The MMI^2 Demonstrator Systems

Deliverable d17 (TA 2)

Esprit Project 2474 MMI^2

A Multi-Modal Interface for Man Machine Interaction with Knowledge Based Systems

Editor:

M.D. Wilson (SERC/RAL)

Created: 12 December 1993

This document is part of a project partially funded by the Commission of the European Communities ESPRIT II programme, as project number 2474. The partners in the project are BIM (Belgium), Intelligent Software Solutions S.A. (Spain), ADR Centre de Recherche en Informatique applique aux Sciences Sociales (France), Ecole des Mines de Saint-Etienne (France), Insitut National de Recherche en Informatique et en Automatique (France), SERC Rutherford Appleton Laboratory (UK), University of Leeds (UK).

Contents

1 Introduction.. 3
2 Overview of the MMI^2 System... 9
3 A development method for Multi-Modal Systems.................... 15

4 Outlines of the Modules in the MMI2 System......................... 17

English Mode..BIM......... 18
French Mode...CRISS....... 20
French Generator..CRISS...... 21
Spanish Mode..ISS.......... 22
Graphical Tools...RAL........ 24
Graphical Network Tool....................................EMSE....... 26
NetTool..BIM......... 29
Graphics Manager...RAL......... 31
Gesture Mode..Leeds....... 32
Command Language..Leeds....... 34
Interface Expert..RAL........ 35
Context Expert..CRISS...... 36
Ellipsis Module...ISS....... 37
Dialogue Controller..BIM........ 38
MMI^2 Global Library...BIM........ 40
Semantic Expert..ISS.......... 41
Domain Expert: formal......................................RAL........ 43
Domain Expert: informal..................................RAL........ 44
User Model...RAL........ 45
Communication Planner...................................ISS.......... 47
NEST: a Network Design Expert SysTem..........EMSE...... 49
NetCortex: A Network Monitoring Applicaton....BIM........ 52

5 The MMI2 Demonstration Scripts... 53
5.1 Conventions used in the scripts 53
5.2 Script for the First MMI2 Demonstrator 54
5.3 First Spanish Demonstration Script 65
5.4 First French Demonstration Script 69
5.5 The script for the second MMI^2 demonstrator 73
5.6 Features illustrated in the first script 80
5.7 Features illustrated in the second script 89

7 References... 92
8 MMI^2 Publications... 93
9 Project Deliverables.. 96
10 Details of Consortium Members.. 100

1 Introduction

The MMI[2] project has developed two demonstrators of co-operative user interfaces to Knowledge Based Systems which employ multiple modes of interaction: natural language, command language, graphics, gesture, direct manipulation and non-verbal audio. The first demonstrator produced in October 1991 is a tool to design local area networks for institutions such as hospitals, universities or offices. The second demonstrator produced in January 1993 is a front end to a program which monitors local and wide area computer networks and allows users to obtain details about the state of the network, logical views of the network and various performance measures. Both demonstrators were implemented primarily in Prolog by BIM with parts in C. The purpose of both demonstrators was to demonstrate multi-modal interaction, to devise an architecture to support a co-operative multi-modal interface, to test the integration of the various technologies required, to define the meaning representation language required by such an architecture, and to investigate the requirements for a co-operative dialogue interface.

This document provides a summary of the systems developed, and the products of the research project. A detailed description of the methodology used, the detail specifications of the systems designed or the reasons for most of the design decisions taken in their development are not presented here. The project has produced 35 deliverables and many publicly available papers which provide these detailed descriptions that are listed at the back of this document with the contact details of the project partners from whom they can be obtained. This document provides an overview of the system architecture, the development method and each of the modules in the architecture. It then provides test scripts used for each of the demonstrators which illustrate features of the co-operative human computer dialogue that they support.

1.1 MultiModality

The MMI[2] project makes a strong commitment to “multi-modal” rather than to “multi-media” interaction in the interface. The term multimodal is sometimes used to describe communication from the user to the system in contrast to multimedia being information from the system to the user. Alternatively, multimodality is sometimes referred to as the use of different human modalities - auditory, tactile, visual, olfactory, taste. 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 media such as radio, magazines, books, television.

The contrast intended here 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 and adds 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).

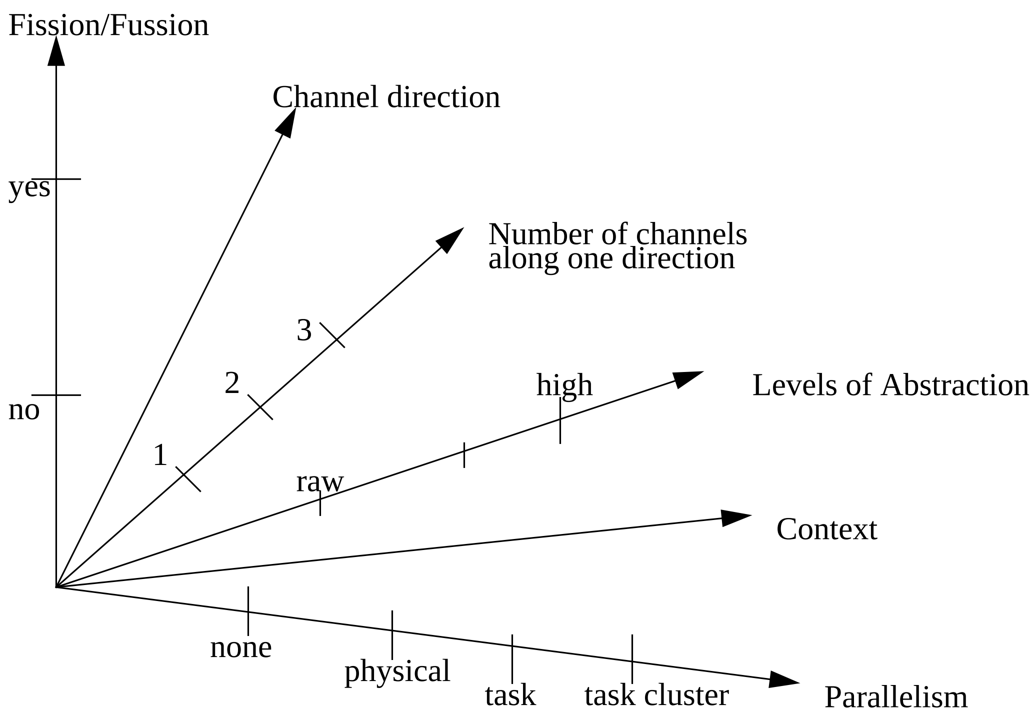


Figure 1: The six dimensions of the MSM framework for interactive systems.

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. This introduced the third conventional multimedia dimension of Parallelism, where both types of phone would be parallel at the physical level. If the phone system was extended to include voice mail or answerphone systems, the Parallelism would reduce to none since the interaction would only be asynchronous with one user at a time. 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 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 , handwritten or spoken (e.g. Cohen, 1992) whereas there is considerable research showing that spoken and keyboard interaction differ in many ways (e.g. Chapanis etal ,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 from 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. These could both be presented as text on the screen for the visual channel, or passed through a text to speech synthesizer 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 the 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? In MMI^2 we have taken the view that different graphical interactions which require different processing models are different modes although they use the same channel. Studies are currently being undertaken outside the project 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, adn 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 systems 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 third 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.

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 a meaning representation which is common to all modes. This 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.The meaning representation language used for all information within the system is called the Common Meaning Representation (CMR). This 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). Thirdly there is a higher level of abstraction used to plan communication in terms of communication forces. At this level, communications acts are labelled as providing such things as apologies, problem reports, justifications, or requests. These are similar to those proposed by Maybury (1991). 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 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 and deixis. There is a context provided by the 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 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, and 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. MMI^2 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.

1.2 Why Multiple Modes ?

The motivation for trying to use multiple 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: Complementary Interface Technologies: Direct manipulation and natural language

	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

The objective of using multiple modes is to allow the user to utilise the strengths of each mode while providing mechanisms for overcoming the weaknesses of each. 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. These issues were addressed in this project by using Wizard of Oz studies at an early stage in the development method to state the natural language requirements (see deliverable D15). These were not entirely successful and there are clear problems in identifying the syntactic complexity, robustness and coverage of the natural language system. These have been addressed in several deliverables, especially D7, D6(TA2) and D15(TA2). They will not be described here further.

Despite these problems, there are considerable advantages from natural language in its abilities to carry complex quantification on both commands and queries as illustrated in the dialogue below:

User: Which machines do not have disks?
System: cmr1 cmr4
User: Add a small disk to every machine that does not have a disk.
System: OK
User: Which machines do not have disks?
System: None
User: Does every machine have a disk?
System: Yes, cmr1, cmr2, cmr3, cmr4.
User: Does every machine have a small disk?
System: No.

and its abilities to express and resolve complex references as in the example below which justify its use.

User: Add a 375Mb Disk to the server.
System: OK.
User: Add *a small disk* to cmr3.
System: OK
User: What is the cost of *the disk*?
System: 1909 Sterling
User: What is the cost of *the server*?
System: 4114 Sterling
User: What is the type of *the disk*?
System: 375 MB Disk.

1.3 The MMI^2 Demonstrators

Within the MMI^2 project a first integrated prototype was produced in 1990 to test the interaction of the components in the architecture. This was designed to perform a simple test script where a user drew a computer network and then asked questions about its properties and performance. The network was represented in an object-oriented database and the system answered the questions using analysis functions derived from human expertise in computer network performance. The answers were presented to the user in a simple dialogue. A second prototype was developed in 1991 which allowed for a more complex dialogue script.

The first demonstrator was produced in October 1991 from these as a tool to design local area networks for institutions such as hospitals, universities or offices using NEST as the application program. This was presented through a demonstration script which mirrors how human experts in network design conduct consultations with clients. This script is presented later in this deliverable to illustrate the complexity of phenomena which the system can handle and the level of co-operation achieved. It is possible to use this system for designing computer networks without following the exact structure of the script since it is not pre-canned, but only an illustration of what is possible. However, the first demonstrator was not produced as an industrial pilot and was not sufficiently robust for general use in the workplace as a tool. The expert system for designing computer networks (NEST) has not been maintained as a tool and is now significantly behind current network technology with the result that it is no longer feasible to seriously use it for its intended task.

Although the first demonstrator successfully showed the viability of the architecture, CMR and component technologies to produce co-operative interaction in a single instance of a design system, it did not convincingly show how tuned the design was to the single application or how much effort would be required to port it to a new application. In order to address these issues a second demonstrator was produced in January 1993 as a front end to the commercial NetCortex program which monitors local and wide area computer networks and allows users to obtain details about the state of the network, logical views of the network and various performance measures. The production of the second demonstrator using the existing modules required only a few person months of effort showing that the interface was viably portable.

Since the use of test scripts to show the dialogue capabilities of the prototypes and first demonstrator had been successful, the approach was followed for the second demonstrator. A demonstration script for this is also included in this document showing the co-operative dialogue which it can support. For both demonstration scripts descriptions are provided of the major features illustrated. Further details of the technologies behind these features is available in the project deliverables and published articles listed at the end of this document.

The architecture of the second demonstrator was a subset of that of the first since it did not include all of the interaction modes, nor a detailed user model. However, the general architecture provides a view of its structure. Equally, the second demonstrator contained graphical presentation tools for maps, the logical structure of networks which were not used in the first demonstrator. The next sections of this report which describe the architecture and the modules within it apply generally to both demonstrators with these exceptions.

2 Overview of the MMI^2 System

The modes available in the MMI^2 system are: for input: English, French and Spanish natural languages, gesture, freehand drawing, direct manipulation of graphics, menu interaction, command language; and for output: English, French & Spanish natural languages, graphics (CAD diagrams and business graphics), non-verbal audio. Figures 3 to 6 provide screen images from the second demonstrator illustrating the graphical tools for displaying business graphics, maps, logical views of computer networks, along with a text interaction window in which natural language can be used. The second demonstrator did not employ the gesture mode, but in the first demonstrator used this in each of the windows, to apply to the objects or text presented.

The architecture of the MMI^2 system can be described as the three layers of Seehiem model for UIMS design (Pfaff, 1985; Duce et al., 1991). The top layer contains the input and presentation modes, the middle layer is the dialogue management layer, and the bottom layer is the application knowledge based system. The resulting architecture is illustrated in Figure 2.

.

The architecture is based around the notion of the “expert module”. Expert modules exist within each of the three layers. The name “expert” should be clearly understood. We are not proposing an architecture of “co-operating experts” or “multiple agents”. What we call an expert is simply a module performing specific tasks and with its own private data structures, and which allows a sufficiently coherent set of processes to be gathered in a single module. While such a notion is clearly not new, the identification of the nature of the basic modules constituting the multimodal interface, and of the interactions between them, has been a crucial step in

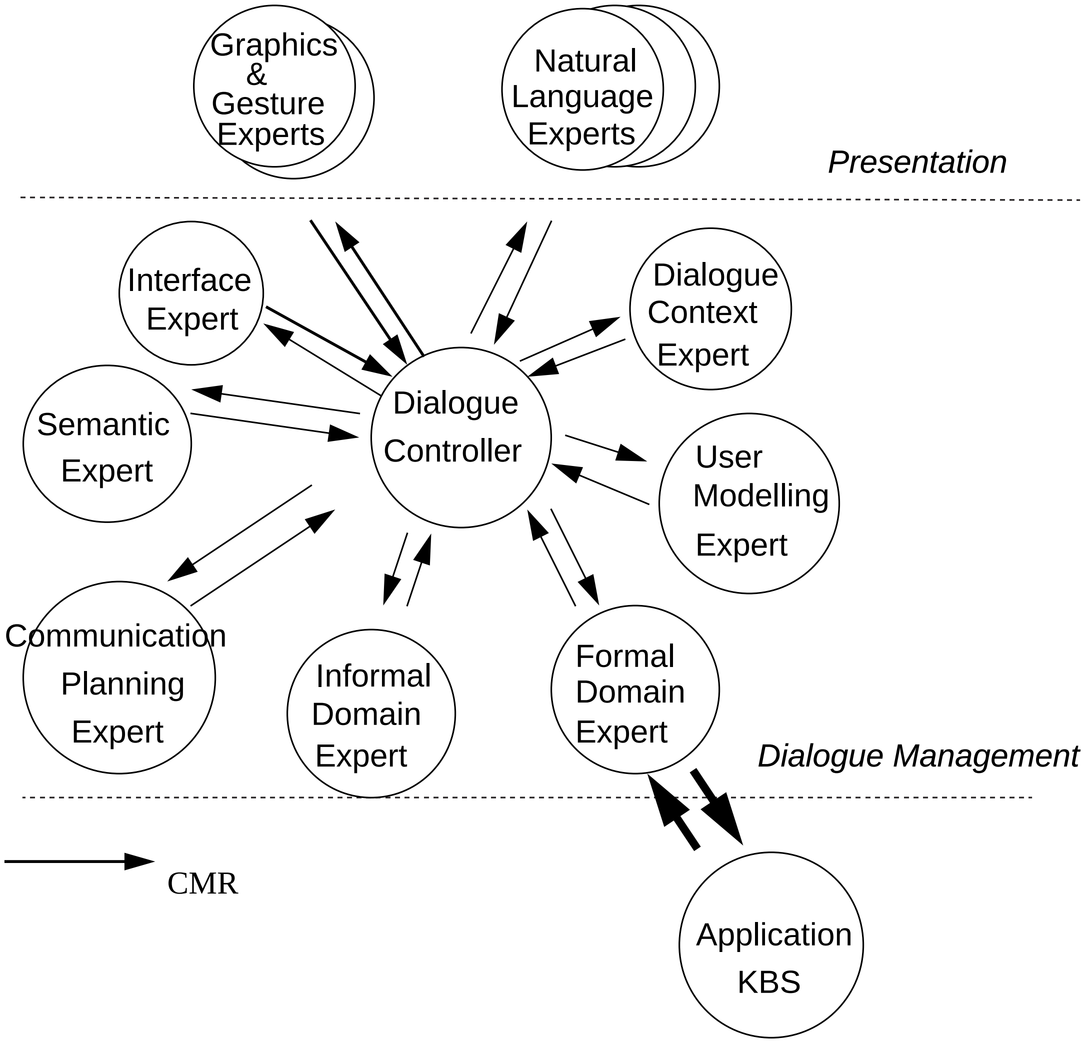


Figure 2: Architecture for the first MMI^2 demonstrator

the project.

A second driving force behind this architecture is that all operations within the dialogue management layer should be performed on a common meaning representation (CMR) which is independent of both the application and any specific mode. To this end, all input from the modes to the dialogue management is cast in the common meaning representation, as is any output to the modes. Since one aim of the project was to develop a toolkit based on the architecture portable between applications, communication between the dialogue management and the application must be in the language of the application. To this end, dialogue management contains a module which maps the common meaning representation onto the application language (Formal Domain Expert).

The next section outlines a development method for multimodal systems which supports the use of the proposed architectute and the three layers in it. This is followed by a section which describes each of the modules in the architecture in detail, describing their functionalities and specifications. The CMR and the functional interface for its evaluation against the worlds of interest in the formal fomain expert, user model, and interface expert are described under the heading of the Global Library. These provide a static view of the system, a more dynamic one can be gained from the scripts which follow and the descriptions of the features they illustrate.

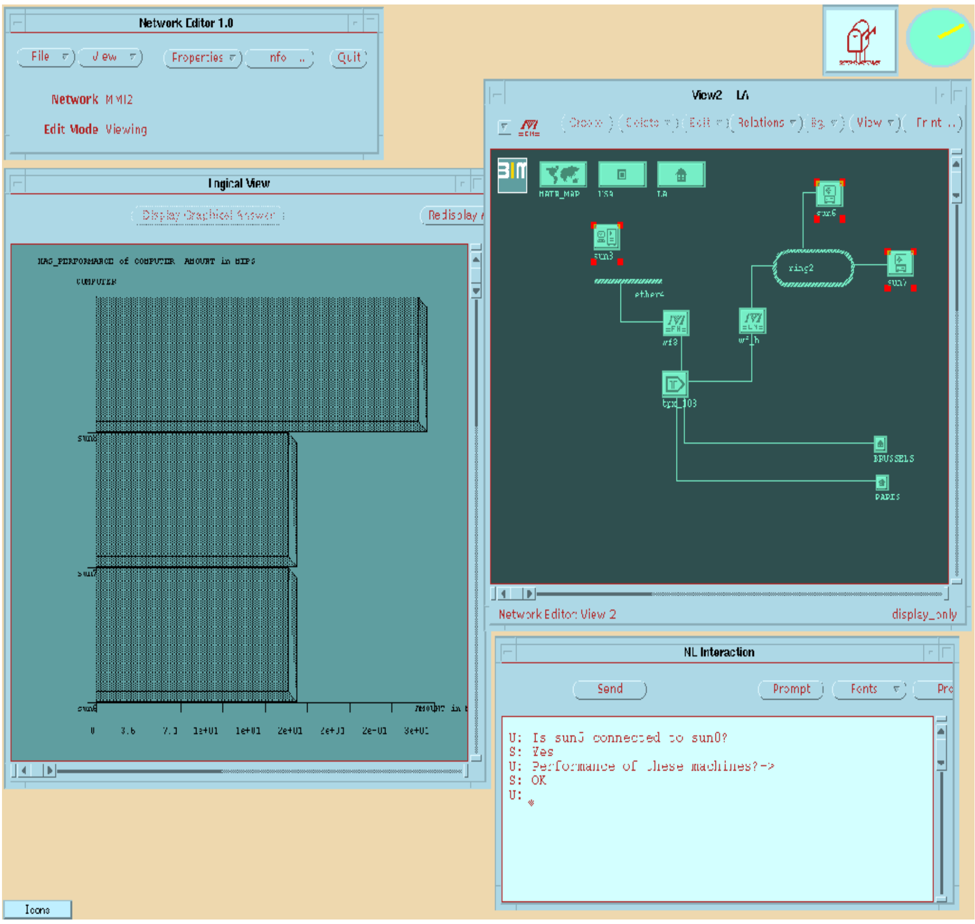


Figure 3: A screen image of the second demonstrator showing the natural language interaction window in the bottom right of the screen, a graphical view of the logical computer network structure and a bar chart of showing system performance.

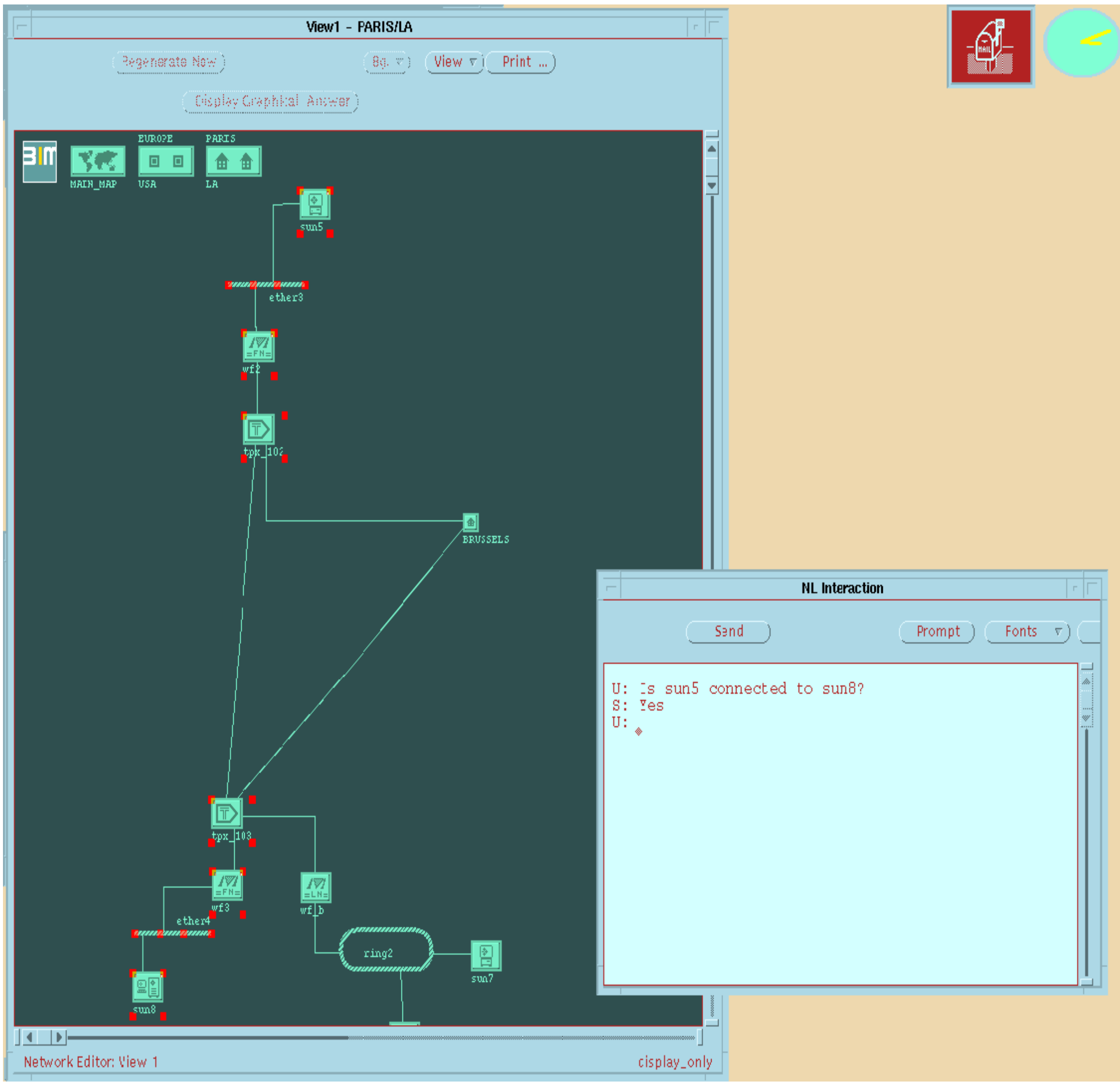


Figure 4: A screen image of the second demonstrator.

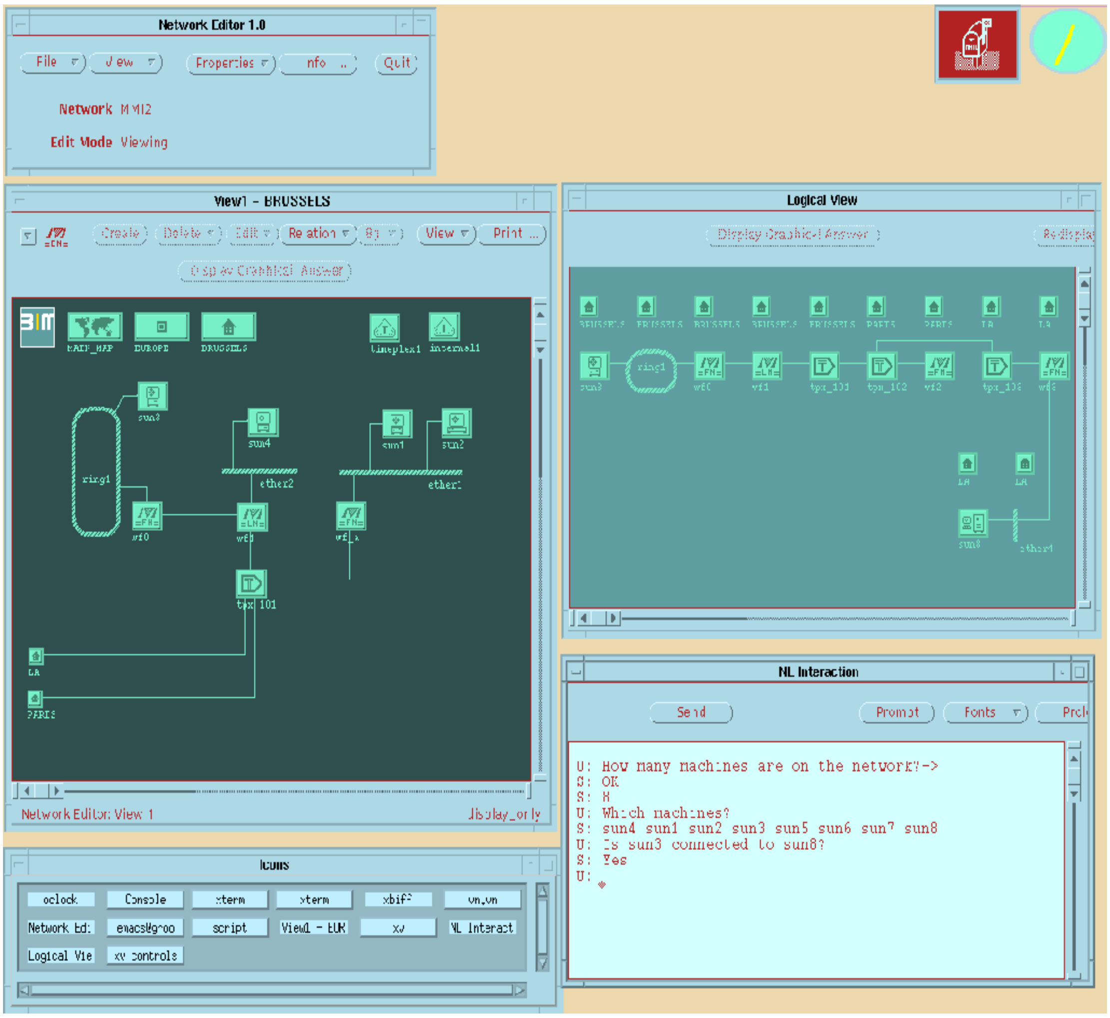


Figure 5: A screen image of the second demonstrator.

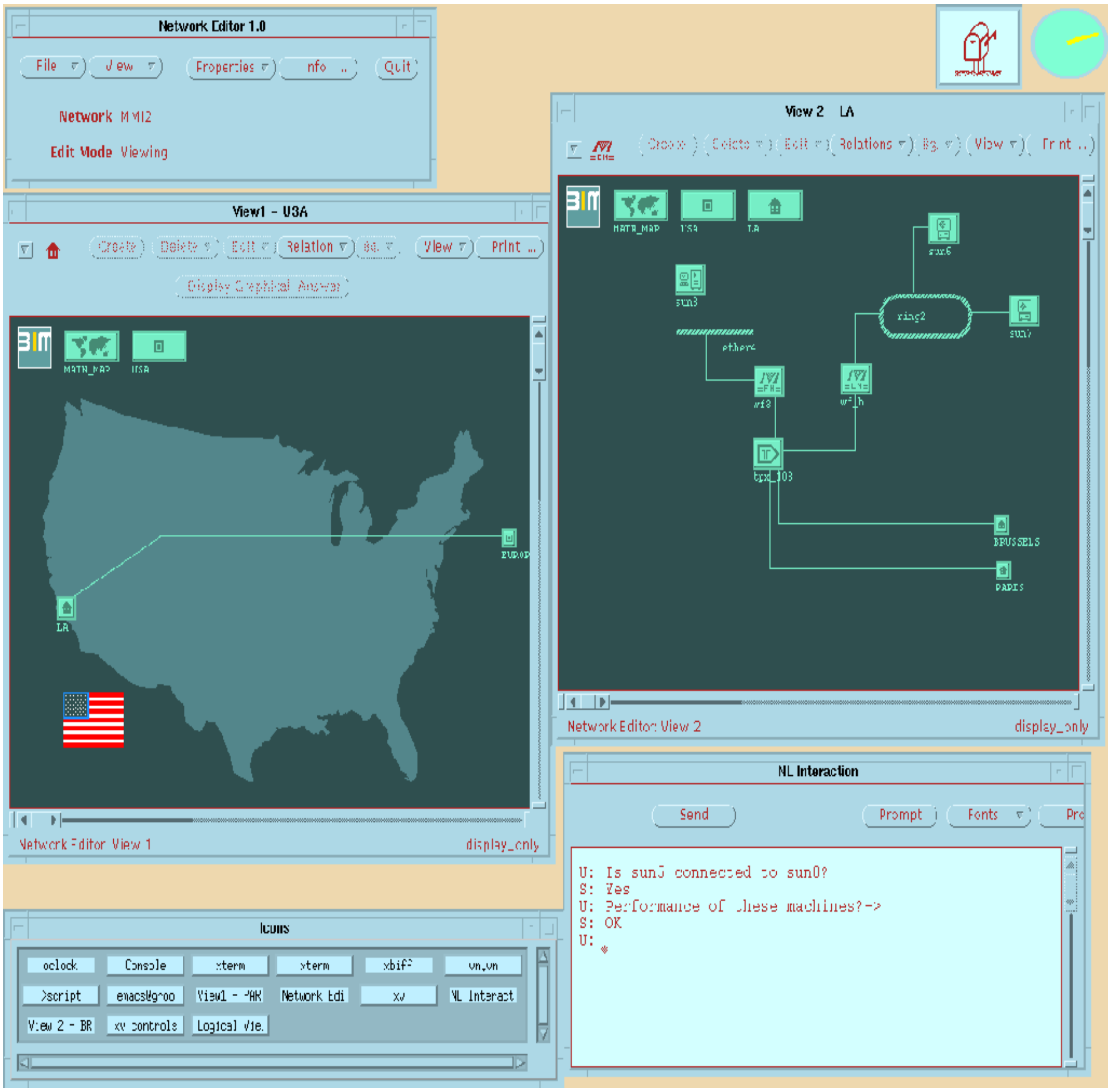


Figure 6: A screen image of the second demonstrator showing the natural langauge interaction window in the bottom right, a logical view of the local network at the mid right, and a logical view of th international network superimposed on a map to provide geographic cues at he mid left.

