ENHANCING MULTIMEDIA INTERFACES WITH INTELLIGENCE

Michael Wilson
SERC Rutherford Appleton Laboratory
Chilton, Didcot, Oxon OX11 0QX, UK
mdw@uk.ac.rl.inf

Abstract

Current software products such as spreadsheets are beginning to include automated graphical design and input checking heuristics to provide added automated functionality. The use of such rules requires higher levels of abstract meaning representation than raw media. They also rely upon representing contextual information describing the domain, task, user and current dialogue. With these technologies interfaces move from being purely multimedia towards multimodality. The MMI2 multimodal system is described to illustrate the representation and architecture required for this class of system and the co-operative dialogue which it can support. A second, simpler multimedia information retrieval and presentation system (MIPS) is then described to show how these technologies of context and abstract meaning representation can be incorporated in commercial multimedia applications to structure and tailor multimedia information to the user.

Keywords: Multimodal User Interface, Multimedia, Intelligent User Interface, automatic presentation

INTRODUCTION

Current interest in multimedia follows the lead offered by large software developers. First generation personal computer applications such as word processors and spreadsheets are now established, and for software manufacturers to grow they must find new applications. Fighting for market share is not sufficient to sustain profit and growth. Either they expand the market for the devices on which their applications run by introducing them to the home and integrating them with television, video, CD and other home entertainment systems or they find new applications in their established office markets (Templeton, 1993). Networked and groupware versions of established office products are a temporary growth area, but when this is saturated a new area must be opened or growth will cease. Similarly hardware manufacturers must foster a need by the user for more CPU cycles, greater memory and disk space or else they too will find their market saturated.

The accepted solution to opening the home market at the same time as increasing both new software applications and hardware demands is multimedia. Digital sound and video consume orders of magnitude more disk space, network bandwidth and CPU time than conventional data types. They also appear compatible with the main TV, cable, film and music industries entertainment output. The provision of multimedia facilities appears to support a new form of publishing. There was no benefit in publishing digital versions of Jane Austin novels as linear electronic text at inflated prices compared to books. The publication of the text of Shakespeare plays in conjunction with video of an acted performance, the sounds of respected actors speaking the lines, pictures of contemporary life and theatres, and historical commentary all interconnected, appears to meet the need not only to experience the play but understand its original context, and in doing so it appears to offer a richer experience (Cotton and Oliver, 1993). This interconnection of media and the provision of links between disparate sources of information follows the vision of Vannivar Bush (Bush, 1945) and Ted Nelson (Nelson, 1988) which are the most commonly cited stimuli to hypermedia.

A contrasting vision is provided by Arthur C Clarke in 2001 of an anthropomorphic computer HAL (Clarke, 1968), which not only provides data, but presents it to the user as appropriate to the user's task, when required. It is this second view which is explored in this chapter. Not only the provision of access to data, but the provision of information. Information theory distinguishes passive data from information which has the defining property that it can be used to change the course of human action (Pierce, 1980). There is a vast amount of data available, locating that which is relevant to a user's task and presenting those aspects of it which convey the crucial information is not a trivial task (Card et al,

1991). Indeed the recent history of artificial intelligence following the HAL vision of computing has been fraught with excessive predictions and frustrated ambition.

In parallel with the recent rise of multimedia computer output has been a growth in novel input devices. Pen based handwriting recognition, speech input of commands, speech to text, CAD interpreted hand drawn diagrams and Jot pen standards are supported by PDA's and recent workstations. Half small personal computers sold are expected to include both voice and pen interfaces by 1998 (Crane & Rtischev, 1993). This combination of input and output media has lead to research in not only multimedia, but also multimodal computer systems which use multiple input and output channels, as well as context information and data abstractions to provide co-operative interaction with users.

This chapter discusses current research in introducing intelligence into the analysis of the input to computer systems and the generation of output from abstract representations in multimodal systems. Although neither the interaction mechanisms of handwriting or speech recognition, nor the context based dialogue control required by such systems even approach marketable standards (even a 99% recognition rate would fail on 6 letters in this paragraph with the result that one word per line would have to be re-entered), they are providing a direction for future developments.

After discussing the definition of multimodal systems an example system, MMI^2 will be described to illustrate their function and practical limitations. Then a multimedia presentation system closer to practical marketability, but including components from such multimodal systems, will be described to illustrate more realistic intermediate developments - MIPS. The purpose of this chapter is firstly to inspire faith in the objectives of multimodal systems, and secondly to show that practical progress can be made in the market towards them by introducing some features from them into simpler multimedia architectures. The use of limited heuristics for automated graphics generation and text correction are already being used as unique selling points to distinguish the existing spreadsheet and word processing applications which are market leaders so that they can provide more functionality than competitors. Further elements of multimodal systems will have to be introduced into multimedia systems as the market develops for similar reasons.

MULTIMODAL AND MULTIMEDIA SYSTEMS

This section describes a framework for classifying multimodal and multimedia systems using different input and output modes, and showing the other variations which arise from their use. The distinction between multimedia and multimodal interfaces is not obvious. Some authors regard multimedia as different presentation media and multimodality as different user input modes. Others make a distinction between the simple media which convey a message (e.g. video, sound, image) and the human sensory modalities which perceive it (e.g. auditory, visual, tactile). In contrast multimedia is sometimes used to indicate an audio visual presentation as in television or films; sometimes it excludes these and only refers to interactive multimedia which can be browsed such as hypertext containing video and images as well as text; or sometimes these are also excluded as being insufficiently rich in there interaction so that only the interaction by users with simulations of objects which can be manipulated as they would be in the world in a manner closer to virtual reality. The term multimedia in a different community is even a label for the use of many mass media such as radio, magazines, books, television.

An important distinction during development is that multimodal systems are designed to be co-operative interfaces which actively choose the most effective and efficient presentation mechanisms for a user; whereas multimedia systems present the information in the medium which the author has provided. The sense of cooperativity intended here follows Grice's (1975) cooperativity principle: "Make your contribution as is required, at the stages at which it occurs, by the accepted purpose or direction of the talk exchange in which you are engaged". From this it follows that both participants in a co-operative interaction have a common immediate aim and the contributions of the participants will dovetail.

The contrast intended here between a multimedia and a multimodal system is best explained within the MSM framework proposed by Coutaz et al. (1993) as a design space for multi-sensory motor systems. This framework is presented from the computer system designer's perspective and differentiates some

obvious features of multimedia while adding those which distinguish multimodality. The framework is represented as a six dimensional space in which systems can be described so that they are not points but occupy a sub-space (see Figure 1).

Figure 1 about here

Three of these dimensions are easily understood in the context of most multimedia systems. Firstly, channel direction is the direction of information passing either from the user to the system or from the system to the user. Secondly, for each direction there can be one or more channels along that direction. Therefore a conventional telephone would allow one channel using audio to pass along each direction from one user to another. A video phone would allow two channels along each direction with sound and vision passing between each end. Both the conventional and video phone would allow synchronous communication between both users at the physical level. Conventional television would allow two channels of audio and video but only in one direction. This introduced the third conventional multimedia dimension of parallelism, the videophone would be parallel at the physical level allowing both audio and video simultaneously and synchronised. A low bandwidth communication channel may only allow sound at one time and video at another at the physical level. Although this may support parallel synchronised use of both media at the task level if the user does not notice the results of the division at the physical level. Similarly input systems may allow only one mode to be used at a time (e.g. speech or pen) at the physical or task levels. More complex forms of parallelism can be introduced than purely at the physical signal level by supporting structured tasks or task clusters in parallel.

Although these three conventional dimensions appear clearly defined from the system's perspective, the exact definition of a mode and its correspondence with a physical channel are not entirely clear. There are those who regard 'natural language' as a single interaction modality whether it is typed , hand-written or spoken (e.g. Cohen, 1992) whereas there is considerable research showing that spoken and keyboard interaction differ in many ways (e.g. Chapanis et al ,1977; Cohen, 1984; Oviatt et al, 1991; Rubin, 1980). If someone is using a device such as MMI^2 which can input and output a subset of natural language the user has to develop a clear mental model of the scope of language which can be used with the device. If the output from the system can be through either a form of canned text or through language generated from a deep logical representation, then the canned text will contain more complex language than the generator can produce. If the generator and the comprehension system have the same complexity then the output from the generator should be used as the basis for the model of what can be input to the comprehension system by the user, and not the canned text. Therefore, to support a clear mental model in users the output from the generator and the canned text should be separated so that users regard them as different, and only use the appropriate one as the basis for their models. Therefore the generated and canned text subsets of natural language should be treated differently and either presented in different windows or by different voices. These could both be presented as text on the screen for the visual channel, or passed through a text to speech synthesiser for output through the auditory channel. Whether they should be regarded as different modes or not is still unclear.

Similar examples are available when using freehand graphics input to a system which is then interpreted and presented back to the user as an object based image, and direct object based graphics with direct manipulation. Should these also be regarded as separate modes while they are both on the same channel? Should pen based gesture be seen as a different mode from direct manipulation of graphics? Various studies are currently being to investigate these issues of the definition of modes more completely (e.g. Bernsen, 1993).

When using a video phone it is possible to send instructions by both speech and image, and to refer in speech to items displayed visually. This process includes the speaker dividing their intended message between two channels - audio and video - and synchronising references between them. This process of the speaker dividing the message is termed the Fission of the intended message. The complementary function of Fusion is performed by the listener who will interpret the video and audio signals and fuse

them together to construct a single comprehended message. In the example of a video phone, the speaker and listener will perform these functions, the device will perform neither. For a device to perform actions based on inputs from two or more different modes such as Bolt's (1980) "Put-that-there" system then the device must perform the fusion. In Bolt's system a naval commander could view a map of ships in an area of sea and command the system to put-that-there while pointing at a ship and then a location. The system would have to fuse the gesture and voice input into a single message in order to perform the action of moving the ship, and then presenting the result on the display. This example performs fusion, but since output is only graphical it does not perform fission. Another example system which performs fission without fusion would be the COMET system (Feiner & McKeown, 1991) which provides explanations of how to fix and use radio sets for soldiers. In this a soldier can select questions about radios from a menu, the answers to these are constructed by the system which then decides on the best channel to present the information. In most cases the information in the message is presented to the user in a combination of images and text; the images showing where on the radio objects are, with salient components highlighted, and the text explaining the actions to be taken on the objects (simple actions such as the clockwise turning of a knob are also represented by conventional images such as arrows). The COMET system performs fission of a message between presentation modes, but does not perform fusion of input since all input is from menus.

