An Architecture for Designer Decision Aiding - DDAS

John Darzentas, Jenny Darzentas, Thomas Spyrou
Research Laboratory of Samos
Dept. of Mathematics
University of the Aegean

Abstract

This paper presents and justifies the design of an architecture of a Designer Decision Aiding System (DDAS) as well as its operation..
The main aim is to provide decision aid to the designer who is tackling a design problem and wants to know which modelling approach(es) out of an array would be more appropriate to apply. Different aspects of the design problem may call for different modelling approaches to be recruited. DDAS architecture accommodates that reality.
The approach to create DDAS is based primarily on principles of Systems Thinking and 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 designer's problem and the modelling approaches.
Systems Thinking offered a means to operationally define the subsystems of that generic design space relevant to the range of design problems recognised by a range of modelling approaches. The links (relationships) of those subsystems to the modelling approaches were elicited primarily from relevant texts and communications with experts. Then an approach based on test score semantics is used to evaluate the role the modelling approaches can play in dealing with the problem contained in the part of the design space selected by the designer (i.e. designer's problem). Finally the system makes recommendations to the designer on the basis of these evaluations.

1. Introduction

In the context of encapsulating and transferring to the designer knowledge relevant to designer problems, including modelling approaches, this paper presents the architecture of a Designer Decision Aiding System (DDAS). In preceding papers [4,5] a justification for the building of such a system was explicated, and the main approach towards defining the philosophy of its architecture was also presented. DDAS is conceived as a decision aiding system which would provide support to designers in order to be able to select the most appropriate modelling technique for their design problem. Using this system the designer will be, in effect, led to express his problem in a way which enables DDAS to match it with the most appropriate modelling approach.

The main component of the approach presented here is a conceptual systemic view of the design space defined through a set of "relevant subsystems" [2,3]. These relevant subsystems were identified with the aid of "rich picture diagrams" [3,7] and from relevant literature and communication with modelling experts. They are presented in a previous paper [5]. The use of rich pictures and of the other material collected to define the subsystems relevant to the problem/situation of concern, is based on Soft Systems Methodology (SSM) [2,3].

The relevant subsystems constitute a set of subsystems of the design space which were identified with respect to the type of problems a range of modelling approaches can tackle. In other words they are relevant to the problem of using these modelling approaches within the "design space". They are actually expressed as activity subsystems.

The approach used revealed associations (links) between the subsystems themselves, in relation to one or more modelling approaches, as well as direct associations between the subsystems and the modelling approaches.

These relationships between the relevant subsystems and the modelling approaches are the main structural/functional component of DDAS's conception. That is the relationships reveal what a modelling approach can do to deal with the action represented by the related subsystem.

These relationships are evaluated on the basis of test score semantics [9,10] in order to produce a recommendation as to what modelling approaches would be more appropriate.

The next section presents in some detail the structure of the system of relevant subsystems, their interrelations with respect to one or more modelling techniques and their relations with the modelling techniques. Section 3 discusses how the relationships between the subsystems, always in relation to modelling techniques, as well as the direct links between the subsystems and the modelling techniques become the constraints in the approach which is based on test score semantics and fuzzy sets. Section 4 gives a description of the general architecture of DDAS from an operational point of view, as well as a description of its use, followed by the Discussion and Conclusions.

2. 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 approach(es) to use to tackle his problem.
This system is defined here as the space which consists of activity subsystems S_j, and their relationships as follows:

$$[S_j, R_{S_j}^{mt}, \Diamond, mt_i, x]$$

$$[S_j, R_{S_jS_k}^{mt}, S_k, mt_i, x]$$

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

$R_{S_k}^{mt}$ is the relationship between S_i, S_k again in relation to mt_i, usually corresponding to pre and post-condition. Finally x is an empirical measure of how much R_s^{mt} is satisfied by mt_i.

The relevant activity subsystems S_j are elicited using Soft Systems Methodology (SSM) [2].

Figure 1 gives a representation of the formal definition of this system.

As an illustration, figure 2 displays part of a conceptual model of the design space as "extracted" from the texts and communications of those involved [1,8], in the attempt to provide an environment for decision aiding. It describes 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 2 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, usually in the form of pre/post-conditions. Figure 3 shows a subset of the relevant activity subsystems identified for the PUM (Programmable User Modelling) and the CTA (Cognitive Task Analysis) modelling approaches after the application of SSM and through the rich pictures of those techniques as described elsewhere [5].

Figure 2 actually evolved from figure 3 by grouping the activities identified accordingly. For instance the highlighted activities in figure 3 form the "is applied" subsystem in figure 2.

The system of relevant activity subsystems is operationally useful in the sense that although the relevant subsystems cannot be mutually exclusive in a mathematical sense, they should be individual semantically and operational so the user can usefully distinguish amongst them. Also they should cover the design space as much as possible.

Presentation of the relevant subsystems and their links will form a "picture" which is shown to the user initially when he is requesting aid for his decisions as to which modelling approach(es) should he use at the current stage. The contents of the conceptual model in that "picture" is also a knowledge model representing knowledge about the design space in general with the possibility of introducing problem specificity.

A form of this "picture" given in figure 4 shows the content of figure 2 where each activity subsystem is expressed with a question which represents a more designer problem oriented view. The question of decision aiding arises when more than one modelling technique are associated to the designer's problem as expressed through figure 4.

Consider the case where a designer using DDAS is shown figure 4. The designer is asked to express his problem through the questions corresponding to subsystems he considers as relevant to his concern. Figure 5 shows a hypothetical subset of figure 2 selected by a designer showing the corresponding relationships considered as relevant to his problem after interaction with DDAS.