3 A development method for Multi-Modal Systems

There have been major advances in methodology for knowledge based systems in the last few years which have been drawn on in developing a methodology for multi-modal interfaces to them (see Wilson et al. 1989 for a review). The overall development methodology that has been followed in MMI^2 is a waterfall with prototyping loops. It follows the overall structure of the KADS waterfall model before it was changed to fit into the spiral risk assessment based approach (Tansley and Hayball, 1993; Shreiberet al., 1993). The main difference is that instead of a single KBS development stream there are three parallel streams, one for the KBS, one for the dialogue management layer and one for the mode layer. This reflects the structure of the MMI^2 system following the UIMS model which places emphasis on the different layers of the architecture rather than simply a KBS architecture with the User Interface as a minor component. Although this lifecycle does not explicitly include risk analysis, the decision to multi-stream it in this way would result from a risk analysis which placed equal risk on all three layers of the architecture.

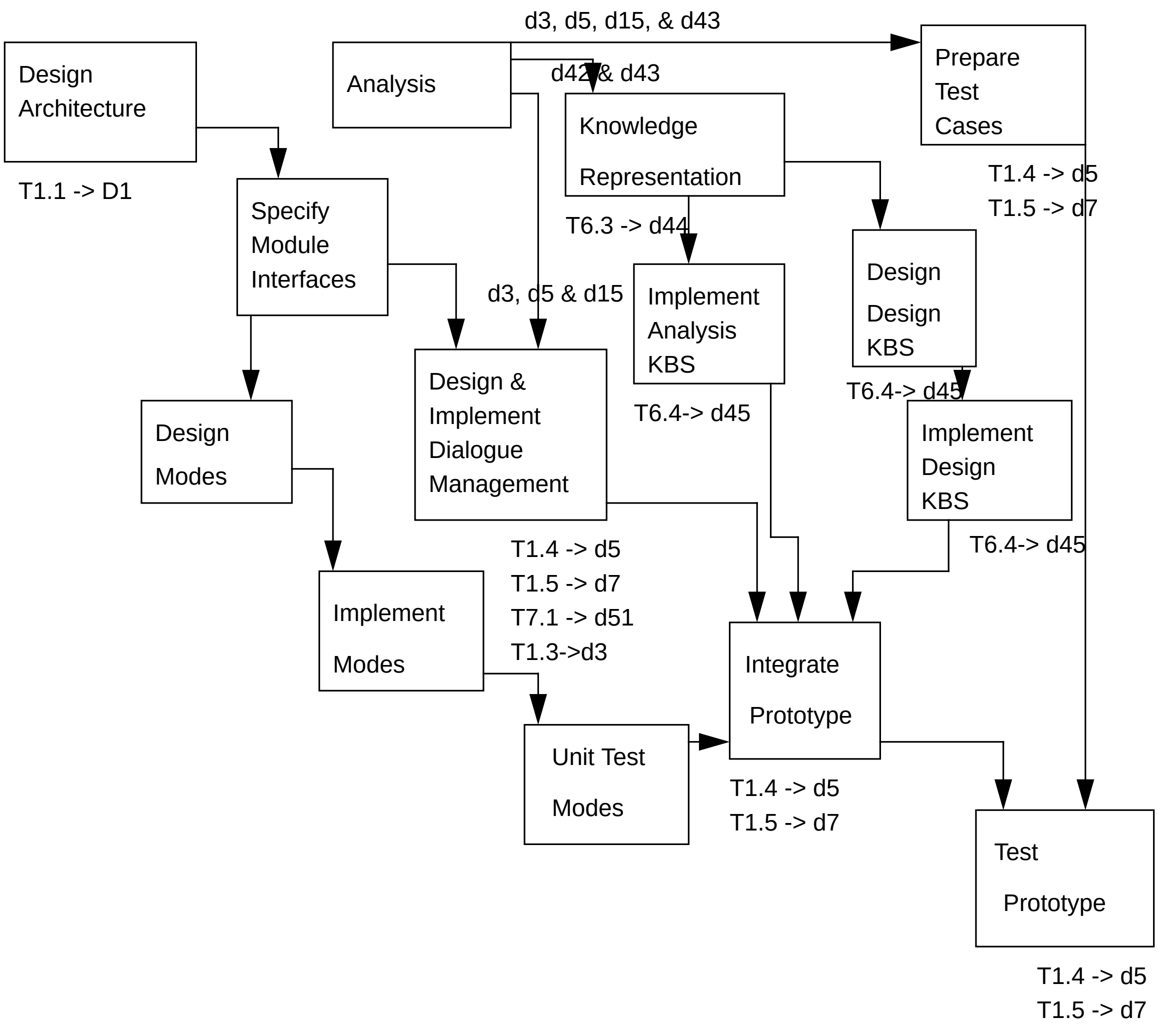


Figure 7: MMI^2 Development Method for Multimodal systems

The initial component of the analysis phase is to determine the scope of the project. This would usually be done prior to any contractual agreement to develop a complete system. Following this Three streams of analysis follow. The first of which determines objective constraints in terms of traditional functional and non-functional requirements. The second of which determines the interaction or co-operation required of the system. This starts with a Wizard of Oz analysis as was used in the MMI^2 development itself. There is now a considerable literature on the use of the Wizard of Oz method for determining natural language and other mode requirements at the interface (e.g. Dahlback et al., 1993). This leads to both the definition of information required at the interface and for the user model. The third stream follows a conventional knowledge acquisition path including the specification of explanation requirements in the task context.

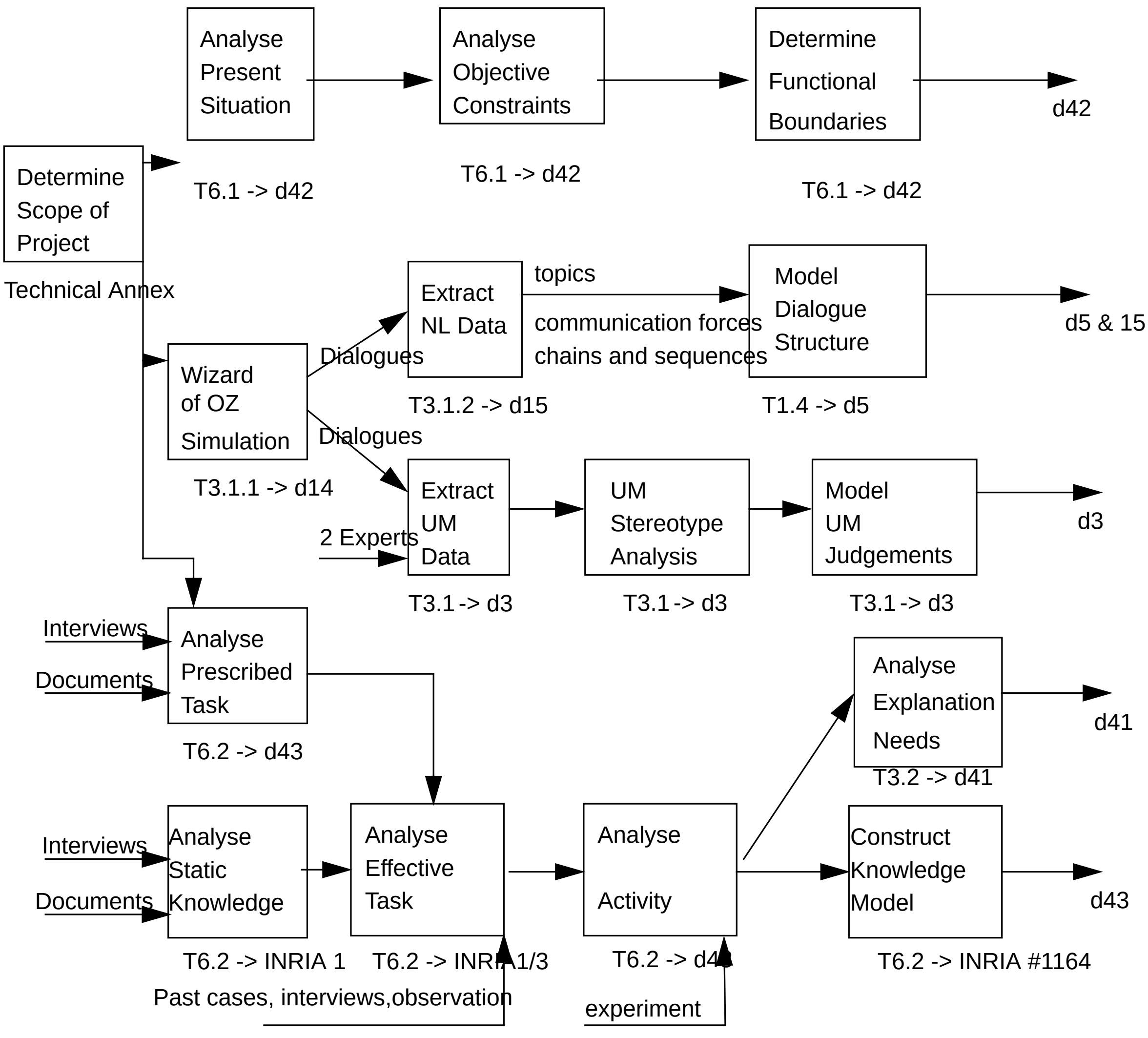


Figure 8: MMI^2 Analysis Method for multimodal systems.

The first stage of design, architecture precedes analysis because the project is a technology demonstrator with the purpose of integrating different user interface technologies, not only to satisfy the requirements of users which would arise from the analysis. Therefore the architecture is to some extent independent from the analysis of user needs, and it is the risks implied by the architecture which have motivated the structure of the development method. For developments following the MMI^2 approach the architecture would be assumed to be the MMI2 one.

The analysis phase assumes that the MMI^2 layered architecture is being used and the role of the developer is to meet the user needs for a particular application. These needs include not only communication, task performance by the system, and explanation.

Figures 7 and 8 include references to the tasks which performed these operations in the development of the first MMI^2 demonstrator (e.g. T6.1) and to the deliverables which resulted from them (e.g. d42). These deliverables are described at the back of this volume and specify the processes and outcomes required in considerably more detail.

An objective of such a method should be to provide guidance for designers as to which modes to incorporate for user input for various tasks in the way illustrated in Table 2 (after Rudnicky, 1993) which would complement the heuristics used within the communication planning and mode generation parts of MMI2 for selecting the mode for system output. However, these guidelines are not based on sufficient data to be strongly argued and all we can advocate is the use of well structured Wizard of Oz analyses during the analysis phase.

Table 2: Matching Input Mode to Activity (after Rudnicky, 1993)

Activity	Speech	Stylus	Keyboard	Mouse
Editing (mark-up)	Harmful	**Good**	Adequate	Adequate
Note-taking, dictation	**Good**	Adequate	Adequate	Harmful
User verification	**Good**	**Good**	Adequate	Harmful
Creating Graphics	Harmful	**Good**	Adequate	**Good**
Form-filling	**Good**	**Good**	**Good**	**Good**
Check-off lists	Adequate	**Good**	Adequate	**Good**
Command and Control	**Good**	Adequate	Adequate	Adequate
Spreadsheets	Adequate	**Good**	**Good**	Adequate
Scheduling	**Good**	**Good**	Adequate	Adequate

4 Outlines of the Modules in the MMI^2 System.

In the following sections each of the modules of the MMI2 system are described. These are intended to be read either as independent items describing the technology of each module or as a whole to provide an overview of the complete system.

English Mode

Author: BIM

Language: ProLog by BIM

Code Size: 14,500 lines

The English natural language mode allows the MMI^2 user to ask questions, issue commands and reply to system requests in English.

The English module was not developed within the MMI^2 project. It was lifted out of the Loqui system (Ostler, 1989), BIM's English natural language interface to relational databases, and ported to the MMI^2 application domain. This involved building a new lexicon and defining the interfaces for communication with other MMI^2 modules.

- Lexicon and Morphology

The English system operates with a preprocessing phase during which a set of stemforms are automatically expanded into a fullform lexicon, using the morphological rules of English. The output of the lexical analysis is a set of lexical entries (containing both syntactic and semantic information) which are dynamically asserted in the Prolog database. The dynamic lexicon serves as input to the parser that is activated subsequently.

- Size of the Lexicon: 468 fullform entries

closed class lexicon - prepositions, pronominals, auxiliary verbs, cardinals, adverbs: 314 fullform entries.

open class words - 107 nouns, 40 verbs, 7 adjectives: 154 fullform entries.

- Parsing

The input to the analysis component is a set of dynamic lexicon entries as computed by the morphological component; the output is a formal semantic representation. The parser operates in a top-down, left-to-right fashion producing a semantic representation from the word representations without an explicit intermediate parse tree, although a feature matrix carrying functional and semantic features is built as the parser proceeds. For each syntactic rule there is a corresponding semantic procedure, following GPSG's compositional treatment of the semantics of English.

The English parser is based on the theoretical principles of Generalised Phrase Structure Grammar (GPSG) but allows mechanisms from other theories as they seem useful, the aim being to develop an implementation of English grammar which is not only theoretically sound but also computationally efficient. Prolog itself is the formal expression language of the English grammar.

- Post-processing

After the parser has produced a semantic representation of the input, this representation is interpreted and transformed via the application of the rules of quantifier scoping. The semantic types are subsequently restricted according to the information of the domain model. Finally the internal structure is transformed to the CMR formalism.

- General coverage

- all major English categories
- nounphrase coordination
- quantifier scoping
- negation
- aggregate functions
- declaratives, interrogatives, imperatives
- relative clauses

● Integration in MMI^2

The English module is connected to several other modules of the MMI^2 system:

Interface expert. The EM communicates with the rest of the system via the interface expert. The IE passes on an atom representing NL input to the EM, which performs linguistic analysis and returns a CMR. The opposite goes for generation: the IE passes on a CMR representing NL output and is returned an atom. Note that there is no full generator in the EM; generation is done via canned text.

Semantic Expert. The parser makes use of 2 types of information from the SE.

- the class hierarchy

- the property restrictions defined on the role fillers of given classes from the hierarchy.

The parser uses these 2 types of information to do basically 3 things: checking compatibility between the parts of the parse tree, figuring out preferred attachments, and restricting general instances of classes to more specific ones, so that domain evaluation of the parser output can be done in a more efficient way.

Domain Expert. The EM calls the domain expert to resolve proper nouns. These are not defined in the lexicon.

French Mode

Author: CRISS

Language: ProLog by BIM, C

Code Size: 17,128 lines (lexicon not included)

The French system takes a sentence typed in standard French and gives an interpretation of it in the form of a CMR expression (CMR is the meaning representation formalism used in the MMI[2] interface). It comprises 3 main modules:

- a Morphological Analyser
- a Syntactic Analyser
- a Semantic Module.

- Morphological Analyser

The first one, the Morphological Analyser (MA), uses a dictionary of 50000 entries and alows to assign to each form (‘word’) of the sentence one or many possible categories with their lexical values. A statistical Markov based filter orders multiple interpretations due to ambiguity. The output of the MA is a list of sequences of categories ordered from the most probable sequence to the less probable.

- Syntactic Analyser

The second module, the Syntactic Analyser (SA), takes one sequence of categories resulting from the MA and tries to build its syntactic structure. In addition, it gives the functional structure of the sentence in order to prepare the next step, i.e. the semantic processing. To build the syntactic stucture of the sentence, the SA uses a context free grammar of the declarative sentence and an adaptation of Earley’s algorithm. The algorithm itself is supported by some linguistic knowledge based on the syntactic behaviour of verbs (a dictionary is used and contains the schemas of the verbs). Interrogative and imperative sentences are transformed into declarative ones before being dealt with. In fact, the syntactic behaviour of verbs gives the functional structure of the sentence.

- Semantic Module

The last module, the Semantic Module, takes a functional structure from the SA and transforms it into a tree structure according to an operator-operand formalism. It then performs different linguistic and logical transformations on this tree (quantification, scoping, verb ‘to be’,...). Finally, a logical expression (CMR) is read from the tree which is the output of the French Parser.

Themes pocessing, based on the approach of the Prague School of Linguistics,is currently being integrated. The idea is to add thematic order information to the CMR generated by the french system. This information can then be used by the context expert for detailed analysis (referencing, topic-focus tracking, etc.).

French Generator

Author: CRISS

Language: ProLog by BIM

Code Size: 15,500 lines (lexicon not included)

0. Overview

The french generator, which is part of the french natural language mode, allows to reply to the MMI^2 user in french natural language. Unlike the french parser, the generator has been developed entirely within the MMI^2 project but, like the parser, it was designed to be as independent as possible from the application and the system. In order to achieve this goal, a general linguistics based component structure was defined that can be applied in different discourse contexts. It is linked to the MMI^2 system through a specialized interface module that translates CMR into the generator's own internal representation. Other links exist through the lexicon (which is domain dependent) and the User Model (see below)

1. Basic features

- dialogue generation (few and short sentences)
- separated approach (first planning then realization)
- only syntactic constraints
- clause-based generation

2. Architecture

The generator essentially executes two main tasks: planning and realization. Each of them can be broken down into subtasks.

2.1 Planning

- *Utterance pre-processing*: this task concerns basically the processing of interrogative sentences.
- *Determination of the functional structure of the clauses*: this is based on verb behaviour (case frames) and allows to give information about the types of complements and how they can be used in the sentence.
- *Linguistic operation processing*: This task addresses issues about the relationships between clauses, like grouping or referential phenomena. It involves three steps. First, the initial clauses are looked through in order to detect all LO's possible. These LO's are then combined to form sets of compatible LO's. Finally, one set of LO's is selected (and applied) taking into consideration stilistic criteria as well as the user's knowledge about the application domain and the language.

2.2 Realization

- *Morphology*: This task is responsible for generating surface forms of words. It uses a large french dictionary of about 55000 entries.
- *Syntactic linearization*: This task determines the syntactical structure of the text at different levels (sentence, clause, and phrase). It consists of reading off the tree constructed during the previous phases to obtain the surface order of words in a phrase and of phrases in a sentence.
- *Surface regularization*: This tasks deals with orthographical smoothing of the entire utterance.

3. Linguistic coverage

The following linguistic constructions and phenomena are handled (in more or less detail) by the generator:

- declaratives
- imperatives
- interrogatives
- noun phrase coordination
- verb coordination
- pronominalization
- subordination
- lexicalization
- adjectives
- verb negation

4. Integration in MMI^2

- *Interface Expert*: The IE passes on a CMR expression (representing the message to be output) to the generator and is returned an atom (representing the NL output).

- *User Model*: The generator calls the UM to get the user's level of knowledge about the application domain and the language in order to generate an adapted text to the current user. For the moment, the UM influences the structure of the utterance rather than its (lexical) contents.

Spanish Mode

Author: ISS

Language: Prolog by BIM, C

Code Size: 15,500 lines

The goal of the SM is to analyse the Spanish text expressions in order to translate them to a CMR (the internal meaning representation formalism of the system). The SM also takes charge of the Spanish generation.

The modular architecture of the SM has the following components.

1. MORFEO

1.1 Morphografic tool

It is written in C and performs the pre-treatment of the input to obtain a string that could be processed by the morphological analyzer. Its mainly functionalities are:

- Amalgamations
- Sign assimilation
- Treatment of capital letters,
- Intelligent processing of the signs of punctuation.

1.2 Morphological analyzer: MORFO-SP

It is written in C and performs the morphological analysis of the string to be parsed. From its input string it offers the list of the standard words corresponding to the input tokens with their morphological features. The morphological analysis is achieved by matching each word of the input with an entry of the dictionary. When it is an inflected word (all the verbal, adjectives and nominal forms) it must be recognized as the combination of a base listed in the dictionary and a suffix of one of the models declared as valid for this base. The morphological processing is able to deal with idioms, compound tenses and compound words, a word to be recognized in the dictionary may be composed by many inflected words with blank spaces between them. When the input string contains amphibologies the output of this module is a list where all the different possible analysis are offered for each position, including, if it is the case, the alternative chaining for the words present in the text.

1.3 The Morphological dictionary

1.3.1 The dictionary of bases: DICO-SPAL

The Spanish dictionary of bases contains one entry for each base of the Spanish words. These bases may contain blank spaces treated as an ordinary character. It is organized as a tree structure of characters with as many maximum levels as number of characters in the longest base to be looked for. Each one of the entries contains ten parameters of morpho-syntactic information. When specifying the base corresponding to an idiom it is possible to mark which one of the words carries the morphological information, to identify some of them by its category or with a wild card and to point out that they may be optional.

1.3.2 The dictionary of suffixes: DICO-SPAL-s

The dictionary of suffixes contains for each model all its corresponding suffixes and for each one of them the morphological features that it carries.

1.4 The markovian filter: MANA

The string leaving from the morphological analysis must be disambiguated before be treated by the syntactic parser. The markovian filtering processing will transform one list with ambiguities to a list of alternative non ambiguous lists, ordered by a ranking of probabilities. Using the filter involves the previous creation of a table of triplets of morphological categories with its corresponding probability degree or the use of the standard tables already available in ISS. When in presence of ambiguous words the filtering algorithm looks into the statistical frequency matrixes searching for the triplets of which the ambiguous category and the two contiguous ones are part. A probability ranking is calculated on the result of the sum of the probabilities of all the triplets containing the different filtering categories of the ambiguous word. When in presence of unknown works the tool is able to pass a message but also to process intelligently the input up to the point to fournish also the correct category assignment.

More than 98% of accuracy in the assignment of categories has been achieved in the evaluation tests of the desambiguation processing performed by MORFEO: MORFO, DICO and MANA, when treating narrative text by a Standard Dictionary of Spanish (SPAL), used as a stand-alone tool outside of the MMI2 interface. MORFEO also has a high performance in its desambiguation processing, being able to treat up to 5000 text characters x minute.

2. THE LEXICON

The list of alternative strings of morphological pieces offered by the markovian filter MANA contains for each token its morphological category, standard word and agreement features. Before be parsed, the string must be augmented with syntactic and semantic features, this information is contained in the lexicon file and it is loaded for each piece from the standard word and morphological category acting as a keys.

3. THE PARSER

The input of the parser is the first list of those ranked as the most provable ordered alternatives by the markovian filtering process. It constructs from one list a functional structure composed by a variable identifier and a feature list corresponding to the analysis of the user intervention. If there is no parsing for this list the second one will be analyzed. The parsing process results from the management of the grammatical rules by two main types of parser strategies.

1) There are two types of grammatical expressions: patterns and rewriting rules with only terminal categories that are used in a sequential and deterministic processing of the input list. In a first step it is covered from the beginning to the end and some elementary groups are reduced to a single element.

2) The greater part of the grammatical rules are integrated in a left-to-right and bottom up procedure with backtracking, whose predictive power is increased by a look-at facility that sanctions only partial parsers that would be created by a top-down procedure.

The grammatical source rules look like a DCG grammar format with only two categories on the right hand side. There is a set of procedures for feature treatment that manages the daughter's feature list to obtain the resulting mother's list. The rules are interpreted from a functional point of view, and the sequential strict semantics of the rewriting format is augmented by instructions for the attachment dependent->top or for the unification of features, directly coded in the proper right hand side part of the rule. There is a compilation procedure of the source grammar that expands the DCG format to standard Prolog rules to be used by the parsing calls.

The following phenomena are treated by the SM:

- Agreement
- Prepositional phrases attachment
- Movements.
- Control.
- Passive.
- Coordination.
- Intrasentential and intersentential ellipsis.
- Assertions, questions and imperatives.
- Pronominal noun phrases.

4. THE PARSER-TO-CMR TOOL

The work of translate the functional structure arising from the parser to the corresponding CMR representation is achieved by the LOGIC program. The conversion is carried out by a recursive top-down left-to-right algorithm that performs these main goals:

- calculation of the u_type
- calculation of the logical quantifier
- translation of features into CMR descriptors and annotations
- ordering of terms, according to the scope of their quantifiers

When a single user intervention with coordination is treated as a coordination of sentences the CMR representation consist of a CMR containing as many CMR_act_analysis as functional sentences obtained.

5. GENIUS

The Spanish generation is done using the canned text technique combined with the passing of arguments to match with variables inserted in the text. This module takes lists of arguments in a predefined order from the communication planner module -CP- and inserts them in the correct places in the output text. A special dictionary translating the language independent terms of the system into Spanish is then consulted for each incoming term and the resulting surface word is inserted in the right place in the text.

The performance and accuracy of the Spanish Mode is very high since the tests demonstrates it takes less than 3 seconds to process a script containing 25 Spanish text interventions embracing 180 words representing the text dialogue of a user in a complete session for designing a LAN for a building.

Graphical Tools

Author: RAL

Language: ProLog by BIM, C

Code Size: 26,646 lines

There are three main classes of graphical tools responsible for the presentation of information and the capture of user input: analysis tools, text tools and the network tool.

Each of the graphical analysis tools produces windows that display a chart through which the user can interact with the system via direct manipulation.

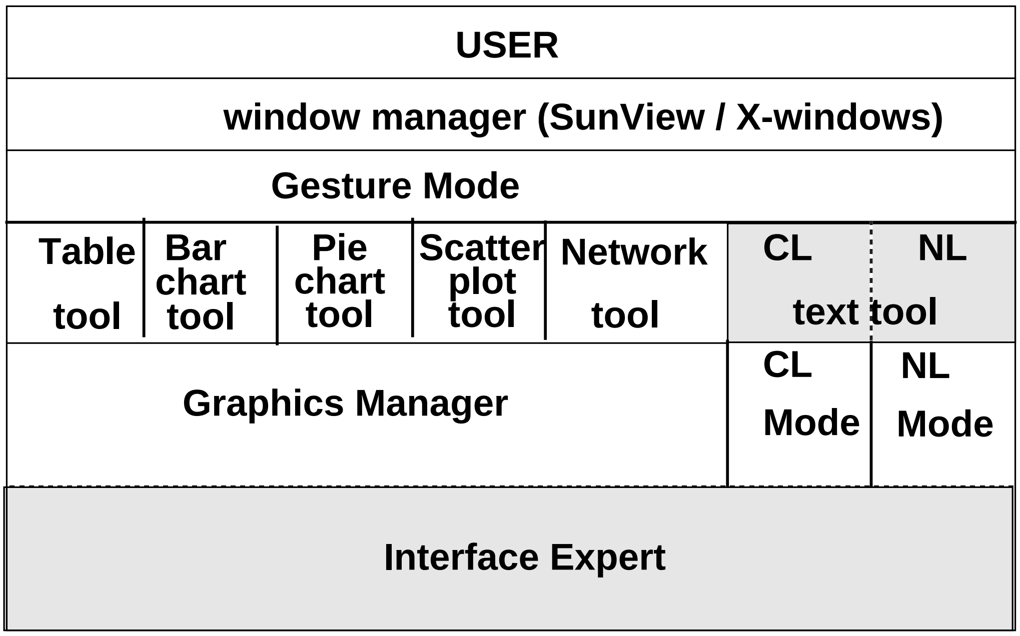
The Graphical analysis tools are general domain-independent modules that display charts of particular types: Bar charts, pie charts, scatterplots/line charts and tables. For each tool, both the system and the user are able to perform the following actions:

- Graphical actions, e.g. “Edit” and “Select”: The user can make a deictic reference to an object represented in the chart, or change the value of an object represented in the chart via direct manipulation on the chart display. The MMI$_2$ system can do similar actions through the Graphics Manager. These actions constitute “moves” in the dialogue between system and user.

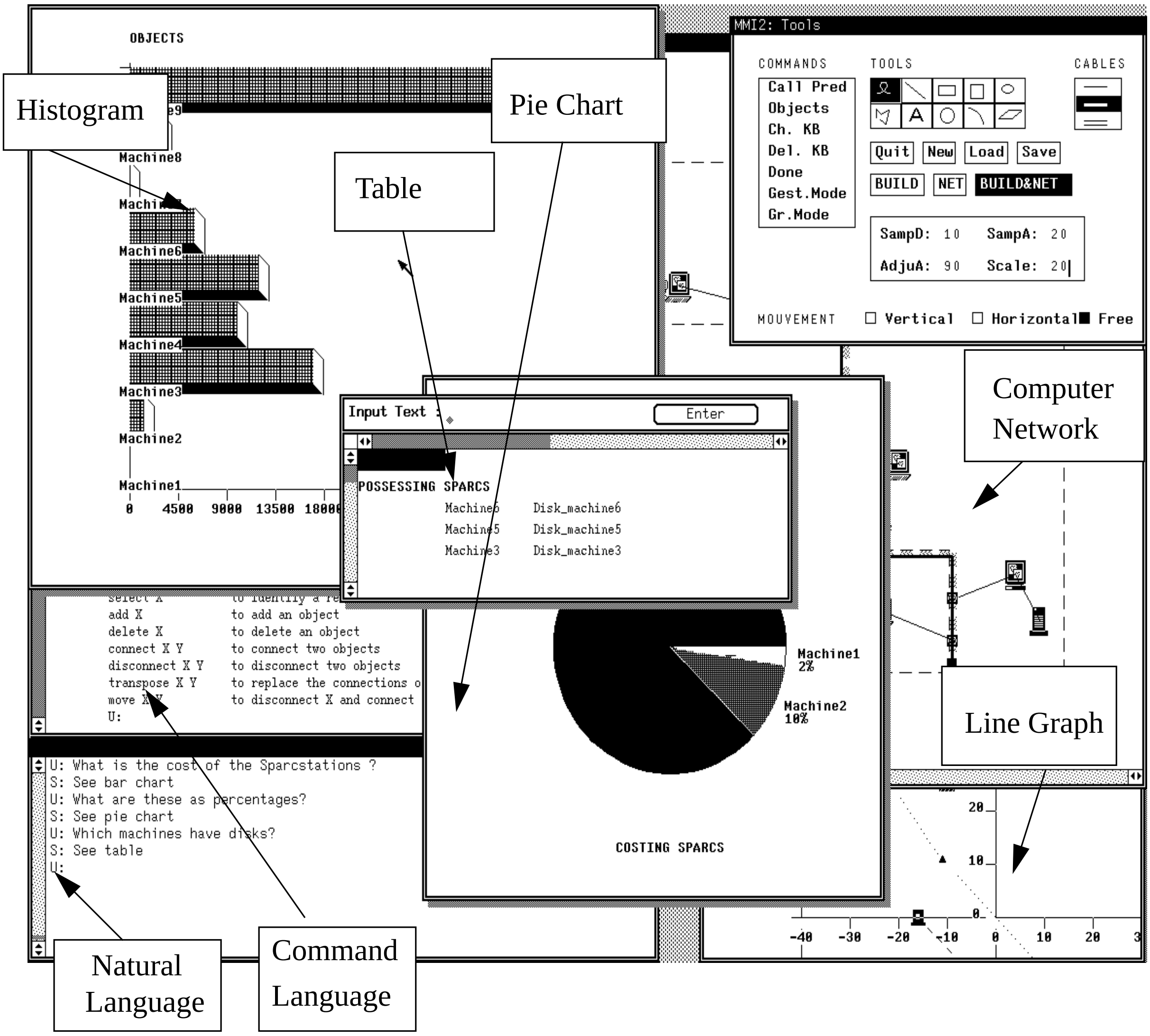
- Change the appearance of the window, (e.g. change the type of filling of the bars on the bar chart, change the size of the pie chart). When the user changes the appearance of a window it is be recorded as a preference in the User Modelling Module and the system will adapt future charts according to this preference. Note, however, that this is perceived as the user tailoring the display rather than as part of the dialogue between system and user.