Fission and fusion are represented on a single scale on the MSM framework so that if both or either is performed the "yes", rather than "no" value would be used. These could be divided onto separate dimensions if the distinction between these two were required to be drawn more clearly.

The video phone system only supports the transmission of raw audio and video information. This may be digitised and compressed, but it is not abstracted into any form of meaning representation. On the fifth scale of the level of abstraction then, such devices would be scored as transmitting raw information. An audio phone system which included an on-line language translation system would be required to abstract above the raw digital signal level and recognise words, relate these to meaning and employ higher levels of abstraction to support the translation. For an example of a voice operated device such as an isolated word speech based command interface to a washing machine, the device would have to abstract to a level of word meaning in order to recognise the meaning of the single commands. This would be a higher level of abstraction than merely storing the raw signal. The Put-that-there system must abstract both gesture and voice input to a higher level than the raw signals in order to fuse the two meanings together and resolve references between them in order to produce a single interpretation of the meaning of the speaker. Similarly, the COMET system must abstract to a high meaning bearing level of representation in order construct its answers to questions, these must later be translated down to low level raw signals in order to present the answers to the user. Therefore the use of higher levels of abstractions is required to support fission or fusion of information in different modes.

The sixth dimension in the MSM framework is that of Context. As there are different levels of abstraction which can be considered, so there are different contexts. The previous dialogue provides a context in which the targets of references in the current utterance can be found. This context must be maintained to resolve anaphora (references to ideas previously mentioned in the textual dialogue) and deixis (references to objects outside the text part of the dialogue presented in images or other modes). There is a context provided by each user themselves, since they have different preferences for the way graphics are presented, whether information should be presented as tables containing exact numbers or as business graphics which provide an overview of the information, or emphasise contrasts, differences or trends. Each user has a different knowledge of the facilities offered by the system and how to use it; they also have different knowledge of the task domain, with different misconceptions of it which require explanations to be tailored to them in order to indicate and correct these misconceptions. Thirdly there is the context of the task the user is performing which will influence the structure of the dialogue, when the system provides the user with the initiative and when it takes it for itself. The dialogue context, user models, and task plans each provide contextual information which can be used to interpret input and to tailor output when it is represented in the appropriate abstraction. MMI2 contains explicit representations of each of these three contexts (in the dialogue context expert, user modelling expert and informal domain expert respectively) which can be drawn on by the modes during comprehension and generation, and by the communication planning components of the system.

Why Multiple Modes ?

The justification for using multiple output media to present information are given elsewhere in this book. The motivation for trying to use multiple input modes rather than relying on direct manipulation or command languages alone is that individual modes have different strengths and weaknesses as illustrated in Table 1 (after Cohen, 1992).

Table 1 about here

The objective of using multiple input modes is to allow the user to utilise the strengths of each mode while providing mechanisms for overcoming the weaknesses of each. In conjunction with multiple output media the use of the corresponding input media will provide maximum engagement with the information. The identified weaknesses of direct manipulation are all overcome by the strengths of natural language. However, natural language introduces new weaknesses which have to be overcome by the development method of the system, particularly in assessing the coverage of natural language required by a task and the presentation of this to users so that they can develop clear mental models of it (e.g. Fraser, 1993; Karlgren, 1993; Veronis, 1991; Wrigley, 1993). These issues are normally addressed by using Wizard of Oz studies at an early stage in the development method to state the natural language requirements (Nickerson, 1977; Dählback et al, 1993).

Despite these problems with their concommitant costs, there are considerable advantages from natural language both in its abilities to carry complex quantification on both commands and queries as illustrated in the dialogue with the MMI^2 system below:

A1) User: Which machines do not have disks?
A2) System: cmr1 cmr4
A3) User: Add a small disk to every machine that does not have a disk.
A4) System: OK
A5) User: Which machines do not have disks?
A6) System: None
A7) User: Does every machine have a disk?
A8) System: Yes, cmr1, cmr2, cmr3, cmr4.
A9) User: Does every machine have a small disk?
A10) System: No.

and its abilities to express and resolve complex references as in the example below which justify its use. In the first example the user issues a command in utterance A3 which uses complex quantification including a universal quantifier and negation which would be very hard to express through direct manipulation or in an artificial command language. In the second MMI^2 dialogue fragment utterance B9 refers to the disk on the server brought into focus as a component part of the machine referred to by its functional role as a server in utterance B7 rather than simply the last disk explicitely mentioned in utterance B5.

B1) User: Add a 375Mb Disk to the server.
B2) System: OK.
B3) User: Add a small disk to cmr3.
B4) System: OK
B5) User: What is the cost of the disk?
B6) System: 1909 Sterling
B7) User: What is the cost of the server?
B8) System: 4114 Sterling
B9) User: What is the type of the disk?
B10) System: 375 MB Disk.

MMI^2 - a multimodal interface for man-machine interaction with knowledge based systems

To illustrate the architecture and knowledge required by multimodal interfaces which support not only multiple channels in both directions and parallelism, but also semantic abstractions, fission & fusion, and the use of rich contexts , an example system will be described. Although other demonstrators such as COMET (Feiner & McKeown, 1991) are more impressive in generating combined multimedia, they do not include advanced co-operative dialogue or input modes. The MMI^2 system was developed with the purpose of demonstrating the architecture and development method required to produce large scale co-operative interfaces to KBS (Binot et al, 1990). The first demonstration task used in this system is that of designing local area computer networks for institutions such as hospitals or universities (figure 2 shows a screen from this system). A second demonstrator task was used to evaluate the generality of the architecture and the portability of the knowledge: the monitoring of wide area computer network performance (figure 4 shows a screen from this system). The overall architecture of the MMI^2 system is shown in Figure 3.

Figure 2 about here

The architecture of the MMI2 system can be viewed as three layers: the mode layer, the dialogue management layer and the application layer. Each mode in the mode layer of MMI^2 has a generator to produce the mode's output from the system generated meaning representation and a parser to produce the meaning representation from user input. The modes supported are English, French and Spanish natural languages, command language, non-speech audio, graphics for displaying CAD diagrams which support direct manipulation and freehand mouse drawing input, business graphics (charts, tables, pie charts, hierarchies), with direct manipulation by the user on these, and pen based gesture (as used in PDAs) on these and the text modes. The natural language modes use conventional natural language processing techniques, the graphics mode uses explicit knowledge about the design of graphic presentations to produce effective and efficient presentations (Chappel & Wilson, 1993; after Mackinlay, 1986). Therefore several channels are clearly used in both directions between the system and the user. These channels can be used in parallel and are synchronised to support both fission and fusion with synchronisation of system output to multiple modes being managed by the Interface Expert.

As described within the MSM framework, the freedom provided to users by multimodal systems firstly relies upon the use of an abstract meaning representation common to all information sent to or received by each mode. The representation used for this must be able to express all such information in order to allow the choice of the most appropriate mode for output and to fuse the input from different modes. In MMI^2 the language is called the Common Meaning Representation (CMR). This language is used to pass between the mode and dialogue management layers of the architecture, allowing a clear interface where different modes can realise (generate images, language, etc.) any CMR description.

The CMR is used to support the fission and fusion of information between modes and to supply a common dialogue context through which to resolve references made within and between modes. Each CMR packet contains one or more CMR_acts, along with the status, mode and time for those acts. Each CMR_act represents an utterance in a discourse (e.g. sentence within a paragraph or speaker's turn of natural language). Each CMR_act contains an utterance type, one or more CMR-expressions, and a slot for mistakes. Each CMR-expression represents a possible interpretation of an ambiguous utterance. Each CMR-expression then contains annotations, a logical formula, and syntactic information. The status field is only used for internal error checking and the time field to co-ordinate input and output at the interface. The mode field identifies the mode through which the packet was recieved as user input or the one for which it is destined as system output. The role of the utterance type is to define what processing, what functional interface call, should be used (either to retrieve an answer, to determine the truth of a formula, to assert the formula in the application, to retract the formula from the application). The annotations and syntactic information are used to convey details of the gender, number of objects mentioned and other features for natural language processing. The formula includes is a typed first order logic with relativised quantification and second order relation symbols as well as the promiscuous reification of objects and events (after Hobbs, 1985). Table 2 shows an example of the CMR for a user's input of selecting an object on the screen and for asking a question about it. The third example shows the CMR which results from the resolution of the referent

in the question to the object selected in a packet without unresolved references which can be interpreted.

CMR is inefficient at encoding bulky media for presentation since it contains the logical information to be presented. The image, video or sound rendition are then selected by the mode to present this information. In the first demonstrator the exact geographical building structure at the level of walls as well as the logical structure of computer networks was encoded in this representation (as shown in figure 2), with a resulting slowing in performance. In the second demonstrator, map information (as shown in figure 4) was not encoded in CMR, merely the logical label of the overall map.

Table 2 about here

Within the MMI^2 system it is reasonable to consider three levels of abstraction. There is the raw input which is typed, or presented through mouse movements as gestures, manipulations or menu selections. Above this there is the CMR which is common to all modes expressing the logical context of communication actions. Thirdly there is a higher level of abstraction used to plan communication in terms of communication forces (Cohen et al, 1990). At this level, communications acts are labelled as providing such things as apologies, problem reports, justifications, or requests. These follow the philosophy of communication acts which are common to intentions that can be expressed in any mode proposed by Maybury (1991) following the notion of communication as an action based endeavour originally proposed by Austin (1962). In addition to these three levels of abstraction there are clearly other local abstractions within the overall system: within the gesture mode strokes are combined into multi-stroke gestures; within the natural language modes there are syntactic abstractions; within the graphics mode pixels are place together into icons to represent objects or into lines and surfaces. However, each of these abstractions is specific to a mode and they are used as steps to relate communications in each mode itself to the meaning representation which is common to all modes. Therefore considering the three levels of abstraction mentioned above provides a clearer view of the operation of the overall system.