The interaction between the designer and DDAS to produce the results shown in figure 5 would be carried out as follows. When the designer selects certain subsystems from figure 4, DDAS, by activating a set of rules, determines which relationships will have to be verified through questioning to the designer. It is important to note that each attribute is associated to one or more activity subsystems, only in relation to modelling techniques, here $R_{AB_1}^{mt_1}$ and $R_{AB_2}^{mt_2}$

The resulting attributes 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.

3. Test score semantics and fuzzy sets in DDAS

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

Assume that the user requesting decision aid has settled with the set of subsystems shown in figure 5, as being relevant to his concern at the current stage of design. The real decision problem will be to evaluate the meaning of the usefulness of each modelling technique which, through the relationships to the subsystems selected by the user, will appear suitable. Following the test score semantics procedure to evaluate each relationship (fuzzy constraint) the user (designer) will provide a score ts_i for each relationship, which will describe the degree to which the relationship is satisfied. Constraint satisfaction means how much the values of the linguistic variables implied in the proposition representing the user's concern satisfy the relationship (fuzzy constraint).

According to this approach his test scores assigned to every relationship will give overall test scores for the groups of attributes 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 [9] 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 (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.

Defuzzification is achieved at the last stage of the aiding process in a similar manner applied to fuzzy control [6]. The membership values of the partial scores (fuzzy numbers or other general functions) are aggregated by making use of, for instance, the centre of gravity [6]. The result of such a crisp value may indicate a conservative tendency. That value together with the result of the use of the minimum operator, where the partial scores are already crisp, however, may offer a more acceptable suggestion.

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 defuzzification is a matter of experimentation in the actual design environments, allowing always, if possible, for subjective intervention by the designer(s).

$[A, R^{mt_1}_{A_1}, \lozenge, mt_1, x]$	$[ts_1]$	= a bit
$[A, R^{mt_1}_{A_2}, \lozenge, mt_1, x]$	$[ts_2]$	= quite
$[A, R^{mt_2}_{A_3}, \lozenge, mt_2, x]$	$[ts_3]$	
$[B, R^{mt_1}_{B_1}, \lozenge, mt_1, x]$	$[ts_4]$	= so & so
$[B, R^{mt_2}_{B_2}, \lozenge, mt_2, x]$	$[ts_5]$	
$[A, R^{mt_1}_{AB_1}, B, mt_1, x]$	$[ts_6]$	
$[A, R^{mt_2}_{AB_2}, B, mt_2, x]$	$[ts_7]$	

Table 1

In the case presented in figure 5 the partial scores are given in table 1. These partial scores, instead of being crisp numbers selected within a range, they could also be expressed linguistically through fuzzy quantifiers, or they could be expressed as fuzzy numbers. The consequence of this is that the partial scores are also fuzzy sets with corresponding membership functions.

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_1. It also shows for the modelling technique mt_1 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 tsi 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.

4. The DDAS Architecture

The decision aiding system is conceived and structured as shown in figures 6 and 7. Figure 6 gives a conceptual and also a functional picture of the decision aiding system. The main modules of the system are described below:

Designer Problem Definition
Here the problem is defined, at the current level, by the designer on the basis of his interaction with DDAS. A design space description via relevant subsystems and their relationships to themselves and to the modelling techniques is provided by the "Design Space" knowledge module (figure 7). The designer selects the set of subsystems relevant to his problem to provide the "Design Problem Space" knowledge module (figure 7). This function also routinely verifies the designer's options.

Figure 6 also describes that process where the designer through his interaction with DDAS produces a description of his design space. It is important to note that the designer's design space description is one way of expressing his problem, amongst many, which is driven by the modelling techniques. In other words the subsystems relevant to the problem will be chosen from a set of subsystems which were identified (elicited) initially because of their relationships to these modelling techniques and to decisions relating to the use of these techniques. The "Design Space" (figure 6) consists of sets of relevant activity subsystems directly relating to specific modelling techniques. The addition operator shown in figure 6 signifies that there are, as expected, overlaps amongst these sets of relevant subsystems. The "Design Problem Space Description" shown in network form in figure 6 is actually what is kept in the "Design Problem Space" shown in figure 7. This is what is used to produce the recommendations for decision (via test score semantics).

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.

Decision models
This part of the architecture contains the function which performs the decision aiding process based on fuzzy sets and in particular on test score semantics. In general 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.

Interface
This module controls the interfacing between the user (designer) and the system (DDAS). It contains six functions which are responsible for displaying the current problem representation and the browsing; input user selections/rejections and evaluations of subsystems and constraints (relationships); and communicate the final recommendations.

Knowledge Modules
These are the "Design Space" and the "Design Problem Space" modules as shown in figure 7. The first contains a description of a general design space in terms of the relevant subsystems and of their relationships amongst them and with the modelling techniques. This is based on representations in a frame - rule based environment. Sets of rules and metarules manage the relevant subsystems network. The second contains the set of relevant subsystems selected by the designer, with their corresponding relationships to each other and to the modelling techniques. This set is basically retrieved from the general design space contained in the first knowledge module and it is the one used by the "Decision Models" module.

Operationally the system works as follows: The designer is requested by the system to identify and describe his problem by selecting a subnetwork (subset of relevant subsystems and corresponding relations) from the overall network (figure 2.) 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.

To summarise, the user is asked to express his area of concern through the relevant subsystem based description of the design space. This is achieved by highlighting those relevant subsystems which he identifies as pertinent to his concern. These relevant subsystems will show their links to the modelling techniques. The links are stored in the knowledge base at varying levels of resolution. The user is required to evaluate with a score those links in relation to how much they (taken now as constraints) constrain the linguistic variables in the proposition which describes his concern.

The functions shown in figure 7 are defined below. The three main functions are:

1 Designer problem definition
Its purpose is to provide a working description of the design subspace relevant to the designer's problem, expressed through a set of relevant subsystems.

2 Application of test score semantics
Applies the test score semantics process in order to evaluate the "meaning" of alternative options (here the modelling techniques) in relation to the designer's problem as expressed in 1.