- Quit or redisplay the window. Again, these are not part of the dialogue between system and user.

The text tools are described in the Interface Expert.



Architecture of the MMI^2 Modes



Graphical tools showing business graphics and text interaction.in the first demonstrator.

Graphical Network Tool

Author: RAL, EMSE

Language: ProLog by BIM, C

Code Size: 26,646 lines

An important tool for graphical interaction in MMI[2] is the Network Tool. Its main characteristics are:

- To allow users to draw buildings and networks either by free hand drawing or by means of graphical tools like lines, boxes, &c.. Buildings and networks are supported by two distinct planar map data structures. However, links between buildings and networks are provided: for example, a machine is inside a room, a cable skirts around walls, &c... For this purpose, some functions which attach attributes to the data structure are provided.

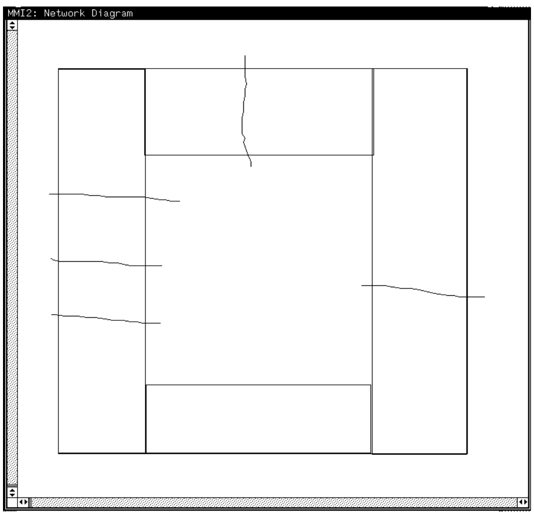
The user can create many floors and put vertical shafts and cables through these floors.

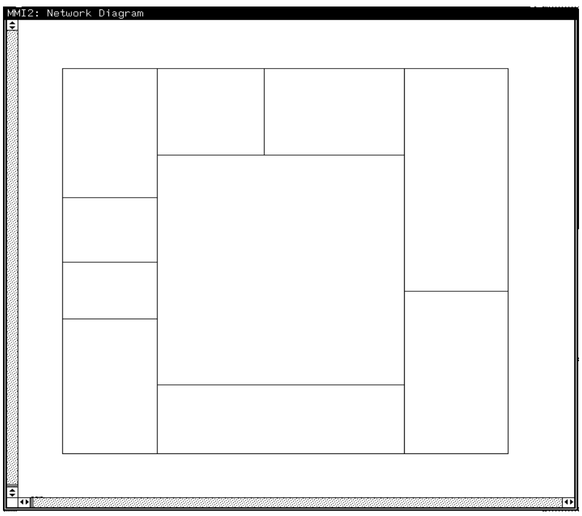
- To preprocess the drawing: the purpose of this process is the correction of the malformations of the unrefined drawing. In fact the preprocessing task has three main functions which are SAMPLING, ADJUSTING and ERASURE ELIMINATION.

- To deduce the semantics of the drawing from the data structures and convey this information

to the graphics dialogue manager. The graphics dialogue manager transforms this information into CMR expressions and sends those to the Dialogue Controller.

- To handle events from user to the gesture module. When a user selects the gesture mode, the Network Tool catches the events, sorts them and sends them to the gesture module to perfom the user action.

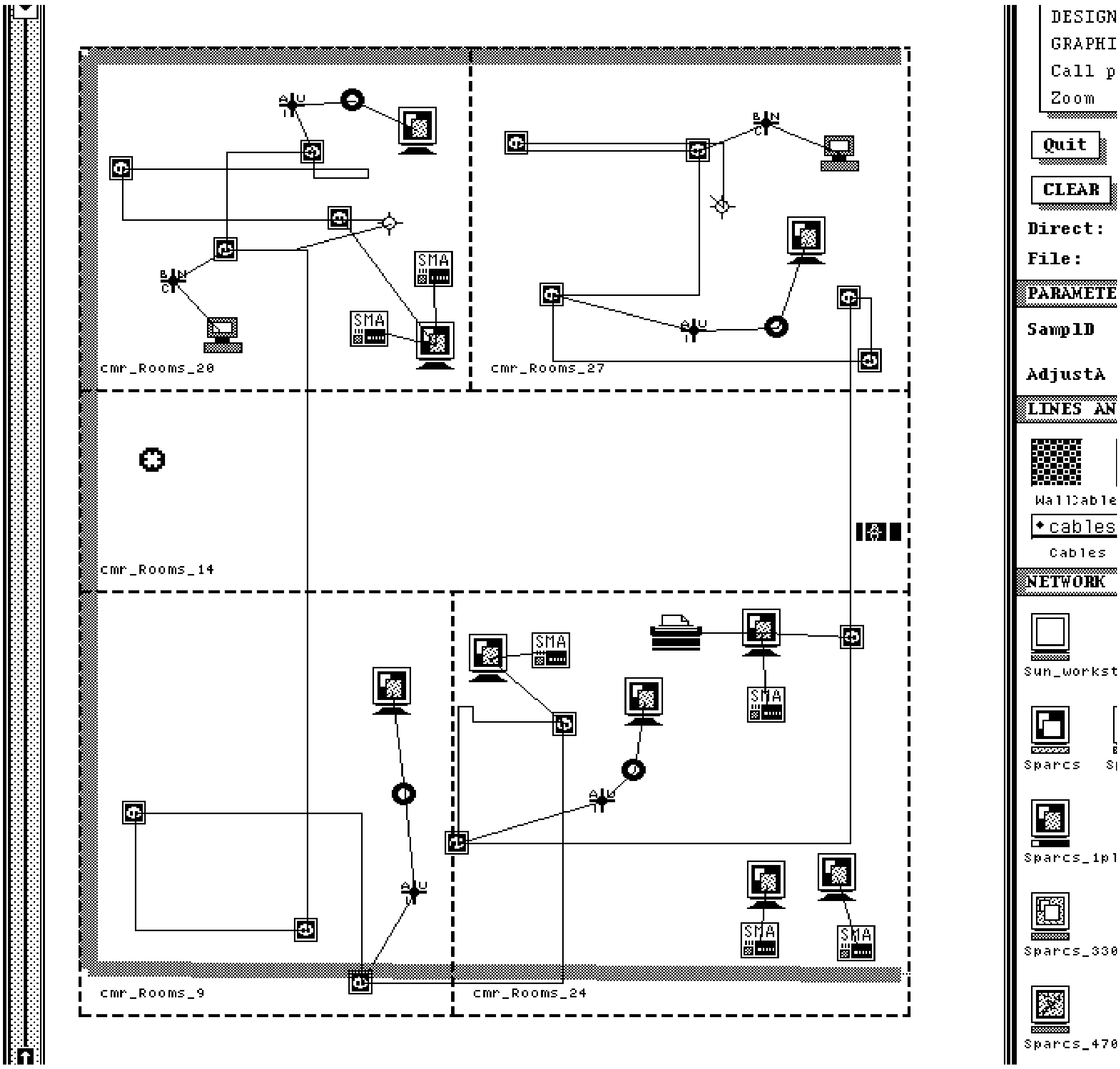




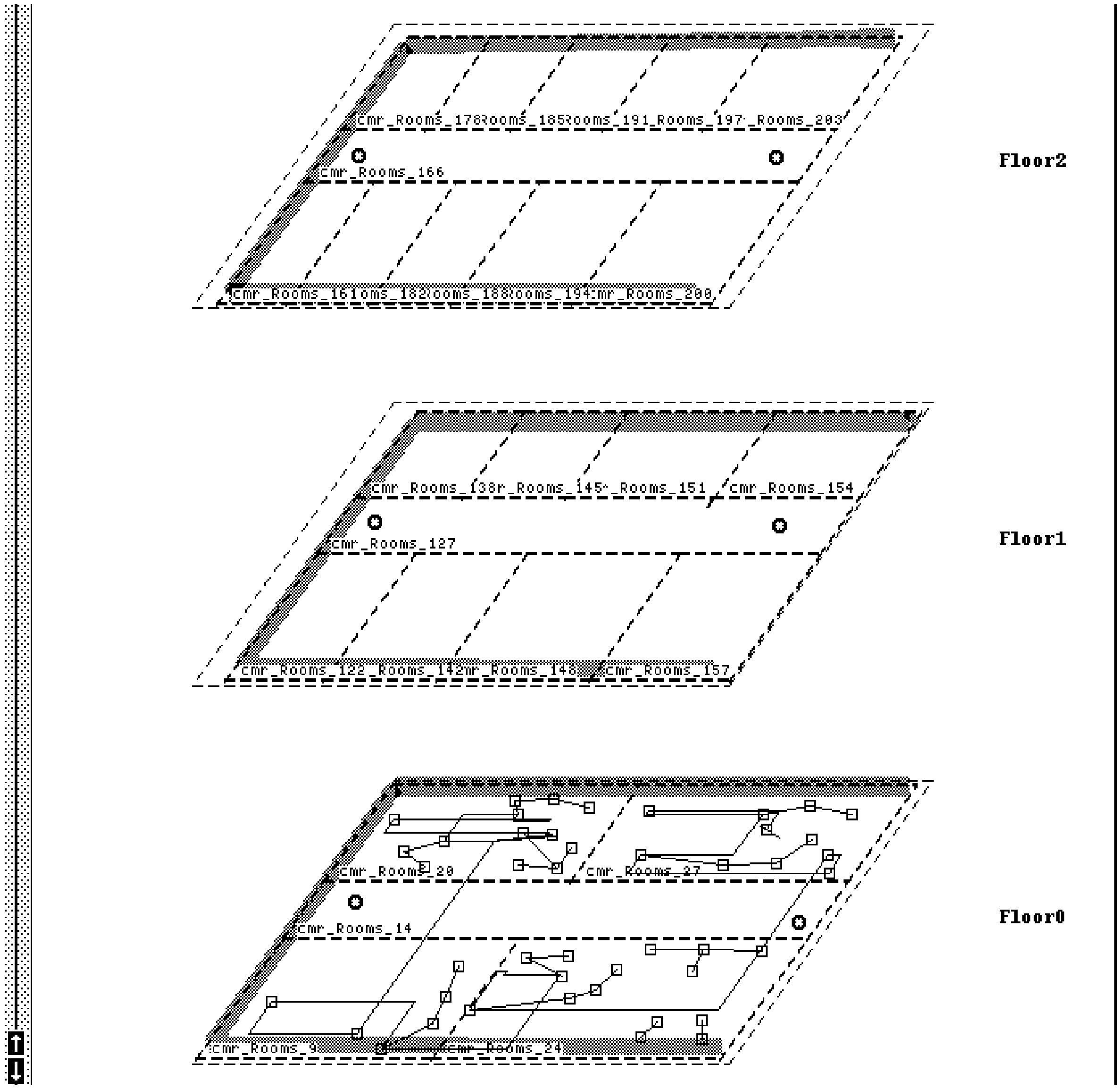
Hand drawn building components are automatically identified as walls as straightened bythe network tool.

● To allow a 3D representation of the buildings and the networks. In fact, the planar map is a data structure representing a 2D drawing, then the Network Tool displays this data in such a way that it looks like a 3D representation.

● To allow the other modules to deal with the graphical objects. Some predicates are implemented allowing the rest of the system to perform actions on the graphical objects (to move, to delete, to reclassify).



The Network Tool showing a layout of typical complexity used to demonstrate the MMI^2 system



A three dimensional view of a three storey building where the lowest floor includes a designed computer network.

NetTool

Author: BIM

Language: ProLog by BIM, C

Code Size:

This tool is essentially the graphical interface from the network edition in NetCortex. It is XView based and defined and generated with Carmen and the Control Board. It provides the user with a view of the containment tree of the network. Navigation through the tree is done by "zooming" into nodes (by double clicking on them). The interface provides 2 windows, each on different parts of the network tree.

The original NetCortex tool is created for editing purposes (ie. dynamically building and changing network descriptions). Again for the purpose of this exercise, we have concentrated on system replies through graphical displays (both full views and highlights in views) and user input through clicks.

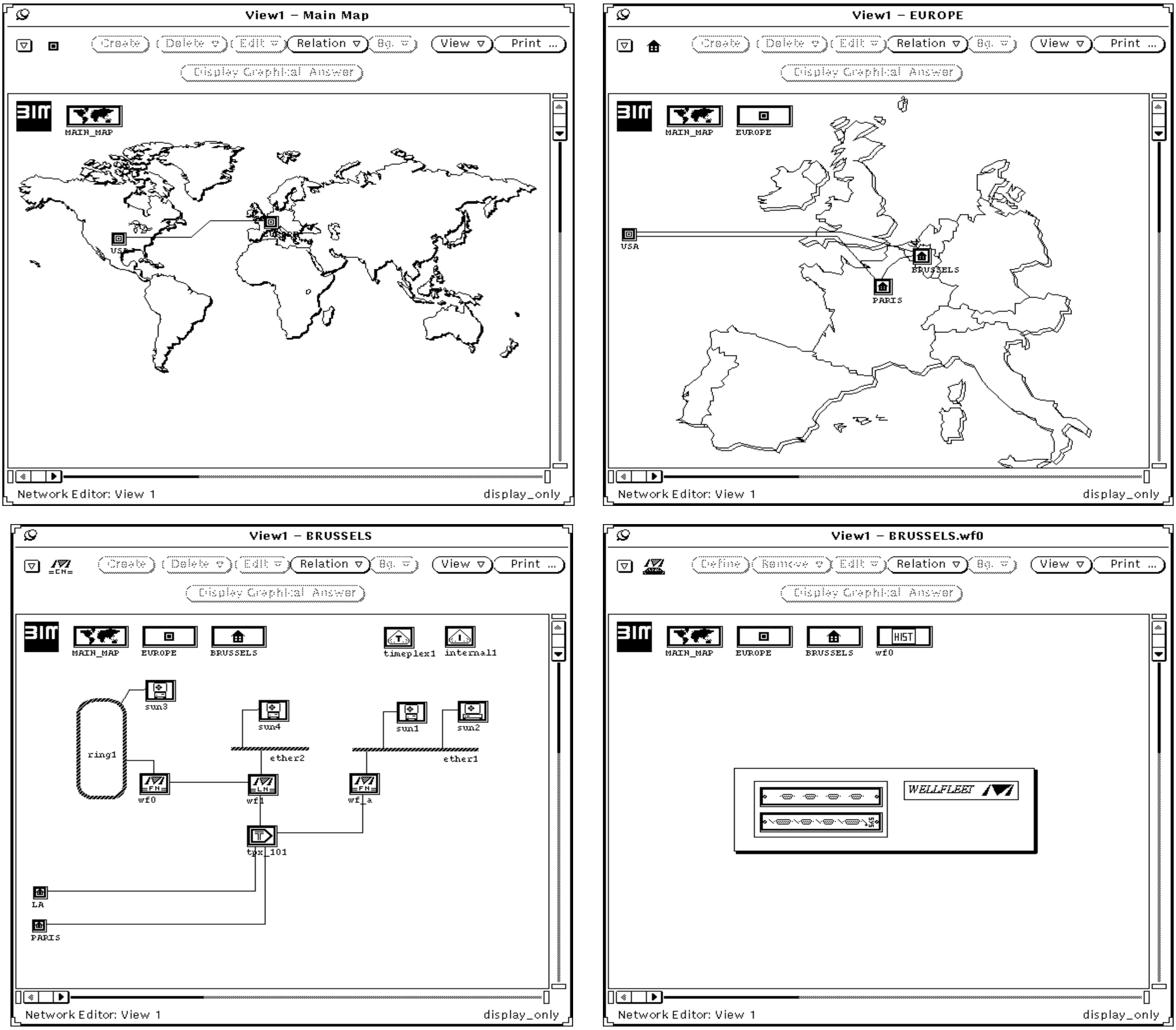
A network description in NetTool is organised in the following way. The top level node is represented by a map of the world, on which a number of icons represent what is called the supersites.

The supersites decompose into sites, which finally contain the actual networks themselves. It is common to label the higher 2 levels with geographical names (e.g. supersites correspond to countries and sites correspond to regions or cities). The networks themselves are grouped into higher level type equipment (workstations, routers, lans etc.) and lower level hardware parts (boards, interfaces, etc).

Links are represented as lines connecting 2 or more symbols. The physical links at the lowest level are represented again as virtual links between the corresponding higher level equipment.

A special type of view on the network's containment tree is provided by the so called double views. These allow 2 interlinked sister nodes of the tree to be inspected within the source view. Higher level virtual links can thus be decomposed down the tree to their corresponding physical connections

For the purpose of MMI^2 NetTool was supplemented with a logical view generation facility. These procedures use the same graphical presentations of network objects as the rest of NetTool. The difference is that they are grouped in a different way e.g. in a particular service subnetwork or along a connection path. The latter type of logical view is always presented in a separate window. The connection type logical view however is whenever possible presented as a highlighted path through the containment tree display in either a single or a double "traditional" NetTool view. If such highlighting is not possible, e.g. if the path runs across more than 2 distinct subtrees in the containment tree, then it is displayed in a separate window. The logical view facility offers the same interactive functionality as the rest of NetTool: logical views are displayed as replies to user requests; its icons can be selected and parsed as user input



Example views of the NetCortex containment tree

Graphics Manager

Author: RAL

Language: ProLog by BIM

Code Size: 6,200 lines

The Graphics Manager is part of the Graphics Mode of MMI^2. The user can communicate with the system through the Graphics Mode, and the system can communicate with the user through this mode. The Graphics Mode consists of a number of graphics tools and the Graphics Manager. The Graphics Manager manages the operation of these tools and handles the communication with the rest of the MMI^2 system. It therefore provides an interface between each individual graphics tool and the dialogue management part of MMI^2.

The Graphics Mode uses several graphics tools to display information to the user. These are: A network tool in which a network and building is displayed, and a number of “graphical analysis tools” (bar chart, pie chart, scatterplot and table). Whenever the user uses one of the graphics tools, the Graphics Manager translates the action into a representation that the rest of the MMI^2 system understands. Similarly, the Graphics Manager takes the system’s representation of the response to the user and finds a way of displaying this response graphically using the graphics tools. It then instructs the graphics tools to display the response to the user.

In summary, the functions that the Graphics Manager performs are:

- Translates user actions on a graphics tool into CMR (the meaning representation language used in the MMI^2 system). For the network tool, this entails translating the planar map structure which represents the network and/or building with the user’s design within the network tool into a CMR structure. This activity requires some application-specific knowledge.

- Translates the CMR representing the system’s response to the user into a graphical action:

  - by requesting a change on one of the existing graphics tool windows. (e.g. displaying a new network on the network tool, moving a machine from one room to another).

  - by displaying a new graphics tool. An example of this type of response is when the Graphics Manager dynamically generates a new chart that will try to display the system’s response in the most expressive and effective way. Heuristics are used to choose and design the most appropriate chart to display the response. These heuristics are based on: the data to be displayed, the current context; the current user; and knowledge of properties of the charts.

- Manages a library of graphical data structures used in some of the tools.

- Provides a library of functions that all the graphical analysis tools use.

- Provides an interface through which other modules of the MMI^2 system can ask about the Graphics Mode. For example, which objects are currently visible to the user? Another example is questions involving reasoning about the relative spatial positioning of the objects visible to the user on the Network tool: e.g. Which machines are to the left of Object x? Thus the Graphics Mode can answer questions about the user’s current interface.

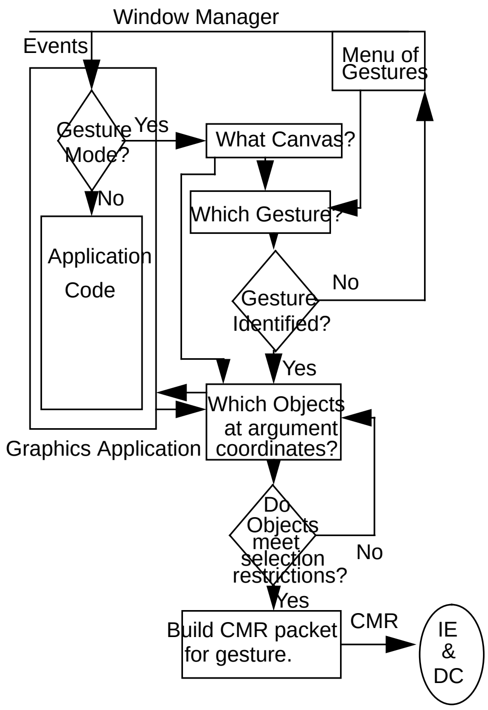
Gesture Mode

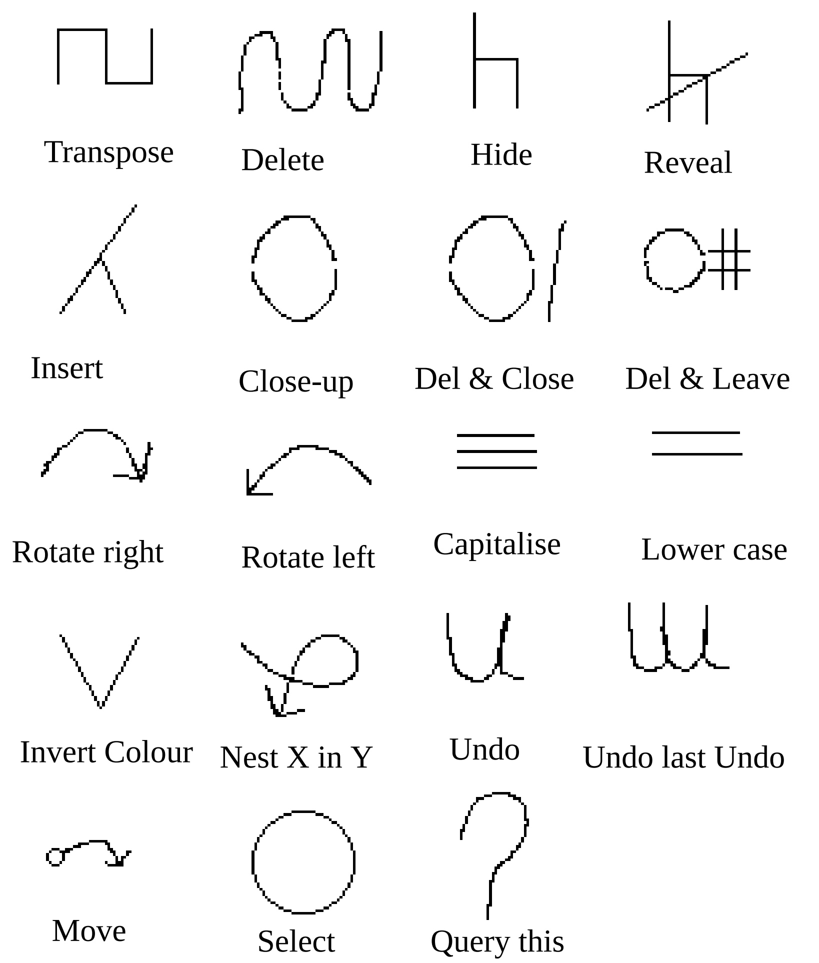
Author: University of Leeds, RAL

Language: ProLog by BIM, C

Code Size: 7,800 lines

The gesture mode of interaction in the MMI² system is intended to allow hand-drawn graphical symbols to be used as input commands to the system. In the illustrative application, of computer system network design, stereotypic symbols are used to modify the drawing of an existing network . The table overleaf shows a typical set of 19 such gestures which have been used as a test set in this project. Motivated by the network design problem, we propose an on-line decision tree approach based on extracted features characteristics, decision rules and number of continuous strokes. The designer gestures are recognised automatically as they are drawn on the screen using the mouse. The advantage of using the decision tree approach is that we may utilise the most important feature variables for each gesture. However, it is non-trivial to automatically extend the decision tree to another set of gestures, so we developed a signature matrix procedure. Each gesture can be considered to be specified as having a charcteristic ‘signature’. This procedure is also based on extracted characteristic features, decision rules and the number of continuous strokes. If all entries of the ‘signature’ vector of the input gesture match with a particular ‘signature’ vector of one of the test set then we classify the observed gesture as that particular gesture. As an option, a menu provides a feedback / correction mechanism which can be used when the automatic procedure misclassifies or fails to identify an input gesture. For new applications or individual development, an algorithm for entering new gestures into the system has been developed. This algorithm captures the reliable features of any drawn gesture and locates the objects/ arguments relative to the gesture. Such a tool would be an essential feature in any gesture interface system. The overall sequences of operations of the gesture mode are illustrated in the diagram below. The core language of the MMI² project is prolog since it is the best general langauge for developing dialogue systems. However, it is not a suitable language for mathematical or statistical modeling. The second language used by the project which has a well supported interface to prolog but which is suitable for mathematical programming is C. Therefore the statistical algorithm for gesture recognition developed at Leeds was coded in C while the other parts of the gesture recognition system were coded in prolog. All of the gesture input system was developed in C.The productivity of the work can be broadly gauged from the amount of commented code produced (see above).





Gestures used for the demonstration.

Command Language

Author: University of Leeds

Language: ProLog by BIM

Code Size: 1,126 lines

A generic interface requires a variety of functional groups of commands. In MMI^2 these are divided into:

- Meta Commands - e.g. access to UNIX commands, creation of macros
- File operations - e.g. loading a saved command file
- User-specific commands - e.g. changing language, user, or command name
- Dialogue - e.g. repeating the last command
- Information - e.g. asking for help or explanation
- Graphical actions - e.g. add an object to a location
- Network design - e.g. compute the cost of the design

All but the last one of these groups can be easily applied to any application, and the CL has been designed in order that new commands can be added simply by an application developer.

Early use

Early in the life of the project, prior to the implementation of the natural language systems, the first version of the CL mode provided a means to develop and test the dialogue management modules as they became available.

Commands across modes

Within unitary or limited mode interfaces, a Command Language (CL) is often chosen as an efficient and powerful means for experienced users to interact with a system. In contrast, as the number of modes of interaction increases, the issuing of commands becomes less tied to one mode, and may be achievable via a number of them. In MMI^2, for example, there are commands in the Graphics and Gesture modes as well as the CL. In such cases, a number of issues become important, if the users are to be served appropriately:

- The format of commands across modes should be as consistent as possible, given the constraints of each mode. For example, in MMI^2, it has not been possible to avoid inconsistency in argument ordering between the Graphics and CL modes.
- Users should not be forced to switch mode too frequently in the execution of a task. Thus, there is a need for some duplication of commands across modes to allow users to persist within a mode of interaction, rather than switch because such a command is not available. For example, in MMI^2, it is possible to issue a a MOVE command in Graphics, Gesture and CL modes.
- Users need to be able to refer (by anaphora, deixis) across as well as within modes. Thus, in MMI^2, anaphora resolution within the Dialogue Controller module will ultimately be able to allow cross- modal reference between input modes, including commands.

Interface Expert

Author: RAL

Language: ProLog by BIM, C

Code Size: 2,900 lines

The current Interface Expert (IE) performs five roles in the MMI^2 system:

1) Main ‘Mode to Dialogue Controller (DC)’ Interface - The Interface Expert gathers the CMR (the meaning representation language used in the MMI^2 interface) produced by each of the MMI2 modes whenever the user interacts with the system using that mode. It then passes the CMR to the DC. This level of interface limits the nesting of calls on user input and clearly separates the modes from the dialogue control function. Also, the Interface Expert takes the returned output from the DC and passes it to the modes for translation; displaying the text mode output itself. In other architectures this role is often called the ’scheduler’.

2) Providing Declarative Knowledge of the MMI^2 Interface The Interface Expert is responsible for the formal evaluation of CMR packets when the contents rely on knowledge of the MMI^2 interface rather than knowledge of the application KBS. There are three types of evaluation which any atomic predicate can be given: querying, asserting or retracting. The Dialogue Controller (DC) decides on the basis of the utterance type of CMR packets which of these three classes of operation should be performed on them. It then calls the Interface Expert for each atomic predicate to be evaluated in the specified way by querying or altering the MMI^2 Interface Expert. This either succeeds, passing back the necessary information to the DC, or fails. If it fails then the DC calls the Domain Expert to evaluate the atomic predicate by querying or altering the application KBS. Therefore these modules are allowing a formal interpretation of user queries and assertions.

In the case of queries, the Interface Expert returns a list of goals which provide a semantics for the atomic formula being queried. These goals may involve other modules: For example, to find out the truth of a predicate querying whether an object is on the left of another object, the Interface Expert returns a goal which involves calls to the Graphics mode for this information.

In the case of assertions of an atomic formula, the Interface Expert defines the necessary changes to the interface required to make the atomic formula true, and carries them out. For example, for an assertion that there is a new user of the MMI^2 interface, the Interface Expert calls the User Modelling module to change the current user.

3) Text Interaction Window - This role requires a window to be presented to the user which will allow user text input and system responses. This window incorporates simple text editing facilities and the gesture mode facilities where they operate on text.

4) A Screen Layout manager - In its screen layout management role the Interface Expert maintains a record of window positions used in the MMI^2 interface and provide locations for new windows on the screen which are mapped to the users task.

