Helping the User to Make Use of the Tools: Transferring Research Results from the Laboratory to the Market, by means of Intelligent Decision Support.

John Darzentas, Jenny Darzentas and Thomas Spyrou
Department of Mathematics, University of the Aegean
GR 83200 Karlovassi, Samos, Greece
email: idarz@aegean.gr

Abstract

This paper describes a system that was built as a proof-of-concept demonstrator for Amodeus 2. The project researched ways and means of developing tools and techniques to aid the interface designer from the point of view of system, user and design considerations. As these products were the results of upstream research, it was also part of the project to find ways of transferring these results to the design community. One way was the DDAS (Designers' Decision Aiding System) whose purpose was to enable designers with specific interface problems to find which technique(s) would be the most appropriate for them to use. The system was designed using an approach based on Soft Systems Methodology (SSM) and fuzzy reasoning, and Intelligent Decision Support Systems (Intelligent DSS).

The usefulness of the system is that it provides a means of transfer of the modelling techniques in direct relation to the concern of the designer. In addition, a beneficial feature of the system is that it encourages designers to think about their problems and to understand them better. It is also a great advantage that the problem descriptions are expressed in natural language which allows the user more degrees of freedom, while at the same time the evaluation procedures are not constructed around quantifying processes thus ensuring that the initial freedom is retained to a large extent until the end of the decision making process. Finally, the system is generalisable to other situations where problems exist, where tools and methodologies exist to help out in those problems, but where a means of helping the problem owner decide what is appropriate for him is lacking.

Acknowledgements: *The work described below was carried out as part of the ESPRIT Basic Research Action Project no 7040, AMODEUS 2*

1. Introduction

Of the problems which arise when dealing with the products of today's technological world, two which are increasingly apparent to policy makers and users alike are those of *usability* and of *transfer of research results* [4,16]. The problem of the transfer of research results, especially where these results relate to usability of computer systems, is the subject of this paper. Thus in this case, the results of research, when successfully transferred to their intended end users, designers of computer systems, can substantially affect the usabilty of the systems designers design and build.

The problem of making computer systems more "user friendly" has received much attention, and no one can deny that computers are now being used by users from all kinds of backgrounds, rather than the "initiated" as was once the case. However, in many systems in use or being designed presently, particularly safety critical systems, such as those in use in defence or in air traffic control systems, *usability,* - the ability to use a system and not make mistakes or be misled by the system -, goes far beyond the concept of "user- friendliness" and has repercussions on safety and security. While technology may leap ahead in bounds producing, for instance, images at faster and faster speeds, when humans interact with the systems, human computer interaction (HCI) research is needed to make sure that those systems take into account the limitations of human capabilities in terms of vision, memory and comprehension, learning new skills and so forth. And unless such research is moved out of the laboratories and exploited in the real world, the knowledge of how to use information and communication technologies (ICT) will continue to lag far behind the knowledge to produce it.

Generally, the problem of moving research results into the market place is a familiar one and driving force behind such European Union funded efforts such as the VALUE programme [18] and initiatives such as ISPO [17]. Typically, funded research takes place in academic or even industrial research centres and their upstream results are often subject to very slow take-up by industry, for a whole range of reasons: for example, "downstream" companies are wary of being "guinea pigs" or simply they are not aware of the potential of the results, and/or if they are, they are not able to envision how these results could be incorporated into their workplace or perhaps do not have the resources to investigate how they might be. It is worth noting that this is sometimes still the case even with specially commissioned research. Of the RACE programme it has been noted by the Commission itself that despite its many successes in advancing communication technologies, they are still far from being in widespread use[19]. The researchers, for their part, are often not equipped in terms of organisation and personnel to transfer their results into industry. Furthermore, their research results are rarely in a form that is easily translatable into a package that can be picked up and utilised by the intended end user.

This paper reports on an actual example of this transfer problem and moreover one where the research results to be transferred related to advances in human computer interaction. The paper describes a methodology which resulted in an active intelligent decision support system (DSS) which was designed to help users, in this case, designers of computer systems, decide which tools and techniques, out of an array produced by HCI researchers, would be most useful for them. The methodology and the resulting DSS is generalisable and could be used for other situations where research results are available but the end user needs a means of helping him decide which is best for him in his specific set of circumstances. Furthermore, these results were of the type that are difficult to transfer because they are not easily packaged for the user.

The methodology [11,12] was influenced by systems thinking and more specifically Soft Systems Methodology (SSM) [8,9], and also by most recent trends in Expert and Decision Support Systems which require the user to participate "actively" in the process of decision making. In the case reported here, where qualitative rather than quantitative knowledge was represented in the knowledge base, a reasoning mechanism was designed based on and thus offering the power of fuzzy logic to evaluate non-crisply defined options [15].

The next section gives a brief background of the transfer of results problem as it occurred in the context of an HCI ESPRIT Basic Research project [2]. Section 3 discusses how the concept of an active intelligent decision aiding system, designed using an SSM based approach incorporating fuzzy logic was used as a vehicle for transfer. Section 4 describes the system, its architecture and how to use it, while in the concluding section, the contribution of this work to the problem areas of usability and transfer of results is discussed.

2. The problems of transfer, what to transfer and how

The AMODEUS 2 [2] project researched ways and means of developing tools and techniques to aid the interface designer from the point of view of system, user and design considerations. These were the products of upstream research, and for best value to be made of this work, the project also investigated ways of transferring these results downstream to the design community.

Three main problems bedevil the transfer work. Those described here are those encountered by the team investigating transfer and assay of the research results. Although this example refers to the HCI domain, it is also valid to say that the complex interplay between the form and content of results and the transfer mechanism between the researchers and users repeat themselves in other domains [24].

The first of these problems refers to the *nature of the research results.* These modelling techniques, coming from wide variety of disciplines, (computer science, cognitive science, ergonomics, etc.) consequently differed considerably in approach, in scope, and in degree of formality, and although all touched on the problems of usability in the design of computer systems, some covered certain areas more than others, and were in turn less concerned with some areas than others. Considerable effort was expended to try to integrate the approaches [2] but the diversity of the approaches and the multidisciplinarity of the research did not lend itself to the establishment of an overarching theory. Several strategies were adopted to aid integration and hence transfer. The most important of these being to work on examples. This was to give a common platform to all the modellers and also to allow designers who were invited at intervals to contribute to the research easier access to the issues under discussion. A further problematic aspect of transfer attempt was that this was ongoing research and that results were being developed, revised and extended continuously. Thus there was no clear definitive and internally coherent "package" of results to transfer.

A second problem, related to the first, concerned the *nature of the design process* [21,22,27,28]. In empirical designer studies that were carried out to try to establish how designers design it was repeatedly found that strict notions of design processes were not adhered to in practice. The formal software engineering view of design suggested a linear process expressed as:

requirements ⇒ design ⇒ specifications ⇒ implementation.

