data would be impossible to read for an inexpressive medium, even if it used perceptual pragmatics. For example:



However, in the next stage of the task where the inspection of a specific path is required, the expressiveness of natural language is too high, i.e. there are no *individuals* to select. Some secondary representation is now required which will provide the necessary limited expressiveness. By using the specific pages identified in the textual description the number of paths can that need to be considered can be reduced. This reduction allows the paths can be displayed using a low expressiveness medium which uses perceptual abstraction, e.g. the representation above.

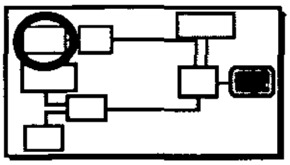


Both tasks show how when expressiveness is used in a task context, it may be required to be in a state of flux, as the different parts of the task dictate. Thus, whilst beginning with high expressiveness, at some later state low expressiveness may be required, or vice-versa. This variation will have important effects on the mental model the user holds about the domain. As has been stressed, only by understanding this relationship can media be allocated based on cognitive evidence. Though this notion is outside the scope of this thesis, it is an important characteristic of media which are actually in use in the interface.

11.2.5.2. E-mail Application Environment

In this example, the framework will be used to design an environment rather than task-specific representations. Whilst the same criterion of matching the expressiveness of the media to the complexity of the task holds, other considerations such as screen real-estate and the displaying of command-options must also be considered.

Stage 1: Domain Description



In this example, the domain description also includes a definition of actions that can be carried out in the environment.

Types

```
Types=message, message-queue
message=destination, subject, owner, text-body
message-queue=message [500]
owner={personal, group, all}
```

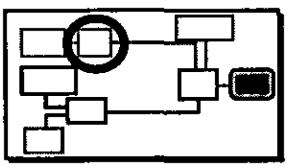
Instances

```
Objects=current message: message, out-queue, in-queue:
message_queue
Actions=new_message, delete_message, add_queue
```

Notes:

```
Outgoing mailing list is often over 100 messages long.
```

Stage 2: Identifying Abstraction Required by Tasks



The interface should support the following behaviours

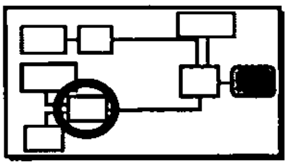
- Task 1: *Delete a specific message from a message-queue.*

Since the message list may so 100 messages long, it is not practical to display messages individually. Thus some form of abstraction is required. However, a specific message must be deleted so specificity is also important.

- Task 2: *Find how many messages in a message-queue are private.*

Again, there maybe over 100 messages in a queue, so individual message display is not practical. In the second stage, specificity is only required on the owner field of the message to allow the number of personal messages to be counted.

Stage 3: Grading Media in Terms of Expressiveness



All media named in Example one, plus:

- The 'perspective wall' (Card et al., 1993) offers an interesting combination of specific and general representation. The perspective wall provides perceptual abstraction on its periphery, whilst giving specificity in its central portion.

Stage 4: Matching Media to Tasks

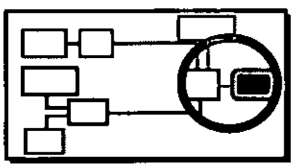


Figure 11.2 shows a possible interface which uses perspective walls with (shade encoding for ownership) to show the two message queues. This provides the relevant level of abstraction for both tasks, i.e. *deletion* and *selection* of e-mail messages based on ownership.

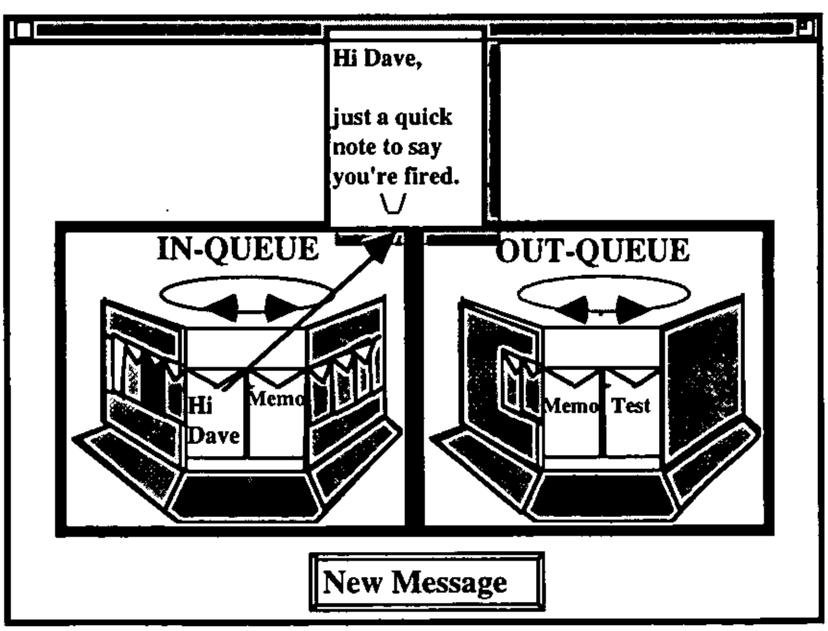


Figure 11.2: Using the Perspective Wall to Display E-mail Messages

The two perspective walls can be rotated independently.

This representation also allows a direct-manipulation style of interaction since objects are displayed individually in the central section. By adding a colour coding key (minimal expressive medium) to the positional system the perceptual abstraction is made possible over semantic aspects of the domain, e.g. message priority.

11.3. Summary

This chapter has described a framework for allocating media to domain tasks. It indicates the application of the theoretical and empirical aspects of this thesis, and shows how *expressiveness* and its human corollary, mental models, play a vital part in the design of user interfaces in order to support effective problem solving. Secondly, the effect output choice has on the interaction style has also been highlighted.

Chapter 12

Conclusions and Further Work

12.1 Conclusion

The following chapter provides an overview of the thesis and a discussion of directions this work could take.

12.1.1 The Thesis in Retrospect

This thesis has addressed the problem of media allocation in multimedia user interfaces, in a variety of domains. The survey of the current multimedia literature identified a number of gaps in the proposed methodologies and terminologies. The most striking deficiency is the lack of consideration given to the user's 'perception' of the interface. The thesis described how this has two important and related impacts on user interface design. Since it has been shown how the external representation of an artefact influences a user's conception of it, then the interface must be designed with knowledge of the nature of this influence. The second aspect determines the dependence of task performance on this conception. The need to address all three parts of this triumvirate along a common dimension, was identified as essential.

The importance of a uniform terminology was demonstrated by analysing the range of literature which can inform the meaning of terms such as multimedia and multimodal interfaces. The need for a deeper and broader discussion of how different media actually encode information was identified, with particular emphasis on how these methods can be harnessed in the appropriate problem solving context. To provide a basis for this discussion, the development and subsequent classification of writing systems was described. This allowed representations that are common in user interfaces to be placed within a descriptive hierarchy which classified representations by the nature of their encoding mechanisms. The coarsest distinction drawn was between systems which encode by perceptual variations in their form and those which encode by using an abstract syntax. From this discussion, the notion of *expressiveness* emerged. This was identified as a function of a medium's representational mechanisms, and describes the amount of domain information a medium can convey. It was also suggested that this property would have an important relationship with the cognitive facilities of users.

To provide a basis for the discussion of tasks that media may support, the notion of viewing human-computer interaction as designing for problem solving, was propounded. This allows for a more focused description of interface design since it can describe the problems to be accomplished, their congruence with human cognitive processes, and how the interface can satisfy both parts. This also made a study of those computer representations which are already used in problem solving appropriate. Consequently, the areas of knowledge-based reasoning and scientific visualisation were culled for related knowledge. Both fields identified the importance of the right level of abstraction in effective problem solving. This observation provided a link with the expressiveness criterion, since the level of abstraction can be seen as the conveying of a certain amount of information.

Given the relationship between tasks and expressiveness, the importance of mental models to problem solving was described. The study covered relevant literature in linguistics, cognitive psychology, and complex process control. Mental models were described as essential in problem-solving, but dependent on the externalisation of the problem domain. The paucity of empirical evidence of the models induced by different representations was shown. This discussion provided the basis for an in-depth study of the relationship between this cognitive aspect of user behaviour, and the design of user interfaces.

Given the deficiencies in the literature, a methodology was proposed which discussed all media at a representational level, describing lexicon, syntax, and semantics. However, a number of differences between the domains represented in the linguistics literature and computer-based domains, were emphasised. The main caveat was that the computer-based domains represent closed-worlds. Consequently, the discussion focused on encoding mechanisms of media, rather than contextual interpretation. From this discussion, the notion of expressiveness was more rigorously defined. This was seen as a result of the encoding mechanisms of media. A distinction was drawn between systems which encoded by physical variations in form, and those which used an abstract type-based system, e.g. natural language. Expressiveness was thus defined specifically as the number of different abstractions of a domain a representation is able to convey. Generally, the less physically constrained the representation is, the higher its expressiveness.

The link between representation and mental models leads to the consideration of how expressiveness influences mental models. This provides the link between the cognitive structures and the interface representation. Given that expressiveness describes how much abstraction of domain information a representation can convey, it was suggested that an induced models will reflect this quality. Thus, if the representation represents information in an isomorphic way, then the mental model will have a similarly simple and specific structure. Conversely, more expressive media which abstract over domain information will induce more abstractive mental models. Given this relationship, the reason that high or low expressiveness is ideal in certain task scenarios becomes clear. It is due to the medium inducing a mental model which is in a congruent form to the abstraction the problem-solving requires. Out of this discussion came the notion of perceptual *pragmatics*. These are a property of physically encoding systems which allow perceptual characteristics to be *read*, rather than perceived. This *reading* is characterised by a coalescing of minimally expressive individuals into a global view, e.g. trends, clusters. Thus, this allows these systems to abstract over domains and therefore increases their expressiveness.

To explore the effect of expressiveness on mental models and consequently problem solving, a preliminary experiment was devised. This investigated the mental models of subjects engaged in long-term problem solving over a range of task complexities, with a variety of media (animation, table, still video, bar chart, graph). The media were grouped into two levels of expressiveness, *low* and *higher* expressiveness. Subject mental models were studied in two parts; declarative knowledge (from verbalisations) and procedural knowledge (from performance data).

Even though this was a preliminary experiment, the results weakly indicated that expressiveness induced the type of mental models that were predicted. Thus, the quality of problem solving is likely to be dependent upon the match of the task complexity and the mental models induced by different levels of expressiveness. For simple tasks which required little abstraction, a concrete mental model was required. The results indicated that the inexpressive media induced such models, as did the higher expressiveness media. However, the two deviated when task complexity increased. In this case, a more abstract mental model was required. It was the more expressive media which induced this kind of model, thus allowing comprehension of the more complex domain concepts and behaviours.

Finally, a framework was described based on the theory and empirical evidence. This allows media to be matched to domain tasks. This is possible by virtue of their expressiveness and the knowledge gained about the relationship between this and the models that they will induce in users. The framework also demonstrated the effect the choice of output representation has on input possibilities. For example, in a typical direct-manipulation interface, the minimal expressiveness nature of the chosen representations, e.g. files, windows, affords a direct-manipulation interaction style. The specifically represented objects can be clicked and dragged at will.

12.1.2 Expressive Interfaces: A New User Interface Paradigm?

Chapter 3 of the thesis made the point that the majority of representations in modern user interfaces are, in the subsequently developed vocabulary, minimally expressive. Whilst the expressiveness approach would deem this acceptable (if the type of mental models required to perform these tasks were simple), interfaces use such representations irrespective of the task to be accomplished. For example, Microsoft's Windows'95, still uses an iconic representation of hierarchical file structures. This does not acknowledge the existence of tasks such as 'show me all the files that belong to David'. The representation must be matched to the abstraction required by the task, but tasks in interfaces are changing.

Buxton (1990) attacked the trend of the standardisation of graphical user interfaces. In his discussion, he asked how user interfaces reflected the abilities of humans when they were using other, non-computer based artefacts, or while engaging in human-human dialogue. A selection of the disparities identified are shown below.

- **Look and Feel**: interfaces do not use binocular vision or touch.
- **The Sonic Finder and Beyond**: interfaces do not provide a sonic landscape which allows navigation through complex information spaces.
- **Handling the pressure**: interfaces do not use input devices that are pressure sensitive.
- **Data Overload**: interfaces do not provide useful information, not copious data.
- **Alone in the corner**: interfaces do not treat users as social beings.

Buxton argued for a more complete interface, stimulating and receptive to all aspects of human activity, both individually and in groups. This is the same claim that was made by Marmollin (1991) for multimedia in the introduction of this thesis, however we are still far from this ideal. Therefore, given the realisation that the GUI paradigm is not sufficient for all tasks, how should interfaces develop?

Clues may be provided by Chapter 3. This described the development of writing systems from simple pictorial representations through to complex abstract representations. This is a good analogy of the future development of computer interfaces since both language and user interfaces are means of communication. Thus, the following parallels can be drawn between these two cases,

- Writing developed recursively through increased use, the same is true of user interfaces;

- The development of writing systems was a result of an increase in social activity and a commensurate increase in the need for communication. The same is true of computer users in an increasingly community based environment (e.g. Internet services; WWW, Chat/Talk, Video Conferencing, MediaSpace). Thus, communication techniques through the interface must evolve to keep pace with this development;

- A second impact on language development was natural evolution in the honing of the language itself. This included the development of more economical symbol manifestations (e.g. Egyptian *demotic* script identified in Chapter 3), or the increased use of complex, non-analogous symbols. In the same way, computer systems must develop to keep pace with the expectations and skills of the user. There is no longer any need for the assumptions which underpin many usability guidelines and interface *organising* metaphors, i.e. a naive user in an isolated, low-bandwidth environment.

