Strategic Knowledge in Design: a Compositional Approach*

Frances M.T. Brazier, Pieter H.G. van Langen and Jan Treur

Vrije Universiteit Amsterdam
Department of Mathematics and Computer Science, Artificial Intelligence Group
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
E-mail: {frances, langen, treur}@cs.vu.nl
URL: http://www.cs.vu.nl/~{frances, langen, treur}

Abstract. In interactive design processes, strategic decisions are made at different levels. To support designers, design support systems need to include corresponding strategic knowledge at these levels. In this paper three levels of strategic interaction and strategic knowledge are identified within a compositional model of design. These levels are identified in reasoning about the manipulation of requirements and their qualifications, reasoning about the manipulation of design object descriptions and reasoning about design process co-ordination. Instances of strategic knowledge illustrate the different levels.

Keywords:
Meta-knowledge, Representing Strategic Knowledge, Compositionality

1. INTRODUCTION

Design is a complex process, in which different types of knowledge play distinct roles. One aspect of the design process is the nature of requirements and their qualifications (e.g., preference relations between requirements). A second aspect is the nature of a specific design description and its properties. A third aspect is the nature of the domain of design objects. A fourth aspect is the nature of a design process itself, the strategies employed to reason about requirements, design descriptions and their interaction. Each of these aspects of design entails different types of knowledge and different types of reasoning behaviour. Therefore, in interactive design, a design support system should support a designer on the basis of knowledge of requirements and their qualifications, knowledge of design descriptions, knowledge of the object domain and knowledge of the strategies that designers employ.

In interaction with a design support system, a designer changes requirements and qualifications of requirements during the design process. For example, a threshold level set in one requirement may be lowered or the qualification of another requirement is changed from ‘hard’ (meaning that the requirement must be satisfied) into ‘soft’ (meaning that the requirement is preferred to be satisfied). A decision to implement changes is made on the basis of many factors, including existing (partial) design object descriptions and an increasing level of understanding of specific aspects of the design problem. Which requirements are changed, when and how, depends on the overall design strategy followed. These changes in requirements, together with the overall design strategy, have impact on the different types of strategies in the subsequent design process. The overall design strategy affects the choice of more specific strategies. Modelling such strategies requires strategic reasoning, strategic knowledge, and strategic user interaction of different types and levels (see also [2,3,4,5]).

This paper distinguishes different levels of reasoning, knowledge and interaction, and shows how these levels can be modelled at different meta-levels within a compositional architecture. The use of meta-levels for this purpose is also a characteristic of both Hori’s

* Please note that this paper is an extended version of [1].

and Oshuga's approach to modelling design [2,3,4]. The distinction between meta-levels is essential to the design of design support systems: it provides a means to reason about interactions between a design support system and a designer. One compositional modelling framework in which different meta-levels can be explicitly modelled is briefly described in Section 2. In Section 3, a generic model of design called GTMD is presented that can be used as a basis for the analysis and modelling of design processes and design knowledge. In Section 4, three levels of strategic knowledge in design are distinguished and it is shown how GTMD models these levels. In Sections 5, 6 and 7, examples of specifications of strategic knowledge in design are presented. These specifications can be combined within GTMD to make a model of a specific interactive design process. The results of applying this compositional approach to modelling strategic knowledge in design are briefly discussed in Section 8.

2 . COMPOSITIONAL MODELLING IN DESIRE

Within the compositional development method DESIRE (DEsign and Specification of Interacting REasoning components), complex tasks such as design are conceptually modelled, specified and implemented [6,7, 8]. Libraries of both generic models and instantiated models are available to support the development of such systems. DESIRE is supported by a software environment that enables the designer to design a system by graphical and textual means. Within the software environment an implementation generator is available to automatically generate executable code from a detailed design, supporting prototyping. The types of knowledge distinguished in DESIRE are described below in Section 2.1; a short description of the use of generic models is given in Section 2.2.

2.1. Compositionality of processes and knowledge

During conceptual and detailed design in the DESIRE modelling approach, the following types of knowledge are distinguished:

- *process composition*
  which includes identification of processes (or tasks) at different levels of process abstraction, their input and output, knowledge of information exchange between processes, knowledge of task sequencing, and knowledge of task delegation.
- *knowledge composition*
  which includes knowledge structures at different levels of knowledge abstraction: compositionally structured information types and knowledge bases
- the *relation* between process composition and knowledge composition.

A short description of these types of knowledge follows; for a more detailed description, see [6], for semantics behind the approach, see [7], and for an overview of the underlying principles, see [8].

Process composition

Process composition includes the identification of the processes and the process composition relation. The identification of processes includes knowledge of a *process* or *task hierarchy* (defining process abstraction levels by process/sub-process relations), and knowledge of the types of information required as *input* and resulting as *output* for each of the processes. Each of the processes is specified as a component (for example, see Figure 1 in Section 3), which is either composed or primitive, and is delegated to one or more agents.

The composition relation is specified by knowledge of information exchange and task control knowledge. Knowledge of *information exchange* between processes defines the types of information transferred between processes. For each of the levels of process abstraction, information exchange between processes is explicitly specified by *information links*. For examples of information links, see the arrows in Figure 1.