However, empirical research, showed that, with one or two exceptions, the product development process had, in reality, relatively little in common with any product design procedures [23]. In real life, the process was one of "muddling through", and that rather than any clear design decisions being taken, there was a kind of "evolving commitment" [33]; design can only be considered as a heterarchical process, a series of reiterations, switching from high level to low level discussion and back, as the design team seeks and receives answers from various inputs at varying levels of granularity [30,31].

Thus there was not even a conception common to both designers and modellers of what constitutes the design activity. nor for that matter between designers themselves and consequently no well defined place within design activity for usability techniques .

The third problem concerned the *interaction between researchers and design practitioners*. That is: the practicalities of actually communicating with the end users for whom these techniques were being developed, namely, the design practitioners. One of the transfer mechanisms was to demonstrate the techniques in specially organised workshops[1,5,32]. The designers were introduced to the modelling, in theory by talks and demonstrations, and in practice, by working through examples and by having the modellers work on real life problems which the designers had been invited to contribute. This last activity took place both at workshops and as large scale investigations lasting several months. However, the pressures of the commercial world being as they are, it was very difficult to get designers to commit to such activities. Even accounting for the fact that the most "enlightened" design teams included Human Factors experts, persons with an especially vested interest in learning about what was going on at the forefront of HCI research, the very richness of the results defied quick and easy transfer.

Since it was difficult to get commitment from commercial designers to participate the techniques were tried out on and by students studying design as part of their curricula. Here the emphasis was on refining of training material that would enable the future generation of designers to use the techniques.

Thus within the project, several types of transfer activity took place. These were the workshops, with all the design planning and organisation that they involved; the working on the common exemplars; and the definition and use of "encapsulated" material [6] to pass on the bare essentials of the techniques.

A further strand of the transfer activity was that which forms the main thrust of this paper. That of investigating the concept of a decision aiding system to act as a transfer mechanism. The next section discusses the concept of the Designers' Decision Aiding System (DDAS) and how its design confronted some of the problems that beset transfer activities.

Although the motivation behind the concept of the DDAS was to act as a transfer mechanism, the actual aid the DDAS was designed to give was that of helping designers decide which technique(s) offered the best value for them. This was a problem that was voiced several times by designers. Their situation, typically, was that they did not have the time or the expertise to "wade through" theory, but wanted useful end results in terms of tools and techniques to help them incorporate usability considerations in their designs, and having understood that there was a variety of such techniques needed some way to distinguishing what was useful for what aspect of design, and how it affected usability related problems.

Moreover, it was stated informally in the workshops that, in some cases, the users were not able to articulate satisfactorily what aspect of usability they were interested in. What they knew is that some part of their designs were not easy to use and they wanted to know what was the reason and how they could correct this and avoid future designs having problems.

While not discounting the value of workshops and other means of transfer, it was felt that a decision support system could offer a further valuable means of transferring results.

3. The concept and design of the Designer's Decision Support System (DDAS)

The first step in undertaking the design and eventual building of the DDAS, was to address the problems engendered by the variety of the modelling techniques as well as the lack of a common perception of the problem space constituted by design activities. Further, the system had to take account of relevant design problems to be able to be useful to designers.

As a starting point for the understanding of the problem situation, a "rich picture" [25,7,9,3] was formed of the design activity space. The views of activities undertaken within design were identified and accommodated in the rich picture in order to form as complete an appreciation of the domain as possible, removing all unsubstantiated referent to sequentiality in design activities. Instead the elements of the design process are shown as components of the design space. These are the artefact itself, specification of the system (tasks) and the interface elements, the communication between them and a continuum of conceptual views of deliberation activity. The process undertaken for the making of the rich picture was via consultations with the modellers and reference to the literature until a consensus was reached, this is described in more detail in [14,10]. An example of a rich picture is shown in fig 1.

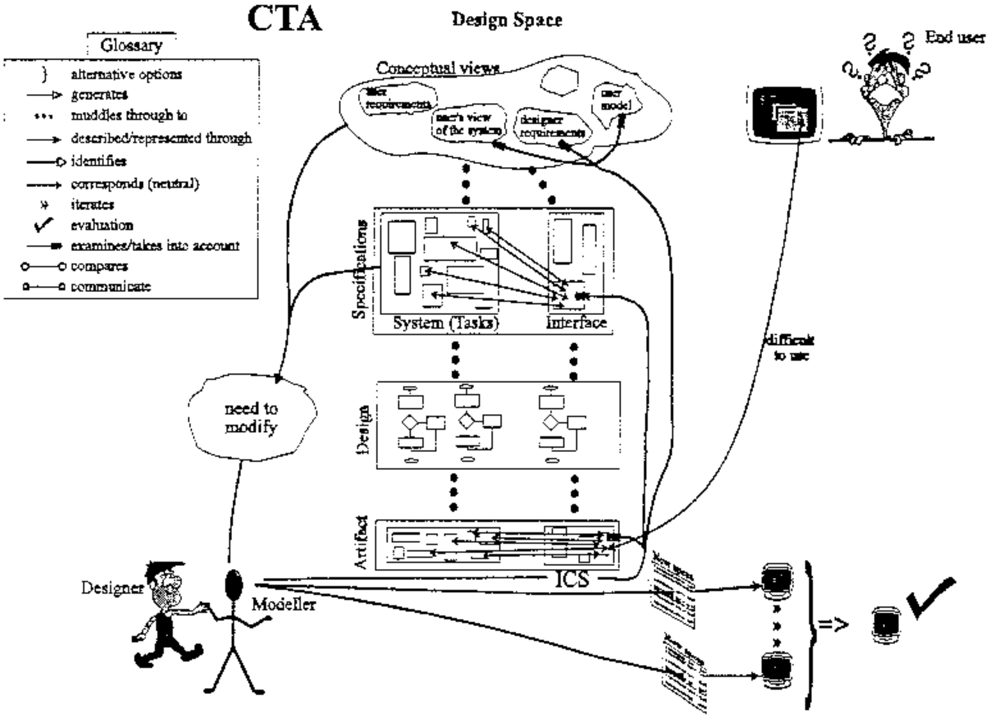

Fig 1. CTA rich picture

The primary benefit of using such an approach is that of gaining insight from learning and debating about the problem situation. It offered an understanding of the activities which constitute design as well as the relationships between the design activities in the design space and the modelling techniques.

The task of identifying and defining the potential of these modelling techniques to design practice i.e. identifying the relationship between the modelling techniques and the problem descriptions they can address was next tackled using the same methods of consultations, literature search and verification [12,13]. The outcome is a conceptual systemic view of the design space defined through a set of relevant subsystems which are identified according to the range of problems that the

AMODEUS modelling approaches can address. The relationships between the relevant activity subsystems and the modelling techniques form the main structural functional component of the DDAS.

