Designing a Designers' Decision Aiding System (DDAS)

John Darzentas. Jenny Darzentas, Thomas Spyrou

Research Laboratory of Santos
Department of Mathematics
University of the Aegean GR
83200, Karlovassi. Samos
Greece

ABSTRACT Ax approach for designing and developing decision aiding systems is presented through the design and definition of the architecture of a system, corresponding to a real problem: that of providing aid to designers The purpose of the decision aid is to provide assistance to computer system designers tackling interface usability problems and needing to know which modelling techniques out of an array would be more appropriate to apply. Different aspects of the design problem may call for different modelling techniques to be recruited. The design of the DDAS accommodates that reality. The architecture of DDAS is based on principles of systems thinking, in particular soft systems methodology, for eliciting and structuring knowledge relevant to a generic design space, and on fuzzy sets through test score semantics for representing and evaluating the meaning of relationships between components of the design problem and the modelling techniques. The advantage of this architecture is that it could be used for other problems -where the decision maker, who is not necessarily an expert, wants to know what tools and techniques are appropriate for him and his particular problem

ABSTRA1T. Cet article présente une méthodologie pour dessiner l'architecture et la réalisation d'un système interactif d'aide a la décision destine a être utilise par concepteurs-réalisateurs (DDAS) Le but de ce système est d'aider les équipes de concepteurs-réalisateurs qui heurtent a des problèmes d' utilisabilite, et veulent savoir quel modèle d'un ensemble peu: leur servir I1 n 'est pas exclus que certains aspects du problème en son totalité peut faire appel a de différents modèles. Le système DDAS, tel qu'il est conçu, peut répondre a cet besoin. L'architecture du DDAS repose sur des principes de la philosophie des systèmes (Systems Thinking) en particulière des systèmes souples (Soft Systems Methodology) qui lui sen de base pour 1' élicitation et représentation de connaissances qui ont pour référent une espace générique conception-réalisation En plus, le système DDAS utilise des conceptes tirées des ensembles flous, a savoir. lest score semantics, pour représenter et évaluer le sens des relations entre les constituents du problème et les techniques de modélisation. Une des bénéfices de cet architecture c'est qu'il peut être utilise dans d'autres situations ou le décision repose sur le choix entre un nombre de techniques ou d'autres outils et utilisateur. qui n 'est pas forcement expert dans ces techniques, veut savoir ou et quand Us conviennent a aider son problème

KEY WORDS. Decision Aiding, Design, Human Comparer Interaction. Soft Systems Methodology, Fuzzy Reasoning

MOT CLES Aide a la décision, conception-réalisation, dialogue homme-machine, ensembles flous, soft systems thinking

I. Introduction.

This paper describes an approach for designing and developing a designers' decision aiding system (DDAS). The decisions to be aided are those which relate to the choice of appropriate tools and methodologies to be used in order to solve certain types of design problems, in this case, usability[1] issues. By designers are meant those who make decisions, judgements and choices for the purpose of intentionally influencing the form, content and function of an artefact, here the *design* of computer systems.

Design tends to be viewed, from the software engineering perspective, as a structured "top-down" process proceeding from requirements capture and analysis through to specification and implementation. This view is useful for the structure it lends, but in practice the nature of design is such that it has been described as a one of "muddling through" (Terrins-Rudge and Jorgensen 1993). and as being "complex, variable and disorderly..." (Hannigan and Herring 1986). Other researchers studying designers at work in an effort to model the design process concluded that 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 (Rouse 1986, Rouse and Cody 1989).

Where the *usability* of the proposed system or artefact is given high priority, there are, generally speaking, two main sources of formalised assistance: guidelines and models.