5) Control of User Events - The Interface Expert provides the main low level control loop for the MMI^2 interface which monitors user input events and controls how these are passed to the modules of the interface which will interpret them.

Context Expert

Author: CRISS

Language: ProLog by BIM

Code Size: 5,000 lines

The context expert is a module of the dialogue system. It contains contextual functionalities that are involved in the contextual processing of each move. These functionalities concern essentially anaphora and ellipses resolution.

The processing of anaphora and ellipses involves several modules in the MMI2 system. Utterances containg anaphorical or elliptical phenomena are represented by so-called incomplete CMRs. The CE contributes to their resolution by proposing possible candidates for completing incomplete CMRs. These candidats are extracted from a representation of the discourse constructed and maintained internally by the CE.

In previous versions of the context expert, contextual functionalities were implemented using the linear structure of the history of moves. This structure was read in reverse order to access information needed e.g. for resolving anaphora.

In the current version of the context expert, the history is no longer the “skeleton” around which are designed contextual functionalities. In the new approach relevant information is accessed through a focusing mechanism. This mechanism emphasizes the role of phrases and relations that link them together (anaphor and ellipsis) in discourse.

The focusing mechanism builds a focus. A focus is here an ordered list of representations (CMR segments) of noun phrases of the discourse. This focus can be interpreted as the set of most salient discourse elements at a given point of the dialogue. The focus is therefore updated at each move.

Updating the focus means four things. First it means attenuating the relative salience (or activation degree) of elements already in the focus. Second it means introducing new elements with their salience found in the utterance being processed. Third it means propagating activation from the new elements in the focus to other elements of the discourse through relations of cohesion (anaphor and ellipsis). Fourth and lastly, it means updating the focus by collecting the elements that have now the highest activation degree.

In order to find candidats for anaphora and ellipses resolution, the CE no longer searches linearly in the discourse history but limits its search to the space delimited by the focus taking into account the activation degree of the elements for its search priorities.

The focus approach is facilitated by the fact that discourse is represented internally in the CE by a formalism that is not standard CMR but a specific modified form of CMR that puts the noun phrase in the center of the representation (and not the utterance, as is the case for standard CMR).

Finally, the interface between the new CE and the other modules respects the same principles as the old CE did.

Ellipsis Module

Author: ISS

Language: Prolog by BIM

Code Size: 25,500 lines

When it comes to intersentential elliptic constituents, fragmentary expressions or utterances where some argument has been elided, the grammar rules construct a constituent structure to which a CMR interpretation with special annotations is assigned. The main work is then performed by the Elliptic Module - ELM. It calls the Context Expert - CE - and passes the necessary information to it in order for it to recover and test suitable candidates from previous interventions to construct a complete CMR.

The ELM module works directly on CMR representations issued from any NL Mode whose CMR's annotations have been adapted to the ellipsis resolution tasks. The ELM module performs its work recovering the elided constituents from previous user interventions being done by the Command Language - CL - or by a Natural Language Mode - EM, FM, SM.

The ELM works on the basis of two kinds of information corresponding to the two types of intersentential ellipsis established:

- **Semantic ellipsis**.

When a mandatory nominal argument of a verb is elided the NLM must offer a CMR where the type ENS is assigned to the elided arguments and its variable is annotated as 'elliptic_sem'.

The ELM calls the CE with the type of the predicate and the indexes of the missed arguments and the CE returns the most provable candidates which then are inserted in right places after being semantically tested.

After the ellipsis has been solved, a loose unification is made between the current CMR and the antecedent in such a way as to prime the values of the first over those of the latter. This leads to the recovering, not only of the omitted arguments of the verb, but also of its possible adjuncts.

- **Structural ellipsis**.

A structural ellipsis holds when the intervention lacks at least the verb . In this case the CMR offered by the Text or Natural Language Modes should include the annotation 4ellip_sem4 and its mode_info slot must contain information on the semantics of the phrases present in the intervention using the special concepts available for that reason in the Semantic Expert Module - SE.

For example, the elliptic intervention 'y de Sala1' ('and of Room1') supports a [GENITIVE] information in the info_slot, and, the intervention 'y a la red ethernet' ('and to the ethernet network') will need a [DEICTIC_PLACE_REF] concept. Taking this information as a basis, the ELM will look, by calling the Context Expert - CE, into the previous interventions for a predicate presenting an argument of this kind.

The candidate is tested in the SE using as arguments the objects present in the correspondent fragment.

If the test fails, the process backtracks over the call to the CE and a new candidate is produced. If the call to the CE fails, the ELM succeds nevertheless returning the input CMR unchanged.

Dialogue Controller

Author: BIM

Language: ProLog by BIM

Code Size: 1,473 lines

The major functions of the Dialogue Controller are to get input from the user and put output from the system. Input from the user always goes through a mode for translation into CMR (Common Meaning Representation). CMR output from the system goes through a mode for translation into the 'language' of the mode. The DC gets input CMRs from and sends output CMRs to the Interface Expert (IE). The DC classifies the input CMR of the user and assigns it a user 'attitude'. The following abbreviations are used: U-user, S-system, w-wants, k-know.

(Uwk,P)User poses question P
(UwSk,P)User tells the system P
(Uw,P)User wants P to be the case

A pair consisting of a user attitude and a CMR is called a 'user desire'. The classification of input is done on the basis of the form of the user input and (perhaps) the form of the previous system output. Questions are assigned the attitude Uwk. Commands are assigned Uw. Assertions are assigned UwSk. User input is placed in the 'dialogue context'. The DC treats the context as a stack: it pushes user desires onto the stack and it decides what to do next by popping the stack and acting on the desire it finds. For example, if the top of the stack is (Uwk,P), the DC finds the answer to P.

The DC also has a classification of system output:

(SwUk,P)System tells the user P
(Swk,P)System poses question P to user
(Sw,P)System wants P to be the case

When the system obtains a desire, the DC pushes it onto the stack. When the desire is popped off the stack, the DC outputs something suitable. For example, if (Swk,P) is popped, the DC sends the CMR for P to the IE, which calls the appropriate mode to translate P into the mode language. The result is a question to the user.

The behaviour of the DC can be described in the following table. The last line will be explained further below. 'C' stands for an arbitrary portion of the context.

ContextAction	New Context
emptyget user input P, classify	[(U attitude,P)]
[(Uwk,P)|C]find that answer is Q	[(SwUk,Q)|C]
[(Uw,P)|C]do P, confirm or disfirm	[(SwUk, con/disfirm)|C]
[(UwSk,P)|C]assert P, confirm or disfirm	[(SwUk, con/disfirm)|C]
[(SwUk,P)|C]output answer P	C
[(Swk,P)|C]output question P, get answer Q	[(UwSk,Q)|C]
[(Sw,P)|C]output command P	C
[(U attitude,P)|C]informal analysis of P	[(S attitude,P1), ...,(S attitude,Pn)|C]

Interpretation in a narrow and in a wide sense takes place in this processing. For example, in answering a user question, the DC must assign a denotation to the proposition the user expresses.We take the application to determine a model that is used in assigning denotations. If the user asks 'Does Machine1 have a disk?', the DC might find that the denotation of the proposition the user expresses is, say, true in the model. But how the DC reacts to the user input is also conditioned by the force of that input, not just by the proposition that is expressed. For example, if the user asserts 'Machine1 has a disk', the system must react in quite a different way. Here the DC does not find whether Machine1 has a disk in the current model; it updates the application so that Machine1 does have a disk in the new model.

Both the formal content and the illocutionary force of the user's utterance play a role in how the system reacts to the user. There is yet a third aspect of the user's utterance that is treated by the system. This can be illustrated with an example.

u: What does Network1 cost?
s: What is the type of Cable1?

Instead of literally answering the question, the system asks the user a question. What has happened here is that the system has decided that the user's question should not be literally answered. (The reason is that a component of the network (Cable1) is underspecified and so lacks a cost; hence the network itself cannot be given a cost.) The information the system uses to decide how to react to the user goes beyond information about the formal content and the illocutionary force of the user's question.

The sort of analysis the system performs on user input in order to support such dialogues as the above is called by us 'informal evaluation'. Its object is to interpret the 'informal' aspects of the user's communication action. This is in contrast to 'formal' evaluation, where the goal is to find the purely formal content (roughly, the model theoretic denotation) of the user's input. The output of informal analysis can be various system desires--to tell the user something, to ask the user something, &c. Thus the analysis enriches the dialogue context. The full behaviour of the DC is therefore more complicated than just answering questions or responding to assertions or commands. This behaviour is described in the last line of the table above. The DC receives a user desire and subjects it to informal analysis, thereby obtaining one or more system desires, which are pushed onto the context.

User input coming to the DC is not always ready for either formal or informal analysis, however. When an input CMR contains anaphoric expressions, these expressions have to be resolved before further processing can take place.

The DC detects anaphoric expressions in a CMR on the basis of the following criteria:

pronouns are anaphoric (they, its, ...)

definite, non uniquely denoting singular nounphrase descriptions are anaphoric (the machine - in case more than 1 machine exists)

non restricted plural nounphrases are anaphoric (the disks)

An anaphoric expression is resolved through relating it to an appropriate antecedent. In order to do this, the DC calls upon the Context Expert (CE) to provide suitable antecedents. The CE, which incorporates rules and criteria for selecting antecedents given a specific anaphor, proposes a number of antecedents in decreasing order of plausibility. The first one accepted is integrated in the CMR expression, and the next anaphor is tackled. When there are no anaphors left, the CMR is resolved and ready for further analysis.

The anaphor resolver supports pronominal anaphora (personal and possessive) and singular and plural nounphrase anaphora. Antecedents must always be explicitly present in the context for anaphora resolution to succeed. The CE does not support set construction for plural anaphora nor can it extract singular antecedents from a set of antecedents.

The relationship between an anaphor and its antecedent can be one of identity as in

user> Which machine is in Room9?
system> Machine4
user> What does it cost?

or a more indirect one, for example 'part of':

user> Which machine is in Room9?
system> Machine4
user> What does the disk cost?

To resolve anaphors such as appear in the latter example, the resolver relies on the presence of relations such as 'part of' between the relevant concepts in the SE.

MMI^2 global library

Author: BIM

Language: ProLog by BIM

Code Size: 3,057 lines

There are two libraries of global predicates in the system, which include general Prolog utilities (e.g. member/2, append/3) and utilities concerning the CMR (Common Meaning Representation). The CMR is the meaning representation language of the project. It is used to represent the communication actions of the user in every mode (graphics as well as NL). Semantically it is based on first order logic with extensions for generalized quantifiers and second order relations. It also provides room for pragmatic annotations that are used to describe the extra-logical properties of communication actions.

There are three important CMR predicates that support the interpretation of user input:

cmr_eval_set(+Var,+Formula,-Set)

Returns the Set of objects that are the denotations of Var for interpretations that satisfy Formula in the application domain. Var is assumed to be free in Formula.

cmr_eval_tv(+Formula,-TruthValue)

Returns TruthValue = true if Formula is satisfied in the application domain. Returns TruthValue = false otherwise. Formula is assumed to be closed.

cmr_update(+Formula,+Annotations,-NewObjects)

Updates application in a minimal way conformable to Formula. Any objects that are created in the application are returned in NewObjects.

These predicates are called by the DC, given its assessment of the force of the user's input. For example, if the input is a wh-question, then the DC calls cmr_eval_set/3, which returns the set of objects in the application that answer the question.

In addition, the CMR library has utilities that allow the syntax of the CMR to be described in Prolog. Then a pretty-printed view of the CMR definition and a set of selection and construction predicates that allow the CMR to be handled as an abstract data structure throughout the system can be automatically generated. These utilities provide some degree of version control and protect the software against inevitable changes in the design of the CMR. A few examples of CMRs and a pretty-printed version of the CMR syntax are included below.

Example CMR and its logical content.

'Does every machine have a disk?'

```
CMR(
   [CMR_act_analysis(
      u_type(polar,question_mark),
      [CMR_exp(
         [anno(x1,[singular,indefinite,neuter]),
         anno(x2,[indefinite,singular,neuter])],
         description(desc(V,x1,COMPUTER,true),
         description(desc(E,x2,DISK,true),
         description(desc(E,x3,POSSESSING,true),
         conj(
            [atom(ARG2,[var(x3),var(x2)]),
            atom(ARG1,[var(x3),var(x1)])])))),
         nil)],
      nil)],
   ok,
   English,
   time)


all _4055:
   COMPUTER(_4055)
   ----------
   some _4105:
      DISK(_4105)
      ----------
      some _4155:
         POSSESSING(_4155)
         ----------
         ARG2(_4155,_4105) &
         ARG1(_4155,_4055)
```

Semantic Expert

Author: ISS

Language: ProLog by BIM

Code Size: 4,500 lines

The SE joins an MMI^2 standard reified world of MMI^2 and a reified world of the application.

The MMI^2 standard world contains conceptual labels representing the reified objects of the interface, the commands, the graphic objects and the word sense representatives that are relevant to the communication for the three languages present in the interface.

The world of the application is represented by means of the relevant terms of the application and the word senses representative for the three NLs that are relevant to the communication in the cooperative tasks.

SE labels, roughly speaking, represent conjointly the possible objects, activities, states and operations of the MMI^2 communication world, legalizing the vocabulary of the CMR formalism.

The SE contains the semantic knowledge on these objects in a double perspective:

- An analytic knowledge that is based on a hierarchy of disjunctive types or classes enhanced when they hold, with two other relations: ROLE_OF and EVAL_ROLE_OF, and with definitional links represented by frames of indexed case arguments containing semantic constraints on their fillers. Roles represent functional perspectives of objects. Eval_roles represent attributive nominalizations of the indexed case arguments of the frame that define reified objects.

- A commonsense knowledge expressed by means of a selected group of relations between labels models the world in a standardized way. These relations are general commonsense relations typically held between the objects in the world (attributes of each class of objects, whole/part relations between objects, furtherly specified in four classes: member part, inseparable part, functional part and unspecific part).

The SE contains a corpus of procedures involving entailment on the analytic knowledge and exploiting the common sense knowledge, that can be used for example in:

- looking up in the hierarchy of classes
- proving the match of constraints
- proving the lacks of fillers of case frames
- giving awareness of the typical relations held between labels in the world, on their parts and attributes and on the equivalence between command and NL window labels

These facilities, which are defined as public predicates, insure the interaction and cooperation of the SE with other MMI^2 modules permitting direct consultation of the SE when they are needed.

The SE supports an on-line information facility on each of its labels, by using se_help/1, obtaining as a response the main relations the label is involved. It also provides a batch facility that creates a documentation file containing most of the available information on SE labels.

MAIN FUNCTIONALITIES

- Assembling application and standard MMI^2 worlds.

- Defining the common labels of the CMR formalism for the representation of input and output communication by the different modes. For the NL modes, this point is especially relevant in the following:

  - constraints in the argument fillers of the case frames.
  - ordering the cases for the three NLs.
  - pragmatic interpretation of coordinations in the

- fillers of a case frame.
- representing nominalizations.
- disambiguating relations that held between nouns in NPs.

The SE is also used by the MMI² Spanish Mode as a source for the compilation process of its NL runtime lexicon. In this case, the SE offers top-down information that is used directly in the parsing process.

- Stating conceptual correspondences between the MMI² commands, the MMI² graphic objects and the rest of the SE conceptual labels. Thus contributing to the multimodal integration.

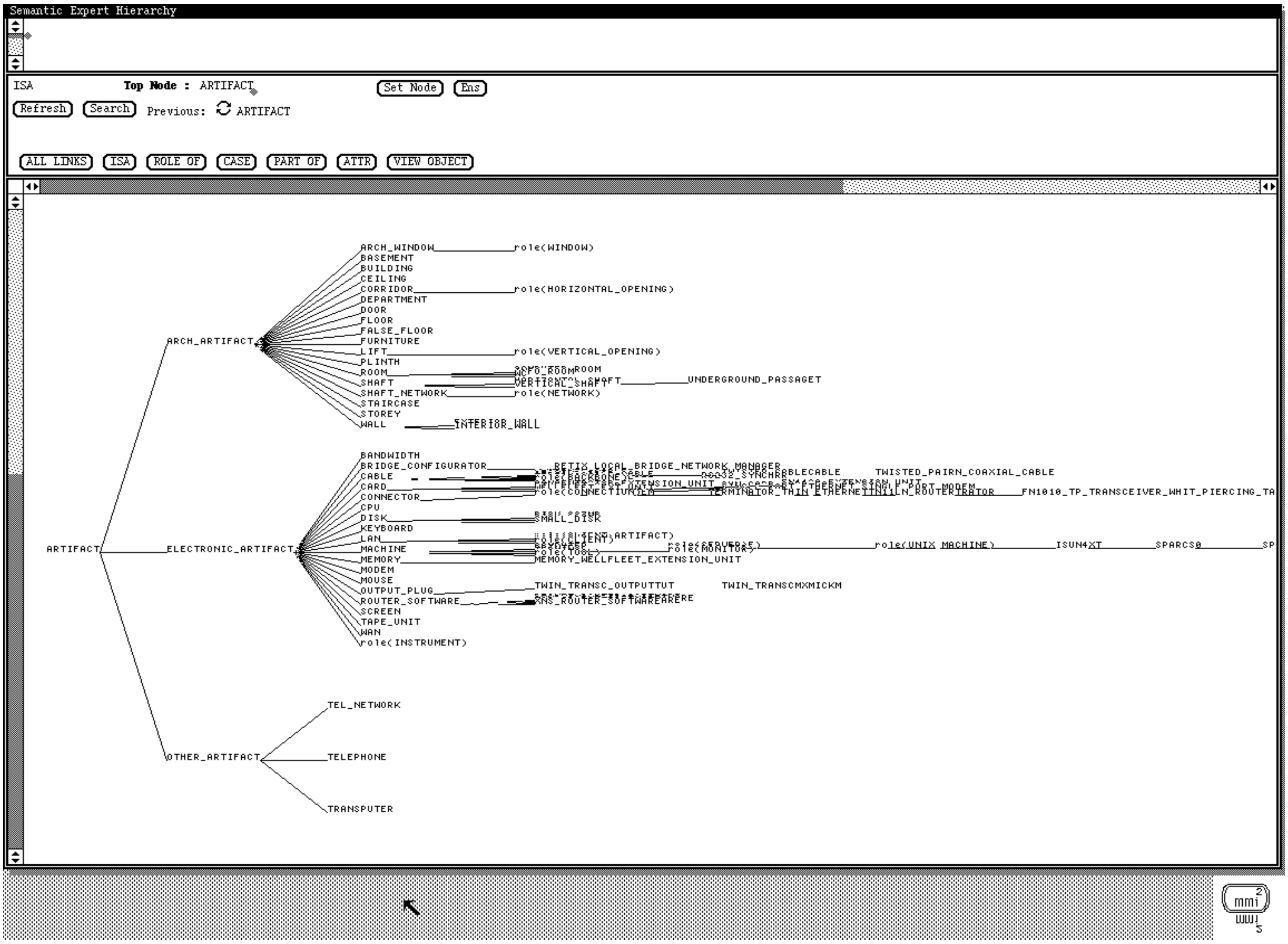
- Offering general predicate facilities to resolve dialogue phenomena, mainly, ambiguities of certain kinds, implicit knowledge presupposition, associative anaphora and reference.

- To help building up the complex concepts necessary for the communication and reasoning of the Interface and the KBS from simple concepts or CMR formulæ.

- To help constructing dialogue cooperative repair communications on command failures.

From an implementation point of view, the SE can be considered a standard reusable component of the interface having standard software interfaces through public predicates allowing the consultation of the SE knowledge and a core representing conceptual facts and procedures that constitute the bulk of its knowledge.

The semantic expert may be viewed through a graphical programmer tool shown below. This figure shows the ISA links from the top node down in the tree. It can be used to show other link types, or all link types, from any node in the network (This allows the user to see the local structure of children from a node in detail while providing a view of the shape of the tree below).



Domain Expert: formal

Author: RAL

Language: ProLog by BIM

Code Size: 3,500 lines

The role of the formal part of the Domain Expert is to provide a formal semantics for the representations of user inputs: that is to say, it gives formal definitions of all possible atomic queries and updates of the underlying application knowledge base. Thus it can be seen as an interface between the Dialogue Controller and the knowledge base for the formal interpretation of user queries and assertions. This is carried out at as far as possible in a general manner, providing a framework for the definition of individual terms and predicates relating to the particular domain under consideration.

Updates to the knowledge base may include both assertions and retractions. The assertion of a negative fact - “Machine 5 does not have a disk” - is supported both in the case where this requires changes to the knowledge base, and also the case where the existing state of affairs is already consistent with the user’s assertion.

All interactions with the knowledge base are carried out in an indirect manner through an intermediate level. This ensures a degree of generality by separating out the information which is specific to an individual knowledge base, thus making the interface more easily extensible allowing for the possibility of use of other and perhaps multiple knowledge bases, including those with pre-existing object instances not created through the MMI[2] interface.

The formal Domain Expert also provides denotations of all symbols used in the internal meaning representation language of the system. This is not restricted to symbols having a denotation comprising objects in the application knowledge-based system, for reference can be made to other objects also; the DE uses procedural attachment to give a denotation to any term which can be used in the CMR as defined in the Semantic Expert (qv).

Of the 900 symbols defined in the Semantic Expert which can be used in CMR constructions, 280 single place object predicates (e.g. SPARCS_SLC/1), 70 relationships (e.g. IS_IN/2) and 5 single place adjectival predicates (e.g. IS_SMALL/1) are defined in the domain expert so that their evaluation is represented in the application KBS. This accounts for all the objects in the KBS, except for 15 intermediate types in the KBS hierarchy which have no corresponding symbol in the SE (e.g. Network_parts). There are also several properties in the KBS which are not represented curently.

A subsidiary role of the formal domain expert is to define translations between units used for quantity relations such as exchange rates for currencies in which different users have costs presented.

Domain Expert: Informal

Author: RAL

Language: ProLog by BIM

Code Size: 1,400 lines

The role of the informal part of the Domain Expert is to provide pragmatic, dialogue-oriented functionalities beyond the narrow range of the formal DE, such as the ability to engage in repair, clarification or explanatory dialogues. This is achieved by inspection of the content of complete utterances, rather than the evaluation of atomic formulas carried out by the formal DE, and takes into account the roles of user input and system output in the overall dialogue. The functionalities addressed were chosen in response to needs identified in the course of study of dialogues in the network design domain, and represent the integration of a more pragmatic perspective into the formal semantic tradition embodied in the formal DE.

Repair:

the identification of error conditions which prevent a user's question or assertion from being handled straightforwardly, with a normal answer or update. The principal method of error analysis is the use of domain knowledge and system metaknowledge in the form of task plans. These define, for various standard tasks in the domain, prerequisite information and constraints on the current state of the domain world. The identification of problems avoids abortive attempts to carry out tasks and allows the system to temporarily take the initiative in the dialogue whilst continuing pursuit of the user's goals.

Explanation:

the determination and provision of replies to user questions that are more informative than formal evaluation alone would give - typically, more informative answers to polar questions than a simple “yes” or “no”. This not only helps increase the user's knowledge of the domain but also avoids unwanted implicatures associated with the bare answer.

Clarification:

assessing user answers to system questions, checking their validity and determining the update required in the case of elliptical replies. This screens out not only unacceptable answers, but also replies which are not actually intended as direct answers to the question.

User Model

Author: RAL

Language: ProLog by BIM

Code Size: 6,200 lines

The User Modelling module dynamically acquires and stores knowledge of the users of the MMI^2 system. This knowledge enables the MMI^2 system to respond more cooperatively to the current user by using knowledge of that user.

The knowledge of the current user that is acquired and stored is:

- The user's general knowledge of the domain of the application - e.g. which domain objects does the user know about?
- The user's knowledge of the MMI^2 system - e.g. which MMI^2 commands does the user know about?
- The user's preferences with respect to the MMI^2 system - e.g. in which currency would the user prefer prices to be given?

Information about the current user is derived from the dialogue. This is represented in a user model for that individual user which is stored permanently between sessions. From the information in the user model, it may be possible to categorise the user as a certain type of user. Having made a categorisation, further knowledge about the user is available because of assumptions that can be made based on stored knowledge about different types of users. The hierarchy of user types allows multiple inheritance from different stereotypes.

Within the individual user model is an explicit representation of knowledge about the user. This is in the form of instantiated predicates. The predicates are determined by knowledge acquisition and represent such concepts as knowledge, misconception, weak knowledge, preference, &c. The user model contains both long-term characteristics about the user (e.g. the language that the user prefers) as well as short-term characteristics (e.g. the user's current domain knowledge).

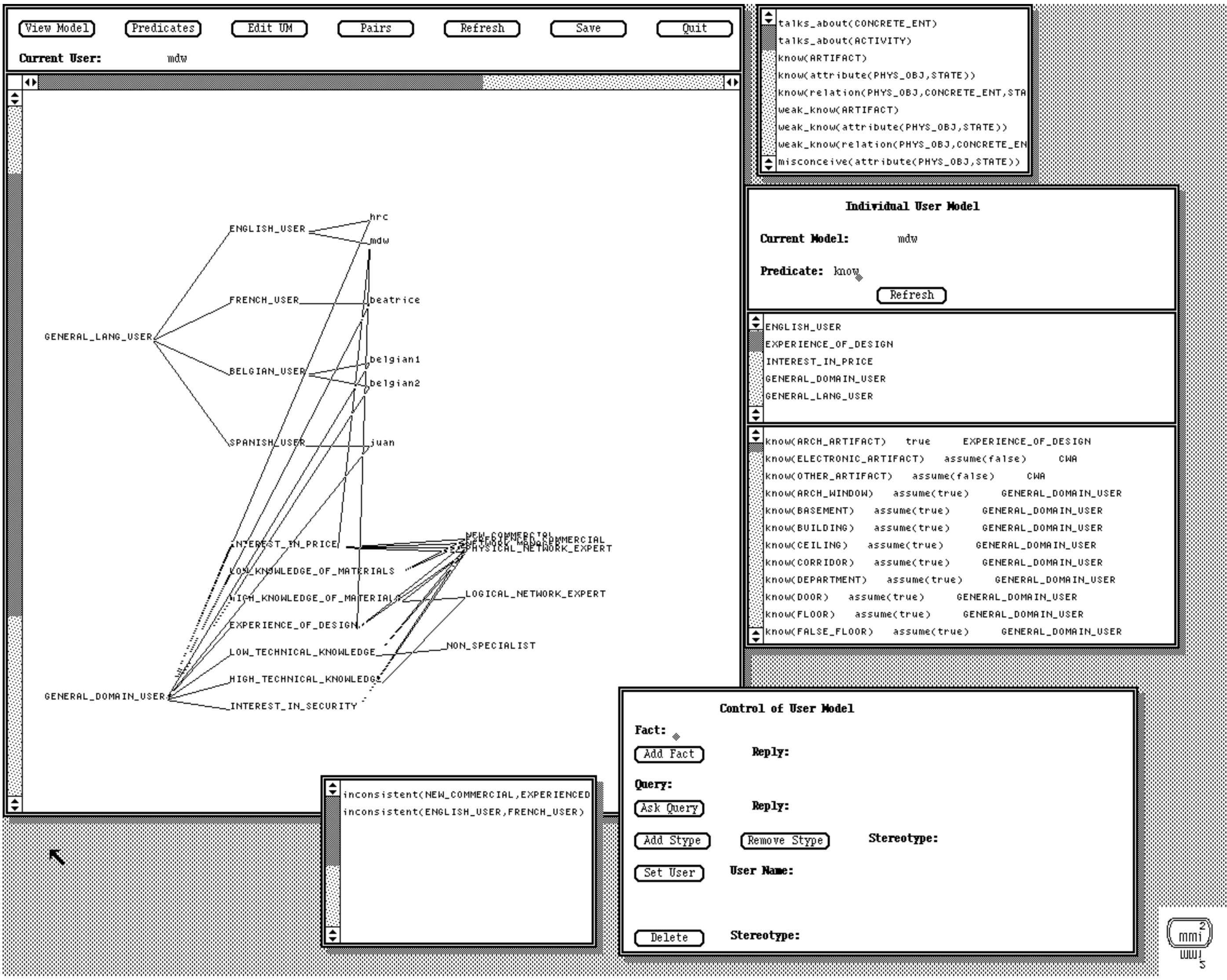
The User Modelling module provides an interface through which other parts of the system ask questions about the current user. For example, does the user know about a particular property of an object? Or, is the user of a particular stereotype? The User Modelling module answers the questions according to what is currently known about the user at that point in the dialogue. The knowledge about the user is used in several ways in the system: for example the communication planning module will give a lengthier explanation to a user who does not know about the area being explained than to a more expert user.

Obtaining Domain dependent knowledge about User Modelling:

A method was developed by Beatrice Cahour and Pierre Falzon at INRIA to obtain knowledge from human experts and users in the application domain that could be used within the User Modelling module. This knowledge was based on how the human experts modelled their interlocutor during simulated dialogues and from questionnaires to potential users in the application domain.

From these studies, the following knowledge was obtained by them and used within the User Modelling module:

- The stereotype hierarchy of users in the domain.
- What domain knowledge would be known by a member of a particular user stereotype.
- The rules that allow a human expert to infer the state of knowledge of an interlocutor from the dialogue.
- The way that the human expert decides that an interlocutor belongs to a particular stereotype.



The user model can be viewed by programmers through a graphical interface to show the inheritance network of user stereotypes (left of the figure) , the knowledge within a particular user model (middle right of the figure), the predicates permitted in user models (top right), and any inconsistencies between beliefs in a user model , or derived from its parents (centre bottom). The beliefs, parents, and rules can all be edited through an editing tool shown in the bottom right corner of this figure.

Communication Planner

Author: ISS

Language: ProLog by BIM

Code Size: 2,900 lines

The role of a Communication Planner (CP) is to decide in which way to convey the system's intentions to the user in such a way as to follow the rules of cooperativity in dialogue.

The system's output communication is conceived in the following way:

Provided with the necessary information from other modules in the interface:

- a response status predicate (RSP) from the informal domain expert (DE),
- a declaration of expectation from the dialogue controller (DC),
- the MMI[2] common meaning representation (CMR) of the user's last utterance,
- user information from the user model (UM),
- its own dialogue history

the CP first determines what communication plan (c_plan) to trigger and in what mode the output is going to be made (NL or Graphics). According to the resulting c_plan, it then captures the required information and builds each of the argumentational roles (a_role) of which the given c_plan is composed.

Example of c_plans:

- request_specification_plan
- inform_specification_plan
- inform_unknown_plan
- deixis_answer_plan

The a_roles are derived from the argumentational level of the dialogue interventions detected in the dialogue corpus derived from the Wizard of Oz tests previously made on network designing tasks. They can be said to be the contextual illocutionary forces of the interventions within a dialogue. The a_roles are composed by one or more communication acts (c_acts) which in MMI[2] are represented by individual CMRs.