With this deeper understanding of the domain and the transfer problem situation that the decision aiding system was attempting to assist with, the task of designing the architecture of the DSS was confronted. In terms of building a DSS, there are three tasks central to development: i) representing the expert knowledge; ii) user problem formulation; and iii) reasoning.

The expert knowledge to be represented is the modelling techniques. As previously stated, these originate from various disciplines and vary in scope of focus: some concentrating on users, some on systems and some on tasks, others on design rationale, etc. Their range of applicability is overlapping and not crisply defined; some techniques are more suited to tackling certain aspects of the design problem space than others. The modelling techniques are research products which are continuously being refined, thus there is no final version of a technique, while the developers of the techniques are variously psychologists, cognitive scientists, computer scientists, etc. whose perspective on design may not necessarily be readily comprehensible to designers. Some techniques require special skills, e.g. knowledge of a programming language; some are aimed at particular members of the design community, e.g. the software engineer, the human factors specialist.

Thus the representation of the expert knowledge needs to be capable of accommodating the multi-disciplinary character of the techniques. It should represent the elements that are common in the sense that they share similar goals, or they explain the same phenomena. It should represent the differences between modelling techniques by capturing their strengths and weaknesses in relation to specific design problems. And finally it should represent the relationships of the various components within a technique and those between different techniques.

In the DDAS this was accomplished by representing the expert knowledge about the modelling techniques and their use in design, through their identified relationships to parts of the possible problems (problem descriptions) they can address. The relevant subsystems and their links form a network of design problem descriptions.

The means of dealing with the question of user problem formulation is closely linked to the representation of the expert knowledge. Two caveats were made here: firstly decision aid is for use with design problems or classes of problem that the Amodeus modelling techniques can handle; and secondly, the designer-user is guided to express his problem by being asked to select descriptions of design problems which relate to his situation of concern. This accords well with recent work in active DSS which show that the user prefers "active" aid from a system, and wants to be prompted [29]. That is to say that a user who has a very well formed idea of what his problem is may not mind expressing it , (though he may wonder if the machine can interpret it as he wants), but a user who just "has a feeling" would have difficulties making that intuitive response to a situation comprehensible to the system. In the case of designers and computer system usability problems this had been shown to be the case.

Thus, bearing it mind the above, the question of user problem formulation was tackled by representing the knowledge in the system in such a way that the designer can express his problem within it: i.e. the designer is shown a variety of problem descriptions and selects those which are most closely resemble his concern.

The resulting attributes and relationships associated with subsystems selected by the designer, can be separated into groups according to the modelling techniques they are associated to. Each of these groups can now be evaluated in order to provide recommendation as to which modelling technique(s) are more appropriate to be used for the particular problem. The evaluation of these groups of attributes is carried out with the aid of test score semantics, this constitutes the third major task area of the development.

The interaction between user and system actually results in a description of the user problem in terms of the activity subsystems which link back into the modelling techniques. Not surprisingly the output shows that several of the techniques would have something to say about this problem. Such output is not succinct enough to be of use to the designer. He requires more intelligent aid. He would like to know when and where a technique is useful. To refine output, the use of fuzzy reasoning, in the form of test score semantics is used, and a recommendation is made to the designer.

Due to the nature of the problem and the method of representation of the expert knowledge, the evaluation is made on the basis of linguistically expressed qualitative as opposed to quantifiable statements. For instance, "technique x is used to resolve problems of ambiguity". The meaning of each of these "discourses" about what each modelling technique can do for a problem or part of a problem is evaluated using test score semantics [34] where the proposals contained in the discourse about a modelling technique are treated as fuzzy constraints. Following the test score semantics procedure each relationship (fuzzy constraint) is evaluated in the explanatory database on the basis of specific templates for the relations. The user provides a score ts_i for each relationship, which will describe the degree to which, according to him (and not the expert who compiled the database), the relationship is satisfied. These two scores per constraint may be combined to one via the an operator such as the compensation operator [35]. According to this approach his test scores assigned to every relationship will give overall aggregated test scores for the groups of attributes and relationships which correspond to each modelling technique. The highest of these overall test scores may be taken as a very good indication that the corresponding modelling technique is the most appropriate currently.

To summarise: the challenges facing the creation of DDAS were threefold;

- to find a common language and use it to describe what each modelling technique can do;
- to describe design problems in such a way that they can be correlated to the relevant abilities of the modelling techniques, using that language;
- to find a way for the system to evaluate the appropriateness of each modelling approach to a design problem.

In order to meet these challenges, an approach was developed for a) the elicitation of expert knowledge about the potential of models; b) representing the meaning of that potential to the client and c) from there recommending to the client the most suitable technique(s). The system's architecture and how the user interacts with the system is described in the next section and an example session is given.

4. The DDAS

4.1. The DDAS Architecture

Figure 2 shows a high level functional architecture of the system. The designer selects the set of subsystems (problem descriptions) relevant to his problem to provide the design problem space knowledge module. The design space module is based on representations in a frame and rule based environment and it is basically retrieved from the general design space contained in the knowledge module. It is the one used by the decision models module. Sets of rules and meta-rules manage the relevant subsystems.