Guidelines are those such as are contained in in-house manuals provided by software houses to their design teams (Apple Computer 1987); military handbooks written to attempt to guarantee some standards in systems designed for use by military personnel (Department of Defence 1986); research papers suggesting principles derived from, for instance, empirical studies conducted to examine particular aspects of usability (Bidgoli 1990, Merchant 1992). etc.: or simply books written for the public (Galitz 1994). In some cases work has continued into automating the guidelines, as in the work by Rouse and Cody (Rouse and Codv 1989). Guidelines are good starting points. However, by themselves, guidelines are not sufficient (Moisier and Smith 1986. Gould 1988, Nielsen 1992). In addition, the capabilities of technology change so fast, that some guidelines very quickly become anachronisms.

The other source of assistance comes from researchers who have concentrated on developing models for the design of components of the interface. Generally these require that the designers translate the design requirements into some intermediate language, examine this for inconsistencies and usability implications, and then eventually translate these statements into a programming language. In this way are caught many design flaws and inconsistencies that would otherwise persist further into the design process.

The Amodeus projects (Amodeus 1994) have sponsored the development of several models of this type including models about users, systems, and tasks, as well as methods for structuring and capturing design commitment and rationale. Once the design community has been alerted to the presence of these models and methods, collective!;, referred to as *modelling techniques,* there still remains the task of transferring modelling to the design community. The concept of a designers' decision aiding system (DDAS) was seen as a means of assisting in this transfer, its objective to guide the designers to select the most appropriate modelling technique(s) for their particular design problem or set of design problems.

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 the design problem in such a way that it could 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 and design and develop the DDAS, 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). It is based primarily on principles of Systems Thinking, and in particular Soft Systems Methodology (SSM) (Checkland 1981, Checkland and Scholes 1990), for eliciting and structuring knowledge and on fuzzy sets through test score semantics (Zadeh 1989) for representing the meaning of relationships between the modelling techniques and the design problem and for reasoning about them.

The next section discusses in more detail the problem environment and the particular difficulties it presented. Sections 3 and 4 present the use of soft systems methodology, and of test score semantics. Section 5 describes the architecture of the system and gives the high level functional specification for its implementation, and finally section 6 presents summary and conclusions.

2. Aiding design: a conceptual view.

The ill-defined, ill-structured nature of design practice imposes real difficulties that impact upon i) understanding and representing the expert knowledge; ii) problem understanding and formulation; and iii) reasoning; - all issues central to the development of a decision aiding system.

Here, the expert knowledge to be understood and represented is the modelling techniques. This knowledge is not 'know-how' or guidelines, but a set of techniques which have been developed to help designers incorporate usability aspects in their designs. The) are multi-disciplinary, 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. It is recognised though that 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. A reason for this great variety is that it reflects the variety[1] which exists in a human activity system such as design.

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. This could be 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 (sub problems) they can address.

This in turn means that the representation of the expert knowledge is such that the designer can express his problem within it. In this way. problem understanding, already a complex process, and in the case of design, made more difficult by the tendency for large parts of design practice to be carried out in an unstructured, almost haphazard, way, can be dealt with by having the designer select the problems (sub problems) that most closely resemble his concern. Thus formulating the problem becomes a matter of restricting the designer to descriptions derived from the expert knowledge, i.e. requesting from the designer to form a description of his problem by selecting relevant descriptions from the knowledge base.

The product of the interaction of the designer with the expert knowledge results in an expression of the design problem against which the modelling techniques have to be evaluated. This evaluation constitutes the aid to the designer and is the result of the reasoning mechanism. Due to the nature of the problem and the representation of the expert knowledge, the evaluation will have to be made on the basis of linguistically expressed as opposed to quantifiable statements.

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 sub problems they can address, was approached in a 'top down" fashion based on Checkland's Soft Systems Methodology (SSM) (Checkland 1981. Checkland and Scholes 1990). Use was made of rich picture diagramming to identify activity subsystems relevant to the modelling techniques (Darzentas et al. 1994), The primary benefit of using such an approach is that of gaining insight from learning and debating about :he problem situation i.e. the relationships between the design action situations in the design space and the modelling techniques.