Examples of a_roles:

- notification
- justification
- negation := regret | decline
- acknowledge
- reply
- interrogation

The following is the structure of the system's dialogue interventions:

system intention := c_plan
c_plan := a_role + a_role*
a_role := c_act + c_act*
c_act := CMR

The RSPs are the set of predicates the informal DE uses as diagnosis of any problem that requires the system to initiate a repair subdialogue. They contain the type of problem at hand plus the information items that caused the reaction. The CP then uses that information to generate the CMRs for each a_role in the c_plan triggered by the situation.

SITUATION <--- C_PLANx --->A_ROLEa/C_PLANy
triggered by has expectation

Examples of response status predicates:

- underspecified (PREDICATE,[X,Y])
- unspecified (PREDICATE,[X,Y])
- unknown (PREDICATE,[X,Y])

The following figure illustrates the c_plan request_specification_plan and a repair dialogue triggered by the underspecification of one of the arguments of the NL command 'Add a workstation to the network':

RSP: underspecified (ADDING,[ARG1])

c_plan:
request_specification_plan :-
regret,
notify,
justify,
interrogation.

Dialogue structured into subdialogues and using the a_roles of each intervention:

Dialogue		**Main dialogue level**	**Subdialogue level**	
1 u:	Add a workstation to the network.	request		1
2 s:	I'm sorry.		regret	2
3 s:	The location and the type of the workstation is underspecified.		notification	3
4 s:	Every addition of a workstation requires specification its location and type.		justification	4
5 s:	What is the type of the workstation?		interrogation	5
6 u:	Sun3/60.		reply	6
7 s:	What is the location of the workstation?		interrogation	7
8 u:	<uses graphics to specify location x>		reply	8
9 s:	<adds the Sun3/60 at location x>	<action>		9
10 s:	ok.	acknowledgement		10

CP treatment of 'Add a workstation to the network'

NEST: a Network design Expert SysTem

Author: BIM, EMSE

Language: ProLog by BIM

Code Size: 8,237 lines

Our objective was to build an expert system for Local Area Network (LAN) configuration using Ethernet - TCP/IP technology. In a first stage, an analysis tool was developed which purpose was to validate a network designed by the user through the graphical interface. The second phase was the conception of a full automatic designing tool which designs simple networks starting from customer requirements (description of building, computers, network requirements, budget and so forth). The currently developed NEST prototype integrates both parts the analysis one and the designing one by using a common database. Below we will present the various components of the existing NEST and introduce the diverse used techniques : objects for the database, prolog for the methods related to the analysis and the designing parts.

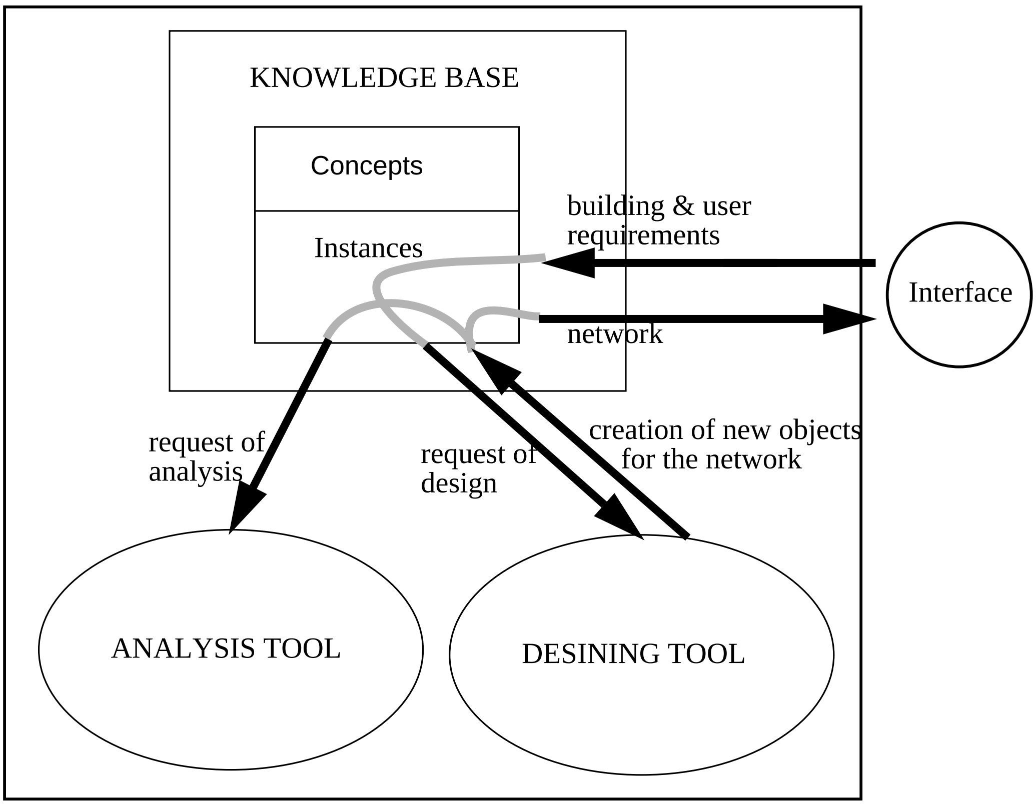
NEST is composed of three main components :

(1) the common database,

(2) the analysis part and

(3) the designing tool.

Before presenting each one separately, let us introduce and justify the language used for the development of NEST : BIM_Probe. It is an object oriented system built on top of ProLog_by_BIM and coupled with a constraint based language. Designing or analysing a network requires to represent all sort of components which can be part of a network : cables, repeaters, bridges, and so on, and also to describe the information related to a building. This information must be represented as various abstraction levels on the components. Indeed, in a first design proposal, it is not need to define exactly the box that must be used, for example a Retix2255M, but it is



The Architecture of NEST

● The analysis tool

This tool can perform various analyses on any fully specified network obtained through the graphical interface or designed by NEST. Five types of analysis can be performed on a network using NEST:

The validity analysis checks if a given network is valid according to the Ethernet technology and reports the found problems. For example, these problems can be (1) some network components cannot be connected ; (2) there are too many or too long segments of cable in the network.

The extensibility analysis. As a quality criterion for a network is its ability to be extended for future customers' needs, an extensibility analysis is provided in the tool. It produces information about how a given network can be extended while respecting the Ethernet protocol.

The client-server relation analysis aims at seeing if the client-server relation is correctly implemented in a given network for example (1) if diskless and server are on the same subnetwork, (2) if each machine has a disk or a disk server, (3) if each server has a disk, ...

The departmentalisation analysis reports the possible fact that machines belonging to a same department are separated by bridges or routers.

The cost analysis computes the cost of a network.

● The designing part of NEST

The implemented tool is a fully automatic one. That means that a full and detailed design is obtained through a single call to it after having entered the complete needed information (building topography, user requirements, budget and so on). The result is the creation of a whole network with selection and location of needed cables, boxes, links to machines and so on.

As the analysis methods, the desining part of NEST is implemented (in ProLog) as methods in the knowledge base. The design method can be seen as a succession of tasks implementing the hierarchical and modular aspects of the network design activity. The tool is implemented as a succession of four main

sufficient to connect a bridge if filtering is required. Accordingly as a lot of objects of real word have to be represented and as specialisation of object is needed, an object oriented language is well suited. These are some of the reasons for which BIM_Probe has been chosen for our application. Others are :

hierarchies of objects can be defined, with slot inheritance so that same properties have not to be redefined for subclasses ;

system flexibility is ensured by the possibility of modifying properties defined in superclasses and by the use of default values or exception slots ;

BIM_Probe is a powerful conceptual modelling tool providing a consistency checking mechanism so that a defined model can be checked in different ways : user defined constraints, absences of required information or wrong types of property values...

BIM_Probe is a programming language built on top of ProLog_by_BIM so that a defined model can really be run and queried. Methods of the analysis and the designing tools are defined in ProLog_by_BIM.

● The knowledge base.

It contains all the definitions of needed objects i.e. both the various network components and the topological information relative to the building(s) [see deliverable d44]. All the characteristics of the used language BIM_Probe are used : (1) The hierarchical structuration of data allowed by object oriented language is highly used so that different concepts share properties and property value ; (2) Specialization is used to particularize definitions for particular classes ; (3)Technical knowledge is expressed by default values given at the class level and inherited by all the instances ; (4) Procedural knowledge is expressed by methods describing the behaviour of the instances ; (5) Constraints are defined in order to reduce the data entry errors and to check that components satisfy the ethernet technology. The flexibility of this approach allows to deal with the fact that network entities are constantly changing due to the constant evolution of network technology ; new components can be easily added by definition of new classes.

tasks described below. Each task involves also the contribution of different modules [see deliverable d45].

*The skeleton selection s*elects the structure of the network i.e. mainly the needed vertical backbones and subnetworks.

The subnetwork repartition configurates subnetworks for each floor of the building (selection of types and location of cables, selection of connection boxes and machines, selection of connectors and so on).

The subnetwork merging : The different subnetworks designed at the previous step are linked to the vertical backbone defined at the skeleton selection stage in order to obtain a network connecting the different machines over the different floors.

The refinement and optimization : According to cost, budget and technological needs, some refinements and optimizations can be performed such as improving the performance by adding new softwares.

Conclusion.

Through the development of NEST, two main problems have been tackled and implemented : the design activity for which artificial intelligence only has relatively weak theories and the LAN configuration for which few expert systems have been created [Metzler & al 88].The emphasis has been to use an object oriented language to represent the data of network configuration and to modelize the design activity by writing methods in prolog and constraints attached to objects.

The state of the system is that all the features mentioned here have been implemented and coupled with the multi mode interface, so that one can design or analyse a network. After the user has drawn a building, located the machines and specified the requirements, NEST can computes a network. This system has been successfully tested. Work continues on providing more complex networks and on optimazing the solution. As NEST is a full automatic designer, the next two years of the development will be concerned with improvement of the flexibility of the tool. Accordingly the user will be allowed to intervene during and after the design process mainly by adding new constraints or providing his own parts of the solutions.

NetCortex

Author: BIM

Language: ProLog by BIM

Code Size:

The NetCortex system is a commercially used computer network management tool from BIM which describes the network as a tree, organised on the principle of physical containment. The system's graphical interface offers a way to navigate through the network tree and inspect the physical connections between the nodes in the tree. The information that a network operator needs with a view to e.g. network reconfiguration or alarm interpretation is present in the tree, or in the information packages attached to the nodes, but he may be forced to inspect quite a few different 'physical' subtrees in order to collect a full picture.

NetCortex provided the second demonstrator application for MMI^2 consisting of 2 parts.

The MIT **(Management Information Tree)** describes a particular network as a tree in which the father-son relation signifies 'physically contains'. This tree is implemented as a collection of ProLog facts. Aside from the containment relation, the MIT holds information on

- Names.
- Class membership.
- Security information.
- Link information. .
- Special relations between hardware parts.
- Poll address.

The **logical view rule base** on top of the MIT computes 2 types of logical views of the network.

- Service oriented logical views defining subnetworks of particular types of equipment.
- Connection oriented logical views defining the communication path between 2 pieces of equipment.

5 The MMI2 Demonstration Scripts

Two demonstration scripts are presented. The first was used for the first demonstrator and the second for the second. The first script is based on a scenario of designing a local computer network for a building. It follows a structure where the user describes what they would like to see on the potential network, then the system asks for other necessary requirements which follow from this such as a description of the building itself. After this the system designs and displays the network over the building diagram. The user then goes on to investigate the network to analyse its properties, then alters the specification. The system then re-designs the network for the new requirements. The user then analyses this design and accepts it. The system finally presents a table of parts required to be ordered for this design.

The second script is based on a scenario of a network manager investigating the state of a wide area network covering the France, Belgium and the USA. The user investigates the locations of some machines and the links between them in the cities of Brussels, Paris and Los Angeles. This requires the use of different resolutions of logical representation from a map based international view down to detailed local connections between machines. The user having located the machine of interest finally investigates the performance of machines the network.

The moves by the user and system are numbered. Some numbers have been omitted as the script has developed, while other numbers have been subdivided into several moves.

The second script is based on a scenario of monitoring a wide area computer network.

5.1 Conventions used in the scripts

Each entry in the script has the form:

<move number><agent><mode><content>

for example:

1 u (CL): name mdw

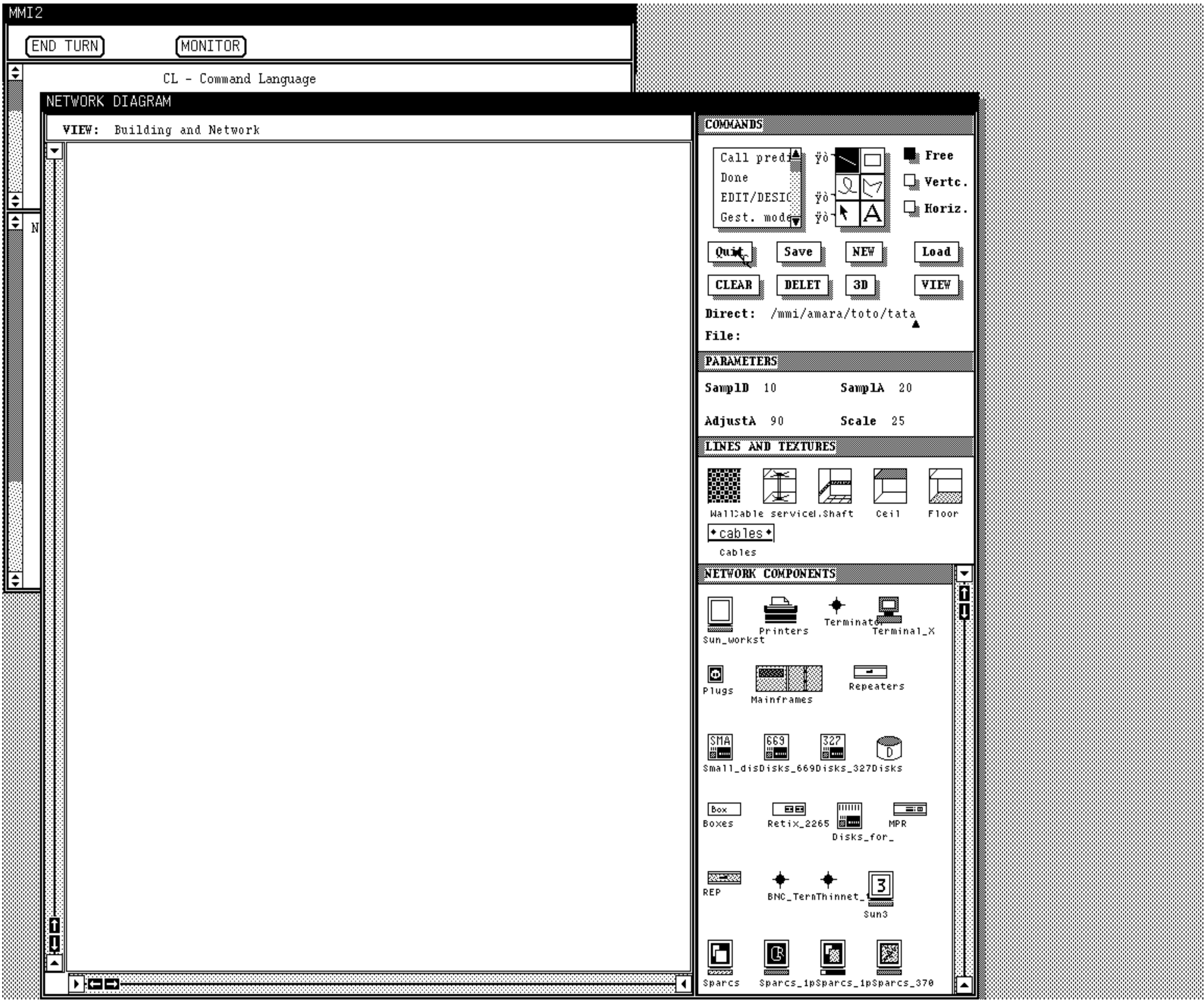
which is the first move, by the user, using Command Language mode, typing “name mdw”. The other modes are abbreviated to: English Mode (EM), Graphics (GR), Gesture (GS), Audio Mode (AM). The content of moves for modes other than EM are described with occasional illustrations. Text is not formatted as it would be on the screen where the natural language and command language would be typed into the appropriate windows of the text tool.

Moves should not be read as always consecutively following each other when multiple modes are used. For example in the moves 44 & 45 the user selects a horizontal shaft and refers to it in natural language “Is using-thin-cable possible in this shaft?”. It is possible to make the graphical selection at any point after the previous utterance, through to the carriage return at the end of this natural language utterance. Although these moves are shown as consecutive in the scripts the can occur in parallel at the physical level of keystrokes or at the utterance level in order to support the synergistic use and fusion of different modes.

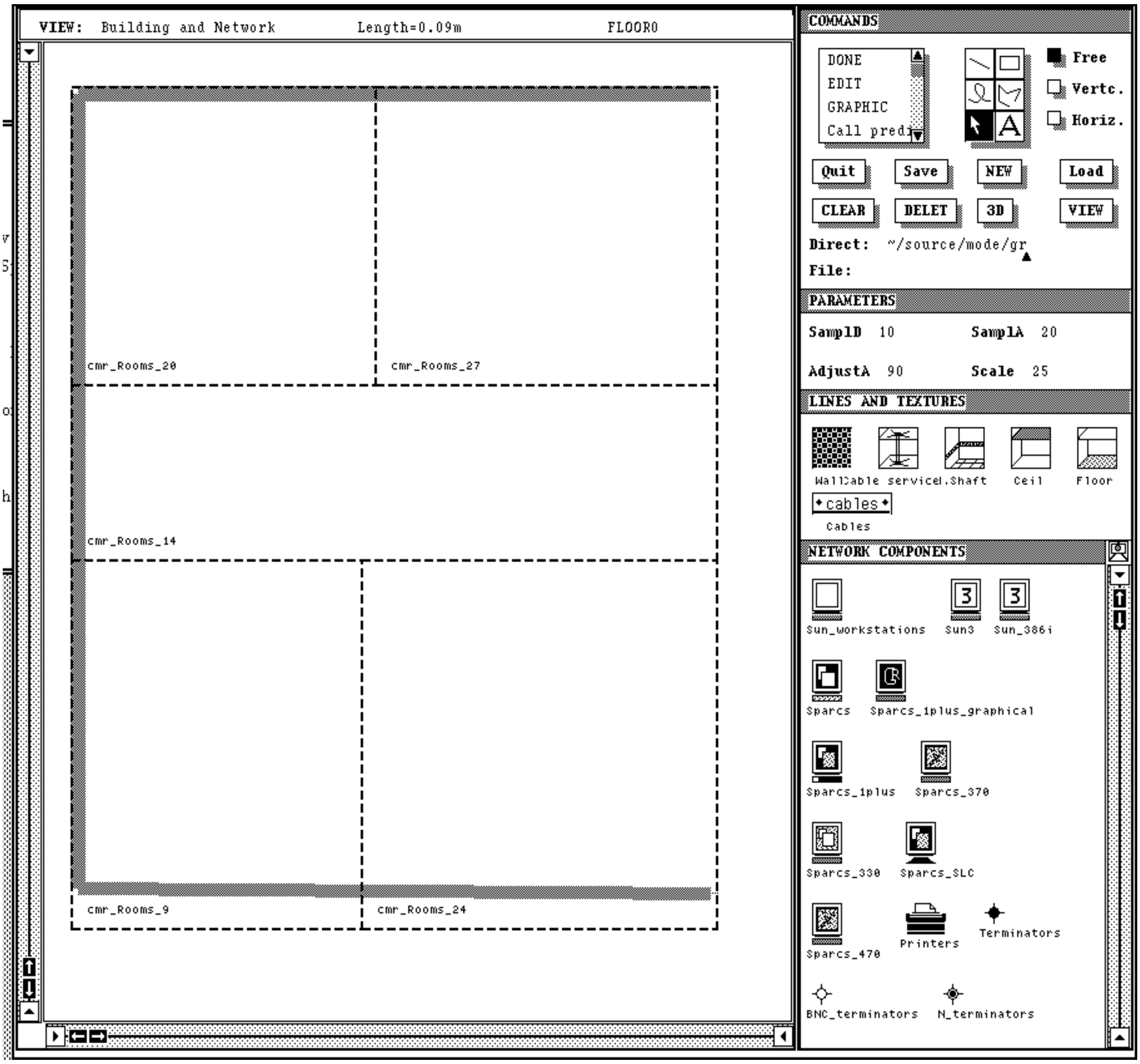
The identifiers for rooms, machines, and other objects used in the script are for illustration only and are not necessarily the ones which will be created or used for any particular demonstration of the system.

5.2 Demonstration Script for the First MMI2 Demonstrator

1 u (CL): name mdw
2 s (EM): ok
3 u (EM): Put 2 X_Terminals on a network.
4 s (EM): ok
5 u (EM): The network should have 4 Sparc_SLCs.
6 s (EM): ok
7 u (EM): Put the network in a building.
8 s (GR): puts up tool
9 s (EM): What are the specifications of the building?

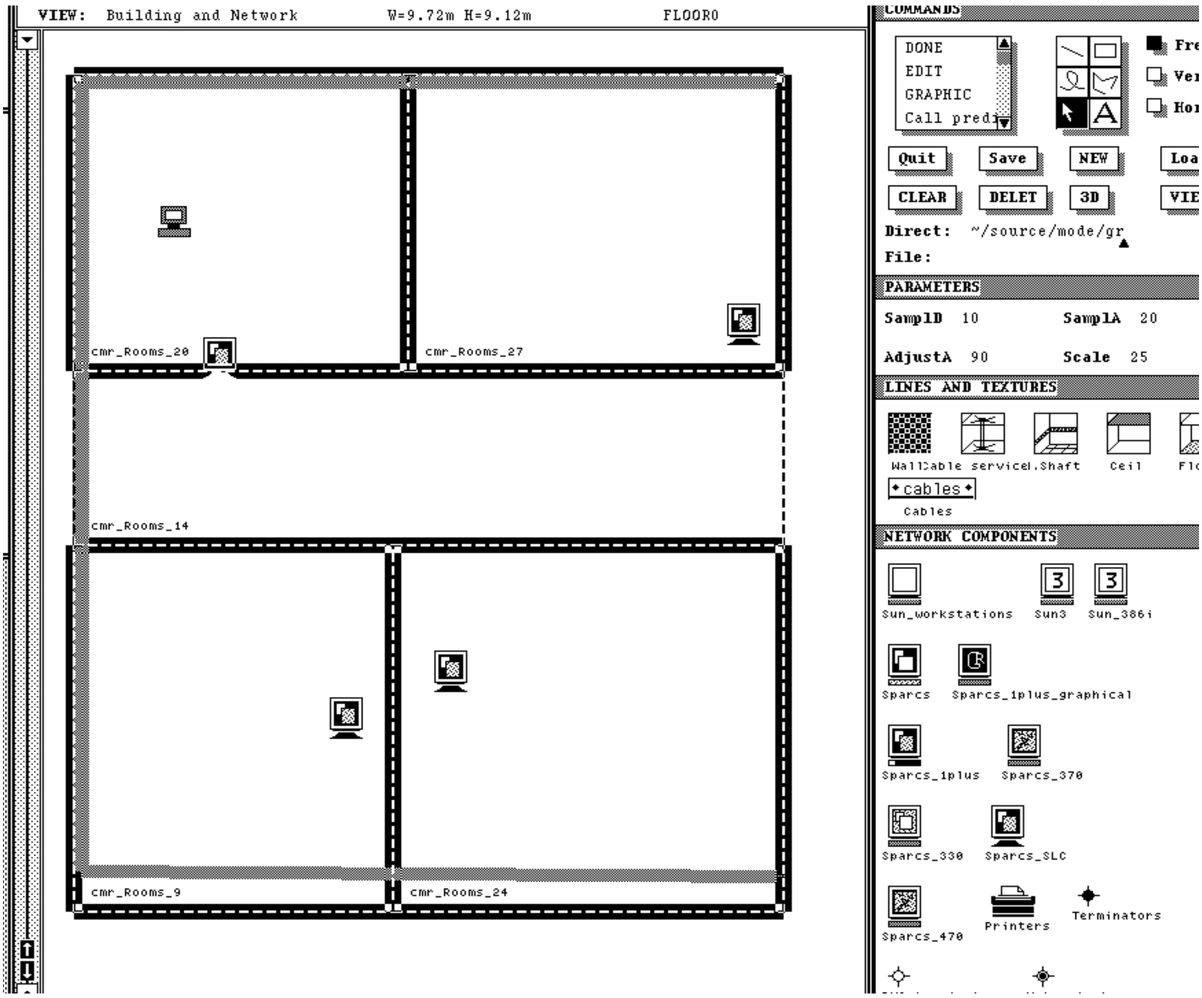


10 u (GR): draws building big enough to have a box, hits Done

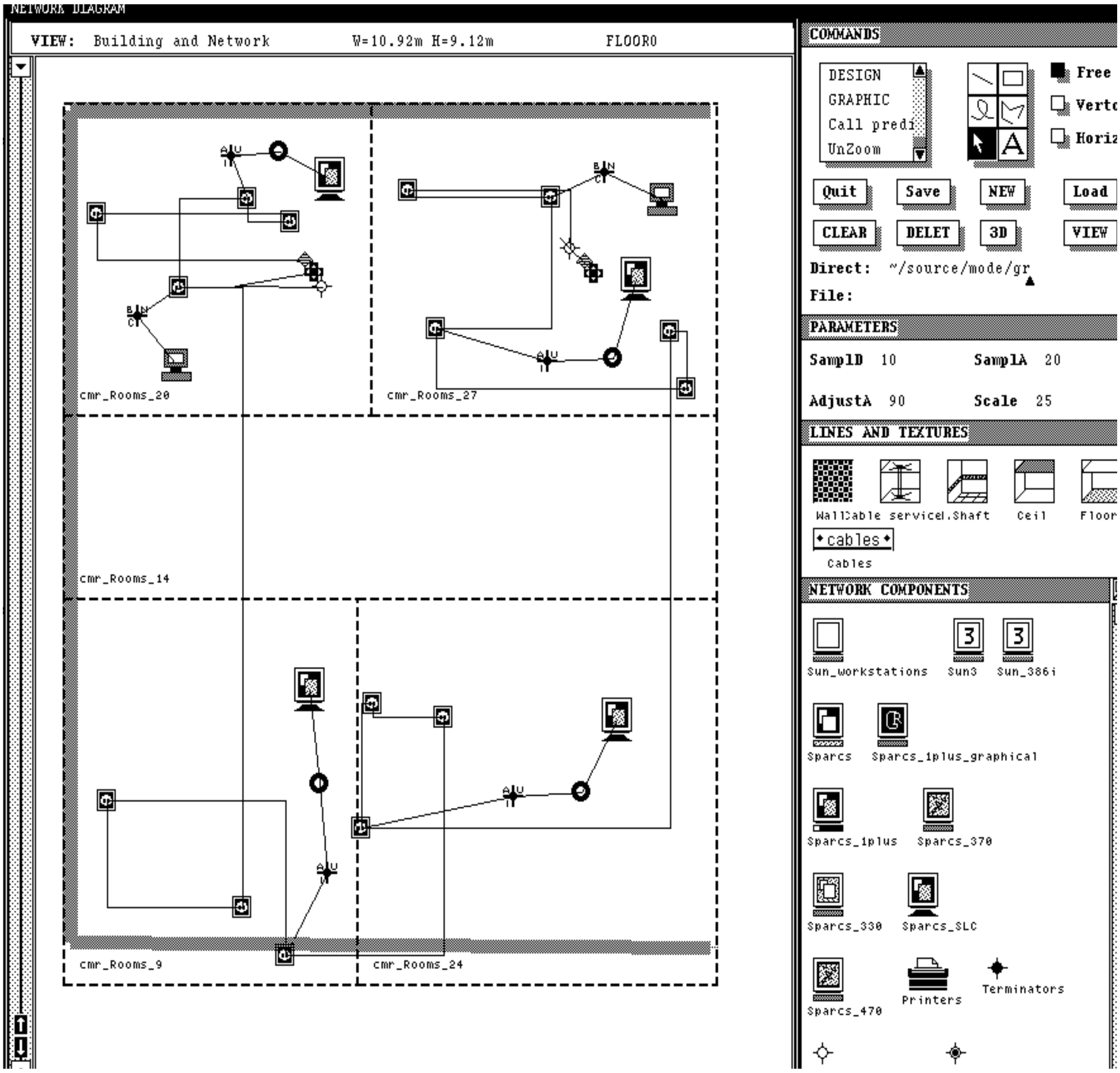


10.1 s (EM): What is the budget of the network?
10.2 u (EM): 1000 Sterling
10.21 s (EM): ok
10.3 s (EM): What is the average cost of installing cables on the network?
10.4 u (EM): 26 Sterling
10.41 s (EM): ok
11 s (EM): Where is Sparcs_SLC1?
12 u (EM): Room1
13 s (EM): ok <displays machine>
15 s (EM): Where is Sparcs_SLC2?
16 u (GR): selects room
17 s (EM): ok <displays machine>
19 s (EM): Where is Sparc_SLC3?
20 u (GR): selects room

21 s (EM): ok <displays machine>
23 s (EM): Where is Sparc_SLC4?
24 u (GR): selects room
25 s (EM): ok <displays machine>
27 s (EM): Where is X_Terminal1?
28 u (GR): selects room
29 s (EM): ok <displays machine>
31 s (EM): Where is X_Terminal2?