3 Interfacing
This is the DDAS function responsible for the communication with the user.

The complete set of system functions is:

1.1 Processing of user selected and system default constraints (links between relevant subsystems).
This function corresponds to the process that aims to combine the information input by the user (in the form of selected subsystems and relations between them), and to reason about this information and all the historical information of the interaction with the user in order to further specify the problem area of the user/designer. This specification of the designer problem is in essence the narrowing down of the decision related design space using a set of rules that drive this reduction since they represent the knowledge necessary to identify problem situations in the above mentioned design space

1.2 Verification of designer acceptance of the conceptual model of function 1.1.
This is a function that compares the results of the function 1.1 with the reaction of the user/designer against these results. It is a measure of the acceptance by the user of the system's understanding of his design problem.

2.1 Request the evaluation of a constraint (user provides partial scores)
This function is the first part of the main module of DDAS, the Decision module. This function is responsible for the specification of those relations that need evaluation from the user. Having under consideration the problem of the user/designer as this was shaped by the iteration of functions 1.1 and 1.2 during the interaction with the user, the function is trying to identify those constraints (links) that, whether or not, pre-evaluated and stored in a data base, they should also be evaluated by the user in relation to the specific problem. Those links are the relations between the subsystems that are contained in the resulting design space that corresponds to the designer's problem. The evaluation of the constraints requested by the function is a search for a subjective measure of how relevant these constraints are to the specific problem through the selected subsystems. This is actually an indirect evaluation of how much these links (constraints) constrain the appropriateness of the modelling techniques (value of the linguistic variable) in the context of the specific problem.

2.2 Aggregation of partial scores
Aggregates the partial scores of the evaluated (by function 2.1) constraints into an overall score.

2.3 Defuzzification
It provides a crisp value by defuzzifying the aggregated test score, to be used for evaluating the modelling techniques in relation to the problem context.

3.1 Display of the current version of representation of designers problems
It shows the subset of the design space relevant subsystems and their corresponding links, selected by the designer after an interaction with the system.

3.2 Browsing the current detailed designer's problem representation
Browsing of detailed representation of designer problem in formal expressions containing the relevant subsystems and their relationships to the modelling techniques and amongst them.

3.3 Input user selection/rejections of relevant subsystems
The main aim of this function is to input from the designer those subsystems that are relevant to his problems. The possible subsystems are presented to the designer via the function 3.1. and he is requested to select those he thinks are directly associated with the problem situation of concern.

3.4 Input user selections/rejections of relationships of 2.1).
This function is responsible for the selection of those relationships between the subsystems that the designer believes that are relevant to his problem situation. Relationships are presented to the user using the function 3.1 or 3.2.

3.5 Input constraint evaluation.
This function is responsible for the input of the evaluations of the constraints that are defined by the function 2.1. These constraints are displayed to the user and the evaluations are input through the activation of the evaluation mechanism.

3.6 Communicate final recommendation.
This is the function responsible for the communication of the final results of the decision process to the designer. The final results are in a form of recommendations relevant to the suitability of the modelling techniques for the particular problem situation faced.

6. Summary and Conclusions

The actual problem to be tackled by DDAS is relatively complex, although not so difficult to express and generalise: namely the evaluation of design approaches to solve design problems in a multi-variable multi-parameter context which includes the designer problem relationship. The approaches 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.

The main aim is to provide decision aid to the designer who is tackling a design problem and wants to know which modelling approach would be more appropriate to apply. The decision aid here would be provided via a decision aiding system DDAS (Designer Decision Aiding System)

This paper presented and justified the design of an architecture of such a system as well as its operation. It also gives high level specification of its functions.

The approach to create DDAS is based primarily on principles of Systems Thinking and in particular Soft Systems Methodology for eliciting and structuring knowledge, relevant to the generic design space and on Fuzzy Sets through Test Score Semantics for representing the meaning of relationships and for reasoning.

Since the modelling approaches can only be used as whole units and not be amalgamated at present, the main need was to define what each approach can do in an operational way and devise a "language" for the designer to be able to link his problem to the abilities of the modelling approaches in order for the decision aiding system to be able to evaluate the appropriateness of each modelling approach to designer problems.

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. Therefore 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 is achieved. Then the application of test score semantics evaluates that role in the part of the design space selected by the designer. Finally the system makes recommendations to the designer on the basis of the evaluations.

The evaluation of the links between the subsystems and the modelling approaches by treating them as fuzzy constraints allows for greater flexibility in representing and also sustains considerably more of the designer's understanding of his problem to be able to aid him in his decision for choice.

The paper also presents an architecture for the decision aiding system and high level specifications of the system functions.

Acknowledgements: This work was funded by the ESPRIT Basic Research Action 7040 AMODEUS (Assaying Means of Design Expressions for Users and Systems)

References

[1] BUCKINGHAM SHUM, S., JORGENSEN, A.H., HAMMOND, N., ABOULAFIA, A.(Eds), Amodeus-2 HCI Modelling and Design approaches: Executive Summaries and Worked Examples Amodeus Project Document: TA/WP16, March 1994.

[2] CHECKLAND, P.B., Systems Thinking, Systems Practice. Wiley, 1981.

[3] CHECKLAND, P.B., SCHOLES, Soft Systems Methodology in action. Wiley, 1990.

[4] DARZENTAS, J., DARZENTAS, J.S., SPYROU, T., Towards a Design Decision Aiding System: (D/DAS). Amodeus Project Document: TA/WP9, 1993.

[5] DARZENTAS, J., DARZENTAS, J.S., SPYROU, T., Defining the Design "Decision Space": Rich Pictures and Relevant Subsystems. Amodeus Project Document: TW/WP 21, 1994.

[6] LEE, C.C., Fuzzy Logic in Control Systems, IEEE Trans. on Systems, Man and Cybernetics, SMC, 20, (2), pp. 404-435, 1990.