The importance of this parallel is also addressed by Gentner & Nielson (1996) in their critique of the Apple Macintosh user interface. They state:

"It is as if we have thrown away a million years of evolution, lost our facility with expressive language, and been reduced to pointing at objects in the immediate environment", (Gentner and Nielson, 1996: pp. 75).

The paucity of expressive representations in user interfaces can be made clear by returning to the representation taxonomy of Chapter 3. This showed classes of different representations based on the way that they encode information. By emboldening the arcs of the graph to show where most representations in user interfaces would be categorised, the spaces left show there are a wide variety of representational forms which are not used. (See Figure 12.1 over).

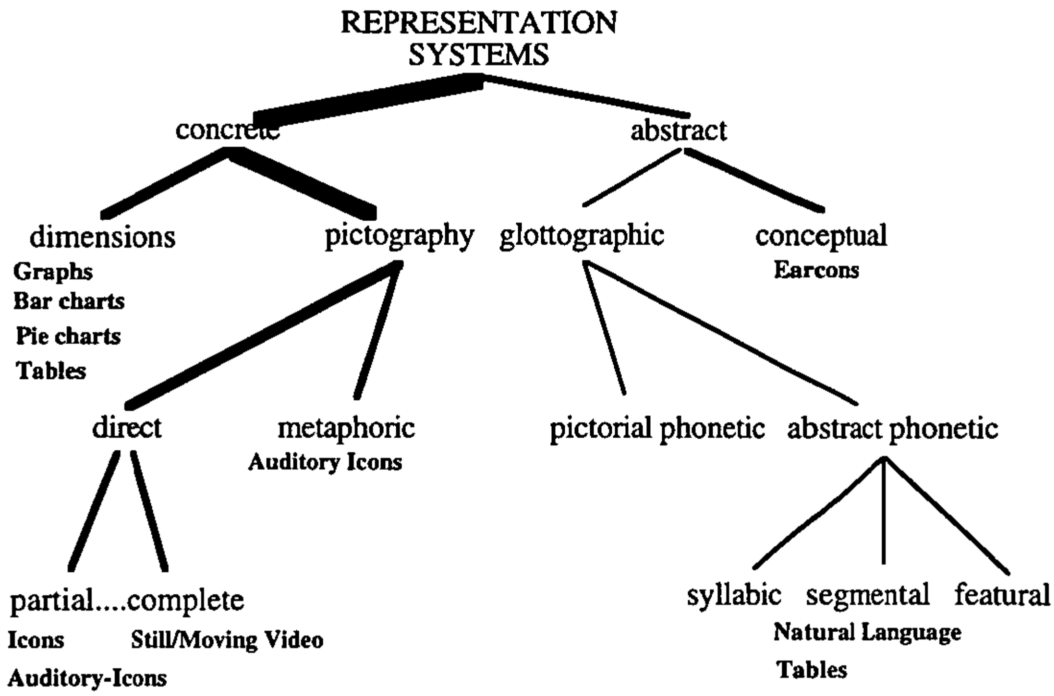
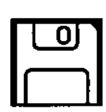


Figure 12.1: The range of representations that are not used

Of course, any increase in the expressiveness of an interface component must be in lock-step with the types of tasks which the user must achieve. Thus, it is wrong for Gentner and Nielson to criticise aspects of the Macintosh interface which are at the right level of expressiveness for the task. For example, the presence of a disk in the floppy drive is represented by an icon:



This is ideal, if presence or not is wished to be known. In this case, more expressive natural language would be unnecessary and perhaps even a hindrance, e.g.

There is a Macintosh disk in the floppy drive.

The Macintosh interface is ideal for such simple task (with a small state space to navigate) but where it suffers is in the articulation of more complex tasks (with a large state space to navigate), e.g. *find me all files belonging to David.* Ironically, it is the UNIX/DOS interface which provides the expressive power to achieve this kind of tasks through its highly configurable and malleable command language, i.e. the command-line interfaces which preceeded the GUI.

In conclusion, the changes in the way computers are used means that interfaces must also change. Only by providing more expressive media to match the more expressive, human-like communication that is becoming more widespread can interfaces provide the user with the support they require. The expressiveness and mental model theory provides a framework for describing media which cover a range of expressiveness, and can therefore be useful in allocating media in such tasks. Care must be taken however, that the positive lessons learnt form the use of inexpressive media; in present graphical user interfaces are not forgotten. This is the challenge for the next generation of user interfaces.

12.2 Further Work

The changes in the way computers are used, and the rapid development of interface technologies will ensure that methodologies for interface design will need constant revision. With this in mind, this section describes how the outlined framework based on the cognitive/representational approach should be developed further.

At a theoretical level, there is still a need for more fundamental investigations of information representation in user interfaces. Of particular interest are the media which allow expressiveness through emergent perceptual pragmatics. Further studies into how obvious these pragmatics are to users are required if they are to relied upon as information carriers in interfaces. Moreover, the study of high expressiveness systems such as natural languages should be fed into user-interface design, particularly given the trend of social computing with its need for more expressive media.

On the empirical front, further studies are required to advance this preliminary investigation. One way is to provide sufficient knowledge of induced mental models for a wider range of media. This knowledge can then be used to validate the theoretical expressiveness grading.

As well as the study of single media, it is important to investigate the effect of different media in combination, as in Alty et al. (1992). The experimental environment did have the facility to display multiple representations of the traffic domain, but there was insufficient time to incorporate this into the main study. The use of multiple media means that the issues of consistency, completeness, coherence, and redundancy outlined in Chapter 1, must be addressed. These four characteristics must be related to the mental model of a domain which will be induced by an interface. Furthermore, the issue of coherence, which was highlighted in the interface examples of Chapter 11, must also be addressed. Here, the expressiveness of an interface changed as the task developed. The effect this change in expressiveness will have on the mental models that have *already* been developed is interesting and requires further investigation. Generally, the experimental study must be developed to ensure more significant results.

By applying the investigative process outlined in this thesis, different media combinations can be classified in terms of mental models. This knowledge can then be used to define to:

- Validate theoretical descriptions of expressiveness;
- Allow comparison of media in different combinations by analysing the models they induce.

If pursued, the second point would provide the evidence that is lacking in the multimedia literature, cognitive justification of the use of multiple media.

The studies should ensure adequate training time and subjects from a wide variety of backgrounds. The types of domains that are investigated should also be varied. These could range from the highly specialised such as process control to the more general such as operating systems. By providing an empirical basis for the classification of a wider variety of media, the interface designer can make more informed decisions about when to use them.

In conclusion, the melding of cognitive investigations of output media into mainstream HCI research should be a priority, particularly with the advent of new output media. This offers an overtly user-centred description of media allocation rather than the task-centred or technology-centred approaches that have been described. This should not be to the exclusion of socially oriented studies such as Suchman (1987), although they themselves argue for an end to the use of cognitivism in HCI. The author sees the key is that non-technological issues should be central to the allocation of media in user interfaces. Whether this is the study of group-working, cognitive theories like mental models, or task analysis; these theoretical studies, validated by empirical investigations, should take the lead in user interface design.

References

Allen, R. E. (1990) "The Concise Oxford Dictionary of Current English". *(R. E. Allen, Ed.), (Oxford, UK: Clarendon Press).*

Alty, J. L. and M. J. Coombs. (1984) "Expert Systems: Concepts and Examples". *(National Computing Centre Publications: Manchester, UK).*

Alty, J. L., C. D. C. McCartney. (1991) "Managing Multimedia Resources in Process Control: Problems and Solutions". *In Proc. of 1st Eurographics Workshop on Multimedia (L. Kjelldahl, Ed.), (Stockholm, Sweden: April), (London: Springer-Verlag)*, pp. 293-306

Alty, J. L., M. Bergen, P. Crauford, C. Dolphin. (1992) "Can Multimedia Become a Benefit Rather than a Feature?". *In Proc. of 2nd Eurographics Workshop on Multimedia, (C. Hornung, Ed.), (Darmstadt, Germany: June), (London: Springer Verlag)*, pp. 27-50

Alty, J. L., M. Rijkaert. (1993) "Process Operators Multimedia Intelligent Support Environment". *Final Technical Report, ESPRIT Project 2397*

Alty, J. L. (1995) "Can We Use Music in the Computer-Human Interface". In Proc of HCI'95: People and Computers X, (D. Diaper, Eds.), (Huddersfield: August), (Cambridge, UK: CUP), pp. 409-424.

Anderson, J. R., G. H. Bower. (1973) "Human Associative Memory". *(Washington, DC: Winston).*

Andre, E., T. Rist. (1993) "The Design of Illustrated Documents as a Planning Task". *In Intelligent Multimedia Interface Design (M. Maybury, Ed.), (Cambridge, MA: MIT Press)*, pp. 94-116.

Andre, E., W. Finkler, W. Graf, T. Rist, A. Schnauder. (1993) "WIP: The Automatic Synthesis of Multimodal Presentations". *In Intelligent Multimedia Interface Design (M. Maybury, Ed.), (Cambridge, MA: MIT Press)*, pp. 75-93.

Apple Computer (1992) "Macintosh Human Interface Guidelines". (Reading, MA: Addison-Wesley)

Arens, Y., L. Miller, C. S. Shapiro, N. K. Sondheimer . (1988) "Automatic Construction of User Interface Displays". *Technical Report, Information Science Institute, University of Southern Calif, Sept.*

Arens, Y. (1991) "When is a Picture Worth a Thousand Words". *Technical Report, Information Science Institute, University of Southern Calif, Sept.*

Arens, Y., E. Hovy, M. Vossers. (1993) "On the Knowledge Underlying Multimedia Presentations". *In Intelligent Multimedia Interface Design (M. Maybury, Ed.), (Cambridge, MA: MIT Press),* pp. 280-306.

Bainbridge, L. (1992) "Mental Models in Cognitive Skill". *In Models of the Mind: Theory, Perspective and Application (Y. Rogers, A. Rutherford, P. A. Bibby, Eds.), (London: Academic Press),* pp. 119-143.

Baker, S., E. Marshall. (1988) "Evaluating the Man-Machine Interface - The Search for Data". *Training, Human Decision Making and Control (J. Patrick, K. D. Duncan, Eds.), (London: Elsevier),* pp. 79-90.

Barnard, J. (1992) "Multimedia and the Future of Long Distance Learning Technology". *In Journ. of Educational Media, Vol. 29(3),* pp. 139-144

Barwise, J., J. Etchmendy. (1991) "Visual Information and Valid Reasoning". *In "Visualisation in Mathematics" (W. Zimmerman, S. Cunningham, Eds.), (US: Maths. Assoc. of America),* pp. 9-24.

Beggs, I., A. Paivio. (1969) "Concreteness and Imagery in Sentence Meaning". *In Journ. of Verbal Learning and Verbal Behaviour, Vol. 8,* pp. 821-827.

Bergan, M. (1995) "The Design and Evaluation of Multimedia User Interfaces in Process Control". *PhD Thesis, Dept. of Computer Studies, Strathclyde University, UK.*

Bernsen, N. O. (1994) "A Research Agenda for Modality Theory". *In Interacting with Computers, Vol. 6(4).*

Bertin, J. (1983) "Graphics and Information Processing". *(Paris: De Greyter).*

Blattner, M. M. (1993) "Sound in the Multimedia Interface". *In Proc. of Educational Multimedia and Hypermedia (H. Maurer, Ed.), (Orlando, FL: June), (VA: AACE Press)*, pp.76-82

Borgman, C. L. (1986) "The User's Mental Model of an Information Retrieval System: An experiment on a prototype on-line catalogue".*IJMMS, Vol. 24,* pp. 47-64.

Bos, E. (1995) "Comprehending User Behaviour Using Psycholinguistics". *In Behaviour and Information Technology, Vol. 14(6)*, pp. 342-357.

Brachman, R.J. (1979) "On the Epistemology of Semantic Networks". *In Associative Networks: Representation and Use of Knowledge by Computers (N. Findler,Ed.), (New York: Academic Press)*, pp. 3-50 .

Brachman, R. J., H. J. Levesque. (1984) "The Tractability of Subsumption in Frame-Based Description Languages". *In Proc. of AAAI-84, (Austin, TX: April)*, pp.34-37.

Brachman, R.J. and J. G. Schmolze (1985) "An Overview of the KL-ONE Knowledge Representation System". *In Cognitive Science, Vol. 9(2)*, pp. 171-216.

Branford, J. D., J. R. Barclay, J. J. Franks (1972) "Sentence Memory: A Constructive vs. Interpretive Approach". *Cognitive Psychology, Vol. 3,* pp193-209.

Bransford, J. D., M. K. Johnson. (1972) "Contextual Prerequisites for Understanding: Some Investigations of Comprehension and Recall".*Verbal Learning and Verbal Behaviour, Vol. 11,* pp. 717-726.

Bresnahem, S., T. Ducas, A. Rubin. (1994) "Cartwheeling in CAMMOTION". *In Hands On, Vol. 17(2)*, pp. 8-19.

Bretz, R. (1983) "A Taxonomy of Communication Systems". *(Englewood Cliffs)*

Brewster, S. A., P. C. Wright, A. D. N. Edwards. (1993) "An Evaluation of Earcons for Use in the Auditory HCI Interface". *In Proc. of INTERCHI '93 (S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel, T. White, Eds.), (Amsterdam: April), (New York: ACM Press)*, pp.221-227.