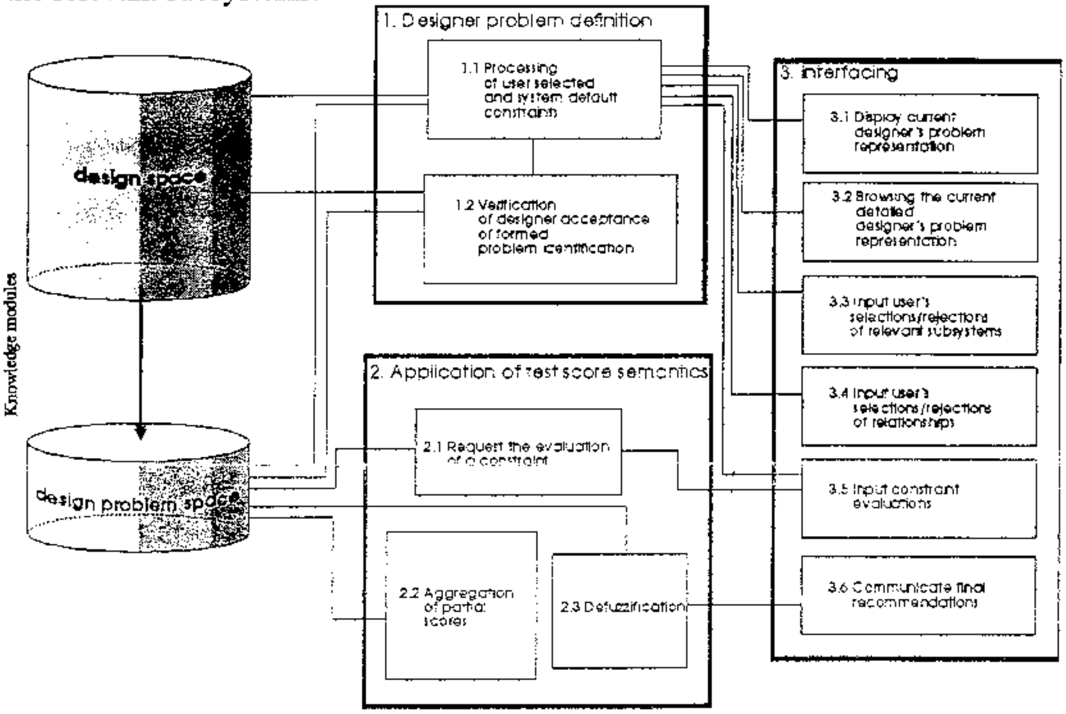

Figure 3. *DDAS architecture*

Fig 2. DDAS functional architecture

Note that modelling techniques are considered as whole units, in other words the modelling techniques are not broken down into operational parts nor there is any attempt to redefine amalgamated techniques from parts of others. The relationships are identified and defined between subsystems of the design space and whole modelling techniques.

The decision models module contains the function which performs the decision aiding process based on fuzzy sets and in particular on test score semantics. This module, having the set of relevant subsystems selected by the designer describing his problem, requests that all the links (constraints) between the subsystems and the modelling techniques etc. are evaluated either by the designer or retrieved from a data base with expert evaluations for some of the links or both. Then following the test score semantics approach this module proceeds with the aggregation of these evaluations followed by the defuzzification of that aggregation towards the final recommendation.

The interface module controls the interfacing between the user (designer) and the system. It contains six functions which are responsible for displaying the current problem representation, browsing and inputting user selections/rejections, evaluations of subsystems and constraints (relationships); and communicating the final recommendations.

Implementation was carried out using CLIPS, an expert system environment developed by NASA and HARDY, a hypertext based diagram editor for Xwindows and windows 3.1. developed by AIAI of the University of Edinburgh [20,26]. Using CLIPS enabled the use of a logic based programming environment needed for the task of manipulating qualitative knowledge, combined with required expert systems features. The use of HARDY enabled the utilisation of visual techniques to interact with the user, making use of graphics and labelled nodes within networks.

4.2. Interacting with the system

The presentation of the problem descriptions to the user; how the user is guided to express/identify his problem and what facilities are available for this; and finally how the system outputs recommendations explained below.

The interaction with the user is based upon two types of presentation elements: the graphic display of the problem descriptions, and the commands that manipulate the interaction.

The problem descriptions are displayed in the form of labelled shapes and are laid out in a series of screens browsable by the user. Two types of shapes are used, one to represent the fact that there exist more specific problem descriptions in the knowledge base, while the other shape represents the most specific expression of a subproblem contained in the knowledge base. In the current version, the former are shown as rhombii, the latter as circles. Shapes may be linked by arcs which denote different types of relationships existing between problem descriptions, for example, green arcs represent "high possible concurrency" and blue arcs, "low possible concurrency".

Commands are displayed as buttons on a toolbar which is permanently on screen. These commands aid the user to choose amongst the available facilities of the system, for example the facility of moving to diagrams/screens that correspond to different levels of analysis is performed by double arrow buttons.

During the interaction the problem descriptions are displayed to the user, so that he can search for and identify those that he considers as most relevant to his problem. The objective is for him to make a selection of these relevant descriptions as a way of expressing his situation of concern. Whilst selecting (by left-clicking on the problem descriptions), the user can also specify the degree of relevance of the descriptions to his problem, and should he change his mind, he can unselect any thing he has already chosen. Each time he clicks on a problem description, its colour changes. Each colour shows the degree of importance of the specific problem descriptions to the user The set of used colours are white, turquoise, yellow, magenta and red, signifying least to maximum importance respectively. This is also the sequence of the colours which appear when left clicking. After red, (most relevant) comes white again and the user can go through this cycle as many times as he wants. In order to guide the user through the network of problem descriptions, these are presented to him at various levels of detail. It is possible for the user to go backwards and forwards between screens (by using the buttons << >>).

The user can ask for comments from the system about the set of the problem descriptions he has chosen (`Comments on Choices`). This facility is available any time during the interaction when selections are made. The comments that the system is able to give are based on the relationships of the problem descriptions that exist in the knowledge base. For instance, the user who has chosen both of the problem descriptions that are parts of a "low-possible-concurrency" relationship, is warned that these problems are not usually concurrent. The problems that are mentioned in the warning messages are highlighted with a black outline in order to find them more easily. The system is flexible in the sense that it allows the user to ignore the warning messages. Should the user want to follow the advice given, he may decide how he wants to solve the implications, by either selecting and unselecting accordingly. Once all the warning messages that the system has to show according to the relationships have been displayed, the system reverts to the normal interaction state where the user can choose/unchoose problem descriptions or choose one of the other available facilities.

A further facility available at any time is that of providing a formatted text description of the set of problem descriptions chosen (`Current State`). The relationships that exist in the knowledge base form the basis for the text description of the chosen problems. Once the text description of the set of chosen problems has been presented, the system reverts to state where the user can choose/unchoose problem descriptions or choose one of the other available facilities.

Should the user want an illustration of a particular problem description, he can obtain examples of use by using the `example` button (or by shift-left-clicking on the problem descriptions in question).This feature can be useful in helping the user decide about how close (if at all) the specific subproblem description is to his own particular problem. Finally, the user is also able to see instructions regarding the use of the system (`Help` button).

When the designer-user feels that the problem descriptions he has chosen describe his situation adequately, he can request a recommendation from the system. The DDAS recommends which of the modelling technique(s) are suitable for his problem. This facility is available whenever no other facility is active. The recommendation is given as formatted text which recommends to the user the most appropriate technique(s). The reasoning behind this recommendation, based upon fuzzy logic, is also given in the formatted text, in order to give the user the justification of the rationale behind the recommendation. The compensation oriented score operator from test score semantics [34] is used to compute the recommendation. For its computation the quantifier value (linguistic or otherwise) that specify how important each chosen most specific problem descriptions is to the user and the quantifier values of how well the modelling techniques satisfy each chosen problem descriptions are used.

4.2 An example

To illustrate some of the system's capabilities, an example detailing a designer's specific problem and how he can handle it using DDAS follows. In this example, a designer's concern is that interface users are often confused by the outcome of clicking a button X. e.g. there can be two different results of clicking the same button X in two different contexts respectively.

The designer wants to resolve this problem. For the sake of the example, it is assumed that he wants the solution to enable the users of the interface to distinguish clearly what are the corresponding effects on the system when a button is pressed. It is also assumed that the designer wants to check that this problem of the design of the interface does not start from a confusion in the requirements.