The second necessity for a co-operative multimodal system is that there is a common reference context for all objects. MMI^2 contains a Context Expert which stores all objects referred to in the CMR representations of the dialogue which pass between the mode layer and the Dialogue Controller and it provides the Dialogue Controller with candidates to resolve diexis and anaphora (e.g. Cohen et al, 1990). Therefore each mode can refer to objects mentioned in other modes where the references will be resolved by the Context Expert as illustrated in Table 2. For example, the user can combine text input and mouse pointing (e.g." Is using thin cable possible in <mouse select> this shaft?") and the system can combine graphical output with text (e.g. "What is the type of <system highlight cable> this cable?").

Figure 3 about here

The range of contexts used in MMI^2 to interpret user input and generate system output is larger than just the dialogue context. Two other domains of knowledge are represented as contexts: the context of the user, and the context of the task being performed. The user model contains a model of the beliefs of the user (Chappel et al, 1992). It monitors all messages passing between the mode and dialogue management layers in CMR and extracts from them beliefs which the user holds (both correctly and incorrectly with respect to the knowledge stored in the KBS in MMI^2 which are assumed to be correct), and the intentions of the user. This user model then acts as a server to other parts of the system which require knowledge of the user, such as the graphics manager for planning effective graphics communication, the natural language generators for generating text, and the communications and informal domain experts for planning multi-modal fission.

Figure 4 about here

The informal domain expert contains plans of tasks the user may wish to perform. These provide a context in which to informally evaluate the user commands and queries, and a model of the position of the user in a task can be drawn on when the Communication Planning Expert is generating system output. When the user asks "What is the cost of the network?", the cost of each item in the network must be known. If some items are under specified for pricing (e.g. a cable has been classed as a generic cable rather than a particular one for which a sales price is listed) such unmet preconditions to the calculation of the price will be determined by the informal domain expert. A second example could be if the user issues a command to "Add a computer to the network?"; in this case the unmet preconditions would be the exact type and location of the machine (e.g. a Sun Sparc2 in Room36). In both these cases the informal domain expert plans provide the context to identify the unmet preconditions that would be passed to the communication planning expert which would in turn continue a co-operative dialogue with the user to meet the objective of their plan by asking for the required information. The dialogue which arises from the use of the task context can be deeply nested since many preconditions may have to be met, but it is clearly directed to the aim of the user which has been identified in the task plan, and thereby conforms to Grice's cooperativity principle.

The communication planning module generates large CMR structures which can be passed to the modes for output to the user. The response the communication planner would make to this example illustrates the use of the third and highest level of abstraction mentioned above: that of communication actions. The reply to the request to "Add a computer to the network?" might be the four statements (as rendered by the English mode):

1) I am sorry.
2) The location and the type of workstation CMR98 are underspecified.
3) Adding a workstation requires the specification of the location and type.
4) What is the type of the workstation CMR98.

These four statements would be associated with the communication actions of Apology, Problem Report, Justification and Request respectively. If the problem had already arisen recently in the dialogue context and the same output produced then the dialogue context would show this and the justification would be omitted. Similarly, if the problem appeared frequently, the apology may be replaced by an exclamation. The context of the user model could also be drawn on to elaborate the justification if it showed that the user was unclear about the types of workstation which were possible answers, so that a list would be provided. This example shows how the task context is used in the informal context expert to trigger co-operative dialogue, and both the context of the user model and the dialogue context are used to modify the system output.

This example of communication planning produces output purely for natural language modes, and does not illustrate multimedia output. The output generated can be directed at different modes depending on which is most effective at conveying the intended class of information. Table 3 shows the relation between different information types and their presentation from the COMET system which is similar to the heuristic rules used in MMI^2.

Table 3 about here

As mentioned above, the informal domain expert supports the interpretation of user input CMR packets against tasks plans. Obviously these do not answer most questions or perform most commands represented in CMR packets. Indeed, no commands are performed as a result of this informal evaluation which merely checks that they can be performed at the time they are issued. The main application program contains expert systems which perform the design of computer networks and store those designs in an object-oriented database. It also contains analysis experts which can analyse the stored design for its potential extensibility and other properties of interest to the application user. To perform the design, analyses, assert building components or design requirements in the database, requires formal evaluation against the database or experts. To answer queries about the objects in the building or on the network requires formal interpretation against this database. This application is

accessed through the formal domain expert which provides a functional interface which includes three operations: Assert, Retract and Goal, to update the knowledge bases, retract information from them, and ask questions of them. This is similar to the level of operation provided in other knowledge based systems (e.g. Guha & Lenet, 1990). For each application domain predicate defined as permitted in CMR within the semantic expert there is an interface in the formal domain expert which maps it to the application itself to support the main operation of the system.

MMI^2 does not include a broad knowledge base of common-sense knowledge (e.g. Guha & Lenet, 1990) but it must include more than just the limited domain knowledge for the demonstrator application for designing computer networks. The interface expert contains information about the interface itself. This is available to answer questions about the interface and its capabilities, but also for the evaluation of predicates in the CMR about the interface. For example, if the user commands the system to "draw a bar chart of the cost of computers on the network" then concepts such as BarChart are not network design concepts, but interface concepts; so that their evaluation is against the domain of the interface rather than network design. The third domain for formal evaluation is that of the user model which is also available for the evaluation or interpretation of predicates in CMR about the user (e.g. to answer questions such has "Who am I?"). All three domains are accessed through the same functional interface which supports the formal evaluation of CMR predicates by the dialogue controller and support the re-use of the modules outside the single application task of local area network design used in the first demonstrator as shown by the development of a second demonstrator in the area of .wide area network management.

Advantages of multimodality

The MMI^2 system described illustrates the architecture required for a multimodal system which supports co-operative dialogue. There are potential modes such as the use of machine vision understanding systems to interpret body posture and facial expression to provide additional data to determine the intended pragmatics of utterances which it does not use; neither does it include continuous output media such as video or sound. Both of these have been considered within the general architectural design and should be compatible with it, with the meaning representation used, and with the view of interaction proposed in the MSM framework.

Such interactive systems as MMI^2 may seem a long way from current multimedia systems but they do provide several advantages. Users can maximise expressiveness by choosing the appropriate input mode: natural language for complex quantification and reference use, freehand drawing for inputting building designs, and pen like gestures for editing, or direct manipulation to move objects in the CAD displays. Secondly, it supports the fusion of multimodal input and the fission of output through the use of a single abstract meaning representation, as well as the use of a single dialogue context to resolve intra- and intermodal referents. Thirdly, the use of the dialogue context, user model, and task model support co-operative dialogue where the user and system have the same aim, while system output is tailored to the user at the point they are in the task and dialogue. Fourthly, the most effective and efficient output mode can be selected for system expression through the use of heuristic communication rules, and the rendition of the abstract meaning representation is generated by further heuristic information design rules within the individual modes (automated graphics design rules or natural language generation system). These four advantages of multimodality arise from the three additional features in the MSM framework (fusion/fission, levels of abstraction towards meaning and context) beyond the three found in conventional multimedia systems (channel direction, number of channels along direction and parallelism).

Limitations of Multimodality

Unfortunately there is a price to pay for the added expressiveness which arises from the multimodal approach. There are serious problems in the use of natural language comprehension modes because of the difficulty with specifying the requirements for a fluent and robust system (Stede, 1993). The conventional techniques for developing natural language interfaces (as used in MMI^2) are very data dependent where the data is collected through Wizard of Oz experiments on potential users in the proposed task. This results in the implementation not of an English natural language interface, but one which understands the limited subset of the English lexicon, grammar, semantics and pragmatics which

are exhibited during that data collection. The engineering issues associated with natural language comprehension are a major current research topic, but at present the development methods do not provide users with a clear mental model of the sublanguage which can be used (Veronis, 1991; Wrigley, 1993).

The second major limitation arises from the abstract meaning representation itself as a medium for encoding graphics and continuous media. Although the CMR has been shown to be sufficiently expressive to represent the information conveyed through the modes it is inefficient. For CAD displays of buildings and networks this included the location of walls, rooms, cables and machines relative to each other. The lines and icons to represent these were generated in the graphics mode. For complex diagrams these representations could be 100,000 lines of CMR logical formulae. For large bodies of text or continuous media to be retrieved these representations would become even larger and less efficient. The CMR is also non-standard, with the result that information must be translated into it for presentation. Further research is required to improve the efficiency whilst maintaining the expressiveness of the meaning representation language. The third problem with developing commercial products incorporating the context models included in MMI^2 to represent the user, task and domain is the effort required to encode them and link the interface to the underlying application. The fourth limitation in applying the approach is in the generality of the output generation rule sets. Although the concept appears to have been proved, further focused projects are required to refine them along with the context sensitive reference resolution rules,

Complete multimodal systems such as MMI^2 are currently only research demonstrators which can produce potent illustrations of multimodal interaction, but are not even robust enough for real user evaluation. However, many components shown in this system are being brought to the marketplace where there is seen to be a need for them. Gesture interfaces of the form used in MMI^2 have now been incorporated in personal digital assistants (PDA) for recognising symbols, even if they are not practically sophisticated enough to interpret cursive writing sufficiently reliably yet. The freehand input of building drawings and their automatic interpretation into objects used in MMI^2 has also been incorporated in many PDAs. Several companies are developing speech input command systems which will fuse their input with that from gesture mode using principles developed in MMI^2. Many spreadsheet developers are including business graphics creation rule sets to automatically generate bar charts, pie charts and graphs from spreadsheet data using rules similar to those used in the MMI^2 graphics manager. Constrained natural language query systems for databases incorporate technologies which are a subset of those used in the MMI^2 natural language input mode, and many of these are starting to incorporate graphical interface tools to help complement natural language, combining modes again using principles seen in MMI^2. It is not practical to move from current application architectures to logic based dialogue centred systems such as MMI^2 in one step. It is necessary to isolate out those aspects which can be combined into more conventional designs to add functionality to them in order to dynamically create presentations which are tailored to the user, task and dialogue context as those in MMI^2 are.