Buckhart, B. J., M. E. Fucso (1996) "Using Animation to Aid Process Flow Visualisation". *In Companion of ACM SIGCHI'96, (M. J. Tauber, Ed.), (New York: ACM Press)*, pp. 21-22

Burger, J. D., R. J. Marshall. (1993) "An Application of Natural Language Models to Intelligent Multimedia". *In Intelligent Multimedia Interface Design (M. Maybury, Ed.), (Cambridge, MA: MIT Press)*, pp. 174-196.

Burton-Roberts, N. (1986) "Analysing Sentences: An Introduction to English Syntax". *(London: Longman).*

Buttigieg, M. A. (1989) "Emergent Features in Visual Display Design for Two Types of Failure Detection Tasks". *Technical Report 8904, EPRL, University of Illlnois at Urbana, Illonois.*

Buxton, W. (1990) "The Natural Language of Interaction". *In 'The Art of User Interface Design' (B. Laurel, Ed.), (London: Addison-Wesley)*, pp 403-416.

Card, S. K., T. P. Moran, A. Newell (1983) "The Psychology of Human-Computer Interaction", (Hillsdale, NJ: LEA)

Card, S. K. (1996) "Pioneers and Settlers: Lessons from Successful User Interface Design". *In Human Computer Interface Design, (M. Rudisill, C. Lewis, P. B. Polson, T. D. McKay, Eds.), (San Francisco, US: Morgan Kaufman)*, pp. 122-169

Carroll, J. M., Thomas, J. C. (1982) "Metaphor and the Cognitive Representation of Computing Systems".*IEEE Trans on Systems, Man and Cybernetics, Vol. 12(2)*, pp. 107-115.

Carswell, D. M., D. Wickens. (1987) "Information Integeration and the Object display. An interaction of task demands and display superiority". *Ergonomics, Vol. 30(3)*, pp.511-527.

Casner, S. M. (1991) "A Task Analytic Approach to the Automated Design of Graphical Presentations". *In ACM Transaction on Graphics, Vol. 10(2)*, pp. 111-151

Chapanis, A. (1981) "Interactive Human Communication: Some Lessons Learned from Laboratory Experiments". *In Man-Computer Interaction: Human Aspects of Computers (B. Shackel, Ed.), (New York: Sijhoff and Noordoff)*, pp. 65-114

Chomsky, N. (1965) "Aspects of the Theory of Syntax". *(Cambridge, MA: MIT Press).*

Clark, H. H. (1972) "On the Process of Comparing Sentences Against Pictures". *In Journ. of Cognitive Psychology, Vol. 3*, pp. 472-517.

Cleveland, W. S., R. McGill. (1986) "An Experiment in Graphical Perception". *In Inter. Journ. of Man-Machine Studies, Vol. 25(5)*, pp. 491-500.

Coll, B., A. Thyagarajin, S. Chopra. (1993) "Experiental Study Comparing the Effectiveness of Computer Graphical Data vs. Computer Tabulated Data". *In IEEE Transactions on Systems, Man and Cybernetics, Vol. 21(4)*, pp. 897-900.

Coll, R. A., J. H. Coll, G. Thakur. (1994) "Graphs and Tables: A Four Factor Experiment". *In Communications of the ACM, Vol. 37(4)*, pp. 77-88.

Cornell, M., B. Parkwoolf, D. Suthers. (1993) "Live Information in a Multimedia Framework". *In Intelligent Multimedia Interface Design (M. Maybury, Ed.), (Cambridge, MA: MIT Press)*, pp. 309-327.

Coury, B., M. D. Boutlette, R. A. Smith. (1989) "Effect of Uncertainty and Diagnosticity on Classification of Multidimensional Data with Integral and Seperable Displays of System Status". *Human Factors, Vol. 31(5)*, pp.551-569.

Coutaz, J., D. Salber, A. Blanford, J. May, R. M. Young. (1995) "Four Easy Pieces for Assessing the Usability of Multimodal Interaction. The CARE properties". *In Proc. of INTERACT '95(Nordby, Helmerson, Gilmore, Arnesen. Chapman and Hall, Eds.), (Oslo, Norway,:August)*, pp. 115-120.

Cox, K. C., G. Roman. (1992) "Abstraction in Algorithmic Animation". *In Proc. of IEEE Workshop on Visual Languages, (Seattle: WA)*, pp.18-23.

Craik, F. I. M., E. Tulving. (1975) "Depth of Processing and the Retention of Words in Episodic Memory". *In Journ. of Experimental Psychology, Vol. 104(3)*, pp. 268-294.

Daiper, D., M. Addison. (1991) "User Modelling: Task Orientated Modelling (TOM) Approach to the Design Model". *In Proc. of HCI'91: People and Computers 6, (D. Daiper, R. Hammand, Eds.), (York: August), (Cambridge, UK: Cambridge Uni. Press)*, pp. 387-399

Daiper, D. (1993) "Liverpool University Task Analysis for Knowledge Description" (LUTAKD) *LUTAKD Tutorial Notes*

De Kleer, J., J. S. Brown. (1983) "Assumtions and Ambiguities".*In Mental Models (D. Gentner, A. L. Stevens, Eds.), (Hillsdale, NJ: LEA)*, pp. 155-191.

Dickson, G. W., G. DeSanctis, M. J. McBride. (1986) "Understanding the Effectiveness of Computer Graphics for Decision Support". A Cumulative Experimental Approach.*In Communications of the ACM, Vol. 29(1)*, pp. 913-923.

DiGano, C. J., R. M. Backer, R. N. Owen. (1993)Logomedia: A Sound-Enhanced Programming Environment for Monitoring Program Behaviour.*In Proc. of INTERCHI '93 (S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel, T. White, Eds.), (Amsterdam: April), (New York: ACM Press)*, pp.301-302.

Dix, A., J. Finlay, G. Abowd, R. Beale. (1993) "Human Computer Interaction". *(London, Prentice-Hall).*

Douglas, S., C. Hundhausen, D. McKeown. (1995) "Towards Emprically-Based Visualisation Languages". *In IEEE Symposium on Language Visualisation.*

Draper, S., D. A. Norman, S. Hollan. (1986) "User Centred System Design", (Hillsdale, NJ: LEA)

Edmonds, E.A., E. MacDaid. (1990) "An Architecture for Knowledge Based Front Ends". *In The Seperable User Interface (Edmonds E. A, Ed.), Computers and People Series, (London: Academic Press)*, pp. 261-269.

Egan, D. E, D. D. Grimes-Farrow. (1982) "Differences in Mental Representations Spontaneously Adopted for Reasoning". *In Memory and Cognition, Vol. 10(4)*, pp. 297-307.

Earnshaw, R. A., N. Wiseman. (1993) "An Introduction to Scientific Visualisation". *(Berlin: Springer-Verlag).*

Falzon, P. (1990) "Human-Computer Interaction: Lessons Learned from Human-Human Communication". *In Cognitive Ergonomics: Understanding Human-Computer Interaction (P. Falzon, Ed.), (London: Academic Press),* pp. 51-66

Faraday, P. (1995) "Evaluating Multimedia Presentations for Comprehension".*In. Doctoral Consortium, SIGCHI '95 (Denver: US, April/May), (ACM Press),* pp. 49-50.

Feiner, S. (1988) "A Grid Based Approach for Automatic Display Layout". *In Proc. of Graphics Interface'88,* pp. 192-197.

Feiner, S., K. R. McKeown. (1993) "Automating the Generation of Coordinated Multimedia Explanations". *In Intelligent Multimedia Interface Design (M. Maybury, Ed.), (Cambridge, MA: MIT Press),* pp. 117-138.

Fish, R. S., R. E. Kraut, R. W. Root. (1992) "Evaluating Video as a Form of Informal Communication". *In Proc. of SIGCHI '92, (New York: ACM Press),* pp. 37-48

Frolich, D. M. (1991) "The Design Space of Interfaces". *In Proc. of 1st Eurographics Workshop on Multimedia (L. Kjelldahl, Ed.), (Stockholm, Sweeden: April), (London: Springer-Verlag),* pp. 53-70.

Furnas, G. W. (1986) "Generalised Fisheye Views". In ACM SIGCHI'86, (Boston, MA: April), pp. 16-23

Gale, S. (1990) "Human Aspects of Interactive Multimedia Communciation". *Interacting with Computers, Vol. 2(2), (London: Butterworth-Heinemann),* pp. 175-189.

Gaur, A. (1984) "The History of Writing". *(London: British Library Reference Section).*

Gaver, W. (1989) "The SonicFinder: An Interface that Uses Auditory Icons". *In Human Computer Interaction, Vol. 4,* pp. 76-94

Gaver, W., T. Moran, A. Maclean, L. Lovestand, P. Dourish, K. Carter, W. Buxton. (1991) "Realizing a Video Environment: EUROPARC'S RAVE System". *In Proc. of SIGCHI '91, (New York: ACM Press)*, pp. 27-36.

Gaver, W. (1993) "Synthesising Auditory Icons" *.In Proc. of INTERCHI '93 (S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel, T. White, Eds.), (Amsterdam: April), (New York: ACM Press)*, pp. 228-235.

Gentner, D., D. R. Gentner. (1983) "Flowing Waters or Teeming Crowds: Mental Models of Electricity". *In Mental Models (D. Gentner, A. L. Stevens, Eds.), (Hillsdale, NJ: LEA)*, pp. 99-128.

Genter, D. and J. Nielson (1996) "The Anti-Mac Interface". *In Communications of the ACM, Vol. 39(8)*, pp. 71-82.

Gershon, N., S. G. Eick (1995) "Visualisations New Tack: Making Sense of Information". *In IEEE Spectrum*, pp. 38-56

Gershon, H., K. Andrews, S. G. Eick, J. Foley, W. Ruh. (1996) "Visualising the Internet: Putting the User in the Driving Seat". *Panel at ACM SIGCHI'96, (M. Tauber, Ed.), (Vancouver, BC: April), (New York: ACM Press), Companion*, pp. 163-164.

Gibbs, S., C. Breiteneder, V. de May, M. Papathomas. (1993) "Video Widgets and Video Actors". *In Proc. UIST '93, (Atlanta, Georgia)*, pp. 179-185.

Gick, M. L., K.J. Holyoak. (1980) "Analogical Problem Solving". *Cognitive Psychology, Vol. 12*, 306-355.

Gilmore, D. J. (1991) "Visibility: A Dimensional Analysis". *In Proc HCI'91: People and Computers VI, (D. Diaper, N. Hammond, Eds.), (Cambridge, UK: CUP)*, pp. 317-330.

Goodman, M. (1968) "The Languages of Art". (New York: Bobbs Merrill)

Grant, S., A. T. Mayes. (1991) "Cognitive Task Analysis". *HCI in Complex Systems (J. L. Alty, G. R. S Weir, Eds.), People and Computers Series, (London: Academic Press)*, pp.148-167.

Green, T. R. G. (1991) "Describing Information Artefacts with Cognitive Dimensions and Structure Maps", *In Proc HCI'91: People and Computers VI, (D. Diaper, N. Hammond, Eds.), (Cambridge, UK: CUP)*, pp. 297-316.

Green, T. R. G. and D. Benyon. (1992) "Displays as Data Structures: Entity-Relationship Models of Information Artefacts". *Technical Report 92/22 (Dept. of Computing, Open University: Milton Keynes, UK)*

Green, T. R. G., S. P. Davies, D. J. Gilmore (1996) "Delivering Cognitive Psychology to HCI: the Problems of Common Langauge and of Knowledge Transfer". *In Interacting with Computers, Vol. 8(1)*, pp. 89-11

Greenberg, S., C. Gutwin, A. Cockburn (1996) "Using Distortion-oriented Displays to Support Workspace Awareness". *In Proc of HCI'96: People and Computers XI, (S. Cunnigham, A. Sasse, R. Winder, Eds.), (August: Imperial College, London), (London, UK: Springer-Verlag)*, pp. 299-314.

Greeno, J .G. (1983) "Conceptual Entities". *In Mental Models (D. Gentner, A. L. Stevens, Eds.), (Hillsdale, NJ: LEA)*, pp.227-252.

Grossen , G. and D. Carnine (1990) "Diagramming a Logic Strategy: Effects on difficult problem types and transfer". *Learning Disibility Quarterly, Vol. 3*, pp. 168-182

Grosz, B. J. (1986) “Attention, Intention and the Structure of Discourse”, *Computational Linguistics, Vol. 12(3), (July/September)*, pp. 175-205.

Haberbeck, R. (1993) "M4 Eval: Evaluation of Multimedia and Multimodal HCI". *In Human Aspects in Computing: Design and Use of Interactive systems and Information Management, (H. J. Bullinger, Ed.), (Holland: Elsevier)*

Hayes, J. R., H. A. Simon (1977) "Psychological Differences Among Problem Isomorphs". *In Cognitive Theory (N. J. Castellan, D. B. Pisoni, G. R. Potts, Eds.).*

Heim, I. (1983) "The Semantics of Definite and Indefinite Noun Phrases", *Ph.D. Thesis, University of Massachusetts, (Amherst, MA).*

Hietala, P., J. Nummenmaa (1992) "MEDUSA-a Multimodal Database Uswr Interface with Computer Supported Learning Aid", Internal Report, (Uni. of Tampere: Finland)

Higgenbotham, J.. (1988) "Contexts, Models, and Meanings: a note on the data of semantics". *In "Mental Representations: The Interface Between Language and Reality" (R. M. Kempson, Ed.), (Cambridge, UK: Cambridge Uni. Press)*, pp. 29-48.