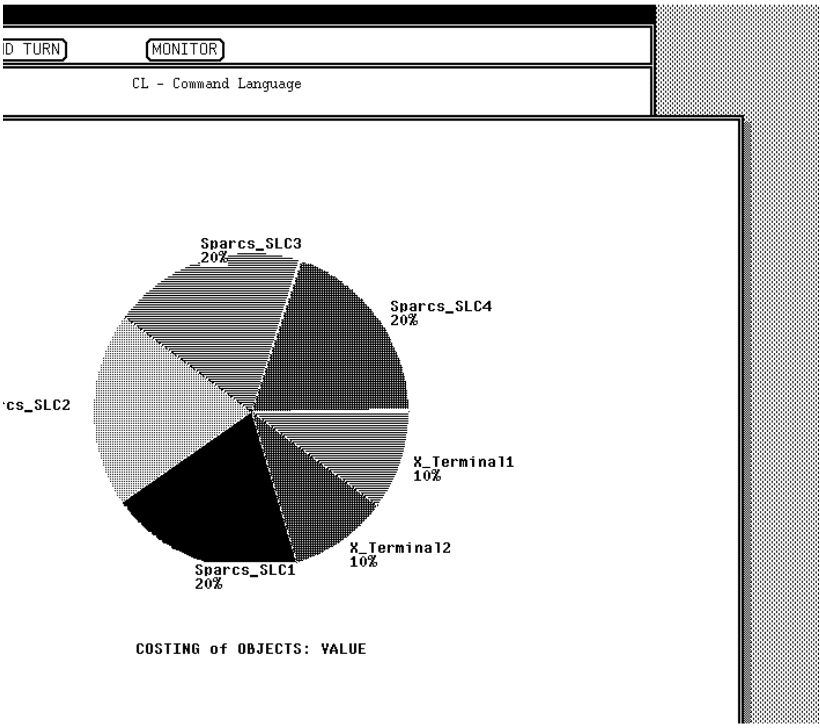


32 u (EM): cmr_Rooms_27
33 s (EM): ok <displays machine>
35 s (EM): ok <displays network>

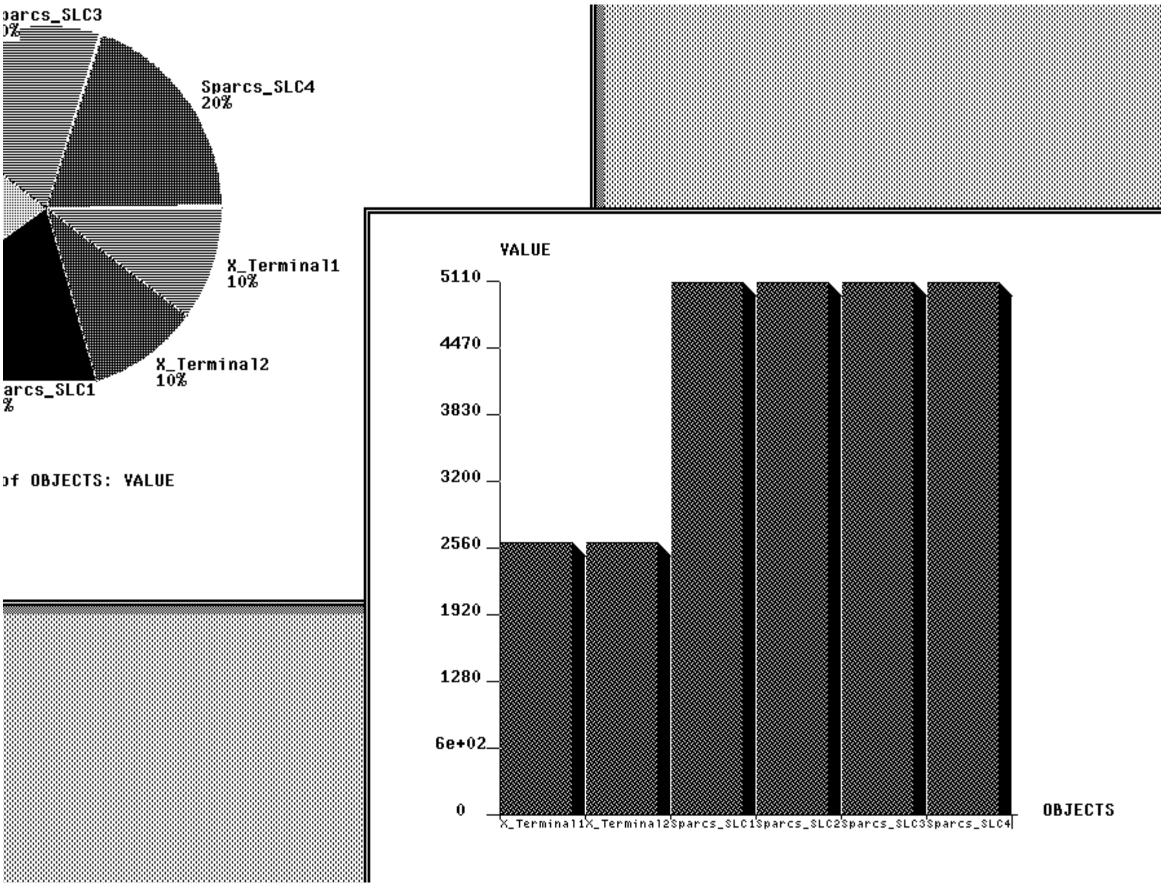


36 u (EM): What do the computers on the network cost?

37 s (GR): pie chart

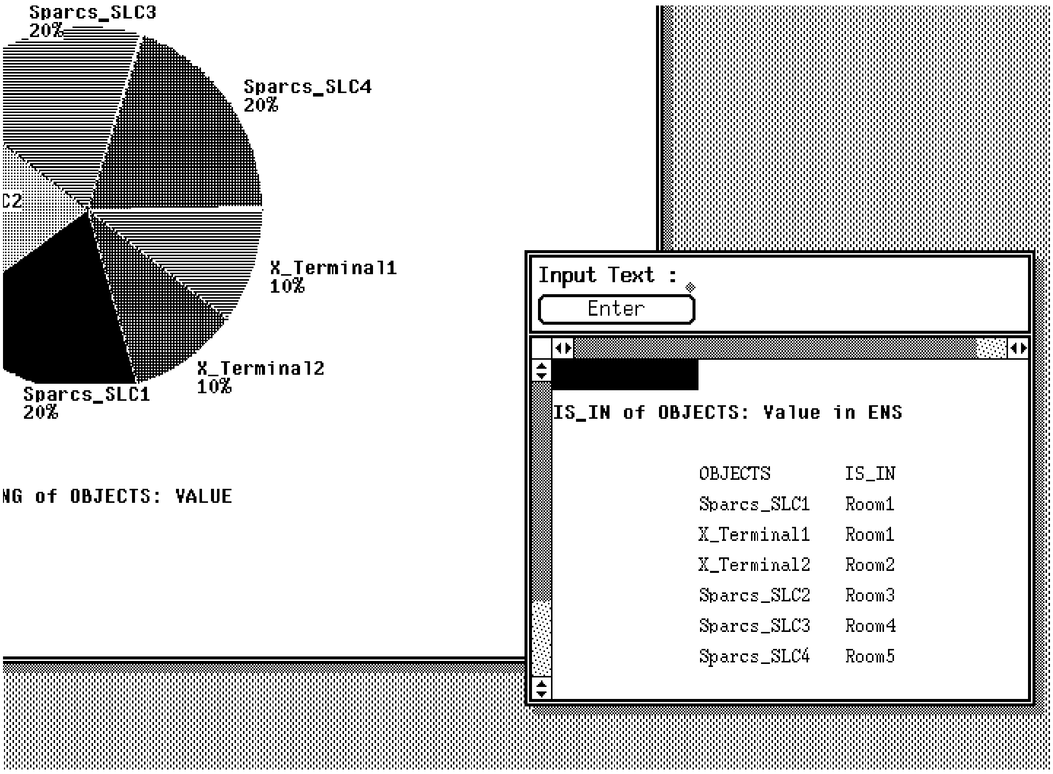


38 u (EM): Display a bar-graph of the cost of the computers.
39 s (EM): ok <displays graph>

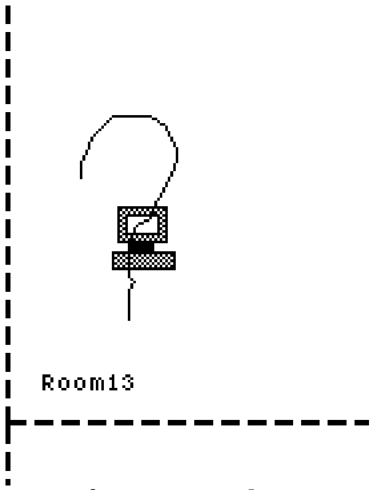


40 u (EM): Which machines are in which rooms?

41 s (GR): table

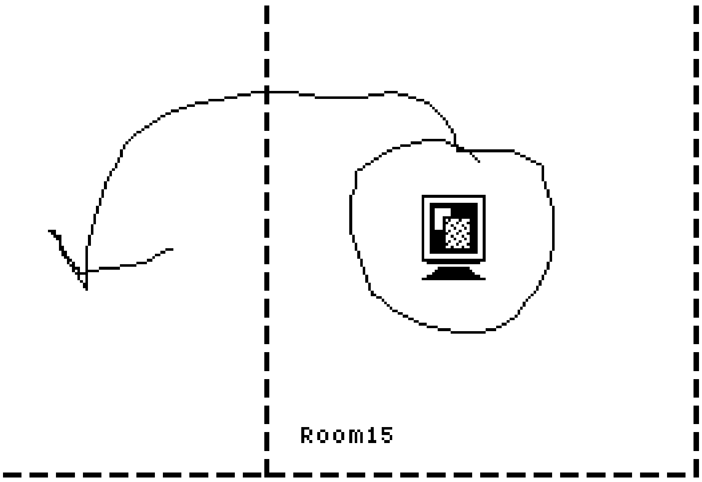


42 u (EM): What is the difference between the network cost and the budget?
43 s (EM): The cost is greater than the budget by 968 STERLING
44 u (GR): selects a horizontal shaft
45 u (EM): Is using-thin-cable possible in this shaft?
46 s (EM): Yes. Using thin cable is possible in any shaft. Thick cable may not be used in horizontal shafts.
46.1 u (GS): help <object>

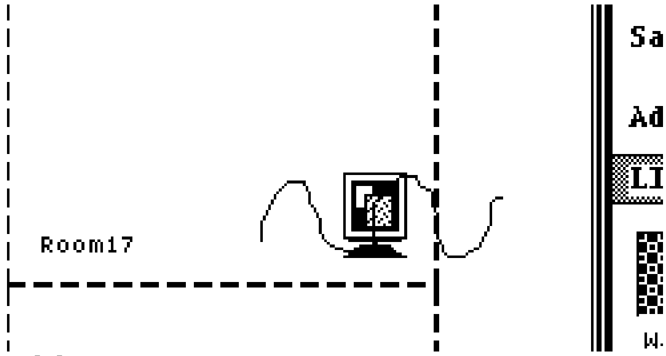


46.2 s (EM): cmr_X_Terminal_2 is a type of X_Terminal. X_Terminal is a type of computer.

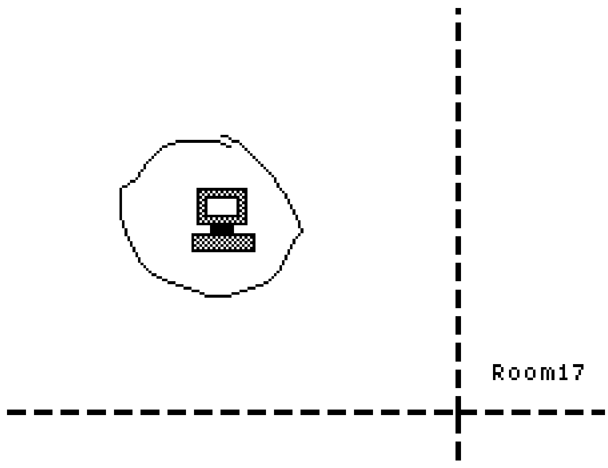
46.3 u (GS) move



46.4 s (EM): ok <object moves>
46.5 u (GS) delete <object>



46.6 s (EM): ok <object deletes>
46.7 u (GS) select <object>



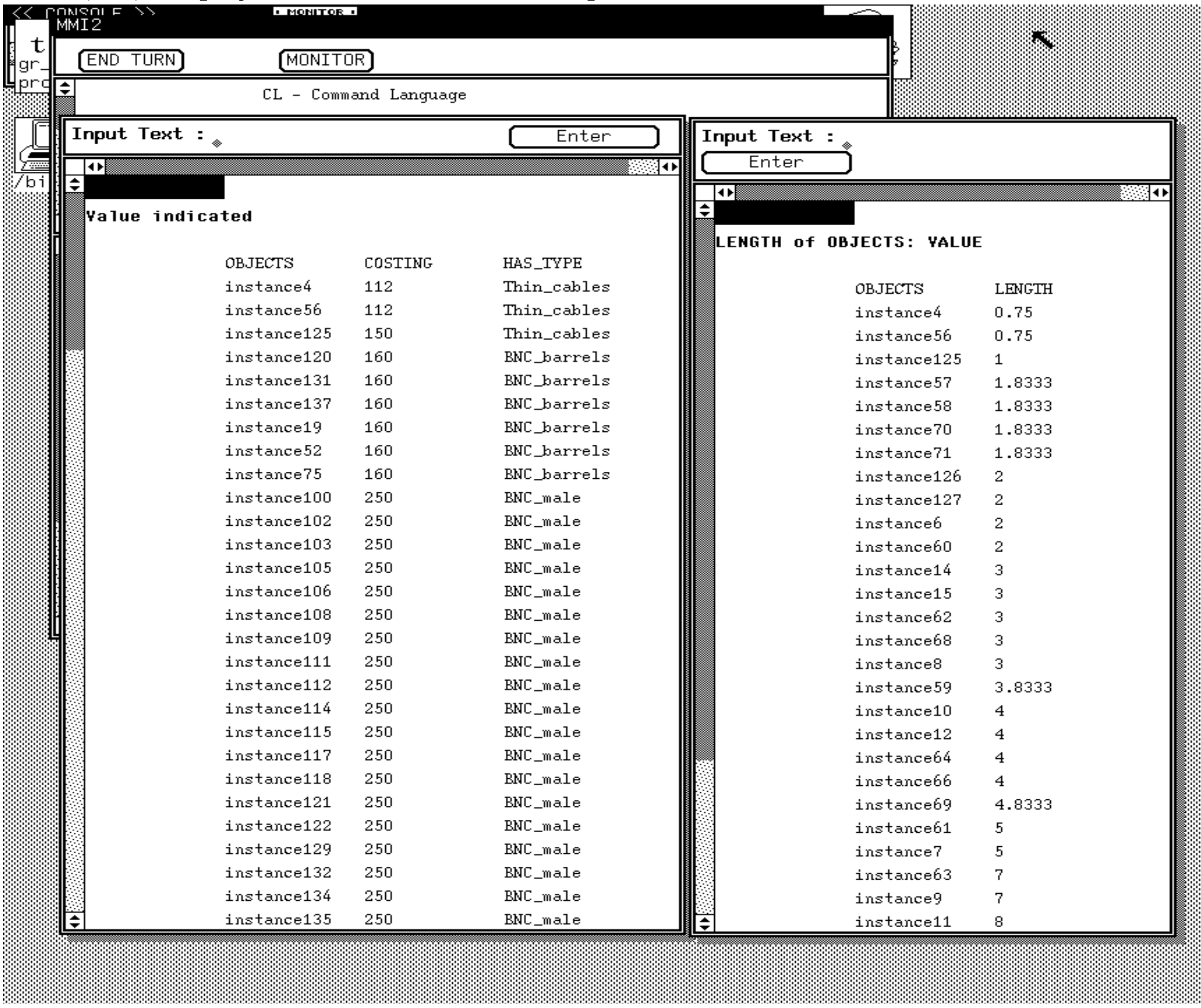
46.8 s (EM): ok
46.9 u (EM): What does this machine cost?
46.10 s (EM): 2680 STERLING
47 u (EM): Add a workstation to the network.
48 s (EM): I am sorry.
49 s (EM): The location and type of the workstation cmr_WORKSTATION_98 are underspec-

ified
50 s (EM): adding a workstation requires specification of location and type
51 s (EM): What is the type of the workstation cmr_WORKSTATION_98?
52 u (EM): Apollo
53 s (EM): The type of workstation is unknown.
54 s (EM): What is the type of the workstation cmr_WORKSTATION_98?
55 u (CL): help WORKSTATION
56 s (EM): A workstation is a subtype of computer, and a supertype of apollo and hp_station and microvax_station and sun.
57 u (CL): help SPARCS_SLC
58 s (EM): A sparcs_slc is a subtype of sparcs.
60 s (EM): What is the type of the workstation?
61 u (EM): Sparc_SLC
62 s (EM): ok
63 s (EM): What is the location of the workstation?
64 u (GR): selects Room1
66 s (EM): ok <adds workstation to network> <displays machine>
67 u (EM): Add a workstation to the network.
68 s (EM): Sorry.
69 s (EM): The location and type of the workstation cmr_WORKSTATION_99 are underspecified.
71 s (EM): What is the type of the cmr_WORKSTATION_99?
72 u (EM): Sparc1_Plus
73 s (EM): ok
74 s (EM): What is the location of the workstation?
75 u (GR): selects Room1
77 s (EM): ok <adds workstation to network> <displays machine>
78 u (EM): Add a workstation to the network.
78.1 s (EM): This is the third time that you have made the same mistake.
79 s (EM): You must specify the type of workstation and location!
79.1 s (AM): <beep>
80 u (EM): Add a Sparc1_Plus in cmr_Rooms_18.
81 s (EM): ok <adds workstation to network> <displays machine>
82 u (GR): adds a disk to the Sparc1_Plus <displays disk machine>
83 s (EM): ok
84 u (EM): cmr_SPARCS_1PLUS_91 is a server.
85 s (EM): ok
86 u (EM): What are the computers on the left of the server?
87 s (EM): <list of computers>
87.1 s (GR): <highlight of computers>
88 u (CL): move cmr_Sparcs_SLC2 cmr_Rooms_9
89 s (EM): ok <graphics moves machine>
89.1 u (EM): Add a Sparc_SLC that has a small disk to cmr_Rooms_18.
89.2 s (EM): ok
89.3 u (GR): <delete machine just created>
89.4 s (EM): ok
90 u (EM): move the machine that has a disk to cmr_Rooms_9
91 s (EM): ok <graphics moves machine>
92 u (GR): adds a workstation to a room <graphic addition>

93 s (EM): ok
94 u (GR): changes a cable from a specific class to a general one
95 s (EM): ok
96 u (EM): What does the network cost?
97 s (GR): highlight cable
98 s (EM): What is the type of this cable?
99 u (EM): thin cable
100 s (EM): ok
101 s (EM): 16947 STERLING
101.1 u (EM): Where is the machine that has a disk?
101.2 s (EM): room
101.3 s (GR): <highlights room>
102 u (GR): selects computer that has a disk
103 s (EM): ok
104 u (EM): what does the disk cost?
105 s (EM): 2345 STERLING
106 u (EM): Where is cmr_Sparcs_SLC_1?
107 s (GR): highlight cmr_Sparcs_SLC_1
108 s (EM): Room2
109 u (CL): delete cmr_Sparcs_SLC_1
110 s (EM): ok <graphic deletion>
116.3 u (EM): Which machines have no disks?
116.4 s (EM): Sparcs_SLC1,Sparcs_SLC2,Sparcs_SLC3,Sparcs_SLC4,
Terminals_X1,Terminals_X2,cmr_SPARCS_1PLUS_0,cmr_SPARCS_1PLUS_1
116.7 u (EM): Add a small disk to every machine.
116.8 s (EM): ok <many graphic additions and links>
111 u (GR): deletes link between a machine and a plug
112 s (EM): ok <graphic deletion of link>
113 u (GR): deletes a box <graphic deletion>
114 s (EM): ok
115 u (GR): deletes a server <graphic deletion>
116 s (EM): ok
116.9 u (EM): What is the average cost of computers on the network?
116.91 s (EM): 2114 STERLING
116.92 u (EM): What is the sum of the cost of computers on the network?
116.93 s (EM): 26345 STERLING
116.94 u (EM): What is the maximum cost of a computer on the network?
116.95 s (EM): 5114 STERLING
119 u (EM): What is the extensibility of the network?
120 s (EM): The network can be extended by adding on Ethernet_cards. You can extended the cable segment segment_id3, segment_id2, segment_id1.
121 u (CL): name hrc
122 s (EM): ok
123 u (EM): What is the extensibility of the network?
124 s (EM): The network can be extended by adding on Ethernet_cards.
The network can be extended by connecting instance155, instance156, instance16,instance21, instance24, instance27,instance30, instance33, instance36, instance39, instance42, instance45, instance115, instance118, instance121, instance124, instance130, instance133, instance136, instance139, instance97, instance98 through empty outputs

You can extended the cable segment segment_id3, segment_id2, segment_id1.
The network network_id3 can be extended according to:
The subnetwork can be extended by 4 segments.
The subnetwork can be extended by 2 segments with connections (not links).
The network network_id2 can be extended according to:
The subnetwork can be extended by 4 segments.
The subnetwork can be extended by 2 segments with connections (not links).
The network network_id1 can be extended according to:
The subnetwork can be extended by 4 segments.
The subnetwork can be extended by 2 segments with connections (not links).
125 u (CL): name mdw
126 s (EM): ok
127 u (EM): client-server problems of the network?
128 s (EM): The diskless machine Sparcs_SLC1 has no server. The diskless machine Sparcs_SLC2 has no server. The diskless machine Sparcs_SLC3 has no server. The diskless machine Sparcs_SLC4 has no server. The diskless machine Terminals_X1 has no server. The diskless machine Terminals_X2 has no server.
129 u (EM): What are the departmentalization problems of the network?
130 s (EM): There is no departmentalizing problem
117 u (EM): What is the connectivity of the network?
118 s (EM): The network component instance56 is connected to elements
to which it can not be connected. It has the ports: AUIF2_2
and the components connected to them are: instance91
131 u (EM): What is the name of the network?
132 s (EM): Network1
133 u (CL): save Network1
134 s (EM): ok
135 u (CL): report Network1

136 s (GR): displays table of all network components and their costs.



137 u (EM): bye

138 s (GR): <windows close and process terminates, saving an updated user model>

5.3 First Spanish Demonstration Script

This is a translation of the first test script presented in the previous section used for demonstrating the Spanish mode.

1 u (CL): name juan
2 s (SM): de acuerdo
3 u (SM): pon 2 terminales X en una red
4 s (SM): de acuerdo
5 u (SM): la red tendra 4 sparcs SLC
6 s (SM): de acuerdo
7 u (SM): coloca la red en un edificio
8 s (GR): puts up tool
9 s (SM): Cuales son las especificaciones del edificio?
10 u (GR): draws a building big enough to have a box. Hits done
10.1 s (SM): cual es el valor de presupuesto de la red?
10.2 u (SM): 200000 pesetas
10.21 u (SM): de acuerdo
10.3 s (SM): Cual es el coste medio de instalacion de cable en la red?
10.4 u (SM): 5000 pesetas
10.41 s (SM): de acuerdo
11 s (SM): Cual es el valor de ubication para el objeto Sparcs_SLC1?
12 u (SM): en Room1
13 s (SM): de acuerdo <displays machine>
15 s (SM): Cual es el valor de ubication para el objeto Sparcs_SLC2?
16 u (GR): selects room
17 s (SM): de acuerdo <displays machine>
19 s (SM): Cual es el valor de ubication para el objeto Sparcs_SLC3?
20 u (GR): selects room
21 s (SM): de acuerdo <displays machine>
23 s (SM): Cual es el valor de ubication para el objeto Sparcs_SLC4?
24 u (GR): selects room
25 s (SM): de acuerdo <displays machine>
27 s (SM): Cual es el valor de ubication para el objeto X_Terminal1?
28 u (GR): selects room
29 s (SM): de acuerdo <displays machine>
31 s (SM): Cual es el valor de ubication para el objeto X_Terminal2?
32 s (SM): En cmr_Rooms_18
33 s (SM): de acuerdo <displays machine>
35 s (SM): de acuerdo <displays network>
36 u (SM): cuanto cuestan los ordenadores de la red ?
37 s (GR): pie chart
38 u (SM): muestra un grafico de barras del coste de los ordenadores
39 s (SM): de acuerdo <displays graph>
40 u (SM): que maquinas hay en las salas ?
41 s (GR): table
42 u (SM): Cual es la diferencia entre el presupuesto y el coste de la red ?
43 s (SM): El coste es mayor al presupuesto en 20000 pesetas
44 u (GR): selects horizontal shaft

45 u (SM): se puede usar cable delgado en esta conduccion ?
46 s (SM): Si. Es se puede usar cable delgado en cualquier conduccion. El cable grueso no se puede usar en conducciones horizontales.
46.1 u (GR): help <object>
46.2 s (SM): <answer>
46.3 u (GS): move
46.4 s (SM): de acuerdo <object moves>
46.5 u (GS): delete <object>
46.7 u (GS): select <object>
46.8 s (SM): de acuerdo
46.9 u (CL): que #
46.91 s (SM): <answer>
47 u (SM): an1ade una estacion de trabajo a la red
48 s (SM): lo siento
49 s (SM): la ubicacion y el tipo de la estacion de trabajo cmr_WORKSTATION_98 estan subespecificados
50 s (SM): an1adir una estacion de trabajo requiere ubicacion y tipo
51 s (SM): cual es el tipo de la estacion de trabajo cmr_WORKSTATION_98 ?
52 u (SM): un apollo
53 s (SM): el tipo de la estacion de trabajo es desconocido
54 s (SM): cual es el tipo de la estacion de trabajo cmr_WORKSTATION_98 ?
55 u (CL): help WORKSTATION
56 s (SM): una estacion de trabajo es un subtipo de ordenador y un supertipo de apollo y hp_-station y microvax_station y sun
57 u (CL): help SPARCS_SLC
58 s (SM): un sparcs SLC es un subtipo de sparcs
60 s (SM): cual es valor de tipo de la estacion de trabajo ?
61 u (SM): un sparcs SLC
62 s (SM): de acuerdo
63 s (SM): cual es valor de ubicacion de la estacion de trabajo ?
64 u (GR): selects Room1
66 s (SM): de acuerdo <adds workstation to the network><displays machine>
67 u (SM): an1ade una estacion de trabajo a la red
68 s (SM): lo siento
69 s (SM): la ubicacion y el tipo de la estacion de trabajo cmr_WORKSTATION_98 estan subespecificados
71 s (SM): cual es el tipo de la estacion de trabajo cmr_WORKSTATION_99?
72 u (SM): un sparcs1_plus
73 s (SM): de acuerdo
74 s (SM): cual es la ubicacion de la estacion de trabajo?
75 u (GR): selects Room1
77 s (SM): de acuerdo <adds workstation to the network><displays machine>
78 u (SM): an1ade una estacion de trabajo a la red
79.0 s (SM): Es la tercera vez que hace el mismo error
79 s (SM): an1adir una estacion de trabajo requiere la especificacion de ubicacion y tipo.
79.1 s (AM): <beep>
80 u (SM): an1ade un sparcs1_plus en cmr_Rooms_9
81 s (SM): de acuerdo <adds workstation to the network><displays machine>
82 u (GR): adds a disk to the Sparcs1_plus <displays disk machine>

83 s (SM): de acuerdo
84 u (SM): cmr_SPARCS_SLC_3 es un server
85 s (SM): de acuerdo
->86 u (SM): cual es el tipo de la caja que esta a la izquierda del server?
->86 u (SM): cuales son los tipos de las cajas que estan a la izquierda del server?
87 s (SM): un RETIX_2265
88 u (CL): move Sparcs_SLC2 Room3
89 s (SM): de acuerdo <graphics moves machine>
90 u (SM): lleva la maquina que tiene disco a cmr_Rooms_18
91 s (SM): de acuerdo <graphics moves machine>
92 u (GR): adds a workstation to a room <graphic addition>
93 s (SM): de acuerdo
102 u (GR): selects a computer that has a specific disk
103 s (SM): de acuerdo
104 u (SM): Cuanto cuesta el disco ?
105 s (SM): 500000 pesetas
106 u (SM): donde esta cmr_SPARCS_SLC_3 ?
107 s (GR): highlights the room
108 s (SM): Room2
109 u (CL): delete Sparcs_SLC1
110 s (SM): de acuerdo <graphic deletion>
111 u (GR): deletes a link between a machine and a plug
112 s (SM): de acuerdo <graphic deletion of link>
113 u (GR): deletes a box <graphic deletion>
114 s (SM): de acuerdo
115 u (GR): deletes a server <graphic deletion>
116 s (SM): de acuerdo
116.1 u (SM): an1ade a cmr_Rooms_9 1 sparcs SLC que tenga disco
116.2 s (SM): de acuerdo <graphic adition of two machine and a disk>
116.3 u (SM): que maquinas no tienen disco?
116.4 s (SM): <answer>
116.7 u (SM): an1ade un disco a todas las maquinas
116.8 u (SM): de acuerdo <many graphic additions and links>
117 u (SM): cual es la conectividad de la red?
118 s (SM): <answer>
119 u (SM): Cual es la extensibilidad de la red?
120 s (SM): <answer>
122 s (SM): de acuerdo
123 u (SM): Cual es la extensibilidad de la red?
124 s (SM): <verbose answer>
126 s (SM): de acuerdo
121 u (CL): name marta
127 u (SM): hay problemas cliente-server?
128 s (SM): <answer>
125 u (CL): name juan
129 u (SM): que problemas de departamentalizacion hay?
130 s (SM): <answer>
131 u (SM): cual es el nombre de la red?
132 s (SM): Network1

133 u (CL): save Network1
134 s (SM): de acuerdo

5.4 First French Demonstration Script

This is a translation of the test script used for demonstrating the French mode. System answers are in English (EM) as the development of the french generator had only just started at the time of demonstration of the first demonstrator.