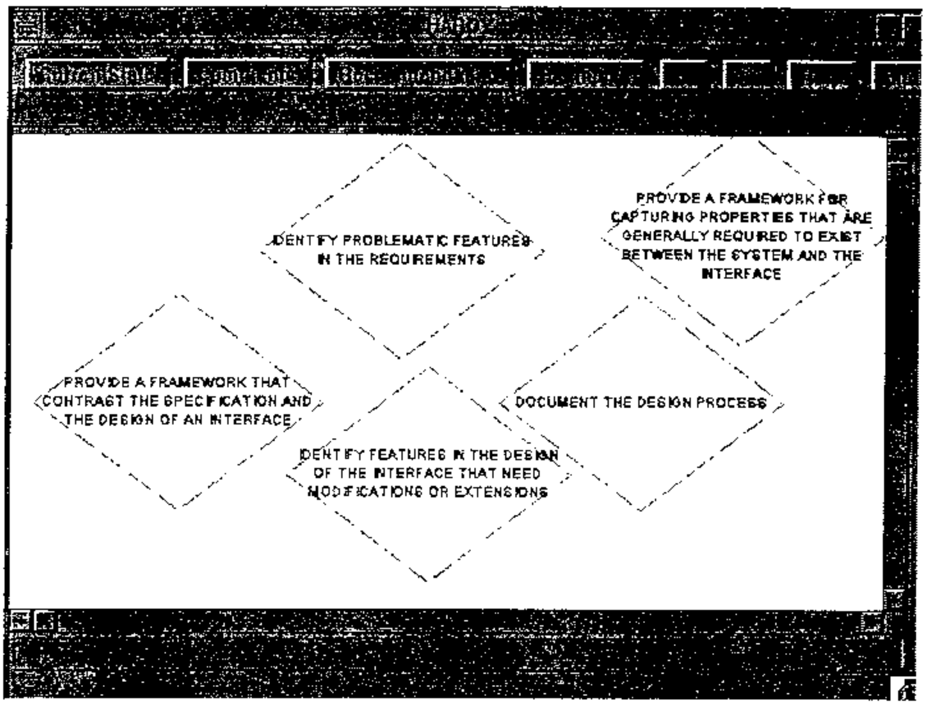

Fig 3. A snapshot of the most general problem descriptions diagram

The designer is firstly presented with a diagram which uses rhombii to represent the most general subproblem descriptions, such as that given in Fig 3. The designer searches through the diagram for labels which come closest to expressing his problem. In this case, he chooses the rhombii with the following labels and assigns to them a degree of relevance:

- IDENTIFY FEATURES IN THE DESIGN OF THE INTERFACE THAT NEED MODIFICATIONS OR EXTENSIONS (red)
- IDENTIFY PROBLEMATIC FEATURES IN THE REQUIREMENTS (yellow)
- PROVIDE A FRAMEWORK FOR CAPTURING PROPERTIES THAT ARE GENERALLY REQUIRED TO EXIST BETWEEN THE SYSTEM AND THE INTERFACE (magenta)

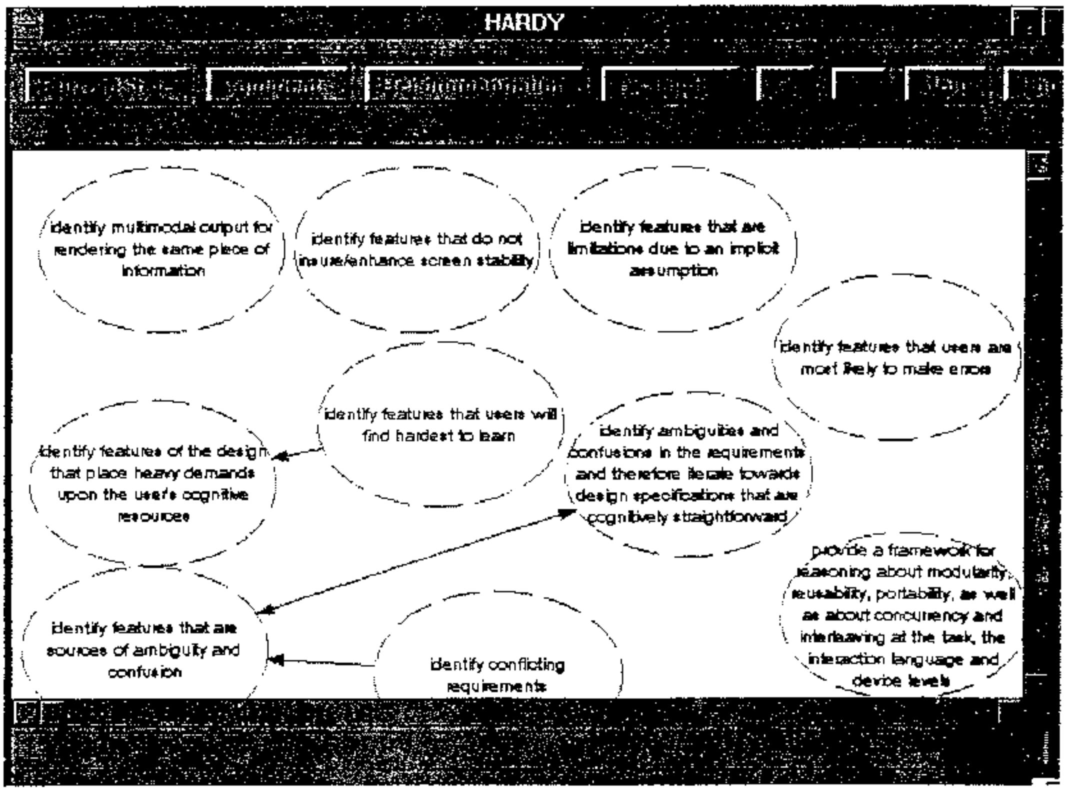

Fig 4. A snapshot of the most specific problem descriptions diagram

The designer presses the button with the label «>>« in order to move to the next diagram with the more specific subproblem descriptions. He is then presented with a diagram which uses circles and arcs to represent the possible subproblem descriptions and the relationships between them, such as that given in Fig 4. The designer searches through the network diagrams for labels which come closest to expressing his problem. In this case, he chooses the circles with the following labels and assigns to them a degree of relevance (colour).

- identify features that are sources of ambiguity and confusion (red)
- identify ambiguities and confusions in the requirements and therefore iterate towards design specifications that are cognitively straightforward (yellow)
- provide a framework for representing and understanding the compatibility between functional (system) state and perceived state (conformance) (magenta)
- provide a framework for representing and understanding the trade-off between what the representation in itself will support and what must be supported by the system (affordance) (turquoise)
- provide a framework for representing and understanding the property of predictability: supporting the system tasks by providing enough information to indicate to user what effect his new actions will have (magenta)

Before going on to choose some more subproblem descriptions from the DDAS diagram, the designer would like to have a commentary from the system about his choices. He clicks on the «COMMENTS ON CHOICES» grey button. This advice is given in a message window as shown in Fig. 5.