MIPS - a multimedia information retrieval and presentation system

The first problem in the implementation of the multimodal approach identified above was the use of natural language as an input mode. If this is replaced with a less free input mode such as menu based form completion the problem of clarity in users' mental models of the limitations on input will be overcome. If the task is limited in potential complexity from a design task to an information retrieval task then the expressiveness required in the dialogue will be commensurately reduced. However, if the dialogue is restricted in this way, then the expressiveness permitted the user is also reduced, but although dialogue will be more predictable, it can still include complex subdialogues.

The second problem identified was the use of the CMR to encode graphics and continuous media for system output. In MMI^2 the logical representation in the CMR carried the identity and relationship of objects to be presented to users. For generated natural language output these representations can be very large. Equally the CMR is a non-standard representation and retrieved information would have to be translated into it. In an information retrieval task it is unreasonable to assume that all the retrieved data will already be written in such a logical language, rather it will be in text or media formats. Queries for information retrieval are normally made in a logical representation (e.g. languages such as

SQL) which can be extended to convey both details about the expected media sought (e.g. a video of New York harbour rather than a textual guide to it) and about the pragmatics of the retrieval process itself (e.g. a cost and time above which the retrieval would not be desired). The major problem comes with the returned data format. Existing databases use many different formats.

To allow any operations on the returned data these would have to be converted to a single format, or a group of formats understood by output modes. Equally, for large amounts of retrieved data which were to be stored (or cached) locally to allow users to browse them, the media would have to be linked together to support the user's task of browsing. Current hypertext systems are closed individual products where the reader can choose routes through a pre-written web but they support links between data on the basis of logical structure and allow the incorporation of different media. Unfortunately, both the content and the link design have usually been completed by the author. A step towards opening up such systems is provided by the Microcosm system (Davis et al, 1993; Hall, this volume) which separates the link structure from the data assets presented in the hypertext thereby allowing users to link pre-existing documents, images and other media items to a web. Since media items are independent they can also be stored in the formats of common presentation tools, providing users with a more consistent system image between the hypertext and other tools on a system. This changes the hyperdocument from being purely an artifact created by an author to one which can be part of an open information system where assets can be re-used in many documents. The next stage in opening up hypertext documents is to represent the link structures themselves in an interchangeable language. This would allow a set of data assets and the link structure to be portable across different presentation and link authoring platforms. This standardisation of open link representation is available in the ISO HyTime standard for hypermedia time based data (Newcombe, 1991; ISO, 1992). These two advances of opening hypertext by separating assets from links and then using standardised representations for both assets and link webs support the portability of hypertext. By employing a HyTime web to represent the logical structure of returned data in the form of links, with the data assets themselves represented in their original stored formats a more efficient portable representation would be achieved for data to be browsed or presented than the CMR used in MMI^2. This arrangement would allow the representation of raw multimedia data, with a logical representation of both queries and the link structure of returned data. A conceptual representation of these logical labels would be required in a domain context model in a KBS although this would not be required throughout an architecture. Similarly, the highest level used in MMI^2 (the communication act level) could also be represented for limited reasoning about communication planning in the form of information presentation design although this too would not be required to be transmitted throughout an architecture, as indeed it was not in MMI^2 where it was limited to the Communication Planning Expert.

While overcoming the problems introduced into multimodal systems by a common meaning representation of both user input and system output by using a task of reduced dialogue complexity, extended logical query languages for user input and a standard hypermedia language which separates data and structure for output we have also maintained the functionality of fission/fusion and multiple channels in both directions from the multimodal approach. However, we also wish to maintain the advantages provided by the use of context for tailoring output and generating it from communication rules. To do this we must maintain contextual models of the domain, task, user and dialogue which can be used to constrain query processing, the linking of returned data into hypertext webs and the presentation of those webs to users, through the use of communication rules which guide the linking and presentation of returned data. The Multimedia Information Presentation System (MIPS) demonstrates this more practical approach to multimedia presentation while drawing on some intelligent communication heuristics.

It is currently possible with commercial products on a PC to retrieve information from SQL servers, data bases and documents across a network whether that information is text, relational tables, images, sound or video. That information can be presented through tools in a commercial windowing system to users for them to read, or cut & paste into multimedia documents.

Unfortunately, the range of different data sources which can be used is limited; the queries to produce the information must be specified for each of the data sources and not as a single query to retrieve the information for each of them; and the tools to present the information will each occupy a different window on the screen and use proprietary presentation styles which differ from each other depending on the source format of the information.

The second currently available form of presenting multimedia information is by authoring it in a proprietary tool into a discrete document or hypermedia network which the user can then browse through. However, the user has no access to documents outside the hypermedia network and are tied to a proprietary representation format.

The MIPS system seeks to combine the best of these two approaches and thereby to overcome the problems of each. MIPS is a presentation tool which includes an access mechanism to distributed heterogeneous data sources. Therefore the access to databases and document stores is provided. It includes a conceptual model of the domain covered by the data sources encoded in a Knowledge Based System which supports a single query tool that can be used to recode a query for each relevant data sources from a single user query. This allows data to retrieved on a topic from all available data sources as a result of a single query rather than requiring individual ones to each data source. The data that is returned from the different data sources will not be presented in different windows for each source, but the data will be combined together into a single relational table, or into a single hypermedia document for presentation. The information will not have to be presented in the stored presentation style since all information will be tailored by a Knowledge Based information design system which will tailor it to the needs of individual tasks and users.

When the information is retrieved it can be included in an existing hypermedia web which will be grown with new information. The growing of this web with new information requires a clear description of the semantics of the information which is provided by the conceptual model of the query in the domain. It also requires skill in designing the web additions which are created by the information design system using the conceptual model of the query.

The hypermedia web itself does not have to be tied to an individual producer's proprietary representation since the data which is the content, and the structure of the web are separated from each other. The data can be stored in any format for which presentation mechanisms are provided, or for which there are available translators to the formats of available presentation mechanisms. This not only frees the data from format constraints but allows it to be used in many hypermedia documents, or by complete different applications. Even the definition of the structure of the web itself does not have to be stored in a proprietary representation, since the format used is that of the ISO HyTime standard which is based on SGML and not one created solely for the system.

There are not currently many HyTime products on the market, but ten years ago when SGML first became a standard there were not many SGML products available. It is expected that in the same way that many government agencies and large companies now use SGML based products to support the interchange of text, large organisations will adopt HyTime to support hypermedia. This will motivate software developers to produce HyTime authoring and presentation systems based on the standard.

The architecture of the MIPS system is shown in figure 5. Although its objective is to support the retrieval of information from heterogeneous distributed databases and to use communication rules implemented in a KBS to organise these and present them, this system must be more market focused than MMI^2 and therefore must address intermediate market needs too. Any hypermedia product at present must be able to support the conventional publishing life cycle of authoring, distribution and reading without adding intelligence in itself. Therefore the MIPS system must be capable of this too, and be able to present pre-written hypermedia documents without any queries. To do this presentation tools, a presentation manager to control the browsing of the web structure and the delivery of data assets to those tools, and a storage mechanism for the hypermedia web are required. The shaded area in Figure 5, shows the architecture required for these functions: presentation tools, a presentation manager, and a HyTime Engine to provide fast access to the object oriented database containing the web link structure.

Once the querying facility is added, it can be used in two ways. Firstly, users can issue queries through the formatted menu interface provided by the query tool. Secondly, authors of hypertext application documents can include pre-written queries in those so that they provide the most up to date information (for example, a table of airline flight costs in a hypertext published tourist brochure can be updated from a remote database when that node is viewed rather than relying on the information provided at the time of authoring). In order to connect the hypermedia tool to existing heterogeneous databases, a

selection and retrieval tool to select the appropriate database as the target for a query, and to format the query for that database is required. Once the query is formatted it must be dispatched to the remote or local database through a communications module, and the returned data must be passed back to the selection and retrieval tool. This must be incorporated into the HyTime web representation by a Web Builder, so that the node can be presented. Again, in Figure 5, the modules to provide this functionality are also shown: Selection and Retrieval Tool, Communications Module and Web Builder. The functionality to support the access of heterogeneous remote data sources will not be discussed in this chapter although it is presented in Behrendt et al (1993).

To support the dynamic construction of screens in response to user queries at run time rather than pre-written queries produced by an author requires a Query Tool for the user to express queries in (also shown in Figure 5). In order to perform the selection of the appropriate database from the heterogeneous set available the system must know what databases are available, what information they represent, what format queries and returned data use, and other information about cost, access time and other non-functional requirements in order to optimise the query. This information is stored in the Knowledge Based System which supports the Query Tool and Selection and Retrieval Tool in the construction of queries. Similarly, the returned data must be constructed into a node, which must be linked to other nodes, and then presented on the screen using the most effective and efficient presentation mechanisms available for that class of data. Again, rules about information design to support this task is stored in the KBS which supports the web builder in constructing the node, and the Presentation Manager in selecting the most efficient presentation tools to render a presentation mechanism.

Figure 5 about here

As with the MMI^2 system, the architecture is clearly divided into three layers responsible for the presentation of information, the dialogue management and the application itself. The application in this view of MIPS is the conjunction of local and remote databases. This clear division is a little confused by the existence of two permanent stores in the dialogue management layer: the HyTime store, and the Knowledge Base; and one in the presentation layer: the files of local data assets for presentation. Given current hardware, the size of video and sound data files, and the speed of transferring them, it is necessary to keep large data assets as close as possible to the presentation tools to provide the speed and quality of presentation demanded by users. MIPS is designed to be run as a client server system with several client machines running presentation tools, with one server responsible for the database access. Therefore to keep data assets close to the presentation tools they are stored on the client machine on which they run, although they can be regarded as data assets cached from those held in databases. The dominant dialogue control mechanism is the web structure itself which restricts the links available to users within the web. Although the hyperdocument may be considered as the application, as an application divided between assets and web, the sole role of the web is to control dialogue. This is therefore represented within the dialogue management layer. The KBS store contains knowledge used by the KBS to select databases and design nodes and presentations. These functions are comparable with the informal domain expertise and communication planning functions in the MMI^2 system and for the purpose of the main application function are also dialogue management functions so their placement in the dialogue management layer is appropriate.