1 u (CL): name beatrice
2 s (EM): ok
3 u (FM): Mettez deux terminaux_X sur un re'seau.
4 s (EM): ok
5 u (FM): Le re'seau a quatre Sparc_SLC.
6 s (EM): ok
7 u (FM): Mettez le re'seau dans un ba^timent.
8 s (GR): puts up tool
9 s (EM): What are the specifications of the building?
10 u (GR): draws building big enough to have a box, hits Done
10.1 s (EM): What is the budget of the network?
10.2 u (FM): 1000 FFR.
10.21 s (EM): ok
10.3 s (EM): What is the average cost of installing cables on the network?
10.4 u (FM): 100 FFR.
10.41 s (EM): ok
11 s (EM): Where is Sparc_SLC1?
12 u (FM): Room2.
13 s (EM): ok <displays machine>
15 s (EM): Where is Sparc_SLC2?
16 u (GR): selects Room1
17 s (EM): ok <displays machine>
19 s (EM): Where is Sparc_SLC3?
20 u (GR): selects Room1
21 s (EM): ok <displays machine>
23 s (EM): Where is Sparc_SLC4?
24 u (GR): selects Room1
25 s (EM): ok <displays machine>
27 s (EM): Where is X_Terminal1?
28 u (GR): selects Room2
29 s (EM): ok <displays machine>
31 s (EM): Where is X_Terminal2?
32 u (FM): Room2.
33 s (EM): ok <displays machine>
35 s (EM): ok <displays network>
36 u (FM): Combien cou^tent les ordinateurs du re'seau?
37 s (GR): pie chart
38 u (FM): Montrez une barchart du prix des ordinateurs.
39 s (EM): ok <displays graph>
40 u (FM): Quelles machines sont dans quelles salles?
41 s (GR): table

42 u (FM): Quelle est la diffe'rence entre le cou^t du re'seau et le budget?
43 s (EM): The cost is greater than the budget by 986 Sterling
44 u (GR): selects a horizontal shaft
45 u (FM): Est-ce qu'il est possible d'utiliser du ca^ble thin dans cette gaine?
46 s (EM): Yes. Using thin cable is possible in any shaft. Thick cable may not be used in horizontal shafts.
46.1 u (GS): help <object>
46.2 s (EM): <answer>
46.3 u (GR): move
46.4 s (EM): ok <object moves>
46.5 u (GS): delete <object>
46.6 s (EM): ok <object deletes>
46.7 u (GS): select <object>
46.8.s (EM): ok
46.9 u (CL): what #
46.91 s (EM): <answer>
47 u (FM): Ajoutez une station au re'seau.
48 s (EM): I am sorry.
49 s (EM): The location and type of the workstation cmr_WORKSTATION_98 are underspecified.
50 s (EM): Adding a workstation requires specification of type and location.
51 s (EM): What is the type of the workstation cmr_WORKSTATION_98?
52 u (FM): Apollo.
53 s (EM): The type of workstation is unknown.
54 s (EM): What is the type of the workstation cmr_WORKSTATION_98?
55 u (CM): help WORKSTATION
56 s (EM): <workstation types>
57 u (CM): help SPARCS_SLC
58 s (EM): a sparcs_slc is a subtype of sparcs
60 s (EM): What is the type of the workstation cmr_WORKSTATION_98?
61 u (FM): Sparc_SLC.
62 s (EM): ok
63 s (EM): What is the location of the workstation?
64 u (GR): selects Room3
66 s (EM): ok <adds workstation to network> <displays machine>
67 u (FM): Ajoutez une station au re'seau.
68 s (EM): I am sorry.
69 s (EM): The location and type of the workstation cmr_WORKSTATION_99 are underspecified.
70 s (EM): Adding a workstation requires specification of type and location.
71 s (EM): What is the type of the workstation cmr_WORKSTATION_99?
72 u (FM): Sparc1_Plus.
73 s (EM): ok
74 s (EM): What is the location of the workstation?
75 u (GR): selects Room3
77 s (EM): ok <adds workstation to network> <displays machine>
78 u (FM): Ajoutez une station au re'seau.
79 s (EM): You must specify the type of workstation and location!
79.1 s (AM): <beep>

80 u (FM): Ajoutez une Sparc1_Plus dans Room3.
81 s (EM): ok <adds workstation to network> <displays machine>
82 u (GR): adds a disk to the Sparc1_Plus
83 s (EM): ok
84 u (FM): Sparcs_SLC4 est un serveur.
85 s (EM): ok
86 u (FM): Quel est le type de la boi^te a`_gauche_de le serveur?
87 s (EM): Retix5565
88 u (CL): move Sparcs_SLC2 Room3
89 s (EM): ok <graphics moves machine>
89.1 u (FM): Ajoutez une Sparc_SLC avec un petit disque dans Room18.
89.2 s (EM): ok
89.3 u (GR): <delete the machine just created>
89.4 s (EM): ok
90 u (FM): De'placez la machine avec un disque vers Room1.
91 s (EM): ok <graphics moves machine>
92 u (GR): adds a workstation to a room
93 s (EM): ok
94 u (GR) changes a cable for a specific class to a generic one
95 s (EM): ok
96 u (FM): Combien cou^te le re'seau?
97 s (GR): highlight cable
98 s (EM): What is the type of this cable?
99 u (FM): Un ca^ble thin.
100 s (EM): ok
101 s (EM): <answer in user's currency>
101.1 u (FM): Ou` est la machine avec le disque?
101.2 s (EM): <room>
101.3 s (GR): <Highlights Room>
102 u (GR) selects computer that has a disk
103 s (EM): ok
104 u (FM): Combien cou^te le disque?
105 s (EM): <answer in user's currency>
106 u (FM): Ou` est Sparcs_SLC1?
107 s (GR): highlights Sparcs_SLC1
108 s (EM): Room2
109 u (CL): delete Sparcs_SLC1
110 s (EM): ok <graphic deletion>
111 u (GR): deletes link between a machine and a plug
112 s (EM): ok <graphic deletion of link>
113 u (GR): deletes a box
114 s (EM): ok <graphic deletion of box>
115 u (GR): deletes a server
116 s (EM): ok
116.1 u (FM): Ajoutez deux Sparc_SLC avec un disque dans Room1.
116.2 s (EM): ok <graphic addition of 2 machines and disks>
116.3 u (FM): Quelles machines n'ont pas de disque?
116.4 s (EM): <machines>
116.7 u (FM): Ajoutez un disque a` chaque machine.

116.8 s (EM): ok <many graphic additions and links>`
119.9 u (FM): Quel est le prix moyen des ordinateurs du re'seau?
116.91 s (EM): 2114 Sterling
116.92 u (FM): Quel est le cou^t total des ordinateurs du re'seau?
116.93 s (EM): 26345 Sterling
116.94 u (FM): Quel est le prix maximum des ordinateurs du re'seau?
116.95 s (EM): 5114 Sterling
117 u (FM): Quelle est la connectivite' du re'seau?
118 s (EM): <answer>
119 u (FM): Quelle est l'extensibilite' du re'seau?
120 s (EM): <answer for this type of user>
121 u (CL): name
122 s (EM): ok
123 u (FM): Quelle est l'extensibilite' du re'seau?
124 s (EM): <answer for this type of user>
125 u (CL): name
126 s (EM): ok
127 u (FM): Proble`mes client_serveur?
128 s (EM): <answer>
129 u (FM): Quels sont les proble`mes de de'partementalisation?
130 s (EM): <answer>
131 u (FM): Quel est le nom du re'seau?
132 s (EM): Network1
133 u (CL): save Network1
134 s (EM): ok
135 u (CL): report Network1
136 s (GR): <displays table of all network components and their costs.

Remarks:

The following sentences cannot be processed normally by the french parser (they have been hard-wired for demonstration purposes):

10.2 and 10.4 : the french analyser doesn't handle numbers

38, 40 and 89.1 : fail due to bugs roughly traced but not yet repaired

42: the parser doesn't handle the "difference entre A et B" construction as case-frame resolution for nouns has not yet been implemented

45: the parser does handle infinitive constructions nor modality

116.xx : the analyser handles only simple cases of quantification and no aggregate operators

5.5 The script for the second MMI^2 demonstrator

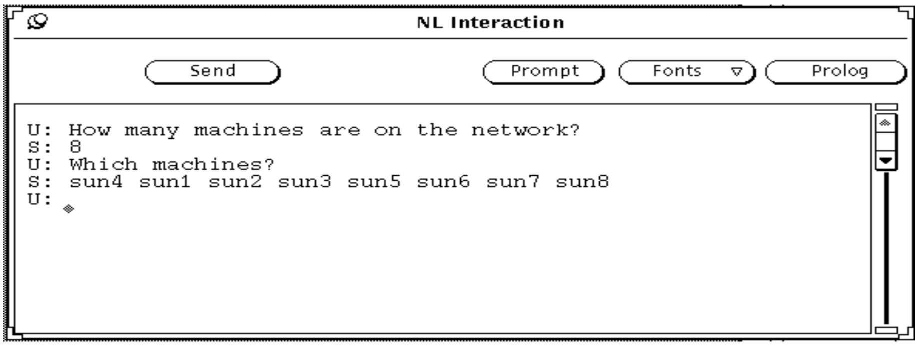
Move1

(1a) U(NL): How many machines are on the network?
(1b) S(NL): 8

Move2

(2a) U(NL): Which machines?
(2b) S(NL): sun4 sun1 sun2 sun3 sun5 sun6 sun7 sun8

Figure 1

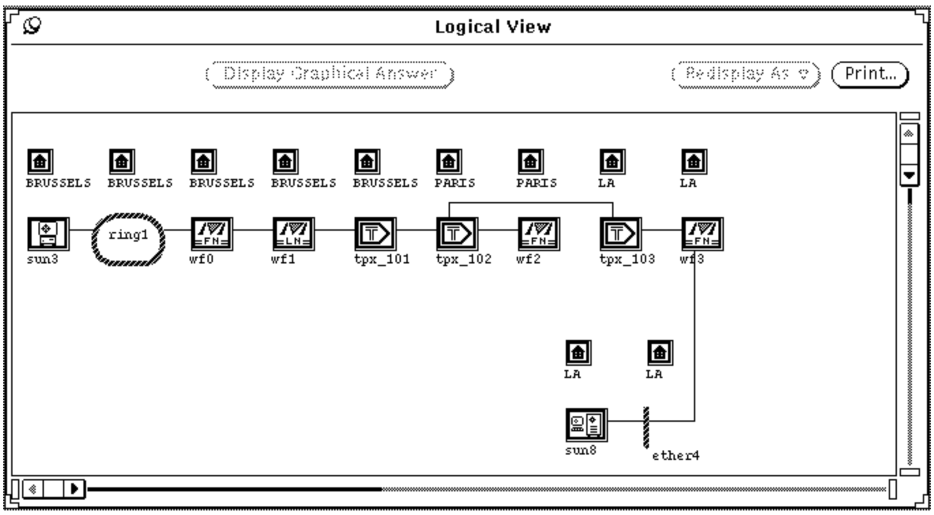


(2c) S(GR): <displays icons representing machines>

Move3

(3a) U(NL): Is sun3 connected to sun8?
(3b) S(NL): Yes
(3c) S(GR): <displays connection path>

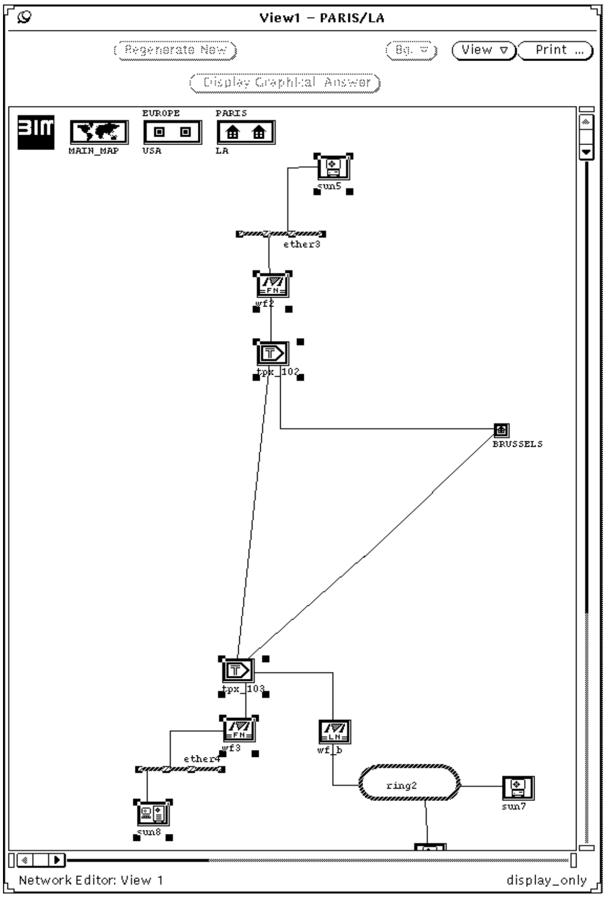
Figure 2



Move4

(4a) U(NL): Is sun5 connected to sun8?
(4b) S(NL): Yes
(4c) S(GR): <opens double view Paris-LA with connection path highlighted>

Figure 3



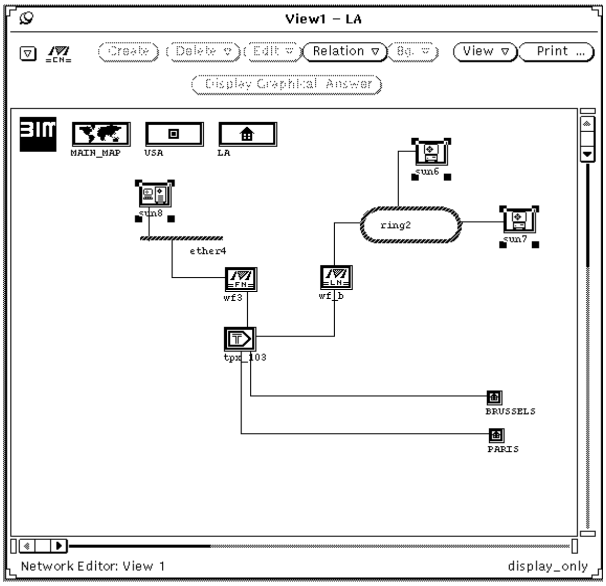
Move5

(5a) U(GR): <click on LA icon in LV window>
(5b) S(GR): <opens LA View1>

Move6

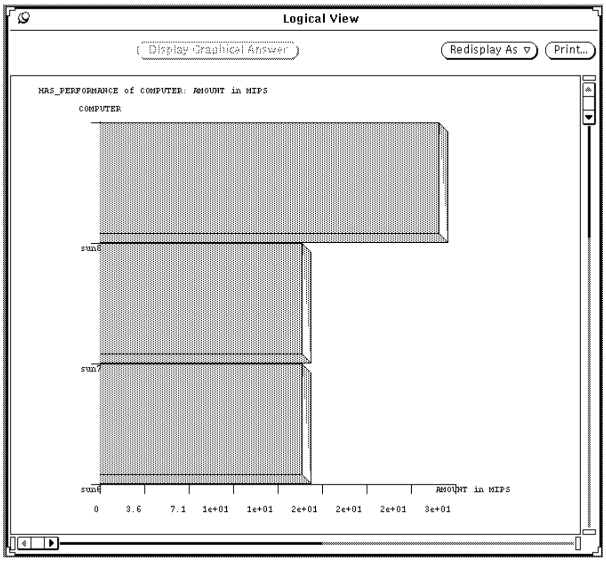
(6a) U(GR): <click on sun6, sun7 and sun8 in LA View1>

Figure 4



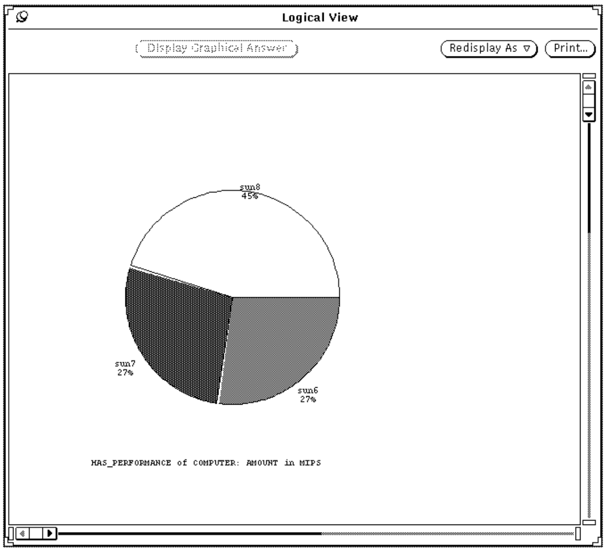
(6b) U(NL): Performance of these machines?
(6c) S(NL): OK
(6d) S(GR): <displays bar chart>

Figure 5



(6e) U(GR): <redisplay as pie chart>
(6f) S(GR): <displays pie chart>

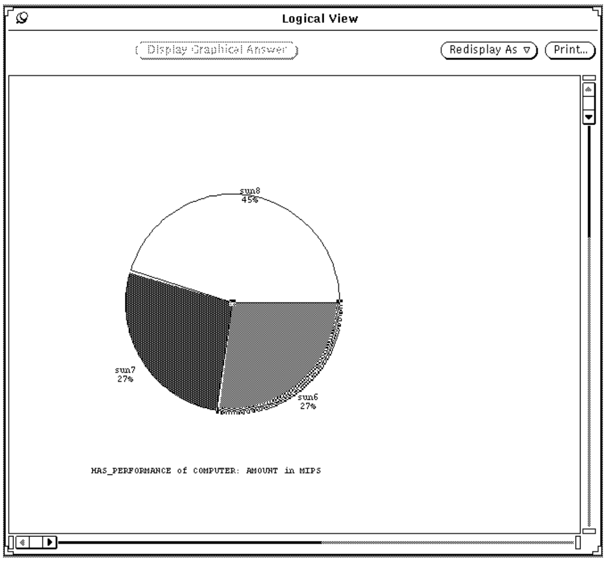
Figure 6



Move7

(7a) U(GR): <click on sun6>

Figure 7



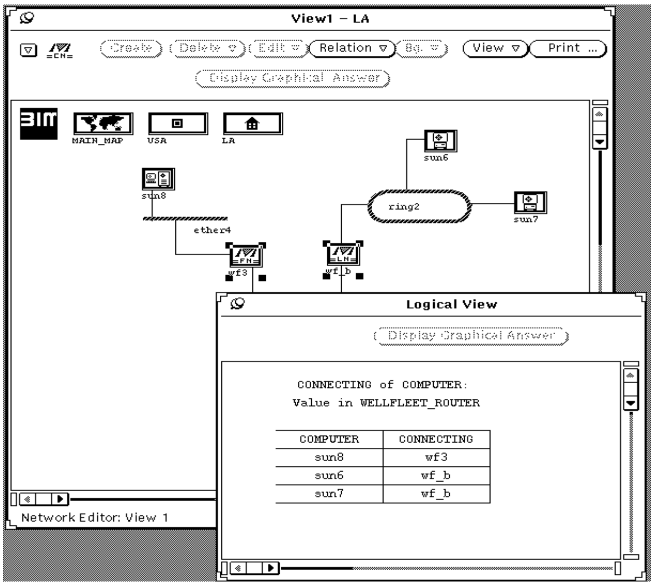
(7b) U(NL): Where is it?
(7c) S(NL): OK
(7d) S(NL): LA
(7e) S(GR): <fronts LA View1>

Move8

(8a) U(GR): <click on wf3 and wf_b in LA View1>
(8b) U(NL): Which machines are connected to these routers?

(8c) S(NL): OK
(8d) S(GR): <displays a table>

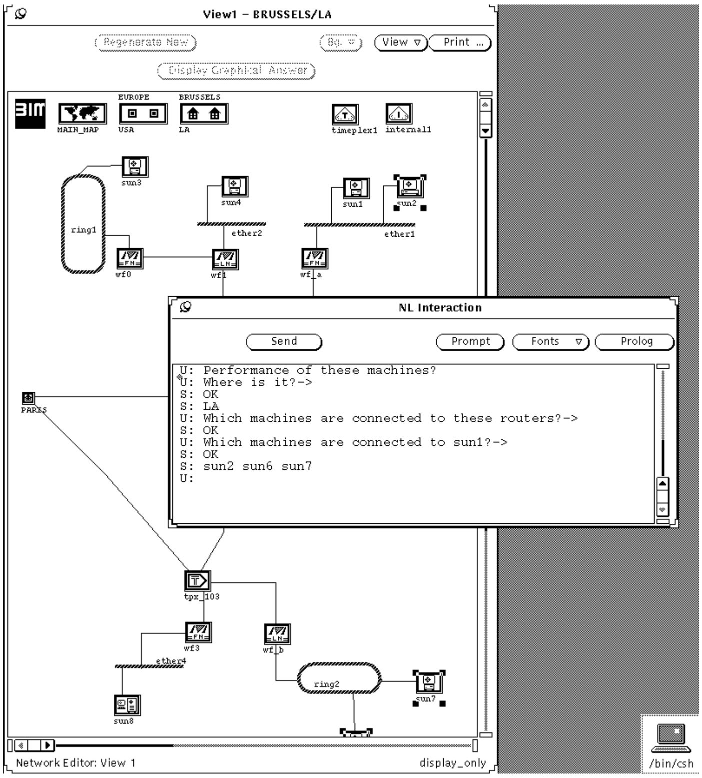
Figure 8



Move9

(9a) U(NL): Which machines are connected to sun1?
(9b) S(NL): sun4 sun3 sun8
(9c) S(GR): <opens double view Brussels-LA with machines highlighted>

Figure 9



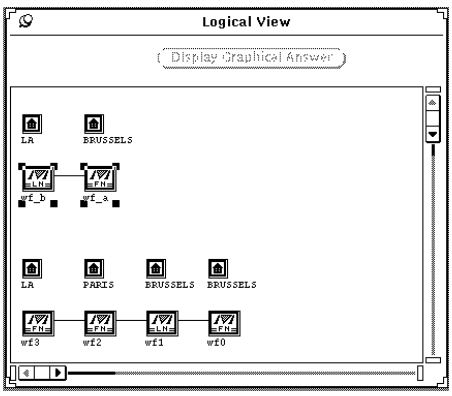
Move10

(10a) U(NL): Subnetwork of routers?
(10b) S(GR): <displays subnetwork>

Move11

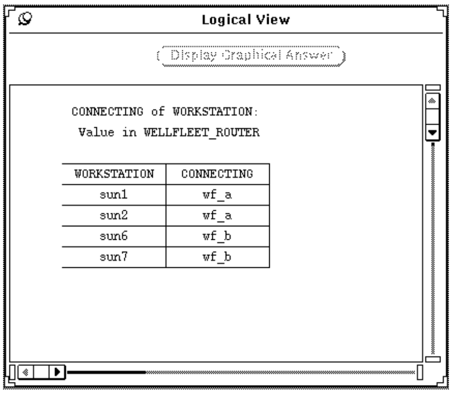
(11a) U(GR): <click on wf_b and wf_a>

Figure 10



(11b) U(NL): Which machines are connected to them?
(11c) S(NL): OK
(11d) S(GR): <displays a table>

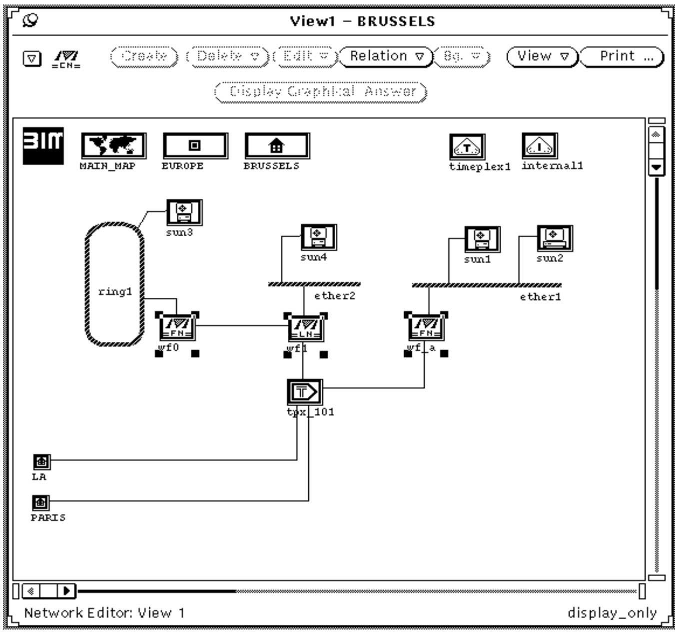
Figure 11



Move12

(12a) U(NL): Average performance of machines?
(12b) S(NL): 18 MIPS
(12c) U(NL): Brussels
(12d) S(NL): OK
(12e) U(NL): the router?
(12f) S(NL): wf1 wf0 wf_a
(12g) S(GR): <opens view Brussels with routers highlighted>

Figure 12



(12h) U(GR): <clicks on brussels history icon>
(12i) U(NL): Sum of performances of its machines?
(12j) S(NL): OK
(12k) S(NL): 72 MIPS

5.6 Features illustrated in the First Test Script

In this section, features of the script are described under three headings: face validity in the task domain, multi-modal interaction, and co-operative dialogue to illustrate how individual features support these three objectives of the project.

FACE VALIDITY IN THE TASK DOMAIN

The structure of the overall script is designed to be credible in the task domain. The task is to design a computer network for a building. An example of a transcript of a Wizard of Oz session of two humans performing the task is provided in Appendix 4 so that a comparison can be made with the test script (see deliverables d14, d15, d43). From the task perspective that structure is:

Moves

1-35:	set network requirements
35:	design network and present design
36-46:	investigate design
47-116.8:	modify design
116.9-132:	investigate modified design
133-134:	save design
135-136:	get purchasing details for network

MULTI-MODAL INTERACTION

There is no obvious categorisation of the features which support or illustrate multi-modal interaction (see Falzon, 1991, Wilson & Conway, 1991). The simplest form is ostensive deixis where a graphical item is indicated (by pointing or highlighting) at the same time as a natural language utterance refers to it.

User Ostensive Deixis:

44 u (GR): selects a horizontal shaft
45 u (EM): Is using-thin-cable possible in this shaft?

System Ostensive Deixis:

97 s (GR): highlight cable
98 s (EM): What is the type of this cable?

One possible advance on this is the possibility for the user to select the mode in which to reply to a system question. Natural language answers are possible (e.g. move 10.2), but they are not always the most efficient:

15 s (EM): Where is Sparcs_SLC2?
16 u (GR): selects room

Similarly, when the system replies to user questions it must choose the most effective mode of presentation, and if that is graphics it must then use the most expressive and effective graphics mode representations for that user at that point in the task and dialogue:

36 u (EM): What do the computers on the network cost?
37 s (GR): pie chart
40 u (EM): Which machines are in which rooms?
41 s (GR): table

Beyond these simple forms of multi-modal interaction would come non-ostensive deixis. This requires the user to mention an object which is in the context, but which has not occurred in the natural language dialogue. It is not clear that any of these have been included in the script, but this may have been an oversight in its construction.

Another simple interaction of natural language and graphics is through the use in natural language of spatial terms which require the graphical representation to support the required reasoning:

86 u (EM): What are the computers on the left of the server?
87 s (EM): <list of computers>
87.1 s (GR): <highlight of computers>

At least two further examples of facilitated multi-modal interaction appear in the script, although these are hard to classify:

System output which includes presenting a graphics tool to allow users the best way of entering information and asking for that information in natural language:

7 u (EM): Put the network in a building.
8 s (GR): puts up tool
9 s (EM): What are the specifications of the building?
10 u (GR): draws building big enough to have a box, hits Done

Explicit request by the user for the answer to be presented in a specific graphical representation:

38 u (EM): Display a bar-graph of the cost of the computers.
39 s (EM): ok <displays graph>

CO-OPERATIVE DIALOGUE

User/System Initiative

The script contains sections where the user has the initiative, the system has the initiative, and where the initiatives are mixed in subdialogue structure:

1-7: User initiative
8-35: System initiative
36-47: User initiative
48-79.1: Mixed initiative

80-95: User initiative
96-101: Mixed initiative
102-136: User initiative

Opening/Closing Structure

The script includes an interaction between dialogue openings and task plans to support the opening period in which the user holds the initiative. Lines 1-7 are an opening to the overall dialogue. During this period the dialogue management system stores the user desires and does not respond to their content, only responds by stating “ok” as an encouragement for the user to continue. At line 7 the system has decided that it knows the major task which the user is trying to perform (that is: to design a network for a building), and takes over the initiative. This reflects the dialogue opening structure observed in the experimental studies of human expert network designers with clients who wished to design networks.

There is no explicit closing structure used by the dialogue manager, and it would trivialise the notion of a closing to use the term to apply to the user’s termination of a session, although it can be performed in natural language, command language or from a menu:

137 u (EM): bye
138 s (GR): <windows close and process terminates, saving an updated user model>

User Plan Recognition

Line 7 triggers the recognition of a task plan for designing a network for a building. This plan contains various prerequisite constraints which are part of the specification of the building. These prerequisites are partly met by the information initially expressed by the user in lines 1-6, and the remainder motivate the system led dialogue down to line 35. The task plan is then completed since a network is designed and displayed to the user.

Similarly lines 47 and 96 cause plans to be identified for the addition of machines and the calculation of a cost although these are in response to an explicit command and a question. In both cases a subdialogue results to complete un-met preconditions on the plan goal.

Argumentation Structure

Within system replies to user moves there is an argumentation structure:

47 u (EM): Add a workstation to the network.
48 s (EM): I am sorry.
49 s (EM): The location and type of the workstation cmr_WORKSTATION_98 are underspecified
50 s (EM): adding a workstation requires specification of location and type
51 s (EM): What is the type of the workstation cmr_WORKSTATION_98?

this example shows a structure of:

48: Apology
49: Problem report

50: Justification
51: System Request

which together become the answer/reply to the user command at 47.

Tailoring System Output to User Domain Knowledge

The system output is tailored to the knowledge the user has shown of the domain. For example, at line 119 the user asks about a property of the design. The reply at line 120 is tailored for this user's knowledge represented in the user model. In contrast, at line 121 a new user is set and that user's model is then drawn on to generate an appropriate, yet different, response to the same question. The reply at lines 123 to 124 show the different content to the same question for a different user. The rules for deriving the user model from the dialogue stream, the prototypes used, and the rules for applying knowledge of the user to generated output were all developed from the Wizard of Oz studies and interviews with experts and users (see deliverables d3, d7, d14, d15, d43).

Natural language output generated by the french generator (not used in this script) is tailored to both the user's domain knowledge and his knowledge of the french language. Reference and elision operations are avoided for users with low domain knowledge in order to have output as explicit as possible, and grouping operations are avoided for users with low knowledge of the french language in order to keep output simple.

Meeting Presuppositions

The simplest form of presupposition checking is to test whether objects referred to exist in the domain of discourse; this is performed by the dialogue controller so that if objects are referred to which do not exist the user is informed of this. For example, if a user asks for the cost of a non-existent object:

u (EM): What is the cost of cmr_Sparcs_789?
s (EM): There is no such object as cmr_Sparcs_789.

At present, no attempt is made to try to determine which object could have been intended by the user given the context in order to either continue processing with the most likely candidate or to present the user with a list of candidates to disambiguate.

A second class of presuppositions which are accounted for are those tied to predicates. The formal evaluation of the predicate for the difference between two amounts returns a value consisting of a number and unit for that number (e.g. number and currency unit). The Communication Planning Expert knows that users prefer a difference to be expressed with a direction between the two input values, therefore it tailors the output to meet this presupposition. For example:

42 u (EM): What is the difference between the network cost and the budget?
43 s (EM): The cost is greater than the budget by 968 STERLING

is produced rather than the formally produced answer:

42 u (EM): What is the difference between the network cost and the budget?

(EM): 968 STERLING

Sub-dialogue Types

It is common for dialogue systems to list types of subdialogue which they can produce with users. These lists often include such items as:

1) Disambiguation subdialogue
2) Topic Change
3) Clarification
4) Correction
5) Repair
6) Explanation.

MMI^2 does not explicitly represent subdialogue structure in the dialogue management system; rather, it is an emergent property of the other aspects of the dialogue management processes.