[7] LEWIS, P.J., Rich Picture Building in the Soft Systems Methodology. *European Journal of Information Systems,* Vol 1, no.5, 1992, pp. 551-360.

[8] YOUNG, R., BUCKINGHAM SHUM, S., DUKE, D., PATERNO, F., MAY, J., BELLOTTI, V. Thoughts and Questions about Transfer and Assaying. Amodeus Project Document TA/IR5, 1994.

[9] ZADEH, L.A., Knowledge Representation in Fuzzy Logic, IEEE Transactions on Knowledge and Data Engng, 1, (1), pp.89-100, 1989.

[10] ZADEH, L.A., KACPRZYK, J.,(eds), Fuzzy Logic for the Management of Uncertainty, Wiley, 1992.

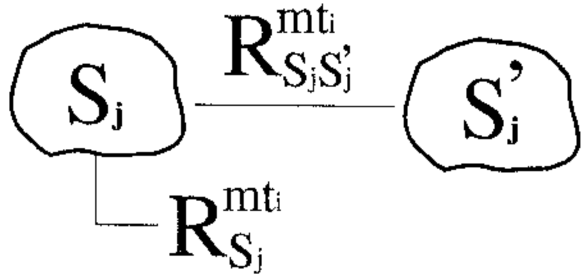


Fig... -11

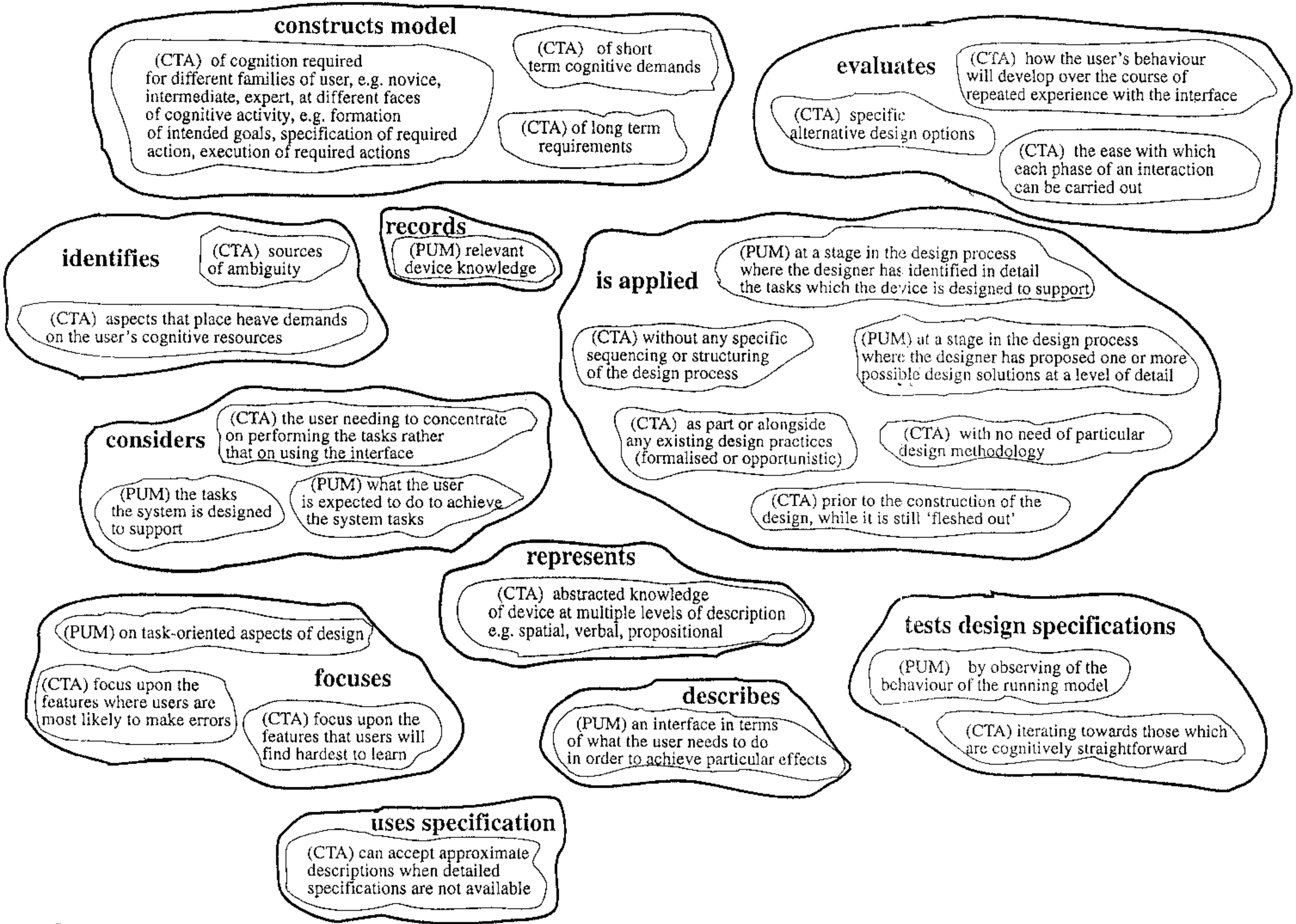

constructs model
(CTA) of cognition required for different families of user, e.g. novice, intermediate, expert, at different faces of cognitive activity, e.g. formation of intended goals, specification of required action, execution of required actions
(CTA) of short term cognitive demands
(CTA) of long term requirements
evaluates
(CTA) how the user's behaviour will develop over the course of repeated experience with the interface
(CTA) specific alternative design options
(CTA) the ease with which each phase of an interaction can be carried out
identifies
(CTA) sources of ambiguity
(CTA) aspects that place heave demands on the user's cognitive resources
records
(PUM) relevant device knowledge
is applied
(PUM) at a stage in the design process where the designer has identified in detail the tasks which the device is designed to support