One of the objectives of the MIPS system is to improve the portability and interoperability of hypermedia systems. Portability is addressed by using existing data formats and presentation tools, and by using a standard representation for the web. To promote interoperability knowledge about the applications with which it must operate has been included in the dialogue management layer, within the KBS: about databases to which it is connected, and about presentation tools available at the local site. Also knowledge has been included about the presentation of data assets, and the construction of nodes, links and presentations, so that data can be integrated into the hyperdocument web. There are several discrete bodies of knowledge that have been included within the KBS, but unlike the MMI^2 architecture where the knowledge is distributed around the system most of it itself a KBS written in Prolog, in MIPS all this knowledge is placed in a single KBS separated from the rest of the architecture. This allows the KBS to act as a server to the rest of the system and for a core presentation

only system to operate without it. However, this does not constrain the complexity of the KBS itself which is now internally modularised.

Figure 6 about here

There are four domains of knowledge which the KBS must know about to perform its functions. Firstly, it must contain knowledge about database access in general and the optimisation of queries. Secondly it must understand the principles of information design in order to construct nodes, links and screens. Thirdly it must understand the language of the application domain since actual queries to databases in an application will use this information, and the actual data to be presented in an application will be in terms of this information. Fourthly, it must have knowledge of the actual presentation tools present at a site and the rules for choosing these for a presentation mechanism. These four domains of knowledge interact so that there is general knowledge about database access and information design, then layered upon this there is knowledge specific to the application domain. The knowledge which was spread through the communication planner, graphics manager and interface expert of MMI^2, here all resides in information design and presentation knowledge. What were the domain experts in MMI^2 here become knowledge about the domain of database access, and the application domain. In order for the KBS to perform its functions, other knowledge which was present in MMI^2 must also be included here. In order to design information it is again necessary to represent the dialogue context, and to classify the user in a user model. Similarly, to permit any portability of the KBS it is necessary to divide four domains of knowledge between task models which describe the functions to be performed in an application domain platform, and then layer application domain knowledge on top of this in a conventional KBS domain model.

These various layers of knowledge required to automate presentation design must be acquired, and cannot all come from an application builder or author who will not be a knowledge engineer. General knowledge about database access and information design exist within the basic MIPS KBS. There is also a core ontology of terms defined which are used in the MIPS document type definition (DTD) for HyTime documents produced for MIPS. On top of this it is necessary for the application builder to extend the ontology for the application domain through a simple tool designed for this purpose. The domain specific rules for database selection and information design can also be entered by the application builder (author) since only a very limited set of these beyond the ontology are required (e.g. constraints on some transitive inferences such as 'travel from A to B, via how many intermediate places'). It is necessary to extend the directory of remote databases available for an application, and indicate the schema of these using terms in the ontology, but this is a form completion exercise and does not require knowledge acquisition. The application builder will also specify the simple user modelling structure by choosing categories of features which can be associated with different user groups in the ontology. The application builder will therefore be required to enter some information into the KBS, but form interfaces are provided to limit the complexity of this operation. These must be evaluated before the system could be viable, and they will undoubtedly have to be changed, but the present system is intended to address this potential problem of overloading the application builder. Once an application has been written in a domain (e.g. Greek tourism) very small additions would be required for further applications. Therefore the knowledge can be seen as one general layer, a second for the application domain and a third for each specific application. The application builder would obviously have to author the HyTime web for an application as well as entering the KBS knowledge. It is probable that some data assets would also have to be created for an application, although these could already exist or be the responsibility of database providers.

Once the application has been created it must be installed at each site. Another layer of customisation is required here to state which actual databases are connected to it, which actual presentation tools are present at the site and which actual user groups are potential users. The preferences of the user groups can also be tailored to the site. This provides a fourth layer of customisation to the user site. The fifth layer of customisation available is to each independent user who can customise their own user model to include their own preferences for database access variables (cost, time etc.), presentation tools to be used, language etc. These can be defaults for users at a site, or can be set explicitly by users willing to devote the effort to this task. The sixth level of customisation available is to an individual session's

dialogue context. This is automatically performed by the KBS which keeps track of the context throughout use and automatically updates the user model of a user on the basis of usage.

These six layers of the customisation process provide the knowledge on which the system can base its retrieval and information design judgements and tailor them to individual users so that they appear intelligent. To perform these customisation steps three different groups of users have been distinguished: application builders who develop the application, site managers who configure the application to a site, and end users who configure the system to their own preferences. Other groups of individuals could also be involved such as specialist asset generation teams (including film directors, graphic artists etc..), remote site database managers, or even domain modellers.

In this application both the problems of heterogeneous database access and those of information design must be addressed. A major worry is the automation of these two areas is the need for general knowledge and ontologies. Stated boldly, the generation of general knowledge systems is a long term goals of the artificial intelligence community and should not be regarded as a solved problem. Within the limited application supported by the MIPS system, and with the deliberate exclusion of any natural language input it is hoped that these issues have been sufficiently constrained so as not to be onerous to application builders. If this is proven not to be so in evaluation, the approach should not be rejected immediately. Artificial intelligence workers such as Guha and Lenat (1990) are developing exactly the form of general knowledge base which could be used to support heterogeneous database access (Lenat & Guha, 1991) and may provide a general basis for the link creation part of the information design process in MIPS applications.

MIPS supports the retrieval of any information stored in data sources. One practical limitation on this is the communication of large media (i.e. video, sound) over computer networks. If this is not practical in real time then the system is only useful for retrieving less bulky media. This is an immediate problem, but even now there are local uses for this technology if not over wide area networks. A second problem with the retrieval of such media is the indexing of it in databases. There are currently no clear standards for the storage of multimedia data, nor for querying it. Querying may seem a strange point to raise, but if a video display is being automatically composed, it could also be automatically edited. That is, a ten minute video of a scene could have a shot extracted from it, or a series of shots which could be put together into a three minute sequence. If the video were indexed using techniques such as that proposed by Burrill et al (in press), then it could have an index attached to each scene and shot not only with the identity of characters, locations and actions in it, but also video attributes such as whether it was an establishing shot, close-up, pan etc. Such indexing on semantic content and video attributes would support the retrieval and composition of sequences according to rules of video direction. This is one example of the forms of presentation design which could be developed for multimedia systems using a MIPS architecture as the starting point; although it has not been developed yet.

The MIPS system is a demonstrator which is considerably simpler than MMI^2. Within the MSM framework of Coutaz et al. described above, it supports only one channel from the user to the system since all user input is through keyboard or mouse selection, but it uses visual and auditory channels for presenting information to the user. It does not support the fusion of user input, but since it can produce several media in reply to a query it could be said to support fission. It has been designed as a multimedia rather than a multimodal system which does not support advanced user input dialogue so this judgement is not surprising. Similarly, parallelism is only supported in system output at the physical level and could only be said to be supported in user input at the task level if queries are pending the return of data while a user continues to browse the hyperdocument. The abstraction of the dialogue supported is more problematic. Outside the KBS there is no abstraction of navigation commands to the web, and little abstraction in data source queries. Within the KBS there is some abstraction of the query to the user's task level but this is mainly to support the use of contexts. The MIPS KBS is designed to accommodate the context of the task, user and dialogue in designing presentations for returned data and linking these into the web. It is this use of contexts in presentation generation which is the contribution of any intelligence in this system. The system has not yet been implemented to a stage which will support evaluation so it cannot be judged whether this is enough. However, an evaluation will take place in Greece of an application in the tourist domain in order to answer this question.

Conclusion

Current multimedia systems present text, images, sounds and video as static objects. The organisation of these objects is determined by the author, and the reader has a comparatively passive role. In order to involve multimedia in more applications it must become more active with the representations used must include meaning which can be manipulated in more complex dialogue structures. The technology currently used to support multimedia presentation does not incorporate any analysis of the signals, any discussion of human-computer communication modalities inherently involves, at some level, the machine's determination of the content of messages, and its need to communicate the content of its own messages. Multimodal systems use different media for input and output, and rely on abstract representations of the information in order to control the dialogue and application. An example multimodal system was described - MMI^2. Such systems are far from the current market but they provide the basis for determining the theory of multimodal communication which is required to be stated in detail if multimedia presentations are to be automatically created.

A second less advanced multimedia system (MIPS) has been described which is an advance on current hypertext, including some intelligence to support the dynamic creation of multimedia documents from retrieved data. This illustrates a second route towards the introduction if intelligent multimedia through open systems which interact with existing data sources. Although this system does not include the advanced dialogue of a multimodal system, its use of context information to dynamically create presentations and the local use of conceptual and communication act levels of abstraction within the information design rules of the KBS may incorporate the most robust aspects of the more exotic system in a way which can be included in commercially supported products.

It is arguable that neither of these systems portrays intelligence as the title suggests. This conclusion is either drawn because of a conviction that it is a misnomer to term any system intelligent or because there are few examples presented that show the systems in operation. If the first is true, the term is used suggestively rather than with any psychological conviction. If the second is the case, on several occasions when MMI^2 has been demonstrated to computer professionals they have refused to believe that it was interpreting the user's input or generating its own output. They claimed that there was either somebody behind the curtain, or the demonstration was so well prepared that the system could not stray outside it. It is this impression of disbelief in those who see the systems which leads to their being called intelligent. Their developers do not make any greater claim.

This chapter has not described various rule sets for dialogue context, user or task modelling, for generating output in different modes nor for selecting media, but references have been provided to sources of this information for those who wish to acquire it. These rule sets for designing system output are still active areas of research but some are being incorporated into existing major software products to provide them an edge in the current market. The heuristics described in this chapter may not currently be available in multimedia products, but they illustrate what may well be available in the future, since the market appears to be demanding the use of these techniques.

Acknowledgements

The research reported in this paper was partly funded by CEC Esprit II grant 2474 to the MMI^2 project and partly by CEC Esprit III grant 6542 to the MIPS project. Organisations involved in the MMI^2 project are BIM (Belgium), ISS (Spain), ADR/CRISS (France), INRIA (France), EMSE (France), SERC RAL (UK), University of Leeds (UK). Organisations involved in the MIPS project are.Longman Cartermill (UK), SERC RAL (UK), STI (Spain), SEMA Group Benelux (Belgium), Herriot-Watt University (UK), Trinity College Dublin (Eire), DTI (Denmark).

References

Austin, J. (1962) *How to do things with words*; editor J.G. Urmson. England: Oxford Univ. Press.