The relationships between the relevant activity subsystems and the modelling techniques contain descriptions of the potential of the techniques in question in relation to a (sub)problem within the design space. These relationships make up 'discourses' about what each modelling technique can do for that problem. The meaning of each of these discourses is evaluated using lest score semantics (Zadeh 1989) where the proposals contained in the discourse about a modelling technique are treated as fuzzy constraints.

To summarise, the problem consists of the design space within which are situated the array of modelling techniques and the designer with his design problem seeking assistance from a variety of avenues, (guidelines, usability testing, etc.) one of which is interaction with the decision aiding system. This interaction results in a description of the problem in terms of the activity subsystems which link back into the modelling techniques. Not surprisingly this output shows that several of the techniques would have something to say about his 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.

3. System of Relevant Subsystems

The system of relevant activity subsystems is the main vehicle for providing a representation of the design space useful for the purpose of aiding the designer in his decision making as to which modelling technique(s) to use to tackle his problem.

This system is defined here as the space which consists of activity subsystems Sj. and their relationships as follows:

$$[S_j, R_{S_j}^{mt_i}, \lozenge, mt_i, x] \qquad [S_j, R_{S_jS_k}^{mt_i}, S_k, mt_i, x]$$

where, Sj is the activity subsystem j. j = 1..N. mt, is the modelling technique i, i = 1..7 currently, $R_{S_j}^{mt_i}$ is the relationship identified within S, in relation to modelling technique i. (◊ denotes that the relationship is actually an attribute of S; which stems out of the properties of mt,).

$R_{S_jS_k}^{mt_i}$ is the relationship between S,. S_k again in relation to mi;, typically corresponding to pre and post-conditions; generality •'specificity level; degrees of concurrency, etc. Finally x is an empirical measure of how much $R_{S_j}^{mt_i}$ is satisfied by mt,.

The relevant activity subsystems Sj are elicited using SSM.

As an illustration, figure 1 displays part of a conceptual model of the design space as "extracted" from the texts and communications of those involved in the attempt to provide an environment for decision aiding (Buckingham et a! 1994, Darzentas et al. 1994, Young et al. 1994}. It shows the relevant subsystems of the design space, as activity subsystems, always in relation to the problem of choosing modelling approach(es) amongst those available, to apply to solve specific design problems.

The conceptual model presented in figure 1 consists of a number of activity subsystems and their relationships to the modelling approaches. There are also implied relationships between the activity subsystems identified with respect to the modelling techniques, (for instance in the form of pre/post-conditions, generality/specificity level, degrees of concurrence, etc.). Figure 2 shows a subset of the relevant activity subsystems identified for the PUMS (Programmable User Modelling) and CTA (Cognitive Task Analysis) modelling approaches (Amodeus 1994) resulting from the application of SSM and through the rich pictures of those techniques as described elsewhere (Darzentas et al. 1994).

Figure 1 actually evolved from figure 2 by grouping the activities identified accordingly; for instance the highlighted activities in the form of the 'is applied" activity' subsystem in figure 1.

The advantages of using a system of relevant activity subsystems are that it offers several features that enhance operationally: although the relevant subsystems cannot be mutually exclusive in a mathematical sense, they can be semantically individual so the user can usefully distinguish amongst them, and they provide good coverage of the design space.

The relevant subsystems and their links forms a "picture" whose contents are in effect a conceptual model of the knowledge about the design space in general as well as problems specific to that space.

Consider the case where a designer using the DDAS system is shown a form of this "picture" in which each activity subsystem has questions attached to them to help the designer navigate through them. He expresses his problem to the DDAS by identifying subsystems he considers as relevant to his concern. The need for decision aiding arises when more than one modelling technique is associated to the designer's problem. When the designer selects certain subsystems, the DDAS system determines which relationships will have to be verified and further questions the designer. Ii is important to note that each attribute may be associated to one or more activity subsystems, only in relation to modelling techniques,

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 and is described in the next section.