Hollan, J. D., E. L. Hutchins, L. M. Weitzman. (1984) "STEAMER: An Inspectable Simulation-based Training System". *AI Magazine(summer), (American Association for Artificial Intelligence)* .

Hollnagel, E. (1993) "The Design of Reliable HCI: The Hunt for Hidden Assumptions". *In Proc. of HCI'93: People and Computers VIII, (J. L. Alty, D. Diaper, S. Guest, Eds.), (Cambridge, UK: CUP)*, pp. 3-15

Holyoak, K. J. (1987) "Surface and Structural Simililarity in Analogical Transfer". *Memory and Cognition, Vol. 15(4)*, pp. 332-340.

Horton, W. (1996) "Designing Icons and Visual Symbols". *In Proc. of ACM SIGCHI'96 (M. J. Tauber, Ed.), (Vancouver, BC, Can.: April), (New York: ACM Press), Companion* , pp. 371.

Howell, R. (1976) "Ordinary Pictures, Mental Representations, and Logical Forms". *In "Synthese", Vol. 33(1)*, pp. 149-174.

Hunt, M. J. (1989) "Speech is More Then Just an Audible Version of Text". *In 'The Structure of Multimodal Dialog' (M. M. Taylor, F. Neel, D. G. Bouwhuis, Eds.), (North-Holland: Elsevier)*, pp. 287-299.

Hutchins, E. (1989) "Metaphors for Interface Design". *In The Structure of Multimodal Dialoge (M. M. Taylor, F. N. Neel, D. G. Bouwhuis, Eds.), (London: Elsevier)*, pp. 11-28

IBM (1996) "Graphics for BBC Televison: Wimbledon Coverage", *BBC Televison, June.*

Jarvenpaa, S. L. (1989) "The Effect of Task Demands and Graphical Format on Information Processing Strategies". *In. Management Science, Vol. 35(3)*, pp. 285-295.

Jacob, R. J. K., H. E. Egeth. (1976) "The Face as a Data Display". *Human Factors, Vol. 12(2)*, pp.189-200.
Jones, B. J. (1995) "Variability and Universality in Human Image Processing". *In "Understanding Images: Finding Meaning in Digital Imagery" (F. Marchese, Ed.), (Holland: Springer-Verlag)*, pp. 199-241.

Johnson, P. (1989) "Supporting System Design by Analysing Current Task Structures". *In "Tasks Analysis", (D. Diaper, Ed.), (London: Ellis-Horwood).*

Johnson-Laird, P. N. (1983) "Mental Models". *(Cambridge, UK: Cambridge Uni. Press).*

Jones, S. (1970) "Visual and Verbal Processing in Problem-Solving". *In Cognitive Psychology, Vol. 1*, pp. 201-214.

Kapolka, M. A. (1991) " A Study in Visual Natural Language: Numbers and Dates". *In "Visual Languages and Visual Programming", (S. Chang, Ed.), (Pergammon Press)*, pp. 47-69.

Keller, P. M., M. M. Keller. (1993) "Visual Clues: Practical Data Visualisation". *(Brussels, Belgium: IEEE Computer Press)*

Kintsche, W., T. A. van Dijk (1977) "Toward a Model of text Comprehension", *Psychological Review, Vol. 85*, pp. 363-394.

Krebbs, R., G. Davies. (1987) "The Design of Signals". *In 'Ecology and Evolution', (London: Addison-Wesley).*, pp. 319.

Lamping J. and R. Rao (1996) "Visualising Large Trees using the Hyperbolic Browser". *In Companion of ACM SIGCHI'96, (D. Tauber, Ed.), (New York: ACM Press), (Vancouver, BC: April)*, pp. 388-389

Lave, J. (1988) "Cognition in Practice". *(Cambridge, UK: CUP).*

Levesque, H. (1986) "Making Believers out of Computers". *In Journ. of Artificial Intelligence, Vol. 30*, pp. 81-108.

Levesque, H. (1988) "Logic and the Complexity of Reasoning". *In Journ. of Philosophical Logic, Vol. 17*, pp. 355-89.

Lewis C., Rieman, J. (1993) "Task-Centred User Interface Design: A Practical Introduction", (London, Heineman), Chapter 2.

Lieberman, H. (1996) "Intelligent Graphics". *In Communications of the ACM, Vol. 39(8)*, pp. 38-48.

Lohse, J., H. Rueter, K. Biolsi, N. Walker. (1990) "Classifying Visual Knowledge Representations: A Foundation for Visualisation Research".*In Proc. of Visualisation90 (A. Kaufman, Ed.), (California), (IEEE Press)*, pp. 131-137.

Mackinlay, J. (1986) "Automating the Design of Graphical Presentations of Relational Information". *In ACM Transactions on Graphics, Vol. 5(2)*, pp. 110-141

Mackinlay, J., J. D. Robertson, S. K. Card. (1991) "The Perspective Wall: Details and Context Smoothly Integrated". *In Proc. of ACM SIGCHI'91, (S. P. Robertson, G. M. Olson, J. S. Olson, Eds.), (New York: ACM Press)*, pp. 173-179.

Macmillan, T. (1989) "Interactive Multimedia Meets the Real World". *In Computer Graphics World, (August)*, pp. 39-46.

Mani, K., P. N. Johnson-Laird. (1982) "The Mental Representation of Spatial Descriptions." *In Memory and Cognition, Vol. 10(2)*, pp. 181-187.

Mannoni, B. (1996) "Bringing Museums Online". *In Communications of the ACM, Vol. 39(6)*, pp. 100-105.

Maies, P. "Autonomous Agents: An MIT Perspective". *Keynote address at HCI'96: People and Computers XI, (August: Imperial College, London).*

Marks, J. (1991) "A Formal Specification Scheme for Network Diagrams that Facilitates Automatic Design" *In Joun. of Visual Languages and Computing, Vol. 2(4)*, pp. 395-414.

Marmollin, H. (1991) "Multimedia from the Perspectives of Psychology". *In Proc. of 1st Eurographics Workshop on Multimedia (L. Kjelldahl, Ed.), (Stockholm, Sweden: April), (London: Springer-Verlag),* pp. 39-52

Maybury, M. T. (1993) "Planning Multimedia Explanations Using Communicative Acts". *In Intelligent Multimedia Interface Design (M. Maybury, Ed.), (Cambridge, MA: MIT Press),* pp. 59-74.

Mayes, A. T., S. W. Draper, A. M. McGregor, K. Oatley (1988) "Information Flow in the User Interface: The effect of experience and recall on the content of MacWrite screens". *In People and Computers IV, (D. M. Jones, R. Winder, Eds), Cambridge: CUP).*

Mayes, A. T. (1992a) "Cognitive Tools: A Suitable Case for Learning". *In Cognitive Tools for Learning, (P. A. M. Kommers, Ed.), (Germany: Springer Verlag),* pp. 7-12.

Mayes, A. T. (1992b) "The 'M -Word': Multimedia Interfaces and Their Role in Interactive Learning Systems". *In Multimedia Interface Design in Education (A. D. N. Edwards, S. Holland, Eds.), NATO ASI Series, (London: Springer-Verlag),* pp. 2-22.

McCleary, G. F. (1983) "An Effective Graphical Vocabulary". *In IEEE Computer Graphics and Applications, Vol. 3(2),* pp. 46-53.

McGonigle, B., M. Chalmers. (1986) "Representations and Strategies During Inference". *In Reasoning and Discourse Processes (T. Myers, K. Brown, B. McGonigle), (London: Academic Press),* pp. 141-165.

McMillan, T. (1989) "Interactive Multimedia Meets the Real World". *In Computer Graphics World, (August),* pp.39-46.

Medyckyj-Scott, D. J., M. Blades. (1991) "Cognitive Representations of Space and the Design and Use of Geographical Information Systems". *In Proc. of HCI'91: People and Computers VI (D. Diaper, N. Hammond, Eds.), (York, UK: August), (Cambridge, UK: Cambridge Uni. Press),* pp. 421-432.

Mezler,J., Shepard, R. N. (1974) "Transformational Studies of the Internal Rotation of Three-Dimensional Objects". *In Theories of Cognitive Psychology: The Loyola Symposium (R. L. Solso, Ed.), (Hillsdale, NJ: LEA).*

Miah, T. (1994) "Multimedia Conferencing at the Desktop". *MHVR, Moscow.*

Moray, N., J. Lootstein, J. Pajak. (1986) "Acquisition of Process Control Skills" . *In IEEE Trans. on Systems, Man and Cybernetics, Vol. 16(4),* pp. 497-504.

Moray, N. (1987) "Intelligent Aids , Mental Models and the Theory of Machines". *JMMS, Vol. 27,* pp. 619-629.

Morin, D. (1992) "Multimedia Overview: Status and Issues". *Internal report for THOMAS-SYSECA, (Saint-Cloud: France).*

Mukherjea, S. and J. D. Foley (1995) "Visualising the World Wide Web with the Navigational View Builder". *In Computer Networks and ISDN Systems, Vol. 27,* pp. 1075-1087.

Newell, A., J. C. Shaw, H. A. Simon (1958) "Elements of a Theory of Problem Solving". *In Psychological Review, Vol. 65*, pp. 151-166

Newell, A. (1982) "The Knowledge Level". *In Artificial Intelligence, Vol. 18*, pp. 87-127.

Nigay, L., J. Coutaz. (1993) "A Design Space for Multimodal Systems: Concurrent Processing and Data Fusion" *In Proc. of INTERCHI '93 (S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel, T. White, Eds.), (Amsterdam: April), (New York: ACM Press),* pp. 172-178.

Nigay, L., J. Coutaz. (1995) "Multifeature Systems: From HCI Properties to Software Design". *Symposium on Multimodal User Interface Design (July), (Edinburgh University: UK).*

Nisbett, R. E., T. D. Wilson. (1977) "Telling More Than We Can Know: Verbal Reports on Mental Processes". *Psychological Review, Vol. 84(3),* pp. 231-259.

Norman, D. A. (1983) "Some Observations on Mental Models". *In Mental Models (D. Gentner, A. L. Stevens, Eds.), (Hillsdale, NJ: LEA),* pp. 7-15.

Norman, D. A. (1986) "Cognitive Engineering". *In User Centred System Design (S. Draper, D. A. Norman, S. Hollan, Eds.), (Hillsdale, NJ: LEA)*, pp. 133-163.

O'Mally, C., S. Draper. (1992) "Representaion and Interaction: Are mental models all in the mind?". *In Models of the Mind: Theory, Perspective and Application(Y. Rogers, A. Rutherford, P. A. Bibby, Eds.), (London: Academic Press)*, pp. 73-91.

Ornstein, R., L. Carstensen. (1991) "Psychology: The Study of Human Experience". *(London: Harcourt, Brace, and Javonovich).*

Paivio, A. (1969) "Mental Imagery in Associative and Learning Memory". *In Psychological Review, Vol. 76(3)*, pp. 241-263.

Paivio, A. (1975) "Perceptual Comparisons Through the Mind's Eye". *In Memory and Cognition, Vol. 3(6)*, pp. 635-647.

Paivio, A. (1978) "A Dual Encoding Approach to Perception and Cognition". *In Modes of Perceiving and Processing Information (H . L. Pickman, E. Saltzman, Eds.), (Hillsdale, NJ: LEA)*, pp. 39-51.

Palmer, S. E. (1978) "Fundemental Aspects of Cognive Representations". *In Cognition and Categorisation"(E. Rosch, B. B. Lloyd, Eds.), (Hillsdale, NJ: LEA)*, pp. 259-303.

Payne, S. J., T. R. G. Green. (1986) "Task Action Grammars (TAG): A model of the Mental Representations of Task Languages". *In Human-Computer Interaction, Vol. 2*, pp. 93-133

Payne, S. (1991) "A Descriptive Study of Mental Models". *In Journ. of Behaviour and I.T., Vol. 10(1)*, pp. 3-21.

Payne, S. (1992) "On Mental Models and Cognitive Artefacts". *In Models of the Mind: Theory, Perspective and Application(Y. Rogers, A. Rutherford, P. A. Bibby, Eds.), (London: Academic Press)*, pp. 103-118.

Pearce, J. (1983) "Graphs are Not Straighforward". *In The Psychology of Computer Use (T. R. G. Green, S. J. Payne, G. C. Van der Veer, Eds.), People and Computer Series, (London: Academic Press)*, pp. 41-56.

Pearce, C. J. (1994) "Dialog Semiosis". *In Advances in Semiotics (N. Johanson, Trans.), (ILL: Bloomington).*

Pfaff, G. E. (1985) "User Interface Management Systems". *In Proc. of the Workshop on User Interface Management Systems, (G. E., Pfaff, Ed.), (Seeheim), (Berlin: Springer-Verlag).*

Pick, H. L., J. L. Lockman. (1983) "Map Reading and Cognitive Skill: Discussion". *In "Spatial Orientation: Theory, Practice, and Application", (H. L. Pick Jr., L. P. Acredolo, Eds.), (London: Plenum Press),* pp.219-224.

Plaisant, C., B. Milash, A. Rose, S. Widoff, B. Schneiderman. (1996) "Lifelines: Visualising Personal Histories". *In Proc. of ACM SIGCHI'96 (M. J. Tauber, Ed.), (Vancouver, BC, Can.: April), (New York: ACM Press),* pp.221-227.

Potosnak, K. (1988) "Do Icons Make User Interfaces Easier to Use?". *Human Factors Special, IEEE Software (May),* pp. 97-99.