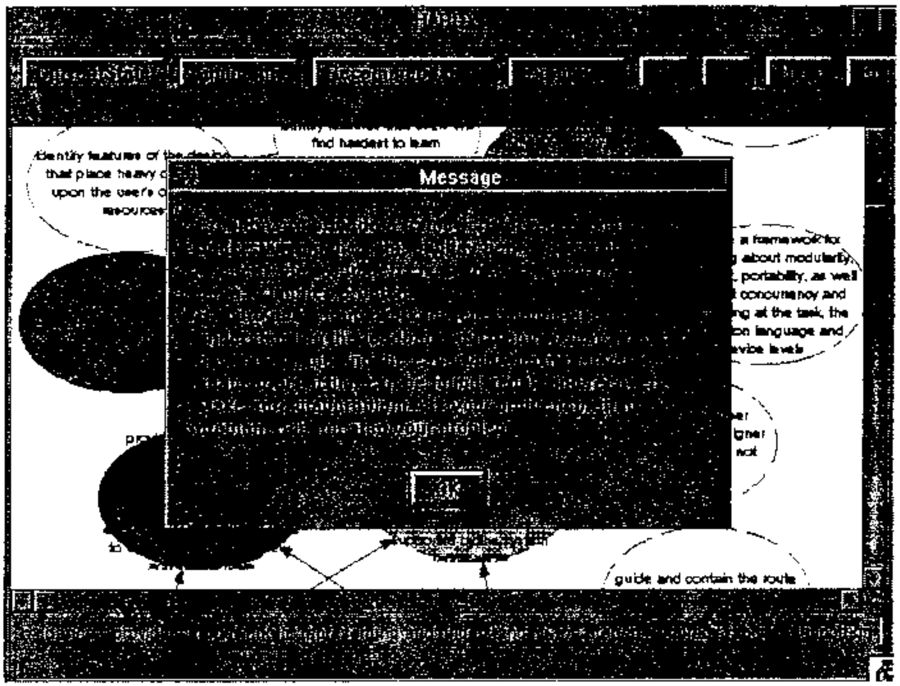

Fig 5. Comments on Choices message window

In this particular case the displayed message comments that according to the system, the subproblem description «*provide a framework for representing and understanding the compatibility between functional (system) state and perceived state (conformance)*» usually implies the one with the label «*provide a framework for representing and understanding the feedback which shows that a mistake has been made and the ease with which an inverse for an incorrect action can be found (repair and recovery)*» and therefore the second could also be chosen.

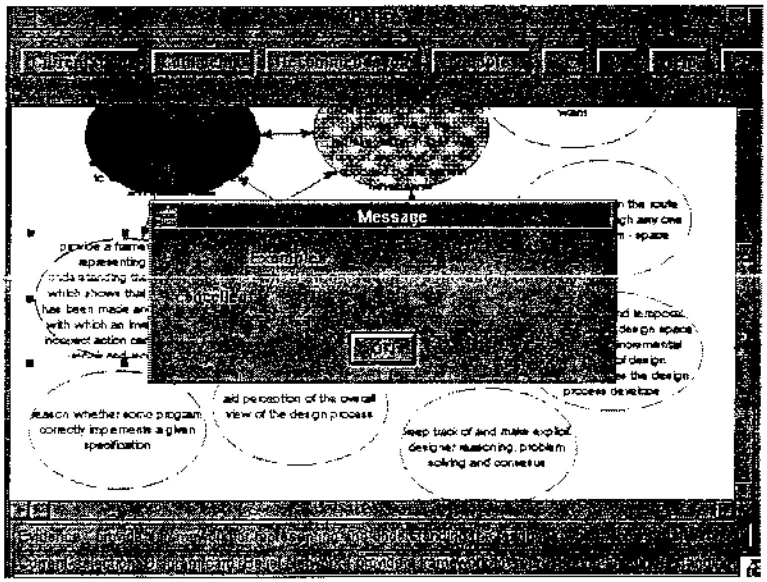

Fig 6. Message window with the available example(s)
for the chosen specific subproblem

The designer can shift-left-click on a subproblem in order to see the available examples (if any) of the specific subproblem. The examples help him understand some characteristic situations that the subproblem should be chosen. The examples are given in a message window as shown in Fig. 6.

Each time the designer wants to see a text description of the chosen subproblem he clicks on the «CURRENT STATE» grey button. A window appears with formatted text which consists of sentences that contain either one selected problem descriptions description or two selected problem descriptions that are related with a type of relationship expressed in words. In this example, a part of the text description that the designer sees is shown on the window in Figure 7

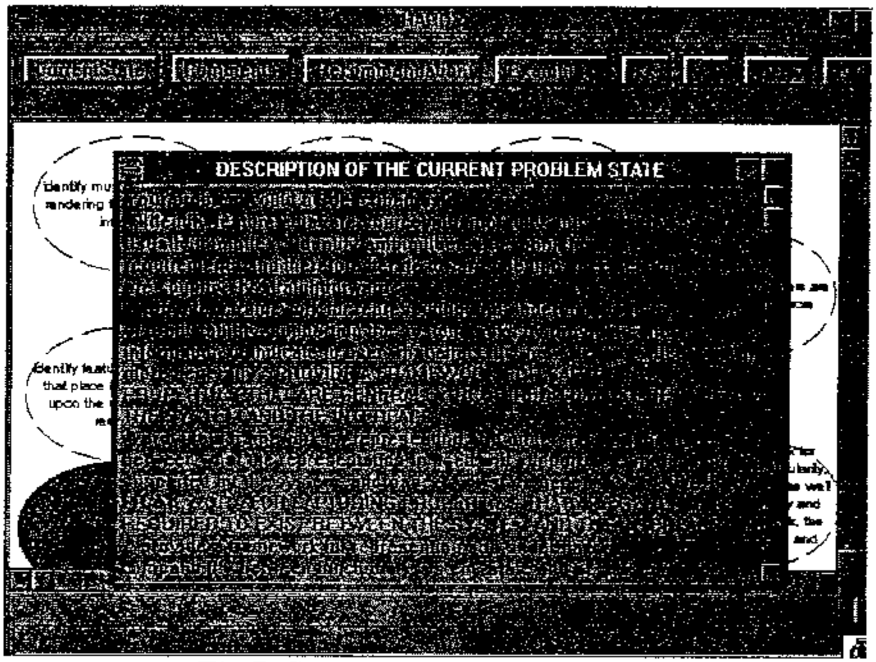

Fig 7. Current state message window

In this way, the system, utilising its knowledge of the design space, and problem descriptions associated with it, prompts the user and aids him to consider subproblem descriptions which may be relevant to his problem of concern and which he has not chosen. The user considers the system's advice and is free to reject it should he not think it relevant.