Behrendt, W., Hutchinson, E., Jeffrey, K.G., Kalmus, J., Macnee, C.A., and Wilson, M.D. (1993) Using an intelligent agent to mediate multibase information access. In *Proceedings of the Workshop on Cooperating Knowledge Based Systems*, Keele, UK September.

Bernsen, N.O. (1993) Modality Theory: Supporting Multi-Modal Interface Design. In *Proceedings of the ERCIM Workshop on Multimodal Human-Computer Interaction*, Nancy, 2-4 November 1993, INRIA: Nancy.

Binot, J-L., Falzon, P., Perez, R., Peroche, B., Sheehy, N., Rouault, J. and Wilson, M.D. (1990). Architecture of a multimodal dialogue interface for knowledge-based systems. *Proceedings of the Esprit '90 Conference*, pp 412-433. Kluwer Academic Publishers: Dordrecht. Also published on CD-ROM by the CEC: Brussels.

Bolt, R.A. (1980) "Put-that-there": voice and gesture at the graphics interface. *Computer Graphics*, 14(3), 262-270.

Bush, V. (1945) As We may Think. *Atlantic Monthly*, July, 101-108.

Burrill, V., Kirste, T., Weiss, J. (in press) Time-varying sensitive regions in dynamic multimedia objects: A pragmatic approach to content-based retrieval from video. *Software and Information Technology* - special issue on Multimedia.

Card, S.K., Robertson, C.G., Mackinlay, J.D. (1991) The Information Visualiser: an information workspace. In *Proceedings of ACM CHI '91: Human Factors in Computing*, 181-188; ACM Press: New York.

Chapanis, A., Parrish, R.N., Ochsman, R.B. & Weeks, G.D. (1977) Studies in Interactive Communication: II. The effects of four communication modes on the linguistic performance of teams during cooperative problem solving. *Human Factors*, 19(2): 101-125.

Chappel, H., Wilson, M. and Cahour, B.(1992) Engineering User Models to Enhance Multi-Modal Dialogue. In J.A. Larson and C.A. Unger (Eds.) *Engineering For Human-Computer Interaction*. Elsevier Science Publishers B.V. (North-Holland): Amsterdam, pp 297-315.

Chappel, H. and Wilson, M.D. (1993) Knowledge-Based Design of Graphical Responses. In *Proceedings of the ACM International Workshop on Intelligent User Interfaces*, pp 29-36. ACM Press: New York.

Clarke, A.C. (1968) *2001: a space odyssey*: a novel. London: Hutchinson.

Cohen, P.R. (1984) The pragmatics of referring and th modality of communication. *Computational Linguistics*, 10(2):97-146, April-June.

Cohen, P.R., Morgan, J. and Pollack, M.E. (1990) *Intentions in Communication*. MIT Press: Cambridge, Mass.

Cohen, P.R. (1992) The Role of natural language in a multimodal interface. In the *Proceedings of the Fifth Annual Symposium on User Interface Software and Technology*, 143-149, ACM: New York.

Cotton, R. and Oliver, R. (1993) *Understanding HyperMedia*, Phaidon Press: London.

Coutaz, J., Nigay, L. & Salber, D. (1993) Taxonomic Issues for multimodal and multimedia interactive systems. In Noëlle Carbonnell (Ed.) *Proceedings of the ERCIM Workshop on Multimodal Human-Computer Interaction*. INRIA: Nancy.

Crane, H.D. & Rtischev, D. (1993) Pen and Voice Unite. *Byte*, 18(11), 99-102.

Dahlbäck, N., Jönsson, A., & Salber, D. (1993) Wizard of Oz Studies - Why and How? In Gray, W.D., Hefley, W.E., and Murray, D. (Eds) *Proceedings of the 1993 ACM International Workshop on Intelligent User Interfaces*, pp 193-200. ACM Press: New York.

Davis, H., Hall, W., Pickering, A., and Wilkins, R. (1993) Microcosm: An open hypermedia system. *Proceedings of INTERCHI '93 Conference on Human Factors in Computing Systems*, ACM: New York.

Feiner, S. and McKeown, K. (1991) Automating the Generation of Coordinated Multimedia Explanations. *IEEE Computer* 24(10); 33-41.

Fraser, N.M. (1993) Sublanguage, register and natural langauge interfaces. *Interacting with Computers*, 5 (4) 441-444.

Grice, H.P. (1975) Logic and Conversation. In P. Cole and J.L. Morgan (Eds.) *Syntax and Semantics*, volume 3, pages 64-75. Academic Press, New York.

Guha, R.V. and Lenat, D.B.(1990) Cyc: a midterm report, *AI Magazine*. 11 (3), 32-59.

Hobbs, J.R. (1985) Ontological Promiscuity. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, Chicago, June 1985, 61-69.

ISO (1992) ISO/IEC JTC1/Sc18/WG8, Information Technology, *Hypermedia/Time-based Structuring Language (HyTime)*. ISO/IEC D18 10744.1.1.

Karlgren, J. (1993) Sublanguages and registers: a note on terminology. *Interacting with Computers*, 5 (3) 348-350.

Lenat, D.B. & Guha, R.V. (1991) Ideas for applying Cyc, *Tech. Report. ACT-CYC-407-91*, MCC: Austin TX.

Mackinlay, J. (1986) Automating the Design of Graphical Presentations of Relational Information, *ACM Transactions on Graphics*, 5(2), 110-141.

Maybury, M.T. (1991) Planning Multimedia Explanations Using Communicative Acts. In *Proceedings of the Ninth national Conference on Artificial Intelligence, AAAI-91*. Morgan Kaufman: Los Altos CA.

Nelson, T.H. (1988) *Hyperdocuments and how to create them*. Prentice Hall: New Jersey.

Newcombe, S.R., Kipps, N.A. and Newcombe, V.T., (1991) The HyTime Hypermedia/Time based document structuring language, *Communications of the ACM*, Vol 34 (11), 67-83.

Nickerson, R.S. (1977) On conversational Interaction with Computers, in *User-Oriented Design of Interactive Graphics Systems*. New York: ACM.

Oviatt, S.L. & Cohen, P.R. (1991) Discourse structure and performance effficiency in interactive and noninteractive spoken modalities. *Computer Speech and Language*, 5(4): 297-326.

Pierce, J.R. (1980) *An introduction to information theory: symbols, signals & noise*. 2nd rev. ed. London: Constable.

Rubin, A.D. (1980) A theoretical taxonomy of the differences between oral and written language. In *Theorectical Issues in reading Comprehension*. Lawrence Erlbaum Assocs.: Hillsdalee, NJ.

Stede, M. (1993) the search for robustnes in natural language understanding. *Artificial Intelligence Review*, 6(4) 383-414.

Templeton, A. (1993) Strategies for Business, In *European Multimedia Yearbook 93*, Interactive Media Publications: London.

Veronis, J. (1991) Error in natural language dialogue between man and machine, *International Journal of Man-Machine Studies*, 35, 187-217.

Wrigley, A. (1993) In defence of sublinguistics, *Interacting with Computers*, 5 (4) 439-440.

Figure Legends

Figure 1: The MSM Framework: A 6-D space to characterise multi-sensory-motor interactive systems (After Coutaz et al., 1993)

Figure 2: An example screen from the first MMI^2 demonstrator showing different interaction modes with the underlying application:

Figure 3: Architecture of the MMI^2 Multimodal Man-Machine Interface for Knowledge Based Systems.

Figure 4: An example screen image from the second MMI^2 demonstrator for monitoring computer network performance showing different interaction modes: a map overlaid with the physical structure of the network; natural language input mode, the logical structure of the computer network.

Figure 5: Architecture of the MIPS multimedia presentation system.

Figure 6: Screen design for a hypermedia tourism application for the Barcelona 1992 Olympics using the MIPS presentation system. The main image is a short video of the site.

	Direct Manipulation	Natural Language

Strengths	1. Intuitive 2. Consistent Look & Feel 3. Options Apparent 4. Fail Safe 5. Feedback 6. Point, Act 7. Direct Engagement with semantic object 8. Acting in'here & now'	1. Intuitive 2. Description including a) Quantification b) Negation c) Temporal Information 3. Context 4. Anaphora 5. Delayed action possible
Weaknesses	1. Description including a) Quantification b) Negation c) Temporal Information 2. Anaphora 3. Operations on large sets of objects 4. Delayed actions difficult	1. Coverage is opaque 2. "Overkill" for short or frequent queries 3. Difficulty in establishing and navigating context 4. Anaphora is problematic 5. Error prone 6. Ambiguous

Table 1: Complementary Interface Technologies: Direct manipulation and natural language

```
(1) CMR(
[
 CMR_act_analysis(
            u_type(phrase([var(x1)]),none),
            [
            CMR_exp(
                        [],
                        identity(var(x1),const(cmr_Shaft0)),
                        nil)],
            nil)],
ok,
Graphics,
time(56,53,23,11,6,1991))

(2) CMR(

[
 CMR_act_analysis(
            u_type(polar,question_mark),
            [
             CMR_exp(
                        [
                         anno(x1,[name(using-thin-cable),singular,definite]),
                         anno(x2,[singular,definite,neuter])],
                        description(desc(E,x1,USING,
                                    identity(var(x1),const('using-thin-cable'))),
                        description(desc(E,x2,SHAFT,true),
```

```
                    description(desc(E,x3,IS_POSSIBLE,true),
                    conj(
                              [
                              atom(ARG1,[var(x3),var(x1)]),
                              atom(ARG2,[var(x3),var(x2)])])))),
                    nil)],
          nil)]
ok,
English,
time(56,53,23,11,6,1991))
```

```
(3) CMR(

[
 CMR_act_analysis(
          u_type(polar,question_mark),
          [
           CMR_exp(
                    [
                     anno(x1,[name(using-thin-cable),singular,definite]),
                     anno(x2,[singular,definite,neuter])],
                    description(desc(E,x1,USING,
                              identity(var(x1),const('using-thin-cable'))),
                    description(desc(E,x2,SHAFT,( identity(var(x2),const(cmr_Shaft0)))),
                    description(desc(E,x3,IS_POSSIBLE,true),
                    conj(
                              [
                              atom(ARG1,[var(x3),var(x1)]),
                              atom(ARG2,[var(x3),var(x2)])])))),
                    nil)]
          nil)]
ok,
English,
time(56,53,23,11,6,1991))
```

Table 2: Three CMR examples resulting from a graphical selection of a shaft (1) , the text utterance "Is using thin cable possible in this shaft ?" (2) , and their fusion (3).