Prathakaram, B., S. V. Raghavar (1994) "Synchronisation Models for Multimedia Presentation with User Participation". *In Proc. of Multimedia '94, (CA: June), (New York: ACM Press).*

Pylyshyn, Z. W. (1973) "What the Minds Eye Tells the Minds Brain: A critique of mental imagery". *InPsychological Bulletin, Vol. 80,* pp. 1-24

Rasmussen, J. (1983) "Skills, Rules and Knowledge: Signals, Signs, and Symbols and other Distinctions in Human Performance". *IEEE Trans. on Man, Systems, and Cybernetics, Vol. 13(3),* pp. 257-266.

Rauterberg, M. (1995) "From Novice to Expert Decision Behaviour: a Qualitive Modelling Approach with Petri Nets". *In Symbiosis of Human and Artefact (Y. Anzai, K. Ogawa and H. Mori, Eds.), (London: Elsevier),* pp. 449-454.

Remus, W. (1984) "An Empirical Investigation of the Impact of Graphical and Tabular Representations on Decision Making". *In Management Science, Vol. 30(5),* pp. 533-542.

Resnick, M. (1996) "StarLogo: An Environment for Decentralised Modelling and Decentralised Thinking". *Demonstration at ACM SIGCHI'96, (M. J. Tauber, Ed.), (Vancouver, Canada: April), (New York: ACM Press), Companion,* pp. 11-12.

Ringelband, O. J., C. Misiak., R. H. Kluwe. (1990) "Mental Models and Strategies in the Control of a Complex System". *Mental Models and HCI, (Holland: Elsevier), Vol 1(1),* pp. 151-165.

Roth, S. F., J. Mattis. (1990) "Data Characterisation for Intelligent Graphics Represenations" *In Proc. of SIGCHI '90, (New York: ACM Press),* pp. 193-200.

Roth, S.F., W. E. Hefley. (1993) "Intelligent Multimedia Presentation Systems: Research and Principles". *In Intelligent Multimedia Interface Design (M. Maybury, Ed.), (Cambridge, MA: MIT Press),* pp.13-58.

Roth, S. F., J. Kobjeychick, J. Mattis. (1994) "SAGE Tools: A Knowledge Based Environment for Designing and Perusing Data Visualisations". *In CHI '94 Demonstration* .

Salber, D. (1994) "Observing Users in Multimodal Interaction" *In Proc. of ACM SIGCHI'94, (Boston, MA: April), (New York: ACM Press),* Companion, pp. 103-105.

Sampson, G. (1985) "Writing Systems". *(London: Hutchinson).*

Sanderson, P. M., A. G. Verhage, R. B. Fuld. (1989) "State-space and Verbal Protocol Methods for Studying the Human Operator in Process Control". *In Ergonomics, Vol. 32(11),* pp. 1343-1372.

Searle, J. R. (1969) "Speech Acts: An Essay in the Philosophy of Language", *(Cambridge, UK: CUP).*

Seligman, D. D., S. Feiner. (1991) "Automatic Generation of Intent Based 3D Illustrations". *In Proc. of SIGGRAPH'91, (Las Vegas, Calif.: July),* pp. 123-132.

Sellen, A. J. (1992) "Speech Patterns in Video-Mediated Conversations". *In Proc. of SIGCHI '92, (New York: ACM Press),* pp. 49-59.

Shannon , Weaver (1949) "The Mathmatical Theory of Communication". Uni. of Illinois Press.

Shepard, R. N. (1975) "Form, Formation and Transformation of External Representations". *In "Information Processing and Cognition: The Loyola Symposium", (R. L. Solso, Ed.), (Hillsdale, NJ: LEA).*
Simutis, Z. M., H. F. Barsam (1983) "Terrain Visualisation and Map Reading. In *"Spatial Orientation: Theory, Research, and Application". (H. L. Pick, L. P. Acredelo, Eds.), (London: Plenum Press)*, pp. 161-194.

Singh, G., C. H. Kok, T. Y. Nijan. (1990) "DRUID: A System for Demonstrational Rapid User Interface Development". *In Proc.of UIST '90: SIGGRAPH Symposium on UI Design, (D. R. Olson: Chair), (Snowbird, Utah: October), (New York: ACM Press)*, pp. 167-177.

Spence, R. (1989) "Decision Support with Multidimensional Icons", *Internal report 89/5, (Dept. of Information Engineering, Imperial College: London).*

Spence, R., M. Parr (1989) "The Cognitive Assessment of Alternatives", *Internal report 90/7, (Dept. of Information Engineering, Imperial College: London).*

Staggers, N., A. F. Norcio. (1993) "Mental Models: concepts for human computer interaction research"*IJMMS, Vol. 38*, pp. 587-605.

Stenning, K. (1978) "Anaphora as an Approach to Pragmatics". *In "Linguistic Theory and Psychological Reality" (M. Halle, J. Bresnan, G. A. Miller, Eds.), (Cambridge, MA: MIT Press)*, pp.163-201.

Stenning, K., J. Levy. (1988) "Knowledge-Rich Solutions to the Binding Problem: a simulation of some human computational mechanisms". *Journ. of Knowledge Based Systems, Vol. 1(3)*, pp. 143-151.

Stenning, K. (1991) "Reasoning with Words Pictures and Calculi: Implementation vs. Justification" *In Situation Theory and its Applications (J. M. Gawron, G. Plotken, S. Tutiya, Eds.), (MAS, Stanford Uni. Press:*, pp. 601-621.

Stenning, K. (1994) "Logic as a Foundation for a Cognitive Theory of Modality" Assignment.*Research Paper HCRC-RP-51, (HCRC Edinburgh University: UK, May).*

Stenning, K, J. Oberlander. (1995) "A Cognitive Theory of Graphical and Linguistic Reasoning: Logic and Implemenation". *Cognitive Science, Vol. 19(1)(Aug/Sept)*, pp. 97-140.

Sternberg, R. J. (1980) "Representation and Process in Linear Syllogistic Reasoning". *In Journ. of Experimental Psychology: General, Vol. 109(2)*, pp. 119-159.

Stevenson, R. J. (1993) "Language, Thought and Representation". *(London: Wiley).*

Suchman, L. (1987) "Plans and Siuated Action". *(Cambridge, UK: CUP).*

Sun Microsystems Inc. (1990) "OPEN LOOK: Graphical User Interface Application Style Guidelines". *(Reading, MA: Addison-Wesley)*

Sutcliffe A., and U. Patel (1996) "3D or not 3D: Is it nobler in the mind". *In Proc of HCI'96: People and Computers XI, (S. Cunnigham, A. Sasse, R. Winder, Eds.), (August: Imperial College, London), (Berlin: Springer-Verlag)*, pp. 79-94

Sventek, J. S. (1992) "An Architecutre Supporting Multimedia Intergration", *ANSA Project, Cambridge, UK.*

Talmy, L. (1983) "How Language Structures Space". *In Spatial Orientation: Theory, Practice, and Application ,(H. L. Pick Jr., L. P. Acredolo, Eds.), (London: Plenum Press)*, pp. 225-282.

Tani, M., K. Yamaashi, K. Tankoshu. (1992) "Object Orientated Video: Interaction with Real World Objects Through Live Video". *In Proc. of SIGCHI ' 92, (P. Bauersfield, J. Bennett, G. Lynch, Eds.), (New York: ACM Press)*, pp. 593-598.

Tarski, A. (1931) "The Concept of Truth in Formalised Languages". *In Logic, Semantics, Metamathematics: Papers from 1923 to 1938, (J. H. Woodger, Trams.), (Oxford, UK: OUP).*

Thorndyke, P. W., S. E. Goldin. (1983) "Spatial Learning and Reasoning Skill". *In "Spatial Orientation: Theory, Practice, and Application", (H. L. Pick Jr., L. P. Acredolo, Eds.), (London: Plenum Press)*, pp. 195-217.

Tortora, G. (1990) "Structure and Interpretation of Visual Languages". *In "Visual Languages and Visual Programming", (S. Chang, Ed.), (New York: Pergammon Press),* pp.3-27.

Trabasso, T., and Riley, C.A. (1975) "On the Construction and Use of Representations Involving Linear Order". *In Information Processing and Cognition: The Loyola Syposium, (R. L. Solso, Ed.), (Hillsdale, NJ: LEA).*

Tubbs, S. L., S. Moss. (1991) "Human Communication". *(London: McGraw-Hill).*

Tufte, R. E. (1983) "The Display of Quantitive Information". *(Conneticut, USA: Graphics Press)*

Tversky, B. (1995) "Cognitive Origins of Graphic Productions". *In "Understanding Images: Finding Meaning in Digital Imagery" (F. Marchese, Ed.), (Holland: Springer-Verlag),* pp. 28-53.

Verhage, A. G. (1989) "Designing Visual Displays to Enhance Operator Knowledge in the Process Control Environment". *PhD Thesis, EPRL, University of Urbana, Illinois.*

Vicente, K. J. (1991) "Supporting Knowledge-Based Behaviour Through Ecological Interface Design". *Research Report (EPRL-91-01), (University of Illinois at Urban: US).*

Weir, G. S. (1991) "Dealing with Complex Systems". *In 'HCI and Complex Systems' (J. L. Alty, M. Coombs, Eds.), (London: Academic Press),* pp. 1-19.

Whorf, B. K. (1942) "Language, Thought, and Reality: Selected Writings of B. L. Whorf". *(Cambridge, MA: MIT Press).*

Williams, D. M. L. (1996) "Multimedia, Mental Models, and Complex Tasks", *Paper presented at Doctoral Consortium of ACM SIGCHI'96, (M. Tauber, Ed.), (Vancouver, BC: April), (New York: ACM Press),* pp. 63-65.

Williams, D M. L., I. Duncumb, J. L. Alty. (1996) "Matching Goals to Tasks: An Approach Based on Expressiveness". *In Proc of HCI'96: People and Computers XI, (S. Cunnigham, A. Sasse, R. Winder, Eds.), (August: Imperial College, London), (London, UK: Springer-Verlag),* pp. 333-349.

Williams, M. D., J. D. Hollan, A. L. Stevens. (1983) “Human Reasoning About a Simple Physical System”. *In Mental Models (D. Gentner, A. L. Stevens, Eds.), (Hillsdale, NJ: LEA)*, pp. 131-153.
Winograd, T. (1972) "Understanding Natural Language". *In Cognitive Psychology, Vol. 3*, pp. 1-191.

Wohl, D. (1992) "Traffic Systems for Engineers and Planners". *(London: Mcgraw-Hill).*

Wong, Y. Y. (1996) "Temporal Typography: A Proposal to Enrich Textual Expression". *In Proc. of ACM SIGCHI'96 (M. J. Tauber, Ed.), (Vancouver, BC, Can.: April), (New York: ACM Press), Companion*, pp. 408.

Woods, D. (1991) "The Cognitive Engineering of Problem Representations". *In 'HCI and Complex Systems' (J. L. Alty, M. Coombs, Eds.), (London: Academic Press)*, pp. 169-188.

Young, R. M. (1983) “Surrogates and Mappings: Two Kinds of Conceptual Models for Interactive Devices”. *In Mental Models (D. Gentner, A. L. Stevens, Eds.), (Hillsdale, NJ: LEA)*, pp. 35-52.

APPENDICES

Appendix A: Experimental Materials

The appendix contains the following materials which are referred to in Chapters 9 and 10.

A.1. Experimental Notes

These notes were used by subjects at the beginning of the experiment,;prior and during the training period.

- Introductory Notes
- Traffic System Description
- Training Material
- Training Questions

A.2. Main Experimental Material

These questions were used in the body of the experiment. The first set appeared on the computer screen whilst the second set were read by the experimenter at the end of each question set.

- Experimental Questions
    - Easy Complexity
    - Medium Complexity
    - Hard Complexity
- Debriefing Questions
    - Easy Complexity
    - Medium Complexity
    - Hard Complexity

A.3 Screen shots

This section shows screen shots of the five experimental media and the control dialog box.

A.4 The Traffic Domain Ideal Model Description

This section contains the functional description of the traffic domain behaviour. This domain is described in detail in Chapter 9.

A.1. Notes

A.1.1. Introductory Notes

Introduction

This experiment will *investigate the effect of problem representation on problem solving success*. You will be required to solve a number of problems in a **computer-simulated road network**. This is done by controlling the red and green times of traffic lights. The simulation is simplified but works in real-time and will stop only on the completion of each problem.

Experimental Environment

In the experiment you will be required to enter new control values into the simulation, and choose which aspect of the simulation you would like to observe

The simulation and experimental environment run on 'Microsoft Windows'. The input values and controls for the simulation output are found on the **control-panel**. To change input values (traffic light red and green times) and which aspects of the simulation are displayed, you will use either the *keyboard* or the *mouse* to alter values or press buttons on the **control-panel**.

You may also be required to arrange windows on the screen to your own satisfaction. If you are unable to do this the experimenter will arrange them for you. All operations are carried by using the number-keys and one mouse button; these are marked with **green** circular stickers.

As a way of obtaining subject data, at certain times, a microphone will record verbalisations you make whilst solving problems. **Therefore, it is essential that you 'think-aloud' as much as possible**. A 'key-stroke' log is also taken automatically by the computer which records all input changes you make.

Questioning Procedure

Questions will appear in the **task-bar** at the top of the screen (this will be pointed out to you). Advancing through questions will be under your control. Questions are grouped into **question sets**. On beginning a **question set** you will be notified as to how many traffic lights will be under your control. Pressing the **R(return)** key will begin the **question set.**

On beginning a question the question text will appear in the **task-bar**. On completion a notifying sound will accompany a **question-completion notice**. Pressing the **R(return)** key will activate the next question.