Otherwise, the system highlights the problem descriptions mentioned with a black outline (Figure 8) to help the user find the problem descriptions that the message refers to.

The user continues in this way, making selections, reading the comments on current choices and reselecting until he is satisfies with what the current selection represents. During this cycle he can get at any time a text description of the current state.

When the designer is satisfied that he has a final set of chosen subproblem descriptions (i.e. he doesn't want to choose any more subproblem descriptions by clicking on them and that he doesn't want to change his belief about the importance he gave to the selected subproblem descriptions, by changing their colour), he then clicks on the «Recommendation» button to get a recommendation about the most appropriate modelling technique(s) for his problem. A window appears with the recommendation. The computation representing the reasoning behind this result is also displayed in the same window for traceablity. This can be transformed to formatted text, in order to give the user the opportunity to understand and justify the system's reasoning (Figure 9).

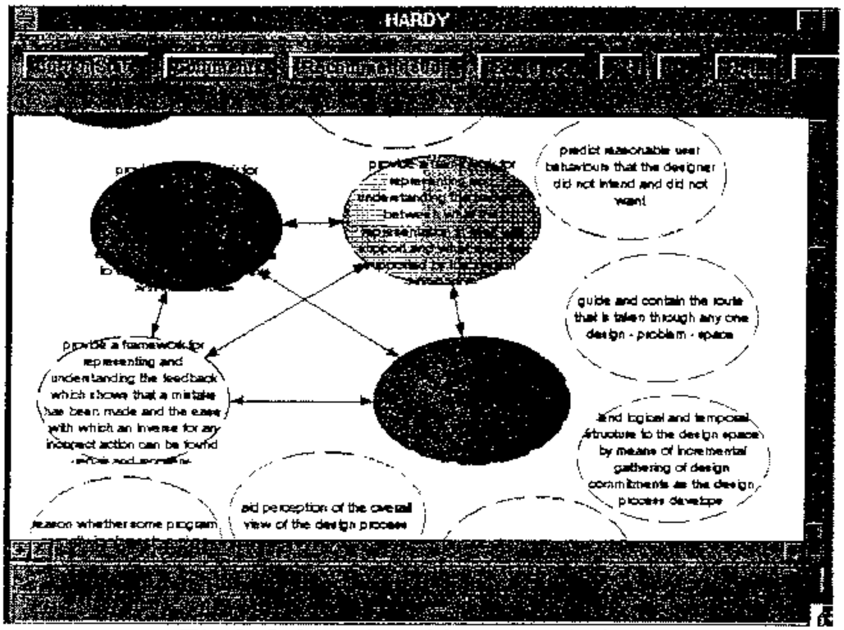

Fig 8. Selected problem descriptions

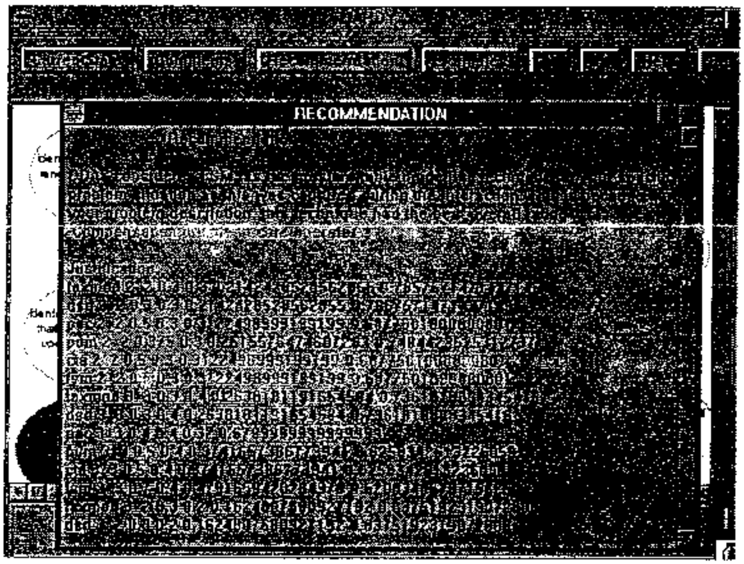

Fig 9. Recommendation message window

5. Discussion and Conclusions

This paper looked at means of transferring research results in cases where these results are not readily exploitable due mainly to their academic nature. The problems encountered in transfer of such work was discussed, and in this case, the research results concerned the usability of computer systems and what can be done to make systems them more comfortable and accessible. A methodology leading to a computer supported aid was given as an example of an effective means of transfer.

The creation of a decision support system to help users chose which results were the most useful for them was a useful tool. Moreover, in addition to the benefits to be gained from other transfer techniques, DDAS offered the following advantageous features:

To the user:

- *it was tailored to user needs*, cutting straight to the heart of user concerns, without the need to undertake to understand theory and rhetoric
- *it acted as a means of accessing the techniques*, deciding which technique(s) were most appropriate was the first step, user then had to find out about how to use the recommended technique(s).
- *it is simple to use and accessible* to anyone with appropriate equipment: no need to attend workshops or demonstration sessions stand alone, no need for human intervention, travel, could be used whenever and whenever convenient.

To Transfer

- *it integrated research results along a common axis*, and furthermore that common axis was directly related to user concerns: "what can this tool for me in my situation". In this way it helped to "package" results for ease of transfer

For the User

- *The user learned more about his problem*

This last was an important benefit that was not envisaged but which is a direct result of the type of vehicle used for the transfer .i.e. the active DSS. The user learning more about his problem, is a significant result in that it impacts the user both in terms of what he learns about the subject (content) and what he was able to see in practice (form): the power of the active DSS to lead the user to question and explore the subject. The "browsing" skills that in a pre-computer age would belong to the person who leafed through encyclopaedias, and which nowadays are translated to those who use the information highway, are harnessed in DDAS to provide the learning forum of such systems. The user is able to search the knowledge base looking for descriptions of problems which are most close to his area of concern. This is a skill that is being developed by all who generally seek information, and its advantage is that the user does not have to come with a problem precisely defined but can make "mini-decisions" through out the process, and learning almost incidentally, what else is on offer.

In terms of transfer and usability the DDAS was conceived specifically to make accessible to the designers techniques for achieving greater degrees of comfort and accessibility. The designers/users of DDAS familiarise themselves with Intelligent Decision Support Systems (Intelligent DSS), and experience for themselves the very real and individually tailored help these systems can provide. The methodology used to design the DDAS overcame difficulties which have been encountered by other transfer efforts, namely that of how to convey results that are widely differing in terms of discipline scope and formality and which are not packaged for transfer. This methodology is generalisable to other results where techniques and tools are available to solve problems but problem owners are unable to efficiently choose which is most appropriate for them.