(CTA) without any specific sequencing or structuring of the design process
(PUM) at a stage in the design process where the designer has proposed one or more possible design solutions at a level of detail
(CTA) as part or alongside any existing design practices (formalised or opportunistic)
(CTA) with no need of particular design methodology
(CTA) prior to the construction of the design, while it is still 'fleshed out'
considers
(CTA) the user needing to concentrate on performing the tasks rather that on using the interface
(PUM) the tasks the system is designed to support
(PUM) what the user is expected to do to achieve the system tasks
represents
(CTA) abstracted knowledge of device at multiple levels of description e.g. spatial, verbal, propositional
tests design specifications
(PUM) by observing of the behaviour of the running model
(CTA) iterating towards those which are cognitively straightforward
focuses
(PUM) on task-oriented aspects of design
(CTA) focus upon the features where users are most likely to make errors
(CTA) focus upon the features that users will find hardest to learn
describes
(PUM) an interface in terms of what the user needs to do in order to achieve particular effects
uses specification
(CTA) can accept approximate descriptions when detailed specifications are not available

Figure 2

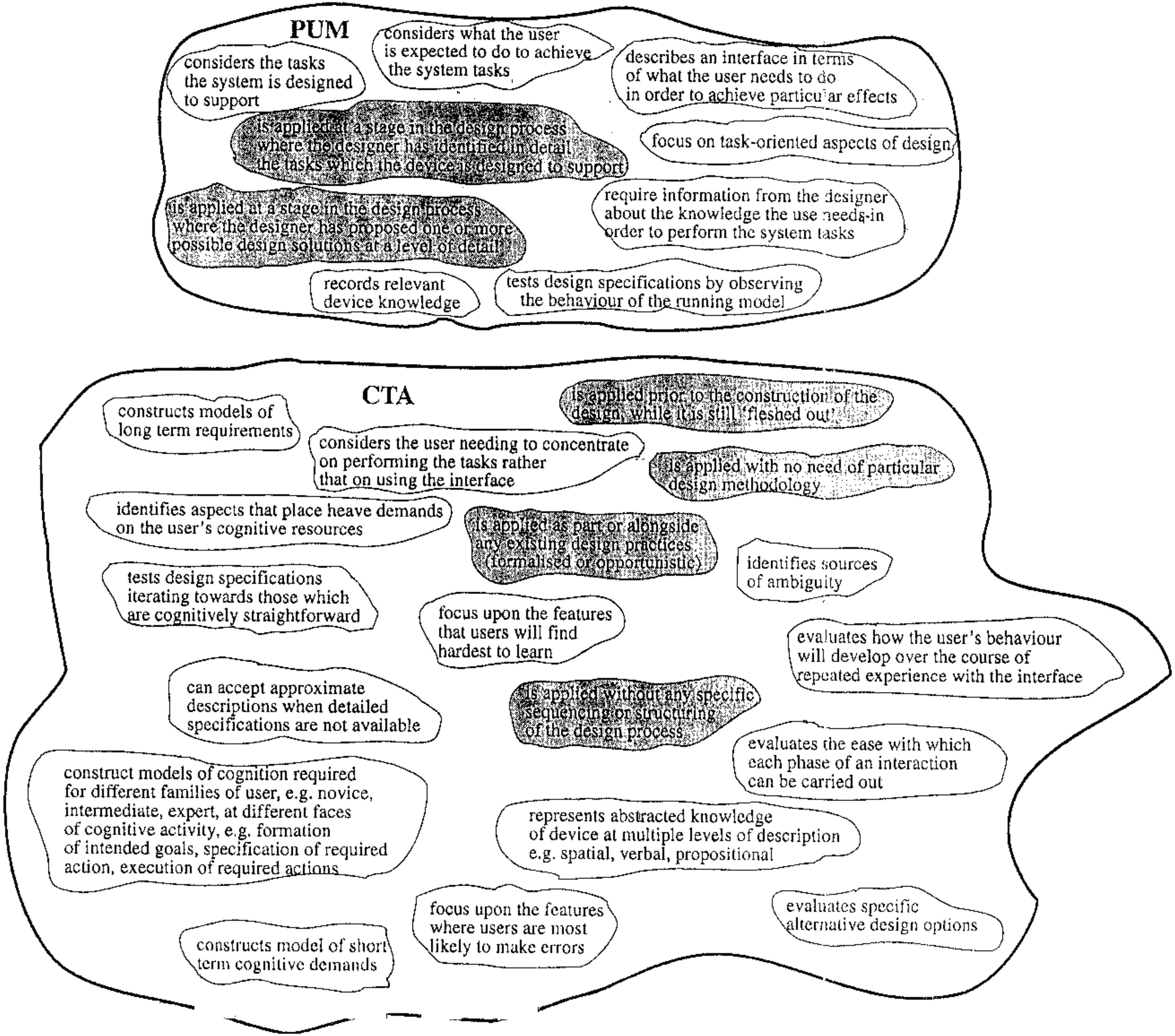
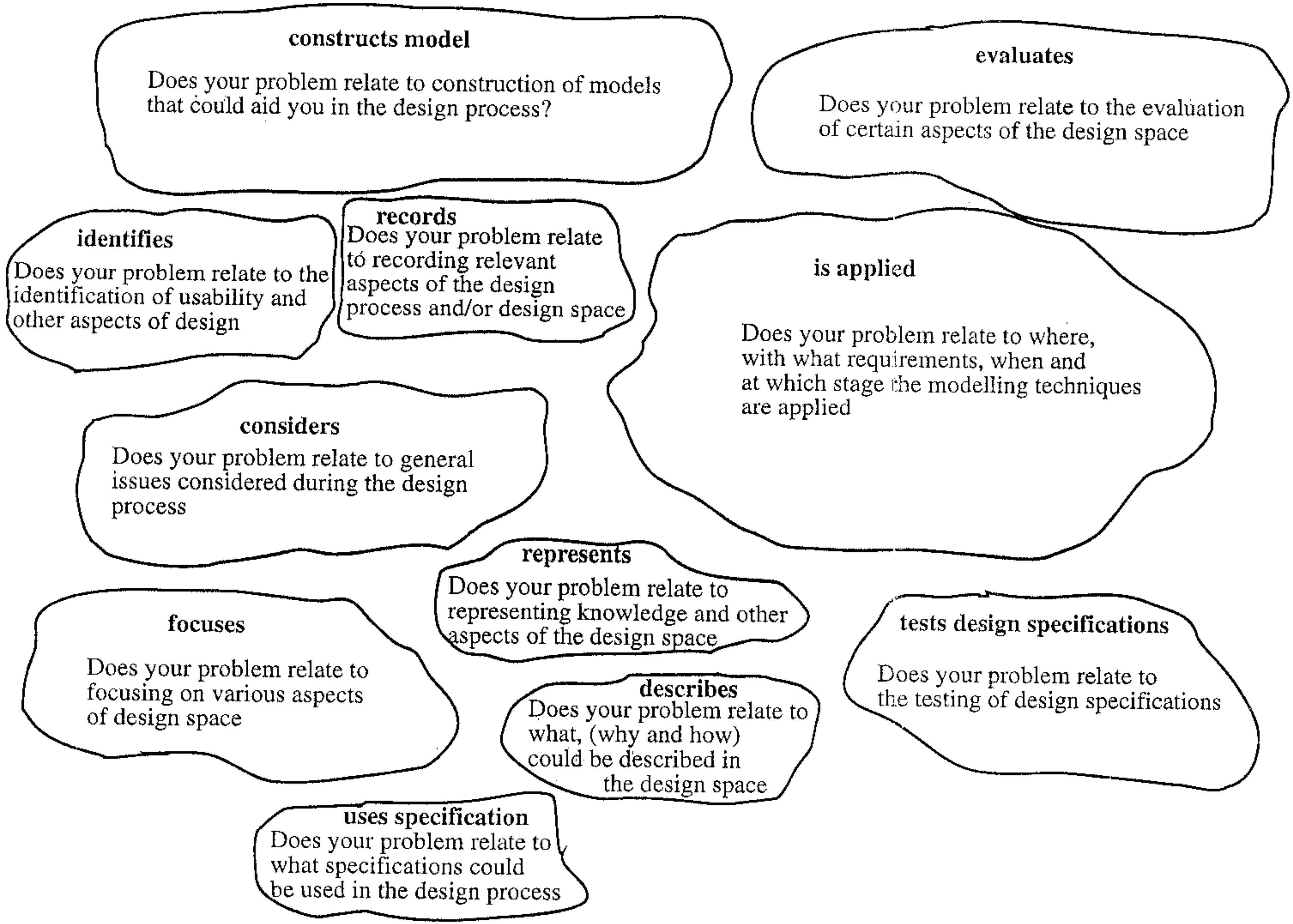