4. Application of test score semantics

Test score semantics have been introduced by Zadeh (Zadeh 1989) for representing the meaning of semantic entities such as propositions, which are viewed as sets of fuzzy (elastic) constraints. which in turn implicitly constrain certain linguistic variables in the propositions.

The representation of meaning of a proposition through test score semantics consists of the following steps-

1) identification of the variables in the proposition; 2) identification of the constraints induced by the proposition: 3) association of a test score to each constraint representing the degree to which that constraint is satisfied: 4) aggregation of the partial scores into a smaller number of scores or a scalar.

in the present work the proposition p of interest ma> be:

p: *Amongst the array of the available modelling techniques there are some which tackle the problem in its present form:* etc.

or even:

p: / *need the most appropriate modelling technique for my problem at this stage.*

It is assumed that p consists of a collection of a number of implicit fuzzy constraints which here are the relationships R between the relevant activity subsystems and the modelling techniques. Also one obvious variable relevant here is •modelling technique'. The constraints induced by p are expressed by the relationships R which as fuzzy relations are kept in an explanatory[1] database (Zadeh 1989).

Conceptually, what is achieved is the evaluation of discourses relevant to a design problem, through a representation of their meaning. The different discourses correspond to the different modelling techniques' approach to the same design problem i.e. to different values of the linguistic variable 'modelling technique'.

The relationships R are usually expressed in text form and are considered as a collection of fuzzy constraints. That is a number of fuzzy relations constituting the meaning of the relationship between the subsystem S. and the modelling techniques in terms of relevancy of those modelling techniques identified and S_i.

Assume that the user requesting decision aid has settled with the set of subsystems as being relevant to his concern at the current stage of design. The real decision problem is to evaluate the meaning of the usefulness of each modelling technique which, through the relationships to the subsystems selected by the user, appears suitable. 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 (designer) may also need to provide a score ts, 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 ma> be combined to one via the minimum or other operator.

According to this approach his test scores assigned to every relationship will give overall aggregated test scores for the groups of attributes and relationships discussed in the previous section, 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.

However, it must be noted that the suggested approach is an attempt to evaluate the meaning of relationships in terms of a proposition expressing concern. In other words it is an attempt to identify the most 'meaningful' action to be taken by the designer in terms of using a modelling technique to proceed with solving his problem. In that context it is worth mentioning that Zadeh suggests that the overall score by itself does not represent the meaning of the proposition of concern, but one has to consider the actual process leading to that score. As a result the overall scores here cannot always reflect the appropriateness of a modelling technique over another in relation to a design situation. For example a modelling technique may be moderately appropriate but it may satisfy[1] (moderately) a large number of links (fuzzy constraints), while another one may strongly satisfy' one or two constraints only. Usually the fact that only a few constraints are very much satisfied is enough to overpower the case of the great number of constraints moderately satisfied in a fuzzy environment. In the case of fuzzy quantifiers the aggregation of the partial scores is achieved using the sigma-count of "modelling technique" Σcount(mod. tech.), where small partial scores are ignored. Zadeh gives a number of standardised rules for the aggregation of the partial scores emphasising also that aggregation could be left to the discretion of the constructor of the test procedure.

The importance of the approach is that it offers a more natural way to evaluating options within a problem situation, in this case a design (sub)problem. by maintaining the more natural linguistic expressions relating to the problem. The choice of the most appropriate aggregation is a matter of experimentation in the actual design environments, allowing always, if possible, for subjective intervention by the designer(s).