At the end of a **question set** a further **notice** will ask you to notify the experimenter.

A.1.2. Experimental Procedure

The experiment is split into **two sessions** and should take in total **about 2.5 hours**. A breakdown of the sessions are provided below.

Session 1

1) Read *subject notes.*

2) Read simulation description.

3) Training period:

   •reading output, controlling output, entering input, questioning

   •training questions (some questions answered verbally)

4) Short break

5) Questions Part 1

6) Debriefing (questions answered verbally)

7) Short break

8) Questions Part 2

9) Debriefing (questions answered verbally)

Session 2

1)Questions Part 3

2) Debriefing (questions answered verbally)

Disclaimer

This experiment is conducted by David Williams, a full-time Ph.D. student in the department of Computer Studies at Loughborough University of Technology.

If at any time you are not happy with the experiment you may leave. You may also retrospectively withdraw at the end of the experiment and your results will be destroyed.

A.1.3. Traffic Simulation Notes

Given at Beginning of Experiment and can be referred to during experiment.

Introduction

In this experiment you will be required to solve problems in a *simulated road network* by operating traffic lights to regulate the flow of traffic. This is done by controlling the length of time a light is red and green (the amber time is fixed). This is called the **phasing time** of the light.

The problems will ask you to make certain variables of the system reach specified values. These important variables are called **target variables** (described below). You will need to control one, two or three of the target variables in a specified way by altering traffic light phasing times. In addition to the target variables, a 4 system **state** variables are present which are directly or indirectly related to the **target values. Take note of these as they will help you in your problem solving.**

Both target and state variables will change at different times. Some will change constantly and others at the end of each red-green cycle of the traffic lights.

Traffic Light Arrangement

Problems take place on a simple road network. Each road has two traffic lights, one nearest the entrance, a third of the way in (light 1), and one nearest the exit, two-thirds of the way in (light 2). Also, a road can carry up to a maximum number of cars per hour, this is called **capacity.**

The majority of questions will concern two roads (which may not have the same flow capacity) at right-angles to each other. The cross-roads between the two roads is at the lights nearest the road exits. In this case, the second road has light 3 nearest the entrance, and light 4 nearest the exit.

Thus traffic enters at lights 1 and 3 and leaves at lights 2 and 4. Since Lights 2 and 4 are at the crossing of the two roads their phasing is opposite. i.e. **when *light 2* is red *light 4* is green and vice-versa**. Traffic only travels along the road it enters, never turning onto any other road.

In some questions the traffic enters the roads at a constant rate. However, there may be a case when traffic enters the part of a road before the first light in a **periodic pattern.** e.g. high flow for two seconds, then low flow for 4 seconds, then high flow for 2 seconds etc.

Traffic System Variable Descriptions

All of the system variables can be displayed selectively by switching them on or off with buttons on the **control panel**. The system's measured variables fall into 3 categories:

1) Input Variables
- •Red time (in time units)
- •Green time (in time units)

2) State Variables-(for each light)
- •Flow rate (in cars per hour)
- •Delay -for each road (in time units)
- •Queue length (in cars)
- •Utility (in a **ratio** of cars)

3) Target Variables (for the whole system)
- •Total Delay (in time units)
- •Total Queue length (in cars)
- •Total Utility (in cars)

In detail:

1) Input -directly controlled from the **control panel**

Red and green time of each light. There will be at least two lights to control and no more than four. **Any phasing change of red or green will only take effect on the next green or red phase.** The green and red times are measured in *time units*.

2) State Variables-indirectly controlled and displayed on the output panel

Delay. This is an estimated time for a car to move along a road. This includes the time that the car is queued at any lights. The value is measured on each simulation step.

Flow. This is a measure of the number of cars passing through a light per 3600 time units (equivalent to an hour). The value is measured on each simulation step.

Queue length. This is the number of cars waiting at a traffic light. The value is measured on each simulation step.

Utility. The efficiency of a light measured as a ratio of the number of cars released by a light in its **amber-green phase** to the cars stored in its previous **amber-red phase**. e.g. a utility of 3 is equivalent to 3:1 and 1:3, 3 times as many cars stored as released, or vice-versa.

3) Target variables-indirectly controlled subject of questions. They are displayed on the output panel.

Total Delay This is the sum of the delay for each road.

Total Queue Length. This is the sum of all queue lengths in the system.

Total Utility. This is the sum of all utility measures for each light.

IF YOU ARE NOT SURE ABOUT ANYTHING IN THIS DESCRIPTION PLEASE ASK THE EXPERIMENTER

A.1.4. Training Material

Training Questions

Worked through with experimenter supervising. Simulation in a general mode where no targets have to be reached.

Investigating the Traffic Simulation

Please answer verbally the following questions by controlling the simulation. The experimenter will answer questions about reading the output and entering input but will give no more information than that contained in the simulation notes.

<u>Controlling Two Traffic Lights with a Constant Input-flow of Traffic</u>
Simulation values are shown in *italics.*

1. Describe *queue-length* on each light?

2. Describe *utility* on each light?

3. Describe *delay* on the road?

4. When do changes in light red or green time come into effect on the simulation?

5. Which of the two lights has a regular, cyclic increase and decrease of *queue length? Why* do you think this is?

6. What is the *flow-rate* of traffic when a light first goes from red to amber to green? Is it always the same?

7. How many simulation steps does it take for traffic to move from light 1 to light 2?

8. How does the *flow-rate* into a light and the *flow-rate* out of a light affect the rate *queue-length* builds and falls.

9. Make the *utility* on light 1 increase? How did you do this?

10. Make the *delay-time* increase?

11. Make the *delay-time* decrease? Once decreased, is this value constant?

12. Make the *utility* on light 2 decrease?

Controlling Four Traffic Lights with a Different Constant Input-flow of Traffic on Each Road

1. Make more cars queue on the 'light 1-2' road than the 'light 3-4' road. Why is this difficult?

2. Which road can allow more traffic through it per hour?

3. What is the relationship between light 2 and 4?

4. Increase the *total-utility* . Why is this difficult?

5. Make the *queue-total* as small as possible. How did you do this?

6. Make the *delay-total* as small as possible. How did you do this?

7. Make the *utility* on light 4 larger than the *utility* on light 2.

8. How quickly do changes in the *red-time* of light 3 affect the simulation? Why is this?

A.2. Main Experimental Material

A.2.1. Experimental Questions

Easy

1. Make Queue Total>120.
2. Make Utility Total<8.
3. Make Delay Total<80.
4. Make Queue Total<30.
5. Make Utility Total>6.
6. Make Delay Total>100.
7. Queue Total>125.
8. Make Utility Total<8.
9. Make Delay Total<85.
10. Make Queue Total>92.
11. Make Utility Total>11.
12. Make Delay Total>123.
13. Utility Total<7.
14. Make Utility Total<12.
15. Make Delay Total<97.
16. Make Queue Total<88.
17. Make Utility Total>8.
18. Make Delay Total>79.
19. Queue Total>112.
20.Make Utility Total<7.
21. Make Delay Total<83.
22. Make Queue Total>91.
23. Make Utility Total>9.
24. Make Delay Total>77.

Medium

1. Queue Total>80 and Utility Total<7.
2. Make Utility Total<8 and Queue Total<40.
3. Make Delay Total<72 and Queue Total<40.
4. Make Queue Total<20 and Utility Total>8.
5. Make Utility Total>6 and Queue Total<30.
6. Make Delay Total>70 and Utility Total<6.
7. Queue Total>120 and Utility Total<6.
8. Make Utility Total<8 and Queue Total<40.
9. Make Delay Total<75 and Queue Total<45.
10. Make Queue Total>22 and Utility Total>9.
11. Make Utility Total>11 and Queue Total<60.
12. Make Delay Total>72 and Utility Total<6.
13. Queue Total>83 and Utility Total<7.
14. Make Utility Total<7 and Queue Total<42.
15. Make Delay Total<67 and Queue Total<40.
16. Make Queue Total>20 and Utility Total<7.
17. Make Utility Total>8 and Queue Total<66.
18. Make Delay Total>79 and Utility Total<10.
19. Queue Total>80 and Utility Total<8.
20. Make Utility Total<7 and Queue Total<48.
21. Make Delay Total<73 and Queue Total<43.
22. Make Queue Total>21 and Utility Total>8.
23. Make Utility Total>9 and Queue Total<66.
24. Make Delay Total>57 and Utility Total<8.

Hard

1. Queue Total<70 and Utility Total<7 and Delay Total>60.
2. Make Utility Total<7 and Queue Total<40 and Delay Total<76.
3. Make Delay Total<67 and Queue Total<40 and Utility Total>8.
4. Make Utility Total>9 and Queue Total<40 and Delay Total>70.
5. Make Queue Total<25 and Utility Total<8 and Delay Total<80.
6. Make Delay Total<90 and Utility Total<9 and Queue Total>80.
7. Queue Total<120 and Utility Total<8 and Delay Total>88.
8. Make Utility Total<10 and Queue Total<60 and Delay Total<84.
9. Make Delay Total<89 and Queue Total<40 and Utility Total>9.
10. Make Utility Total>7 and Queue Total<40 and Delay Total>60.
11. Make Queue Total<21 and Utility Total<9 and Delay Total<87.
12. Make Delay Total<78 and Utility Total<9 and Queue Total>82.

A.2.2. Debriefing Questions

Read by experimenter at the end of each set of questions. Questions within each set asked in random order.

Debriefing: Easy

•*This must be recorded.*
•*Pen and paper must be provided.*
•*The monitor must be switched off.*

"From your experience with the system"

Draw the light arrangement in the four light system. Show any detail you think necessary.

Describe the progress of a car along one of the roads. Provide as much detail as possible.

What effect does changing the red-time of a light have on it's utility and the overall delay time for that road?

Describe how the *utility* value of a light relates to cars passing through it or stopping at it.

Describe how the *delay-total* calculated for either road. What sequence of input changes would be required to increase this value?

Are the two utility values and *delay-time* closely related for either road.

Even if light the second light of a road has a high *red-time* and a low *green-time* describe a situation where the *queue_length* at this light would not increase. (apart from the red-time of light 1 being very high, and the green-time very low). Roughly, what sequence of input changes would be required to achieve this?

How could the traffic flow volume be made the same for both roads, given that one road has a higher traffic capacity?

What effect does changing the *green-time* of an input light (1 or 3) have on it's *utility* and the *utility* of the subsequent light?

What was the most difficult aspect of the system to understand or to control?

Debriefing: Medium

•*This must be recorded.*
•*Pen and paper must be provided.*
•*The monitor must be switched off.*

"From your experience with the system"

What will make *utility* increase on a light? How can this be achieved by changes in the inputs.

What input settings will make *total_utility* increase?

What input settings will make *total_delay* decrease?

Describe how the *delay-time* is calculated for either road. What sequence of input changes would be required to increase this value?

Are the two *utility* values and *delay-time* closely related for either road?

What will happen to the *utility* of a light if its *queue_length* is increasing at constant rate? What effect would an increase in the *red-time* of the light have on its *utility*?

How could the traffic flow volume be made the same for both roads, given that one road has a higher traffic capacity?

Describe the progress of a car along one of the roads. Provide as much detail as possible.

If a certain input sequence cause the *total-queue* of the system to increase, what will happen to the *total-utility*?

Can you think of an another physical system which exhibits similar behaviour to the system described by the system?

What was the most difficult aspect of the system to understand or to control?

Debriefing: Hard

•This must be recorded.
•Pen and paper must be provided.
•The monitor must be switched off.
•Periodic input flow

"From your experience with the system"

Can you draw the pulse-pattern for traffic flowing into the two roads. Did the patterns start at the same position for each road?

What input settings will make *total_utility* increase?

What input settings will make *total_delay* decrease?

If *utility* increases on the first light how will *delay-time* for the road be affected, given the second light is letting nearly all cars through?

Describe how the *delay-time* is calculated for either road. What sequence of input changes would be required to increase this value?

Are the two *utility* values and *delay-time* closely related for either road?

What will happen to the *utility* of a light if its *queue_length* is increasing at constant rate? What effect would an increase in the *red-time* of the light have on its *utility*?

Describe the progress of a car along one of the roads. Provide as much detail as possible.

If a certain input sequence cause the *total-queue* of the system to increase, what will happen to the *total-utility*?

Can you think of an another physical system which exhibits similar behaviour to the system described by the system?

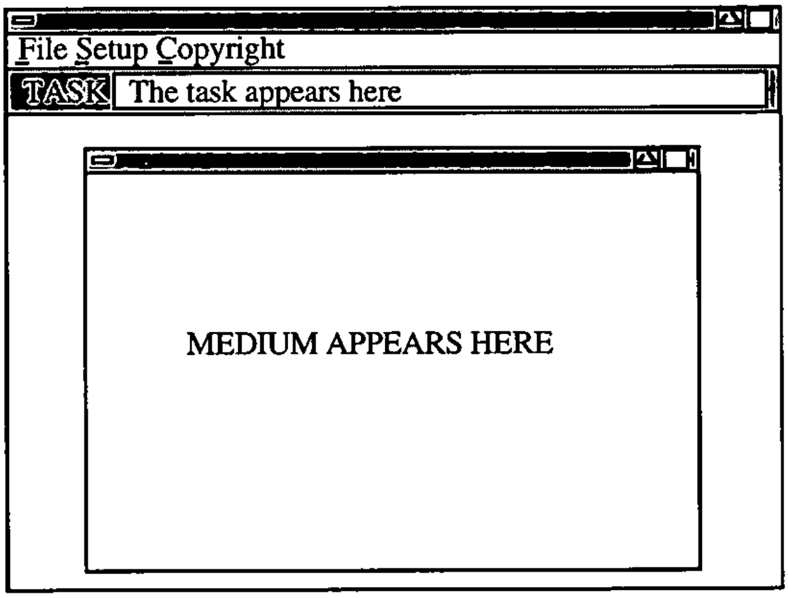
What was the most difficult aspect of the system to understand or to control?