PUM
considers the tasks the system is designed to support
considers what the user is expected to do to achieve the system tasks
describes an interface in terms of what the user needs to do in order to achieve particu'ar effects
is applied at a stage in the design process where the designer has identified in detail the tasks which the device is designed to support
focus on task-oriented aspects of design
is applied at a stage in the design process where the designer has proposed one or more possible design solutions at a level of detail
require information from the designer about the knowledge the use needs in order to perform the system tasks
records relevant device knowledge
tests design specifications by observing the behaviour of the running model
CTA
constructs models of long term requirements
is applied prior to the construction of the design, while it is still 'fleshed out'
considers the user needing to concentrate on performing the tasks rather that on using the interface
is applied with no need of particular design methodology
identifies aspects that place heave demands on the user's cognitive resources
is applied as part or alongside any existing design practices (formalised or opportunistic)
identifies sources of ambiguity
tests design specifications iterating towards those which are cognitively straightforward
focus upon the features that users will find hardest to learn
evaluates how the user's behaviour will develop over the course of repeated experience with the interface
can accept approximate descriptions when detailed specifications are not available
is applied without any specific sequencing or structuring of the design process
evaluates the ease with which each phase of an interaction can be carried out
construct models of cognition required for different families of user, e.g. novice, intermediate, expert, at different faces of cognitive activity, e.g. formation of intended goals, specification of required action, execution of required actions
represents abstracted knowledge of device at multiple levels of description e.g. spatial, verbal, propositional
focus upon the features where users are most likely to make errors
evaluates specific alternative design options
constructs model of short term cognitive demands

Figure 3



constructs model
Does your problem relate to construction of models that could aid you in the design process?
evaluates
Does your problem relate to the evaluation of certain aspects of the design space
identifies
Does your problem relate to the identification of usability and other aspects of design
records
Does your problem relate to recording relevant aspects of the design process and/or design space
is applied
Does your problem relate to where, with what requirements, when and at which stage the modelling techniques are applied
considers
Does your problem relate to general issues considered during the design process
represents
Does your problem relate to representing knowledge and other aspects of the design space
focuses
Does your problem relate to focusing on various aspects of design space
describes
Does your problem relate to what, (why and how) could be described in the design space
tests design specifications
Does your problem relate to the testing of design specifications
uses specification
Does your problem relate to what specifications could be used in the design process