$[A, R_{A1}^{mt_1}, \Diamond, mt_1, x]$	$[ts_1]$	badly
$[A, R_{A1}^{mt_1}, \Diamond, mt_1, x]$	$[ts_2]$	not so well
$[A, R_{A2}^{mt_2}, \Diamond, mt_2, x]$	$[ts_3]$	
$[B, R_{B1}^{mt_1}, \Diamond, mt_1, x]$	$[ts_4]$	very well
$[B, R_{B2}^{mt_2}, \Diamond, mt_2, x]$	$[ts_5]$	
$[A, R_{AB1}^{mt_1}, B, mt_1, x]$	$[ts_6]$	
$[A, R_{AB2}^{mt_2}, B, mt_2, x]$	$[ts_7]$	

Table 1

The example presented in table 1 shows a selection made by the designer of two subsystems A and B as the relevant ones to his problem, together with the relationships and attributes of the subsystems in relation to modelling technique mt_i It also shows for the modelling technique mt_i that the test score can be fuzzy quantifiers linguistically expressed by the designer.

Here the minimum operator has been used for the aggregation of the scores x in the relational data base and the corresponding test scores ts_i given by the designer.

Hence for each modelling technique having a connection through a number of relationships to designer's problem, there is an overall score. The highest of these scores is an indication that the corresponding modelling technique would be the most appropriate. This recommendation could be justified b> presenting the overall scores of" the other modelling techniques and information about a number of other factors such as max and mm partial scores per technique; subsystems corresponding to these partial scores; etc.

5. DDAS architecture

Figure 3 shows a high level functional architecture of the system. The designer selects the set of subsystems 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 metarules manage the relevant subsystems.

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 pans 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.

Operationally the system works as follows: The designer is requested by the system to identify and describe his problem by selecting a subset of relevant subsystems and corresponding relations from the overall system of subsystems representing the design space. This is achieved by the functions 3.1. 3.2, 3.3 and 3.4. His selections now' form the current description of the problem, and this is carried out by functions 1.1 and 1.2. This current problem description consists of the selected relevant subsystems and their relationships. On the basis of those the system through function 2.1 proceeds to ask the designer to evaluate the relationships (links), now taken as fuzzy constraints. Function 3.5 inputs these evaluations. The system at this point through function 2.2 continues applying the test score semantics approach by aggregating the partial scores (evaluations) and defuzzifying the overall test score through function 2.3. This is finally communicated to the designer via function 3.6.

6. Summary and Conclusions

This paper presented a methodology for providing decision aiding in a situation where the problem owner is a computer systems designer, the problem space is human computer interaction and the decision aiding is to actually aid the designer to choose amongst a number of modelling techniques available for designing interfaces, and in general, usable systems.

The actual problem tackled by the decision aid is relatively complex, although not so difficult to express and generalise: namely the evaluation of modelling techniques to solve design problems in a multi-variable multi-parameter context which includes the designer problem relationship. The techniques are multi-disciplinary and although most of them can deal with most aspects of a generic design space at various levels, they all identify themselves as more efficient at specific areas of the design space than others,

A blending of systems thinking and fuzzy representation of meaning was used in a methodology for developing an intelligent decision aiding system for designers of computer systems and especially of human computer interaction subsystems.

Systems thinking offered a means to operationally define the subsystems relevant to the generic design space. The links of those subsystems to the modelling approaches were elicited primarily from relevant texts and communications with experts. In this way the primary purpose of "translating" the design space into a mode where the role of the modelling approaches in the design space is described in an operational way was achieved. Then the application of test score semantics can evaluate that role in the part of the design space selected by the designer, enabling the system to make recommendations to the designer on the basis of the evaluations.

The evaluation of the links between the subsystems and the modelling approaches carried out by treating them as fuzzy constraints allows for greater flexibility in representation and also sustains considerably more of the designer's understanding of his problem.

The approach is generic in the sense that it can be applied to develop intelligent decision support in many problem areas where the correspondence between tools/methodologies and problems must be established and evaluated in order for the problem solvers/decision-makers to decide *what* they should use to tackle their problems and *when* and *where.*

Work is in progress upon the implementation of the system whose architecture is described in this paper with a "proof-of-concept" demonstrator. This has been developed using CLIPS, an expert system environment developed by NASA and HARDY, a hypenext based diagramming and development tool developed by the

A1AI of the University of Edinburgh. Validation trials with designer; are due to commence shortly.