REFERENCES

[1] Aboulafia, A. Nielsen, J. Jorgensen,A.H. Evaluation Report on the EnEx1 Workshop AMODEUS Project Document TA/WP2

[2] Amodeus, ESPRIT Basic Research Action 3066, 1989-1992, AMODEUS (Assimilating Models of Designers Users and Systems) and ESPRIT Basic Research Action 7040 AMODEUS II 1992-1995 (Assaying Means of Design Expressions for Users and Systems) Documentation available by anonymous ftp (ftp.mrc-apu.cam.ac.uk) or by www (http://www.mrc-apu.cam.ac.uk/amodeus/qref.html).

[3] Avison, D.E., Golder, P.A.,Shah, H.U. Towards an SSM toolkit: rich picture diagramming, *European Journal of Information Systems*, Vol 1, no. 6., 1992, pp.397-307.

[4] Bangemann, M. et al: Europe and the Global Information Society Recommendations to the European Council June 1994, Report available via http://www.echo.lu/eudocs/en/.bangemann.html

[5] Buckingham Shum, S. Hammond, N. Jorgensen, A.H., Aboulafia, A. EnEx3: The ISLE Modelling Transfer Exercise Amodeus Project Document TA/WP22.

[6] Buckingham Shum S., Jørgensen A.H., Hammond N. and Aboulafia A.(Eds), Amodeus-2 HCI Modelling and Design approaches: Executive Summaries and Worked Examples , *Amodeus Project Document: TA/WP16.*, 1994

[7] Checkland, P.B. The development of Systems Thinking by Systems Practice- a methodology from an action research program. Progress in Cybernetics and Systems Research, Vol. 2, Trappl, R.,Hanika, F de P., (eds) Hemisphere, 1975, pp.278-283.

[8] Checkland P.B. Systems Thinking, Systems Practice, Wiley, New York, 1981.

[9] Checkland P.B., Scholes J. Soft Systems Methodology in action, Wiley, New York, 1990.

[10] Darzentas J., Darzentas J.S., Spyrou T. Defining the Design "Decision Space": Rich Pictures and Relevant Subsystems , Amodeus Project Document: TA/WP 21, 1994.

[11] Darzentas J., Darzentas J.S., Spyrou T. Fuzzy Reasoning and Systems Thinking in a Decision Aid for Designers in Proceedings of Second European Conference on Intelligent Techniques and Soft Computing, Aachen, pp.1609-1614, 1994

[12] Darzentas J., Darzentas J.S., Spyrou T. Designing a Designers' Decision Aiding System (DDAS): a Designers' Decision Aiding System, Journal of Decision Systems vol 4. No 1, pp. 9-22, 1995

[13] Darzentas J., Darzentas J.S., Spyrou T. An Architecture for Designer Decision Aiding in Brannback, M. and.Leino, T (Eds) DSS-Galore, Abo Academy Press, Abo Ser. A. 427, pp115-132, 1995.

[14] Darzentas, J., Darzentas, J.S. Spyrou, T. Towards a Design Decision Aiding System: (D/DAS). Amodeus Project Document TA/WP9.

[15] Darzentas, J. Spyrou, T. Tsagaris, C.: Evaluating Option -related Text Descriptions for Decision Aiding.in Proceedings of the 6th Meeting of the European Working Group on Decision Support Sytems, Samos Greece, May 1995

[16] European Commssion: Information and Communication Technologies Partenership available via http://www.echo.lu 1995

[17] European Commssion: Information Society Project Office (ISPO) part of Commission's Action Plan as laid out in *Europe's Way towards the Information Society*: (COM(94) 347) July 1994

[18] European Commission VALUE (1989-1993) & II (1992-94) Programmes for dissemination and exploitation of Research Results

[19] Gannon, P. Whatever happened to that (multi-billion) thing called RACE? Network Europe November 1994

[20] Giarratano J. and Riley G. Expert Systems: Principles and Programming. PWS Publishing, Boston, MA., 2nd. edition, 1994.

[21] Grudin, J. Interactive Users: Bridging the gaps between developers and users, IEEE Computer 24, 4 1991 pp.59-69

[22] Grudin, J. Interface: an Evolving Concept. pp110-119. *Communications of the ACM* pp83-99, Vol 6, no.4, April 1993.

[23] Hannigan, S. Herring,V. The role of Human Factors Inputs to Design Cycles: Deliverable A1.2b, HUFIT CODE HUFIT/8-HUS-11/86, 1986

[24] Laopodis, V. Fernandez, F.F. Gritzalis, D., Darzentas, J. :Evaluation of ICT Polivcy Options by the Public, Using the Awareness Scenario Workshops Methodology presented at 2nd European Conference on the Evaluation of Information Technology, Henley U.K. July 1995

[25] Lewis, P.J. Rich Picture Building in the soft systems methodology. *European Journal of Information Systems*, Vol 1, no.5, pp. 551-360, 1992.

[26] NASA Johnson Space Center, Houston, TX, "Clips Programmer's Guide, Version 6.0, JSC-25012", June 1993.

[27] Nielsen, J. A meta-model for interacting with computers. *Interacting with Computers* pp 147-160, vol 2, No.2 1990

[28] Nielsen, J. The Usability Engineering Life Cycle pp12-22 Vol 25,No 3, IEEE Computer, March 1992

[29] Raghav Rao H., Sridhar R., Narain S. An active intelligent decision support system - Architecture and Simulation. *Decision Support Systems*, 12, pp. 79-91, 1994

[30] Rouse, W.B. and Cody, W.J.: On the design of man-machine systems: Principles, practices and prospects. *Automatica 24,* pp. 227-238, 1988.

[31] Rouse, W.B. and CODY, W.J.: A theory-based approach to supporting design decision making and problem solving. pp.291-306 *Information and Decision Technologies*, 15, 1989.

[32] Shum, S. HAMMOND, N. The "EnEx1" Modeller-Designer Workshop: Implications for Encapsulation, Amodeus Project Document TA/WP4

[33] Terrins-Rudge, D. Jorgensen, A.H. Supporting the Designers, Reaching the Users in Byerly, P.F. Barnard, P.J. May, J. (eds) Computers, Communication and Usability: Design Issues, research and methods for integrated services. pp87-98 Elsevier 1993

[34] Zadeh L.A. Knowledge Representation in Fuzzy Logic, *IEEE Transactions on Knowledge and Data Engng* 1 n° 1, pp. 89-100, 1989.

[35] ZIMMERMANN H.-J., ZYSNO P., Latent Connectives in Human Decision Making, Fuzzy Set and Systems, 4, pp.37-51, 1980.