Figure 4

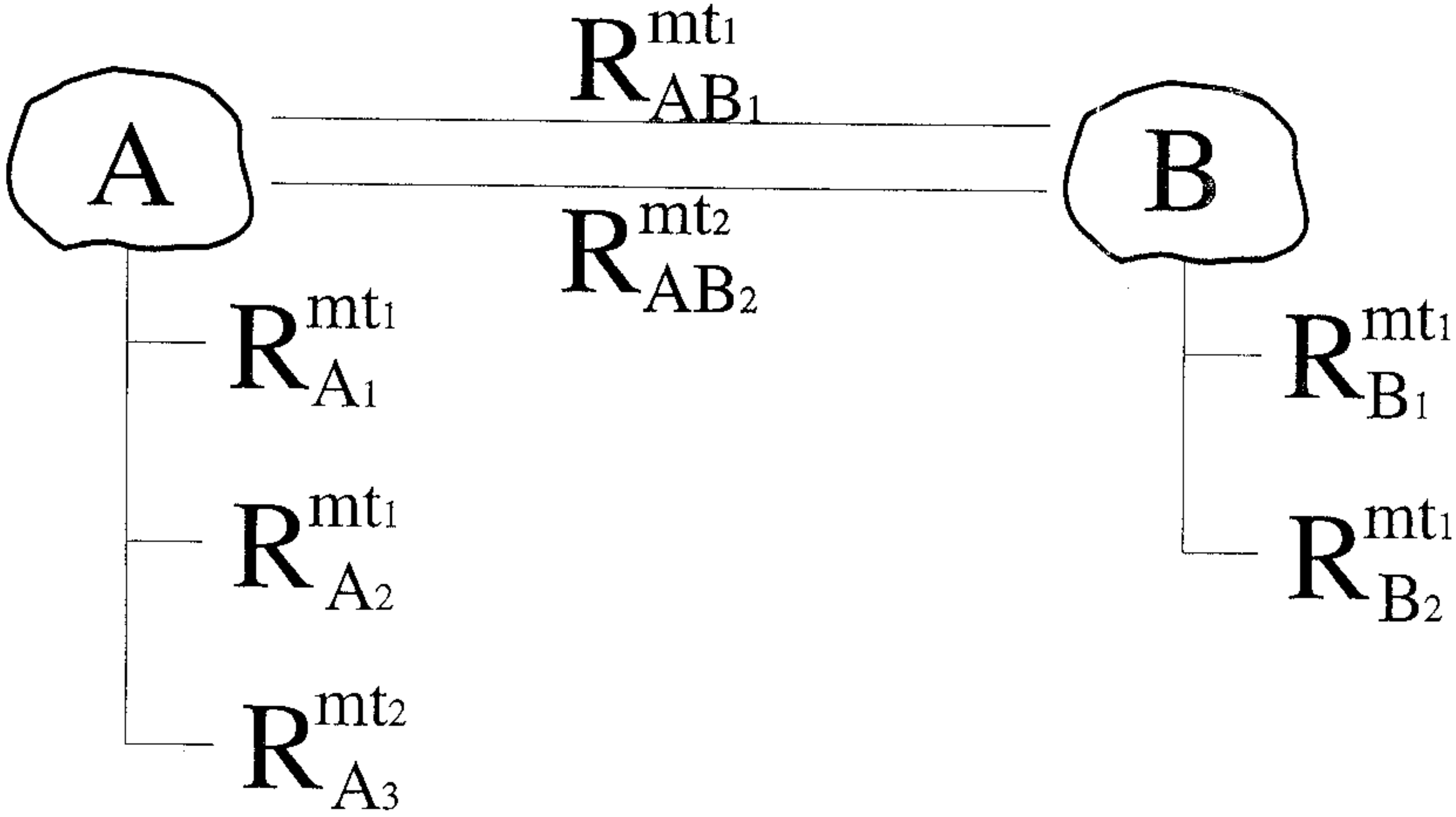


Fig 5

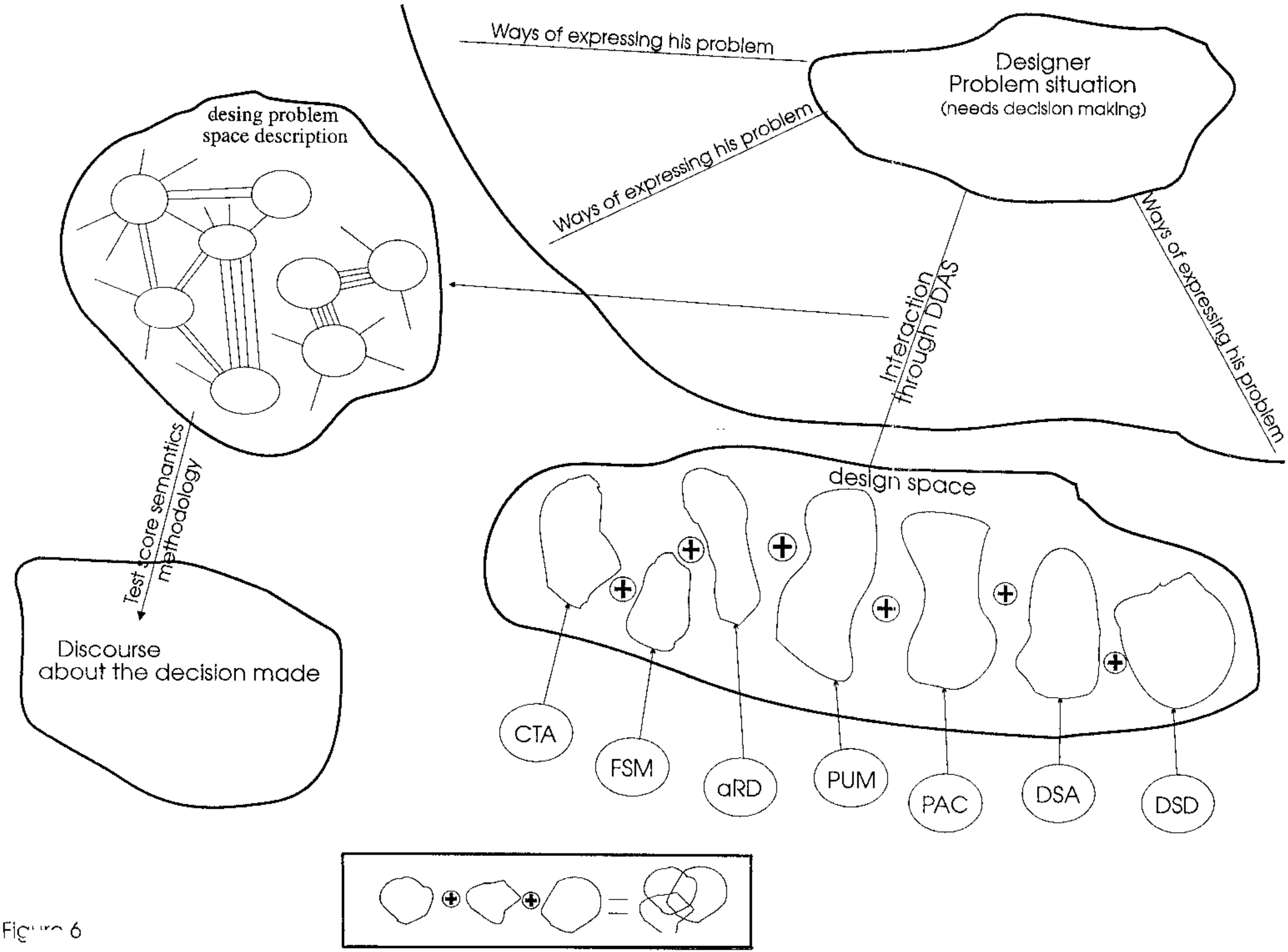

Ways of expressing his problem
Designer
Problem situation
(needs decision making)
desing problem
space description
Ways of expressing his problem