Acknowledgements: *The work reported on in this paper was funded by the ESPRIT Basic Research Action 7040 AMODELUS (Assaying Means of Design Expressions for Users and Systems}*

7. References

AMODEUS. 1994 : ESPRIT Basic Research Action 3066. 1989-1992. AMODEUS (Assimilating Models of Designers Users and Svstems! and ESPRIT Basic Research Action 7040 .AMODEUS 11 1992-1995 (Assaying Means of Design Expressions for Users and Systems! Documentation available by anonymous ftp (ftp.mrc-apu.cam.ac.uk).

APPLE COMPUTER. 1987 . *Human Interface Guidelines: the Apple Desktop interface,* Addison-Wesley,

BIDGOLI H. 1990: “Designing a user-friendly interface for a decision support system ». *Information Age,* pp.148-154.

BUCKINGHAM SHUM S., JORGENSEN A.H., HAMMOND N. AND ABOULAFIA A.(Eds), 1994 : « Amodeus-2 HCI Modelling and Design approaches: Executive Summaries and Worked Examples ». *Amodeus Project Document' TA/WP16.*

CHECKLAND P.B., 1981 . *Systems Thinking, Systems Practice,* Wiley, New York.

CHECKLANDP.B., SCHOLESJ.. 1990 : *Soft Systems Methodology-in action,* Wiley, New York.

DARZENTAS J.. DARZENTAS J.S., SPYROU T.. 1994 : « Defining the Design "Decision Space": Rich Pictures and Relevant Subsystems ». *Amodeus Project Document: TAiWP 2J.*

DEPARTMENT OF DEFENCE. 1986. *Human engineering procedures guide.* Military- Handbook MIL-HDBK-XXX, Washington DC.

GALITZ W.O.. 1994 : *It's lime lo dean your windows: Designing GUIs that work,* Wiley.

GOULD J.D., 1988 : « How to Design Usable Systems ». *Handbook of Human-Computer Interaction,* M.Helander (ed), Elsevier.

HANKIGAN H. HERRING V.. 1986 : *The role of Human Factors Inputs to Design Cycles,*Deliverable AI.2b. HUF1T CODE, HUF1T/8-HUS-11.'86.

LEWIS P.J.. 1992 : « Rich Picture Building in the Soft Systems Methodology ». *European Journal of Information Sysiems. 1 n° S,* pp. 551-360.

MERCHANT A.N.. 1992 : « Expert Systems: a design methodology », *Computers. Environments and Urban Sysiems 16.* pp. 21-41.

MOISER J.K. SMITH S.L., 1986 : « Application guidelines for designing user interface software», *Behaviour and Information Technology 5.* pp. 39-46.

NIELSEN J.. 1992 :« The Usability Engineerins Life Cycle ».IEEE *Computer 25 n° 3,* pp. 12-22.

ROUSE W.B.. 1986 : «On the value of information in system design: a framework for understanding and aiding designers », *Information Processing and Management 22 n° 2,* pp 217-228

ROUSE W.B.. CODY W.J.. 1989 : « A theory- based approach to supporting design decision making and problem solving ». *information and Decision Technologies 15.* pp 291-306.

TERRTNS-RUDG; D.. J0RGENSEN AH.. 1993 : « Supporting the Designers. Reaching the Users». in. *Computer; Communication and Usability: Design Issues. research and methods for integrated seduces Byerly* P.F.. Barnard P.J.. May J. Eds. Elsevier, pp 8~-98.

YOUNG R., BUCKINGHAM SHUM S.. Duke D.. PATERNO F. MAY J., BELLOTTI V, 1994 : «Thoughts and Questions about Transfer and Assaying». *Amodeus Project: Document TA 7R5*
ZADEH L.A.. 19S9 . >. Knowledge Representation in Fuzz; Logic *//. IEEE T'-a/sjzcnoris or, Knowledge and Data Engng 1 n°1.* pp 89-100