(If subject is using the static video medium) Did you find the pictures useful?

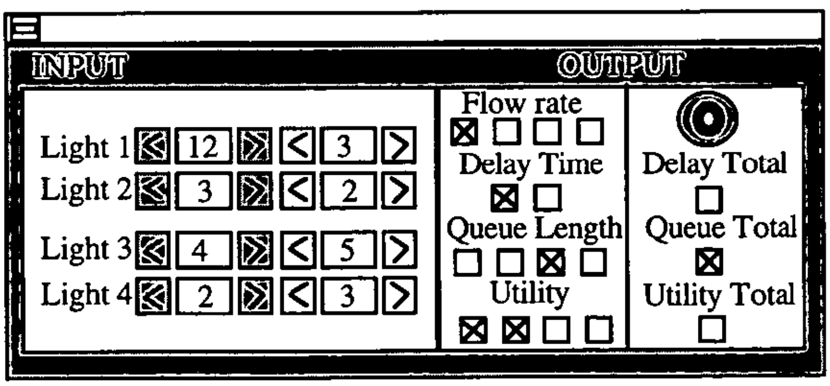
A.3 The Experimental Media

The following section contains renderings of the experimental media.

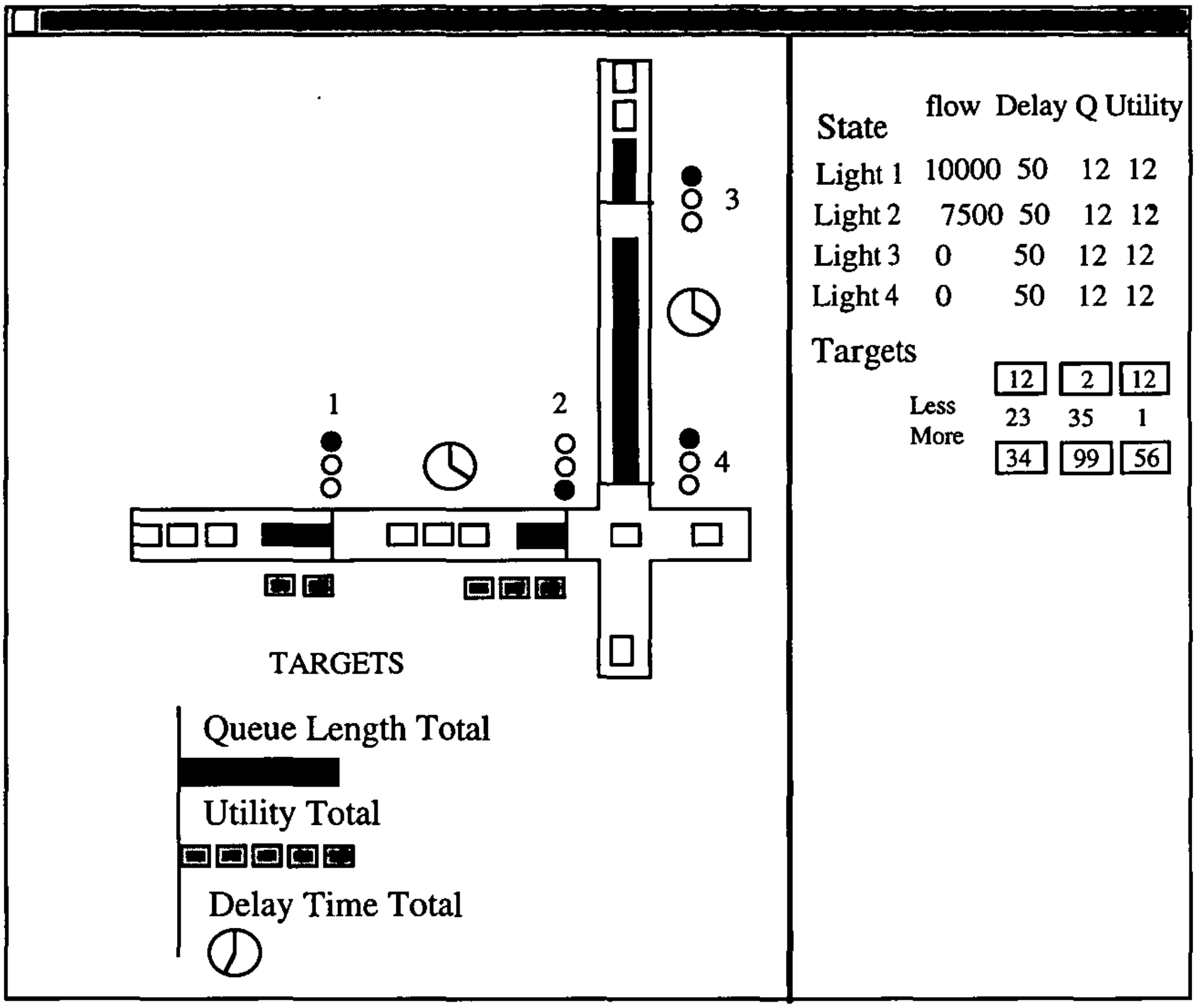
All media were shown within the following screen layout.



A.3.1 Control Dialog Box

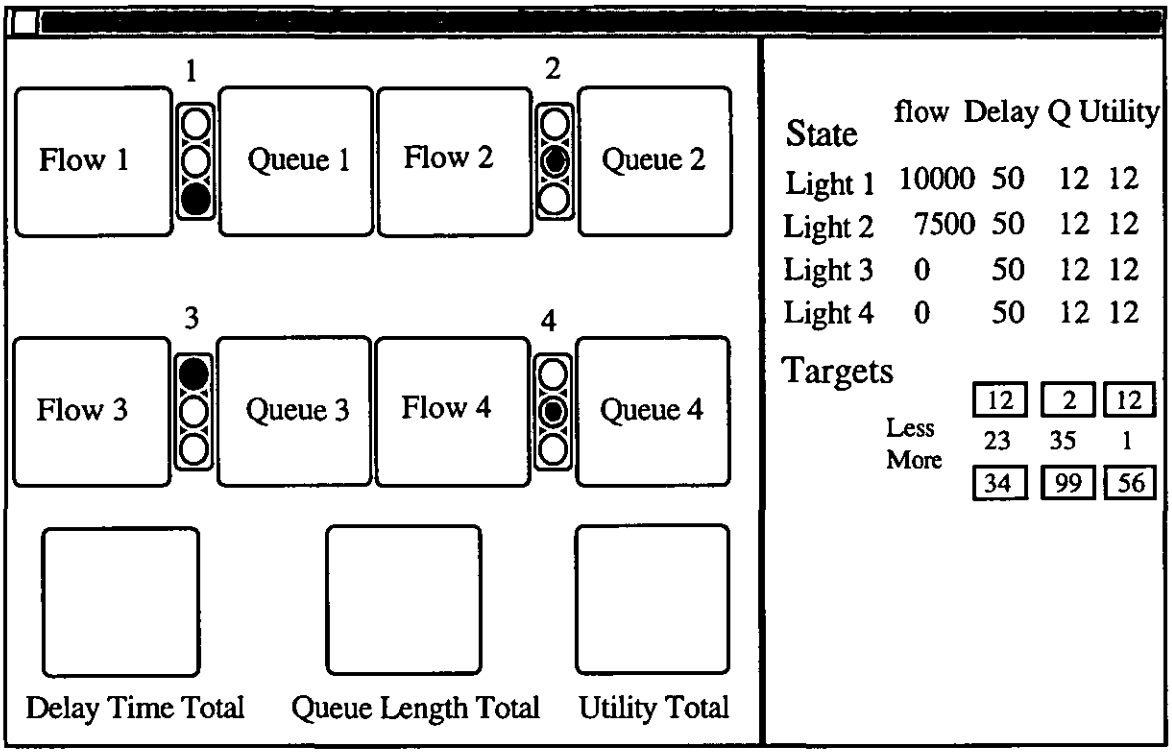


A.3.2. Animation



A.3.3. Static Video

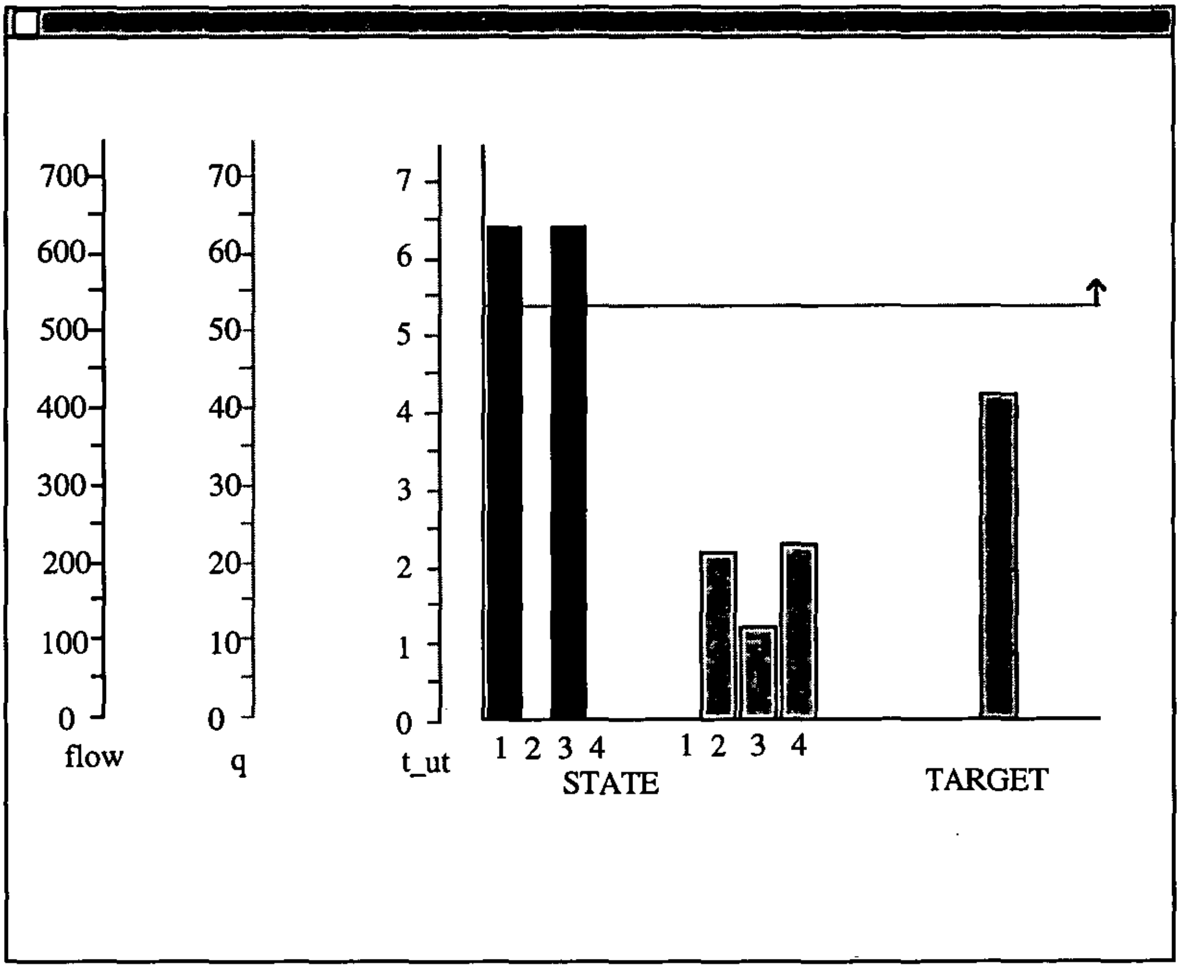
(For example images, see example in Chapter 6)