Ways of expressing his problem
Interaction
through DDAS
Test score semantics
methodology
design space
Discourse
about the decision made
CTA
FSM
aRD
PUM
PAC
DSA
DSD

Figure 6

Knowledge modules

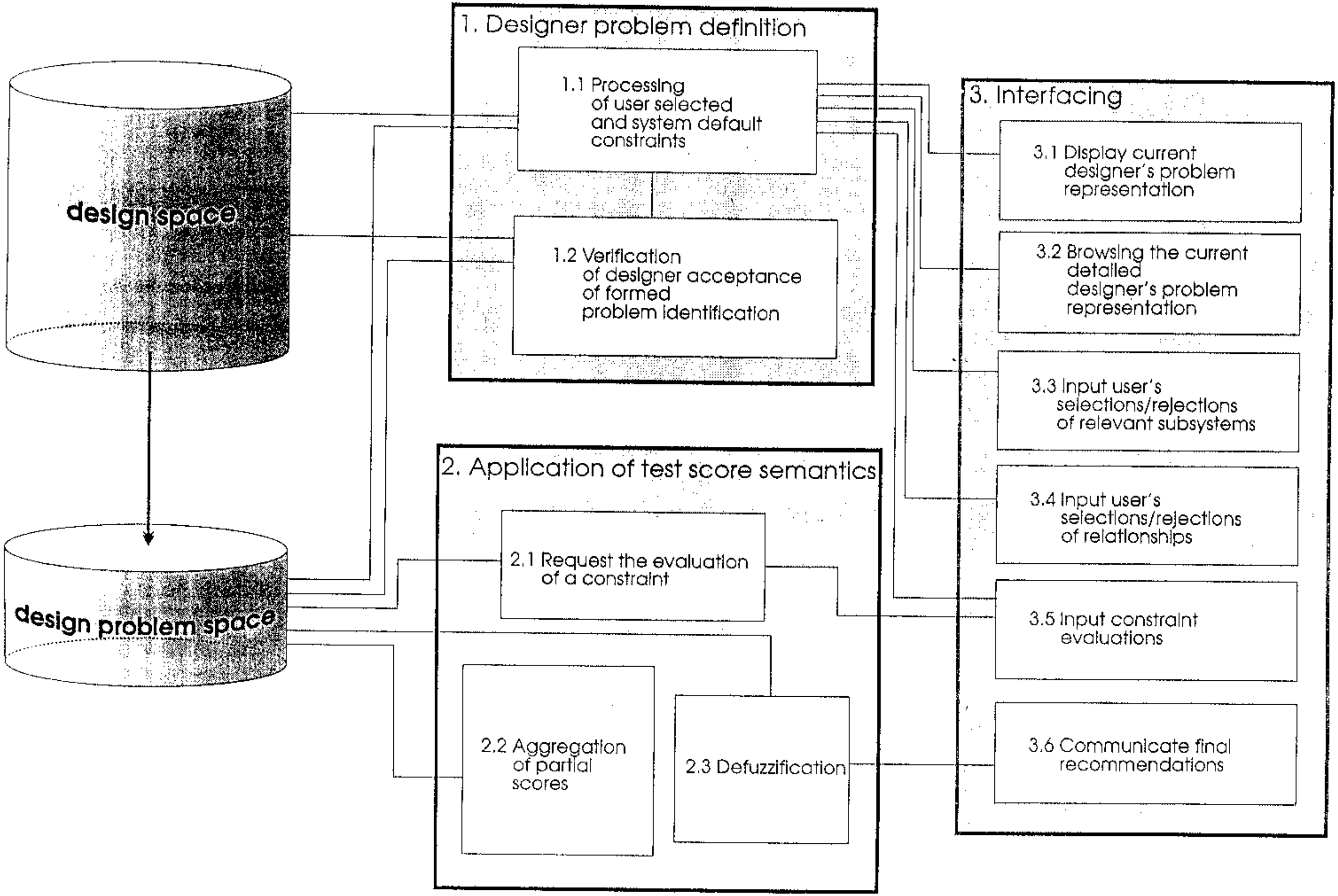


Figure 7