Information Types	Presentation Style
Location Information Physical Attributes	Graphics Only
Simple Actions Compound actions	Text and Graphics
Conditionals	Text for connectives text and graphics for actions
Abstract Actions	Text Only

Table 3: The relation between different information types and their presentation (after Feiner & McKeown, 1991)

Figure 1

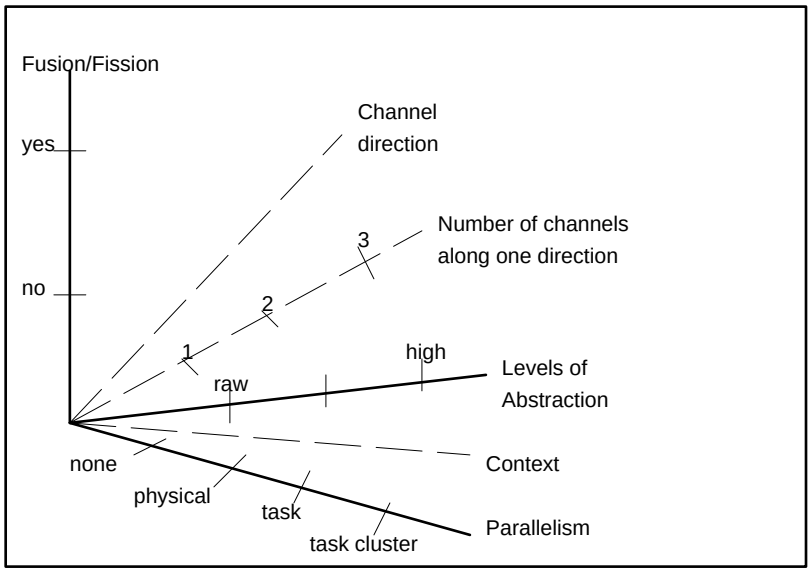

Fusion/Fission
yes
no
Channel
direction
Number of channels
along one direction
3
2
1
high
raw
Levels of
Abstraction
Context
none
physical
task
task cluster
Parallelism

Figure 2:

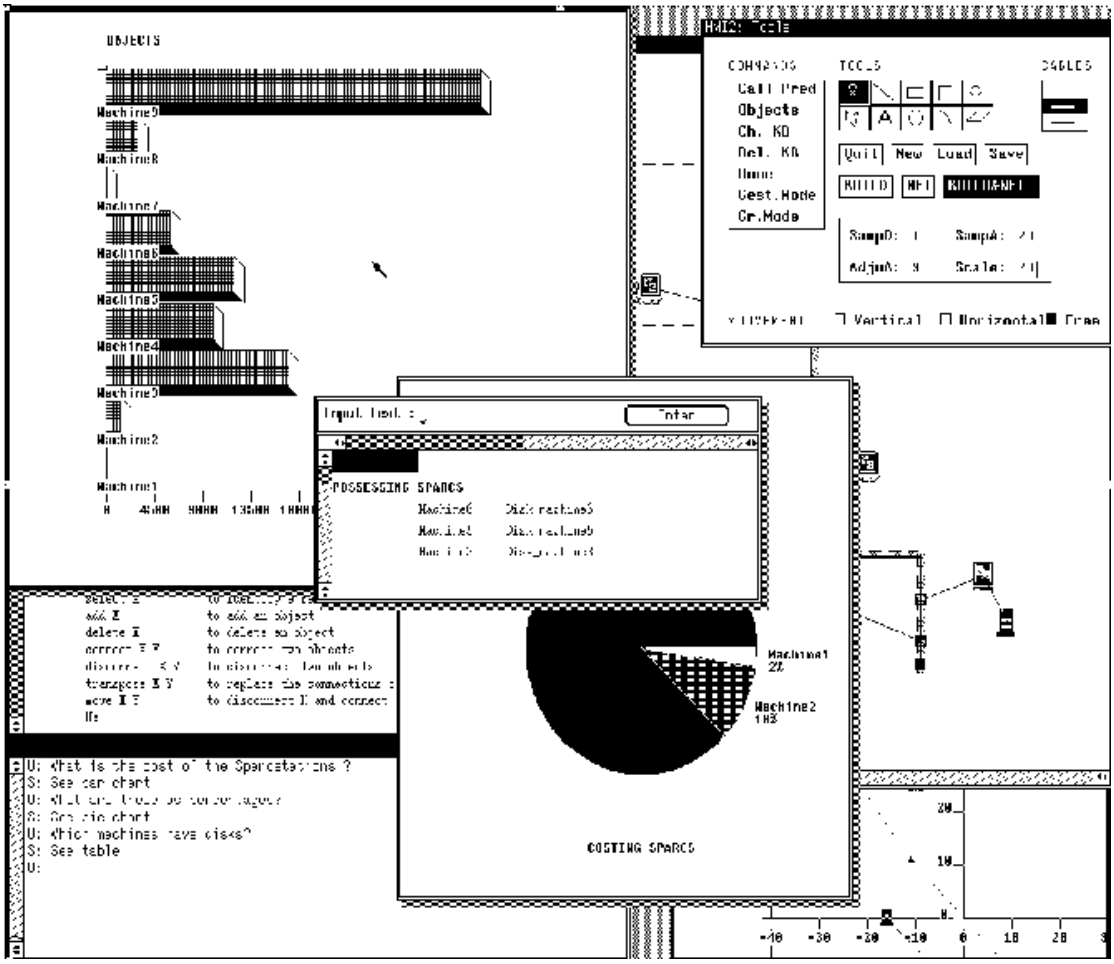

OBJECTS
Machine9
Machine8
Machine7
Machine6
Machine5
Machine4
Machine3
Machine2
Machine1
Call Pred
Objects
Ch. KB
Del. KB
Name
Dest.Mode
Cr.Mode
Quit
New
Load
Save
Vertical
Horizontal
Free
Input Task:
Enter
POSSESSING SPARCS
Machine6
Machine5
Machine4
Machine3
Machine1
2X
Machine2
1X
COSTING SPARCS
U: What is the cost of the Sparcstations ?
S: See bar chart
U: Give pie chart
U: Which machines have disks?
S: See table
U:
20
10
0
-40
-30
-20
-10
0
10
20

Figure 3

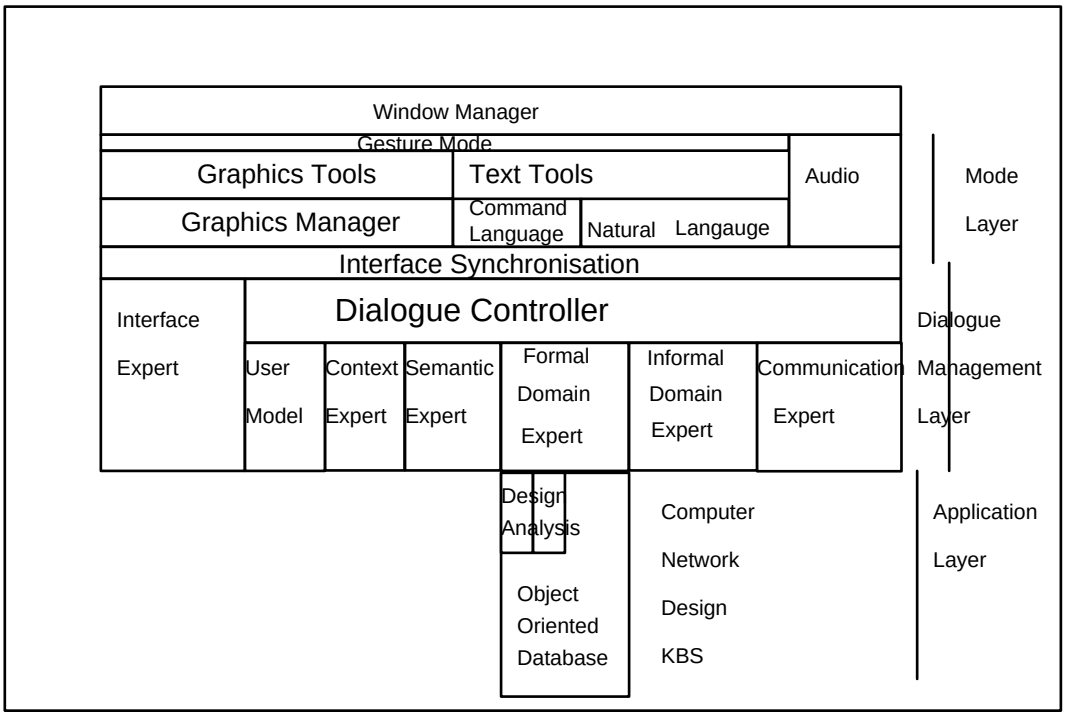

Window Manager
Gesture Mode
Graphics Tools
Text Tools
Audio
Graphics Manager
Command Language
Natural Langauge
Mode Layer
Interface Synchronisation
Interface Expert
Dialogue Controller
User Model
Context Expert
Semantic Expert
Formal Domain Expert
Informal Domain Expert
Communication Expert
Dialogue Management Layer
Design Analysis
Object Oriented Database
Computer Network Design KBS
Application Layer

Figure 4:

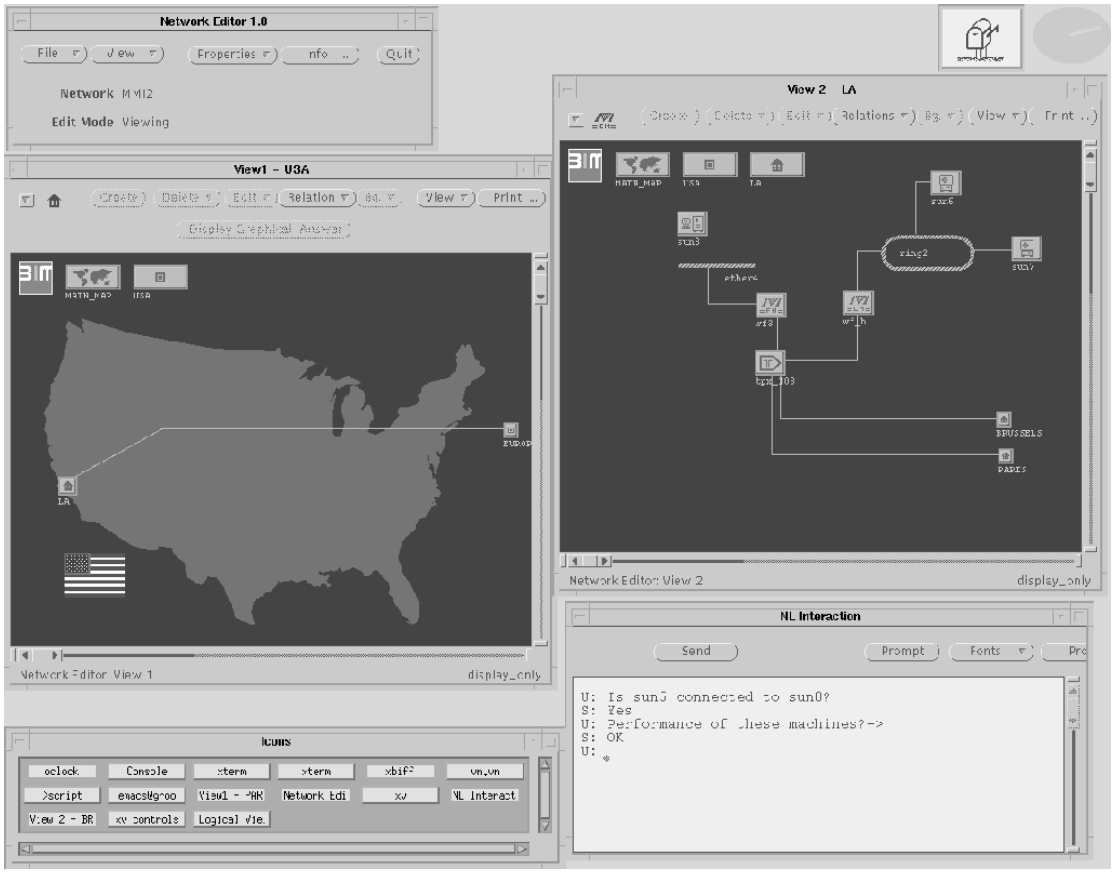

Network Editor 1.0
File
View
Properties
Info
Quit
Network MMI2
Edit Mode Viewing
View1 - USA
Create
Delete
Edit
Relation
View
Print ...
Display Graphical Answer
MATH_MAP
USA
EUROP
LA
Network Editor View 1
display_only
View 2 LA
sun5
sun3
ring2
sun7
ether4
fd2
bpx_103
BRUSSELS
PARIS
Network Editor View 2
display_only
NL Interaction
Send
Prompt
Fonts
U: Is sun3 connected to sun0?
S: Yes
U: Performance of these machines?->
S: OK
U:
Icons
oclock
Console
xterm
xterm
xbif
vm.vm
Xscript
emacs@groo
View1 - MM
Network Edi
xv
NL interact
View 2 - BR
xv controls
Logical Vie

Figure 5

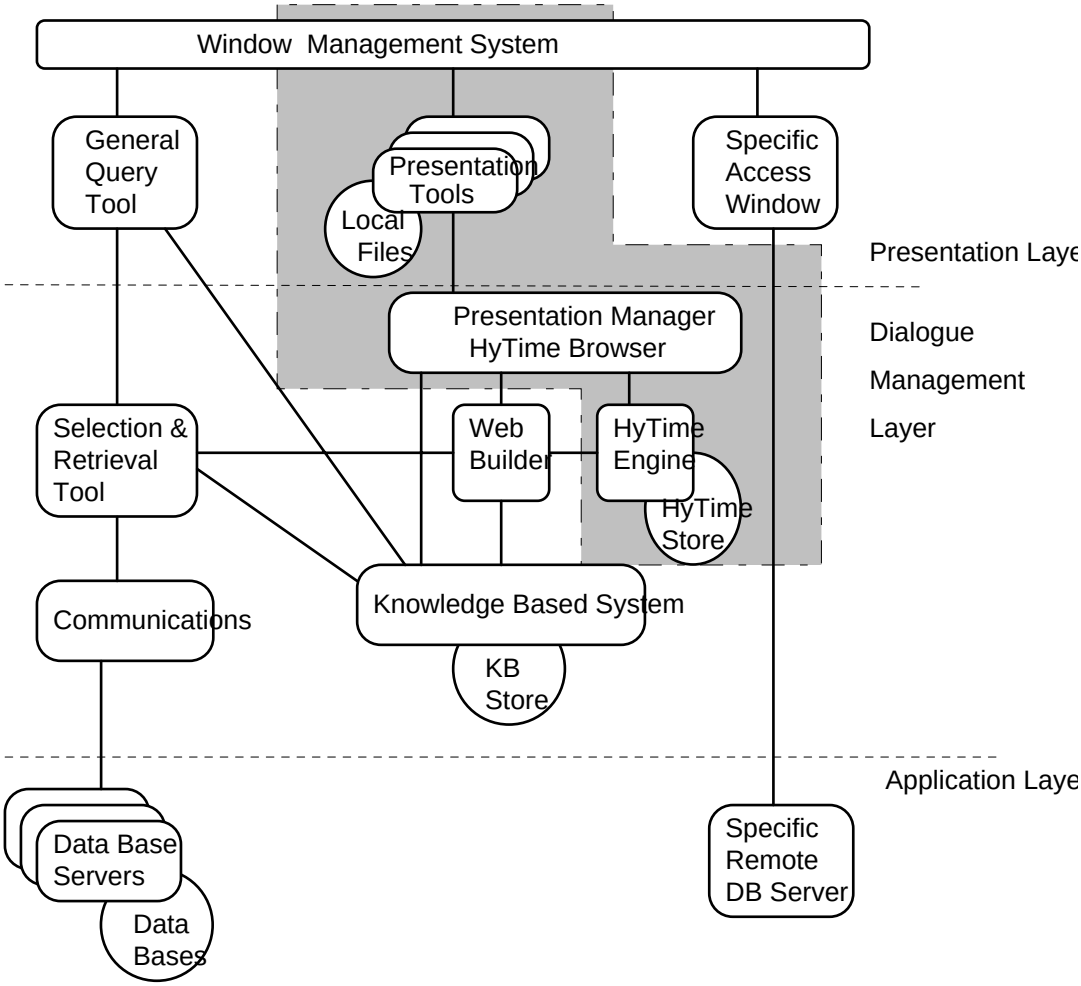
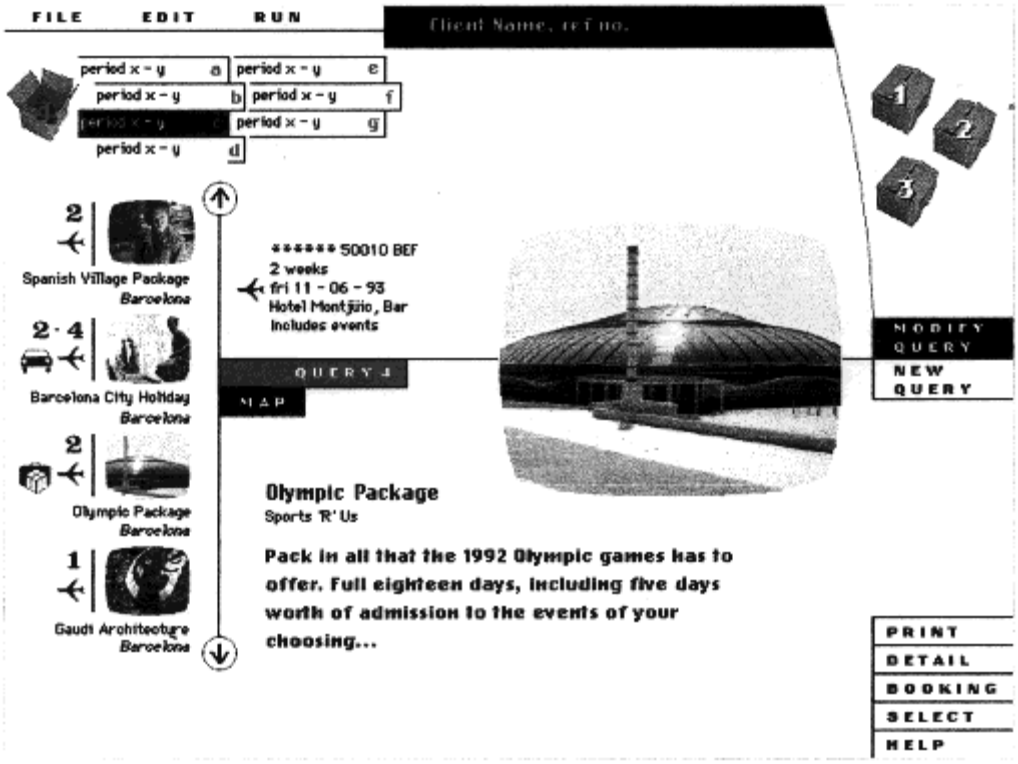

Window Management System
General Query Tool
Presentation Tools
Local Files
Specific Access Window
Presentation Laye
Presentation Manager
HyTime Browser
Dialogue
Management
Layer
Selection & Retrieval Tool
Web Builder
HyTime Engine
HyTime Store
Communications
Knowledge Based System
KB Store
Application Laye
Data Base Servers
Data Bases
Specific Remote DB Server

Figure 6:



Brief Biography of the Author

Michael Wilson is a Chartered Psychologist holding BSc and PhD degrees in Experimental Psychology. Since 1983 he has undertaken research into Human Computer Interaction at the MRC Applied Psychology Unit, Cambridge and Knowledge Engineering at the SERC Rutherford Appleton Laboratory where he is currently head of the Intelligent User Systems Section. At present he is the

RAL team leader in the Esprit projects MMI^2 and MIPS, and acts as a monitor of research for the SERC. He has published over fifty book chapters and journal articles on task analysis, modelling human-computer interaction, knowledge acquisition, and multimodal and multimedia user interface design.