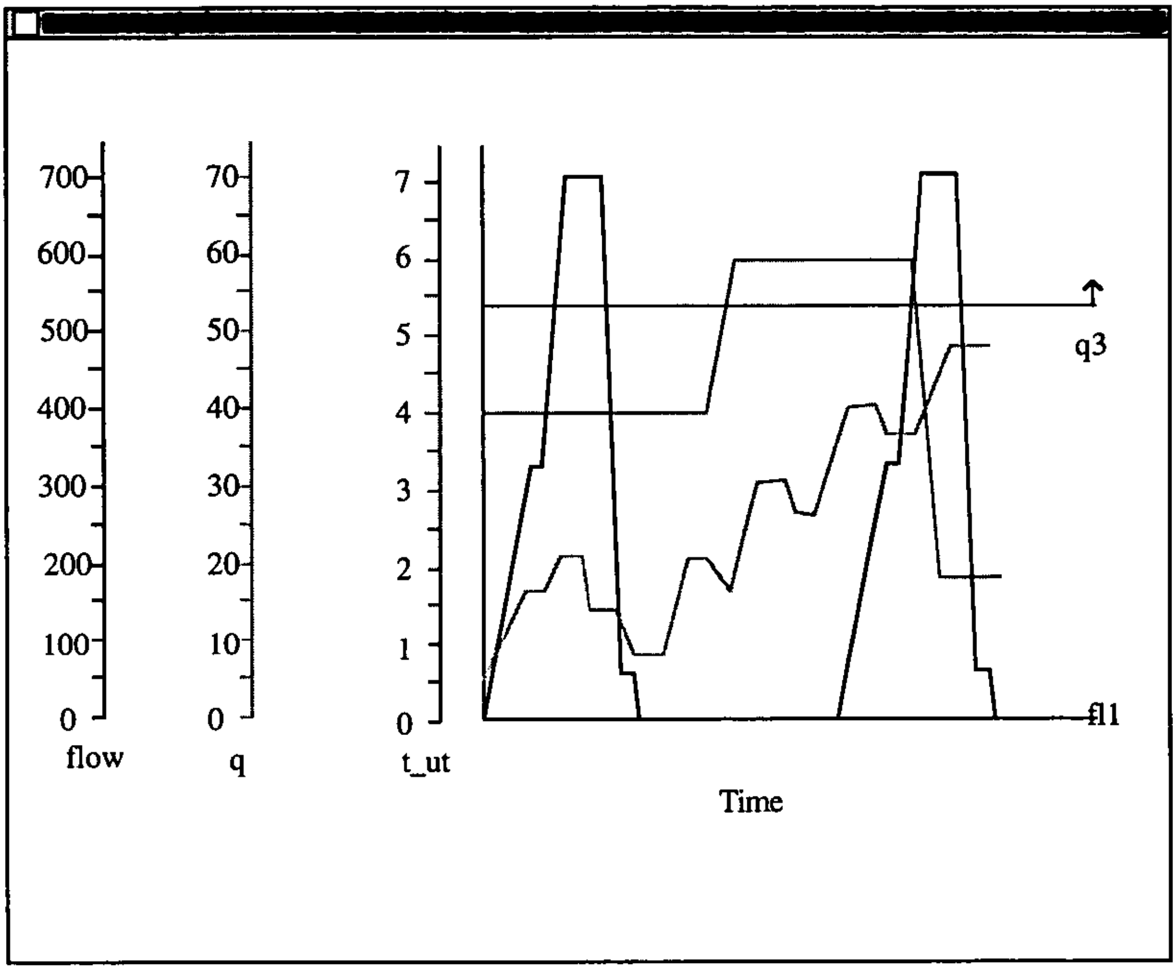
A.3.4. Dynamic Table

	STATES (L1 L2 L3 L4)	TARGET	AIM
Flow rate	0-10000-7500-100 0-10000-7500-100 0-10000-7500-100 0-10000-7500-100		
DELAYS (R1-R2)	0-15-0-15 0-15-0-15 0-25-0-0 0-25-0-0	30 30 25 25	0 Less More 15
QUEUE LENGTHS	0-15-0-15 0-15-0-15 0-25-0-0 0-25-0-0	30 30 25 25	12 Less More 0
UTILITIES	0-15-0-15 0-15-0-15 0-25-0-0 0-25-0-0	30 30 25 25	25 Less More 0

A.3.5. Dynamic Bar Chart



A.3.6 Dynamic Graph



A.4 The Traffic Domain Ideal Model Description

The ideal model of the traffic domain is used as the basis for comparing the declarative knowledge subjects possess about the domain. The model describes the relationships between the input, state and target variables. Relationships are proportional (+) or inversely proportional (-) and can have a number of constraints attached (Shown in the diagrams by each '•' statement below a relationship). These constraints allow a tighter definition of the relationship in terms of other related system characteristics.

In the descriptions the following nomenclature will be used, abbreviations are shown in brackets:

Input Variables

- The average traffic flow into the road a (fina);
- The average traffic flow into the road b (finb);
- The green time of light x (Gx);
- The red time of light x (Rx).

State Variables

- The maximum queue length on light x for a green/red cycle (Qxmax);
- The instantaneous queue length on light x for a green/red cycle (Qx);
- The rate at which the queue length of light x is changing (Qxrate);
- The instantaneous utility value of light x (Utx);
- The instantaneous delay value on road a/b (D1/2);
- The average traffic flow into light x (fx).

Target Variables

- The instantaneous total queue length of the system (QT)
- The instantaneous total delay time of the system (DT)
- The instantaneous total utility of the system (UTT)

Constraint Terminology

A number of abbreviations are also used to describe constraints on relationships

- (-/+)Δx : positive or negative change of a variable x;
- <, > : as a binary operator (Less than or greater than), as a unary operator (decrease or increase)
- <<, >> : as a binary operator (much less than or greater than), as a unary operator (large decrease or large increase)
- c : the number of cars.
- t+x : an event happens x simulation steps later.
- change: the dependent variable's value must change.
- or : both relationships hold.
- l=G or l=R: A light l is green (G) or a light is red (R).

These are combined, for example:

- Rx>>: The red/green time of light x must greatly increase;
- ΔcRx : The change in the number of cars queued on light x .

The following model is defined for one road only, since both roads are identical in behaviour.

A.4.1. Input -> State Relationships

	R1	G1	R2	G2	R3	G3	R4	G4	*fina*
Q1Max	+ •cR>	- •cR<cG							+ •cR> •t+10
Q2Max	- •R2<< •t+10 •cG2>cR2 •G1<< + •R2>> •t+10 •cR2>cG2 •G1>>	+ •R2<< •t+10 •cG2>cR2 •G1<< - •R2>> •t+10 •cR2>cG2 •G1>>	+ •cR>	- •cR<cG			- •cR<cG	+ •cR>	
Q1Rate									+ •t+10
Q2Rate									
Q1									

Q2									
Ut1	+ •R<<G •cR<<c G •change - •R>>G •cR>>c G •change	+ *G<<R •cG<<cR •change - •G>>R •cG>>cR •change							+ •R<<G •cR<<cG or *G<<R •cG<<cR - •R>>G •cR>>cG or •G>>R •cG>>cR
Ut2	+ • t+10 •cR<<c G •change - • t+10 •cR>>c G •change	- • t+10 •cR<<cG •change + • t+10 •cR>>cG •change	Ut1	Ut1			Ut1	Ut1	
D1	+ •Qmax> •-ΔQ2 max>ΔQ 1max	- •Qmax< •+ΔQ2 max>ΔQ 1max	+ •Qmax> •-ΔQ2 max>ΔQ 1max	- •Qmax< •+ΔQ2 max>ΔQ 1max			- •Qmax< •+ΔQ2 max>ΔQ 1max	+ •Qmax> •-ΔQ2 max>ΔQ 1max	
Fl1									+ •t+10 •l=G •Q=0
Fl2									+t+20 •l1=l2=G •Q1=Q2 =0
Fina									

A.4.2. Input Variable -> Target Variable Relationships

	DT	QT	UTT
R1, R2	+	+	+ (-)
G1, G2	-	-	- (+)
fina	+	+	

A.4.3. Target Variable -> Target Variable Relationships

	QT	UTT
DT	+ • t+some time	+ • cR>cG • change • t+some time - • cG>cR • change • t+ some time
QT		+ • cR>cG • change • t+some time - • cG>cR • change • t+ some time

A.4.4. State Variable -> Target Variables Relationships

	DT	QT	UTT
Utl and UT2	+ • cR1>cG1 • change • change in others is less • t+some time - + • cG1>cR1 • change • change in others is less • t+some time	+ • cR1>cG1 • change • change in others is less • t+some time - + • cG1>cR1 • change • change in others is less • t+some time	+
Fl	+ • t + 10 • F2<F1 • Other queues don't move	+ • t + 10 • F2<F1 • Other queues don't move	+ • t+10 • cR2>cG2 • change
q1max, q2max	+	+	+ (-)

A.4.5 State Variable -> State Variable Relationships

	Q1Max	Q2Max	Q1Rate	Q2rate	Q1	Q2	Ut1	Ut2	D1	Fl1	Fl2
Q1Max							+ (-)		+ •Qmax in s. dir. •-ΔQ2 max>ΔQ1max	+ •if fout is less •t+10	
Q2Max	• •G1>> •R2<< • t+10 •cR1>> •ΔcR2<< ΔcR1							+ (-)	+ •Qmax> •-ΔQ1 max>ΔQ1max		
Q4Max											
Q1Rate	+						+				

Q2Rate		+						+		+ •if fout is less •if Q2>0 •if l=G	
Q4Rate											
Q1									+		
Q2									+		
Q4											
Ut1	+ •cR1>>cG1 •change - •cG1>>cR1 •change	+ •t+10 •cR2>>cG2 •R2>>0 •change	+						+ •cR>cG •Qmax> •-ΔQ2max>ΔQ1max change - cG>cR •Qmax> •+ΔQ2max>ΔQ1max change		
Ut2		+ •cR2>>cG2 •change - •cG2>>cR2 •change		+			- •cR1>>cG1 •cR2>>cG2 •change or op'site + •cG1>>cR1 •cG2>>cR2 •change or op'site		+ •cR>cG •Qmax> •-ΔQ1max>ΔQ2max •change - cG>cR •Qmax> •+ΔQ1max>ΔQ2max •change	+ •t+10 •cR2>>cG2 •change - •t+10 •cG2>>cR2 •change	
Ut4											
D1	+ •Qmax> •-ΔQ2max>ΔQ1max	- •Qmax< •+ΔQ2max>ΔQ1max	+ •Qmax> •-ΔQ2max>ΔQ1max	- •Qmax< •+ΔQ2max>ΔQ1max			- •Qmax< •+ΔQ2max>ΔQ1max	+ •Qmax> •-ΔQ2max>ΔQ1max			
f1											+ •Q2=0 •t+10

Appendix B

Using the Visual C++ Environment

This appendix gives a brief overview of the Visual C++ Windows programming environment in which the experimental environment was developed and used. This section assumes some Windows programming experience with Microsoft's Software Development Kit.

In general, programming is carried out using the object-oriented language C++. A number of libraries and a skeletal application infrastructure are provided allowing the relatively straightforward programming of full Windows applications.

B.1 What is a Visual C++ Application?

A Visual C++ application is an application in Microsoft Windows which has been designed and developed using the Microsoft Foundation Class Library (MFC), and the Visual Workbench and Application Studio Windows-based development tools. The components of this process will now be described in more detail.

B.1.1 Application Structure

At the heart of the C++ application framework are the concepts of documents and views. A *document* is a data object and is created by the New or Open commands on the File menu and is typically saved in a file. A *view* is a window object through which the user interacts with a document. The key objects in a *runtime* application are:

• The document(s)

• The document class (derived from CDocument[1]) which specifies application data.

• The view(s) specifies how the user sees the document's data. A document can have any number of differently configured views attached to it. Views can be defined from a number of standard types, e.g. CScrollview, CEditview.

• The view class specifies interface rendering , e.g. dialogues, and interaction, e.g. responses to key presses and mouse behaviour.

B.1.2 Microsoft Foundation Class Library (MFC)

The MDC is an object-oriented library built on top of Microsoft's Windows Development Kit (SDK). A large amount of SDK functionality is accessed via slightly different headers. The root object of the MFC is a CObject and from this come all the CDocument, CView, CFile access and device contexts. Most programming in Visual C++ involves deriving objects from MFC classes and providing application specific behaviour.

B.1.3 Visual Workbench

Visual Workbench is the centre of the development process where actual coding takes place. From this environment the Application Studio can be accessed along with compiler and build tools. The application-specific behaviour of interactive objects is defined by their respective message-handlers which are attached to objects using the *ClassWizard* application.

B.1.4 Application Studio

The Application Studio is used to design the visual interface by supporting the creation of application *resources*, e.g. dialogue boxes, menu bars, icons, bitmaps, and accelerator keys. A number of predefined behaviours are provided for these objects including check boxes, radio buttons, and menu-ticks/greying. However, the actual application-specific behaviour of these resources, i.e. how they affect the document data, is not defined in Application Studio.

---

[1] See Microsoft Foundation Class Library description.

B.2 Debugging and Expert Tools

An extensive debugging environment is present in the Visual Workbench application and as separate Windows applications:

• **Visual Workbench debugger**: Allows the setting of breakpoints and probing of data values during execution. Applications can also include debugging routines which are part of the MFC library.

• **Codeview**: Mixed source/object code debugger which allows the setting of breakpoints and probing of data.

• **Dr. Watson**: Resident application which dumps system state information to a file after a General Protection Fault (GPT). This includes the assembly instruction that caused the GPT and all registers values.

B.3 Running Applications

Applications can either be executed within the development environment or stand alone.

B.4 An example: The Experimental Environment

To make the Visual C++ development process clear, there follows a brief description of the experiment application using the terms that have been described.

The important components of the application are:

CApp: Media Experiment
/*Defines Media_EDoc as a Multiple Document Interface (MDI) and defines a number of media combination, e.g. animation and static table, bar chart and video, etc.*/
CDocument Media_EDoc;
/* Contains the traffic simulation itself, file access, and keystroke log recording routines. */
CView DynGphVw;
/* The dynamic graph rendering and behaviour.*/
CView TableVw;

```
/* The dynamic table rendering and behaviour.*/
CView StVidVw;
/* The static video rendering and behaviour.*/
CView BarVw;
/* The bar chart rendering and behaviour.*/
CView AnimVw;
/* The animation rendering and behaviour.*/
CDialog ContrlDlg;
/* Phasing-time input box behaviour affecting Media_EDoc data*/
```

Resource File Generated by Application Studio

Icons;
Static Video Images;
Dialog boxes:
    Experimental completion;
    Input box;
    Next Question.