Since the dialogue structure in MMI^2 is that of a push down stack of system and user desires the dialogue structure can contain infinitely nested subdialogues. For example, a variation on lines 47-81 in the script could be to allow the user to nest requests for workstations to be added to the network:

U (EM): Add a workstation to Room1.
S (EM): I am sorry.
S (EM): The location and type of the workstation cmr_WORKSTATION_98 are underspecified
S (EM): adding a workstation requires specification of location and type
S (EM): What is the type of the workstation cmr_WORKSTATION_98?
U (EM): Add a workstation to Room1.
S (EM): I am sorry.
S (EM): The type of the workstation cmr_WORKSTATION_99 is underspecified
S (EM): adding a workstation requires specification of location and type
S (EM): What is the type of the workstation cmr_WORKSTATION_99?
U (EM): Add a workstation to Room1.
S (EM): I am sorry.
S (EM): The type of the workstation cmr_WORKSTATION_100 is underspecified
S (EM): adding a workstation requires specification of location and type
S (EM): What is the type of the workstation cmr_WORKSTATION_100?
S (EM): Sparc_SLC
S (GR): (EM): ok <adds workstation to network> <displays machine>
S (EM): What is the type of the workstation cmr_WORKSTATION_99?
S (EM): Sparc_SLC
S (GR): (EM): ok <adds workstation to network> <displays machine>
S (EM): What is the type of the workstation cmr_WORKSTATION_98?
S (EM): Sparc_SLC
S (GR): (EM): ok <adds workstation to network> <displays machine>

There are different classes of system answer to user questions which could be equated with this conventional approach to subdialogue structure:

Command Driven:
90 u (EM): move the machine that has a disk to Room1
91 s (EM): ok <graphics moves machine>

Information seeking (by system):

10.1 s (EM): What is the budget for the network?
10.2 u (EM): 1000 Sterling

Information seeking (by user):

104 u (EM): what does the disk cost?
105 s (EM): 2345 STERLING

A tutorial answer:

44 u (GR): selects a horizontal shaft
45 u (EM): Is using-thin-cable possible in this shaft?
46 s (EM): Yes. Using thin cable is possible in any shaft. Thick cable may not be used in horizontal shafts.

An unspecified task prerequisite answer:

47 u (EM): Add a workstation to the network.
48 s (EM): I am sorry.
49 s (EM): The location and type of the workstation cmr_WORKSTATION_98 are underspecified
50 s (EM): adding a workstation requires specification of location and type
51 s (EM): What is the type of the workstation cmr_WORKSTATION_98?

or:

96 u (EM): What does the network cost?
97 s (GR): highlight cable
98 s (EM): What is the type of this cable?
99 u (EM): thin cable
100 s (EM): ok
101 s (EM): 16947 STERLING

An inappropriate type (unknown) answer by the user to a system question:

51 s (EM): What is the type of the workstation cmr_WORKSTATION_98?
52 u (EM): Apollo
53 s (EM): The type of workstation is unknown.
54 s (EM): What is the type of the workstation cmr_WORKSTATION_98?

Ambiguities in the mode representation of the user input are represented as multiple CMR Expressions with a formula for each ambiguous interpretation within a single CMR Act, itself within a single CMR packet which represents the communication act by the user. These are then resolved at the level of dialogue management which selects a single CMR Expression to act upon. This currently curcumvents the need for disambiguation subdialogues although it will not suffice for all cases where some sub-dialogue will have to be introduced.

Failures at the mode level are handled by passing back a message as part of the CMR expression to the dialogue management. For example,

u: What is the meaning of my like?
s: Parsing failed.

u: Add a computer to the doormat?
s: The word "doormat" is unrecognised.

Where the input is well formed but the system fails to answer a query the system resorts to its final, least cooperative answer. For example, when there is no network, and the user asks:

u: What is the cost of the network?
s: The answer cannot be determined

Deontic Language

Users frequently use modals such as ’should’ and ’can’ in English. The CMR does not represent modality explicitly, nor does the application support modal reasoning. However, many times when modals are used they are redundant to an instruction. In order to support a more natural use of language than the formal aspects of the CMR, the natural language modes can absorb these modal operators to produce CMR which is purely declarative in force, and does not include modal formulæ. Through this mechanism line 5 of the script includes a deontic “should”:

5 u (EM): The network should have 4 Sparc_SLCs.

Epistemic Reasoning

The script does not demonstrate any advanced reasoning about what it knows, but one fragment addresses this issue:

51 s (EM): What is the type of the workstation cmr_WORKSTATION_98?
52 u (EM): Apollo
53 s (EM): The type of workstation is unknown.
54 s (EM): What is the type of the workstation cmr_WORKSTATION_98?
55 u (CL): help WORKSTATION
56 s (EM): A workstation is a subtype of computer, and a supertype of
apollo and hp_station and microvax_station and sun.
57 u (CL): help SPARCS_SLC
58 s (EM): A sparcs_slc is a subtype of sparcs.
60 s (EM): What is the type of the workstation?
61 u (EM): Sparc_SLC

Here the system is recoursing to its knowledge of its own intensional definitions of types and describing this to the user. An interesting contrast exists here in an early version of the Spanish script where lines 55 & 56 were replaced by:

55 u (SM) Cuales son los tipos de estaciones de trabajo?
56 s (EM): SPARCS_SLC
where the question is not “what types of workstation are there?” which is conveyed by “help WORKSTATION”, but “what are the types of the workstations?” which becomes a question about the types of workstations which have denotations in the model theory rather than about the types which exist in the intensional definitions, of which instances can be created.

Second order reasoning

The CMR used in MMI^2 is a first order logic. However, this has been extended to allow the use of certain second order relations:

116.9 u (EM): What is the average cost of computers on the network?
116.91 s (EM): 2114 STERLING
116.92 u (EM): What is the sum of the cost of computers on the network?
116.93 s (EM): 26345 STERLING
116.94 u (EM): What is the highest cost of a computer on the network?
116.95 s (EM): 5114 STERLING

Quantification

Within the approach taken to updating from the CMR both existential and universal quantification can be used, e.g.:

116.7 u (EM): Add a disk to every machine.
116.8 s (EM): ok <many graphic additions and links>

Negation in Queries

Similarly, negation can be used in queries (see the next section for an account of negation in updates as a contrast):

116.3 u (EM): Which machines have no disks?
116.4 s (EM): Sparcs_SLC1,Sparcs_SLC2,Sparcs_SLC3,Sparcs_SLC4,
Terminals_X1,Terminals_X2,cmr_SPARCS_1PLUS_0,cmr_SPARCS_1PLUS_1

Natural Language Shortcuts

When the system has been used by domain experts and users in initial informal trials, they have reported that they do not wish to type complete lengthy natural language queries. The Command Language is designed to offer an alternative to overcome this problem, but the English Mode also offers the facility for users to abbreviate natural language queries. For example, interactions about properties such as:

40 u (EM): Which machines are in which rooms?

41 s (GR): table

can be replaced by :

u (EM): Machine locations?
41 s (GR): table

or queries about the identity of existing objects can be contracted:

u (EM): What machines are on the network?
becomes:
u (EM): machines?

and

131 u (EM): What is the name of the network?
becomes:
u (EM): Networks?

5.7 Features illustrated in the second demonstration script

Information retrieval

One objective of the demonstration script is to illustrate the practical usefulness of an MMI2 type interface for information retrieval. Data which are more or less directly present in the application, are sometimes very cumbersome to retrieve using a classical graphical interface. Adding text and the possibility of mixing text and graphics greatly improves interface quality. The first 2 moves in the script illustrate how text interaction can be an enourmous timesaver as opposed to the graphical alternative for certain types of - simple - questions.

(1a) U(NL): How many machines are on the network?
(1b) S(NL): 8
(2a) U(NL): Which machines?
(2b) S(NL): sun4 sun1 sun2 sun3 sun5 sun6 sun7 sun8

The remainder of the script shows how information retrieval is also optimised by multimodal input and output, by techniques of appropriate response determination and by addition of rules on top of the database.

Multimodal interaction

The demonstration script shows the combination of text mode and graphics mode both for user input and for system output.

For user input, the main mechanism for mixing modes is anaphora resolution. The user selects one or more objects graphically, then refers to them with an English pronoun or definite description. Single and multiple clicks are accepted from all graphics windows, i.e. the 2 network tool windows (move 6, move 8), the chart tool window (move 7) and the logical view window (move 11).

(6a) U(GR): <click on sun6, sun7 and sun8 in LA View1>
(6b) U(NL): Performance of these machines?

(8a) U(GR): <click on wf3 and wf_b in LA View1>
(8b) U(NL): Which machines are connected to these routers?

(7a) U(GR): <click on sun6>
(7b) U(NL): Where is it?

(11a) U(GR): <click on wf_b and wf_a>
(11b) U(NL): Which machines are connected to them?

The multiple selection in move 6 is of a new type as it passes on the complete set of selected objects as a possible antecedent to the Context Expert, rather than introducing the selections one by one. The special type of anaphoric relation, where anaphor and antecedent are bound through the part-of relation, is illustrated in move 12.

(12c) U(NL): Brussels
(12d) S(NL): OK
(12e) U(NL): the router?

The response determination procedures from the CP were reused in the new demonstrator and are illustrated in the script. System output is either sent to the graphics mode, or to the text mode or (most often) is a combination of text and graphics:

- a yes/no question about connectivity with an affirmative reply and a graphics display of the connection path, either through highlighting (move 4) or through logical view generation (move 3).

(4a) U(NL): Is sun5 connected to sun8?
(4b) S(NL): Yes
(4c) S(GR): <opens double view Paris-LA with connection path highlighted>

(3a) U(NL): Is sun3 connected to sun8?
(3b) S(NL): Yes
(3c) S(GR): <displays connection path>

- a wh-question with the reply displayed either in the form of a chart (move 6, move 8), as a full network subtree (move 7), through highlighting the relevant objects in a network display (move 9, move 12) or by generating a logical view (move 10).

(6b) U(NL): Performance of these machines?
(6c) S(NL): OK
(6d) S(GR): <displays bar chart>

(8b) U(NL): Which machines are connected to these routers?
(8c) S(NL): OK
(8d) S(GR): <displays a table>

(7b) U(NL): Where is it?
(7c) S(NL): OK
(7d) S(NL): LA
(7e) S(GR): <fronts LA View1>

(9a) U(NL): Which machines are connected to sun1?
(9b) S(NL): sun4 sun3 sun8
(9c) S(GR): <opens double view Brussels-LA with machines highlighted>

(12e) U(NL): the router?
(12f) S(NL): wf1 wf0 wf_a
(12g) S(GR): <opens view Brussels with routers highlighted>

(10a) U(NL): Subnetwork of routers?
(10b) S(GR): <displays subnetwork>

If a graphical display is already on the screen and becomes relevant again at a later stage in the session, the system will simply front the window containing the display, rather than recreate it (move 7 above).

Natural Language interaction

The full functionality of the English Mode in the original MMI2 demonstrator was retained.

Some of its features are illustrated in the script, such as the analysis of second order predicates, the processing of possessive pronouns and the possibility to use shortcut English queries (move 12).

(12a) U(NL): Average performance of machines?
(12b) S(NL): 18 MIPS

(12i) U(NL): Sum of performances of its machines?
(12j) S(NL): OK
(12k) S(NL): 72 MIPS

Graphical interaction

The original interface of the network editor is still intact in its graphical information retrieval function. The user can navigate through the network tree by clicking on iconic objects and getting a display of the subtree dominated by that object (move 5).

(5a) U(GR): <click on LA icon in LV window>
(5b) S(GR): <opens LA View1>

Knowledge base interaction

All user system interactions with queries about connections between network objects (move 3, move 4, move 8, move 9) or about logical subnetworks (move 10) address the knowledge base built in an MMI2 fashion on top of the network database.

(3a) U(NL): Is sun3 connected to sun8?
(3b) S(NL): Yes
(3c) S(GR): <displays connection path>

(4a) U(NL): Is sun5 connected to sun8?
(4b) S(NL): Yes
(4c) S(GR): <opens double view Paris-LA with connection path highlighted>

(8b) U(NL): Which machines are connected to these routers?
(8c) S(NL): OK
(8d) S(GR): <displays a table>

(9a) U(NL): Which machines are connected to sun1?
(9b) S(NL): sun4 sun3 sun8
(9c) S(GR): <opens double view Brussels-LA with machines highlighted>

(10a) U(NL): Subnetwork of routers?
(10b) S(GR): <displays subnetwork>

6 References

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

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

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.

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

Cohen, P.R. (1992) The role of natural language in a multimodal interface. UIST '92, 143-149. ACM Press: New York.

Coutaz, J., Nigay, L., & Salber, D. (1993) The MSM framework: A design Space for Multi-Sensori-Motor Systems. In Proceedings of EWHCI '93, East West Human Computer Interction Conference, Moscow, August 1993.

Dahlback, N., Jonsson, 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.

Duce, D.A., Gomes, M.R., Hopgood, F.R.A. and Lee, J.R. (1991) *User Interface Management and Design*, Eurographic Seminar Series, Springer-Verlag: Berlin.

Feiner, S. & McKeown, K. (1991) Automating the Generation of Co-ordinated Multimedia Explanations. IEEE Computer 24(10): 33-41.

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

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

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.

Pfaff, G.E, (1985) *User Interface Management Systems*, Eurographic Seminar Series, Springer-Verlag: Berlin.

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.

Rudnicky, A. (1993) matchine Input Mode to the Task, BYTE, 18(11), pg 100.

Schreiber, G., Wielings, B. and Breuker, J. (1993) KADS: A principled approach to Knowledge-Based System Development. Academic Press: London.

Tansley, D.S.W and Hayball, C.C. (1993) Knowledge-based systems analysis and Design: A KADS developer's Handbook. Prentice-Hall: London.

Wilson, M.D., Duce, D.A. and Simpson, D. (1989) Life cycles in Software and Knowledge Engineering: A comparative review. Knowledge Engineering Review. 3 (4), 189-204.

7 MMI^2 Publications

External Publications:

Barreras, J., Gatius, M., Perez, R., Ruiz, J.C., Trotzig, D. (1991) The Spanish Mode in MMI2 - An integrated module for linguistic domain and sublangauge modelling. ISS Report distributed at ESPRIT Conference ’91.

Barreras Joan (1993) "Resolucion de elipsis y tecnicas de parsing en una interfaz de lenguaje natural". VIII Congreso del SEPLN, Granada, Septiembre 1992. Boletin del SEPLN no. 13, Febrero 1993.

Ben Amara, H., Peroche, B., Chappel, H., Wilson, M.D. (1991) Graphical Interaction in a Multi-Modal Interface. In Proceedings of Esprit ‘91 Conference, pgs 303-321, Kluwer Academic Publishers: Dordrecht.

Ben Amara, H., Nahed, A., Peroche, B. (1992) Un outil graphique pour une interface multi-modes. INFORSID ’92, Clermont-Ferrand (France).

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. In Proceedings of Esprit ‘90 Conference, 412-433. Kluwer Academic Publishers: Dordrecht.

Cahour, B. (1989) Pragmatic Modelling in consultation dialogues, Proceedings of the Work With Display Units Second Conference, Montreal, 11-14 Sept, 1989. Elsevier: Amsterdam.

Cahour, B., Chappel, H., Falzon, P., Ringland, G.A. and Wilson, M.D. (1989). Literature Review on User Modelling, Rutherford Appleton Laboratory Report, RAL-89-108, Oct. 1989, pgs 63.

Chappel, H.R. (1992) Improving the use of Knowledge based Systems with Explanations. Position paper at the ECAI Workshop on Explanation, ECAI, Vienna, August 1992.

Binot, JL., Debille, L., Sedlock, D., Vandecappelle, B., Chappel, H. & Wilson, M.D. (1992). Multimodal Integration in MMI2: Anaphora resolution & Mode Selection. Proceedings of WWDU '92, Berlin, September, 1992.

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, pp297-315.

H Chappel, M D Wilson (1993) Knowledge-Based Design of Graphical Responses Proceedings of the ACM International Workshop on Intelligent User Interfaces, 29-36, Orlando, Florida, January 1993

Dami, S., Lallich-Boidin, G., (1991) “An expert-system for french analysis within a multimode dialogue to be connected”, 23 p., RIAO 91 conference, Barcelona.

Darses, F. (1991) Constraints in Design: Towards a methodology of Psychological Analysis based on AI formalisms. In D. Diaper et al. (Eds) Human-Computer Interaction: INTERACT ‘90, 135-139. Elsevier Science Publishers: Amsterdam.

Doe, G.J., Ringland, G.A., Wilson, M.D. (1992) A Meaning Representation Language for Co-operative Dialogue. Proceedings of the ERCIM Workshop on Experimental and Theoretical studies in Knowledge Representation, 33-40, Pisa, Italy, May 1992.

Falzon, P. (1991), Analysis of multimodal dialogue for domain, explanation and argumentation knowledge, in M.M. Taylor, F. Neel & D.G. Bouwhuis (Eds.),Proceedings of the second Vencona Workshop on Multi-Modal Dialogue, September 1991 - ISSN 1018-4554.

Jouve, C. & Balfroid, F. (1992) Un systeme a base de connaisance pour la conception de reseaux: NEST. In Proceedings of the 12th International Conference on AI and Expert Systems: Avignon.

Kuipers, E. & Wilson, M.D. (1992) A Man-Machine Interface for Multimodal Interaction with Knowledge Based Systems (MMI2). Proceedings of the East-West HCI Workshop, Vol 2, 373-378 St Petersburg, August 1992.

Lallich-Boidin G., Henneron G., Palermiti R (1991), "Analyse du francais: achevement et implementation de l’analyseur morpho-syntaxique", 125 p., Cahiers du CRISS n. 16, Grenoble, 1991.

Nahed, A, (1992) Utilisation des cartes planaires pour la realisation de plans: aspects algorithmiques. GROPLAN ’92. Nantes (France).

Perez, Ricard (1989) V Congress of the SEPLN. Technical Session, October, 6. Barcelona. Note in SEPLN, Bol. 8. December, 1989.

Perez Ricard (1991) "El Proyecto MMI2: La construccion de un interface multimodal para interacciones hombre/maquina" Boletin del SEPLN no. 10, Julio, 1991

Perez Ricard (1993) "El significado en el interfaz multimodal MMI2" VIII Congreso SEPLN, Granada, Septiembre 1992. Boletin del SEPLN no. 13, Febrero 1993.

G A Ringland, H R Chappel, S C Lambert, M D Wilson, G J Doe (1992) Approaches to Self-Explanation and System Visibility in the Context of Application Tasks, In Proceedings of Expert Systems '92 (12th annual conference of the BCS Expert Systems Group), Cambridge, England, December 1992. Cambridge: CUP.

Ruiz Anton, J.C. (1991) “Un parser basado en la sintaxis de dependencias”. Procesamiento del Lenguaje Natural, 9, pp. 169-176.

Trotzig David (1991) "Fuerzas ilocucionarias y niveles argumentativos en la generacion de dialogo hombre/maquina" VIII Congreso SEPLN, Granada , Sep 1992. Boletin del SEPLN no. 13, Febrero 1993

Wilson, M. .D. & Falzon, P. (1991) Multimedia and Multimodal Systems: Architectures for Co-operative Dialogue, Proceedings of the ERCIM Workshop on Multimedia System, Lisbon, Nov. 1991, pgs 149-153.

Wilson, M.D., and Conway, A. (1991) Enhanced Interaction Styles for User Interfaces, IEEE Computer Graphics and Applications, vol 11, pp 79-90.

Wilson, M.D., Ringland, G.A., Wickler, G. (1993) Cooperative Dialogue in Multimodal Interfaces. In Proceedings of ERCIM Workshop on Multimodal Human Computer Interfaces, INRIA Lorraine, Nancy, October.

Wilson, M.D., D. Sedlock, J-L. Binot, P. Falzon, (1991) An Architecture For Multimodal Dialogue, in M.M. Taylor, F. Neel & D.G. Bouwhuis (Eds.), Proceedings of the second Vencona Workshop on Multi-Modal Dialogue, 1991 - ISSN 1018-4554.

Wilson, M.D. (1993) Building Intelligent Multimedia Interfaces. In Proceedings of BCS Multimedia Systems and Applications Conference, December, Leeds. BCS: London.

Wilson, M.D. (1993) Knowledge Engineering. In M. Bartosek (Ed.) Proceedings of SOFSEM '93. CCS: Prague, pp 357-373..

Wilson.M.D. (in press) Enhancing Multimedia Interfaces with Intelligence. In J.A. Vince and R.A. Earnshaw (Eds) Multimedia Systems and Applications. Academic Press: London.

Wilson, M.D. (in press) MMI2: Experience with co-operative multimodal human-computer dialogue. In J. Carbonell (Ed.) Multimodal Human Computer Interaction, INRIA: Roquencourt.

Wilson, M.D. (forthcoming) Graphics requirements for MultiModel Interaction. In Proceedings of the European Graphics Network Workshop on Graphics and Knowledge Engineering, Bonn, Feb 1994.

Video

Kuijpers, E. & Wilson, M. (1992) MMI2: A Multi-Modal Interface for Man Machine Interaction with Knowledge Based Systems. 16 minutes running time. VHS format. CRISS: Grenoble.

8 P2474 Project Deliverables (available from the consortium partners)

d1: Literature Review and General Architecture, All partners (April, 1989)

Part one of D1 provides a general literature review of the state of the art at the start of the project in 1989. This covers Multimodal interfaces, each of the interaction modes, dilaogue management, user modelling, knowledge based systems and the distribution of knowledge through the system.

Part two of the deliverable describes the initial overall arhiecture of the MMI2 system.

d2: Common Meaning Representation, BIM (November, 1989).

D2 defines the common meaning representation (CMR) used by the dialogue management layer of the system and by each ofthe modes to support the resolution of references and the fussion of user input in different modes.

d2(TA2): The Context Expert, CRISS (June 1993).

Detailed description of the latest version of the Context Expert module: theoretical background, topic/focus mechanism, activation and saliency,single and plural anaphora, ellipsis, software architecture.

d3: User Modelling for Multi-Modal cooperative dialogue with KBS, Beatrice Cahour & Helen Chappel (February 1991).

The user modelling expert is defined as a module which overhears the user/system conversation and updates itself on the basis of the user's knowledge. The overidable default multiple inheritence stereotype system is specified. A methodology is defined for populating such stereotype systems, and the case study ofthe first MMI2 demonstrator in the network design domain is used to illustrate this.

d3(TA2): Advanced Dialogue Management: Argumentation Coordination, ISS (June 1993).

d4(TA2): Advanced Dialogue Management: Ellipsis, ISS (June 1993).

d5: First prototype of integrated dialogue management, All partners (February 1991).

A description of the first prototype produced in 1990 which illustrates the use of multiple modes and simple dialogue handling. This prototype only allows users to design the network themselves since it does not include the design expert system NEST. However, the design can be invetigated using the analysis experts, and by interogating the object oriented database repreesntation of it. The dialogue supported consists of a series of question and answer pairs, and a nested subdialogue, and illustrates anaphora handling,ellipses, mixed mode response and mode switching.

d5(TA2): Advanced Dialogue Management: The Interpretation of Theme, Evert Kuijpers (June 1993).

A bibliographic study of the theory of theme, followed by a first try at applying the Prague approach to the french system in MMI2.

d6(TA2): Dialogue Robustness. Graham Doe and Michael Wilson. (Jan. 1993)

The properties of the dialogue which can be supported by the first demonstrator are investigated in detail i nthis deliverable. An attempt is made to place the sub-langauge used in the natural language modes in the context of human-human natural language, and the desired set of requirements on language at the syntactic, semantic and pragmatic levels.

d7: The first MMI2 demonstrator: a Multi_modal Interface for Man Machine Interaction with Knowledge based Systems. All partners (December, 1991).

A summary of the first MMI2 demonstrator in 1991 for computer network design. This is a similar document to D17, describing the experts in the architecture and the demonstration script. This illustrates complex multi-modal interaction, deeply nested dialogues and co-operative interaction.

d7(TA2): Spanish Mode Analysis, ISS (June 1993).

A description of the Spanish Natural Language comprehension system.

d10(TA2): French Language Generation, Samir Dami (July 1993).

A description of the French Natural Language Generation system.

d12(TA2): The second MMI2 Demonstrator Appplication, S. Bescos, L Debille, S Koneci (Nov. 1992).

A detailed description of the second MMI2 demonstrator in the computer network monitoring domain. The method of porting MMI2 to a new domain is described.

d13(TA2): Explanation in MMI2. Helen Chappel and Michael Wilson (Dec 1992).

A specification of the role of explanation in MMI2 dialogues.

d14: Description of the Experimentation. Beatrice Cahour (April, 1989)

Details of psycholinguistic style experiments and Wizard of Oz experiments conducted with users in the network design domain as part of the analysis phase of development.

d15: Extraction of NL data, INRIA, BIM, CRISS, ISS. (June 1990) .

Details of the natural language data extracted from users through the Wizard of Oz and Knowledge Acquisition studies in the netwoek design domain.

d15(TA2): Evaluation, All partners (Dec 1993)

Reports of several studies to evaluate different aspects of the MMI2 system.

d16: English Mode Task 3.2, BIM, (Oct 1990) .

Details of the English natural Language comprehension and generation system.

d16(TA2): Industrialisation, All partners, (Dec 1993).

Exploitation plans for MMI2 from all partners.

d17(TA2): The MMI2 Demonstrator Systems, All partners (Dec 1993).

This document, reviewing both demonstrators of the MMI2 system.

d18: French System: Achievement and implementation of morpho-syntactic parsers, G.Lallich-Boidin, G.Henneron, Rosalba Palermiti (January 1990).

Detailed specifications of the French Natural language parsers used for the French comprehension system.

d20: From a syntactic structure to a CMR expression, E. Kuijpers. G. Lallich-Boidin & J.Rouault. (April 1991).

Detailed specifications of the conversion of the output of the French Natural language parsers into the MMI2 CMR used as thesecond part of the French comprehension system.

d24: Spanish Mode. ISS (Dec 1991).

Detailed specification of the Spanish natural language comprehension and generation system.

d28: Command Language Specification - F.Arshad, N.Drakos, & N.Sheehy (June 1990) .

d29: A description of the Command Language Interpreter, Nikos Drakos, F. Arshad & N.Sheehy (June 1990)

d31: Evaluation of the Compatibility of the Command Language with Natural Language. Mark Howes and Christine Parker-Jones (Feb. 1992).

d32: Gesture Mode, Mark Howes, Nawal Ghali, Noel Sheehy, Kanti Mardia & Michael Wilson. (April 1991).

A detailed specification of the algorithms used to interpret pen like gestures made with a pointing device on the graphics tools in the gesture mode.

d34: Icon/image/menu/string Tools, EMSE (January 1991).

Detailed specifications of the network graphics tool primatives.

d36 & d40: Building Gesture and Graphics Tools. Mark Howes, Nawal Ghali, Noel Sheehy, Kanti Mardia, Tim Hainsworth, Helen Chappel, Michael Wilson, Helmi Ben Amara, Abdelfata Nahed and Bernard Peroche. (Feb 1992).

A comparison af different algorithms for gesture recognition, a description of the graphics manager, and of the network tool.

d38: The Graphical Representation of Structured Representations, H.R. Chappel & M.D. Wilson (Sept. 1991).

Detailed specifications of the process used to automatically generate business graphics presentations in the graphics mode.

d42: Specification of the Application, Fabienne Balfroid (April, 1989)

An initial specification of the computer network design application.

d43: Knowledge Acquisition, Francoise Darses, Fabienne Balfroid, Christine Jouve. (May 1990)

An account of the knoweldge acquisition process used to develop the computer network design application and details of the knowledge acquired.

d44: Knowledge Representation - Fabienne Balfroid, Graham Doe, Christine Jouve (June 1990).

A detailed specification of the knowledge representation used in the objcet oriented database, network design expert system NEST, and the informal domain expert.

d45 : Description of NEST, a Network design Expert SysTem, Fabienne Balfroid & Christine Jouve (Sept. 1991).

A detailed description of the design of the NEST application.

d51: Interpretation Module. G. Doe, D. Trotzig, D. Sedlock & M.D. Wilson (October 1991).

A description of the formal evaluation and interpretation functions used in the dialogue controller on the formal domain expert, user model and interface expert, and the informal evaluation performed by the informal domain against task plans. This includes specifications of the dialogue controller, formal an informal domain experts and the communication expert.

Deliverable numbers which are not presented here were used in the first technical annex to the CEC contract (1989 to 1991) to refer to deliverables to be produced in the second period of the project and were re-numbered in the the second technical annex indicated as TA2 (1992-1993).

9 Details of Consortium Members

BIM

BIM
510 Chaussee de Louvain
B - 1930 Zaventem
Belgium

Telephone: +32 2 719 26 11
Fax: +32 2 725 47 83

Contact Person: Lieve Debille
e-mail: ld@sunbim.be

Intelligent Software Solutions S.A.

ISS
Josep Tarradellas, 15
08 029 BARCELONA
Spain

Telephone: (34) 3 410 04 05
Fax: (34) 3 410 05 18

Contact Person: Ricard Perez

ADR Centre de Recherche en Informatique applique aux Sciences Sociales

CRISS
Universite Pierre Mendes-France
BP 47
38040 GRENOBLE CEDEX 9
FRANCE

Telephone: +33/76.82.54.06
Fax: +33/76.82.56.75

Contact Person: Evert P.J. KUIJPERS

Ecole des Mines de Saint-Etienne

EMSE
158 cours Fauriel
42023 Saint-Etienne cedex 2
France

Telephone: +33 77 42 01 23
Fax: +33 77 42 00 00

Contact Person: Prof. Bernard Peroche
e-mail:peroche@fr.emse

Insitut National de Recherche en Informatique et en Automatique

INRIA
BP 105 Rocqquencourt
78153 Le Chesnay
France

Telephone: (33) 1 39 63 54 22
Fax: (33) 1 39 63 53 30

Contact Person: Pierre Falzon

Current Address:

Pierre Falzon
CNAM-LENET
41 rue Gay-Lussac
75005 Paris

Tel. +33 1 43 54 18 27
Fax +33 1 43 25 36 14
Email falzon@cnam.cnam.fr

University of Leeds

Department of Psychology
University of Leeds
Leeds LS2 9JT
UK

Telephone: +44 532 431 751
Fax: +44 532 336 017

Contact Person: Mark Howes
e-mail: markh@psychology.leeds.ac.uk

Rutherford Appleton Laboratory

Informatics Department
Rutherford Appleton Laboratory
Chilton
Didcot
Oxon
OX11 0QX
UK

Telephone: +44 235 44 66 19
Fax: +44 235 44 5831

Contact Person: Michael Wilson
e-mail: mdw@inf.rl.ac.uk