Task control is explicitly modelled within components by *task control knowledge*. Task control knowledge includes not only knowledge of which tasks should be activated when and how, but also knowledge of the goals associated with task activation and the extent to which goals should be satisfied. Components may be either continually capable of processing new information (*awake*) or conditionally capable of processing new information (*active*), depending on task control knowledge. Comparably information links may be either continually capable of transferring new information or conditionally capable of transferring new information, depending on task control knowledge. As a result the need for parallel or sequential processing may be determined dynamically.

Knowledge composition

Knowledge composition includes knowledge of how *information types* are specified and structured (according to knowledge abstraction levels), and knowledge of how *knowledge bases* are specified and structured (according to knowledge abstraction levels). The information types, required as input or generated as output of a process, are specified by explicit naming. The same holds for the knowledge structures (information types and knowledge bases) used internally in a component. During knowledge acquisition appropriate compositional structures for domain knowledge are devised: information types and knowledge bases can be composed to information types and knowledge bases at a higher knowledge abstraction level. Compositional knowledge structures and compositional process structures in principle are defined independently. Their relation is specified by references within each process or task to the knowledge structures to be used.

Within a knowledge structure, concepts identify objects and relations distinguished in a domain (*domain-oriented ontology*), but also to express the methods and strategies employed to perform a task (*task-oriented ontology*). In detailed design, concepts and relations between concepts are defined based on *order-sorted predicate logic*. Units of information are represented by the ground *atoms* defined by the information type. The role information plays within reasoning is indicated by the level of an atom within a component: different (meta)levels may be distinguished. In a two-level situation the lowest level is termed *object level*, and the second level *meta-level*. Meta-level information contains information about object level information and reasoning processes; for example, for which atoms the values are still unknown (*epistemic information*). Similarly, processes which include reasoning about other processes are modelled as meta-level processes with respect to object level processes. Often more than two levels of information and reasoning occur, resulting in meta-meta-...-level information and reasoning.

2.2. Knowledge Acquisition: the Role of Generic Models

Generic agent and task models are used to structure knowledge acquisition. These models are generic with respect to both the task and the domain, and as such can be refined by defining more specific task structures (*specialisation*, by extending the task hierarchy) and by defining specific domain knowledge (*instantiation*, by adding detailed specifications of knowledge structures). The generic model of design presented in Section 3 is an example of a generic task model.

A specific task model is most often the result of negotiation with an expert and user: a *shared task model* is acquired on the basis of existing generic models of the type of tasks required. The shared task model, a mediating model [9] used both to structure knowledge acquisition (in the development phase) and the interaction between the user and the system (when the system is used), is an agreed model: a model agreed to be applicable by both the system designer and the user. In general, three different levels of interaction can be distinguished [10]:

- object level interaction,
- interaction at the level of strategic preferences, and
- interaction at the level of task modification.

Object level interaction is not uncommon to knowledge-based systems: it entails interaction about factual information, for example, specific facts about a given world situation. The factors on which design decisions are based are often, however, of a slightly different nature. Strategic preferences refer to, for example, goals, heuristics, and assumptions. Such information is meta-information with respect to the factual information on which object level interaction is based. Once a model has been designed (i.e. tasks and knowledge structures defined, interaction, delegation and control specified), a user may decide that the model needs adaptation. Interaction about the redesign of a task model is known as interaction at the level of task modification.

3. A GENERIC MODEL OF DESIGN

Analyses of design processes are often based on models of design tasks, design systems and designers' approaches (see, for example, [3,11,12,13,14,15]). In this paper, a generic model of design introduced by [16] is used to analyse the role of strategic knowledge within design processes. Refined and improved versions of this model have been used to analyse different types of design (sub-)tasks in a number of domains; for example, conflict management in design [17], re-design of knowledge-based systems [18], elevator configuration design [19] and design rationale [20]. This generic model of design, as Oshuga's model of design [3], distinguishes reasoning about requirements, reasoning about design object descriptions and reasoning about the design process as a whole.

The generic model of design assumes the existence of a problem statement in the form of a set of requirements and qualifications of requirements (including requirements based on a client's needs and desires), in addition to knowledge of the manipulation of requirements, knowledge of the domain, knowledge of the manipulation of design object descriptions and knowledge of design strategies. In a design process, design object descriptions (DODs) are devised on the basis of requirement qualification sets (RQSs). In other words, requirement qualification sets guide the development of design object descriptions by limiting the space to be explored.

In the generic model, a *requirement* is a statement about necessary or desired characteristics of the artefact to be designed, whether formulated abstractly (e.g. in terms of needs, desires or wishes) or concretely (e.g. in terms of observable or measurable properties of the artefact). Requirements can often be grouped into sets of requirements which are directly related to a specific *view* of the artefact to be designed.

Requirement qualifications may be used to define criteria with which a (partial) design object description can be evaluated (e.g. human-friendliness, robustness, modularity, environment-friendliness, etc.). Requirement qualifications can also be used to define *preferences* on requirements, expressing the relative importance of these requirements. For example, requirements may be qualified as hard, which means they must always be satisfied, or, for example, as soft, which means that their satisfaction is desired but not essential.

Requirements often change during the process of design. Which requirements are considered when depends on the design strategy employed with respect to the manipulation of requirements and their qualifications. An example of a strategy for requirement qualification set manipulation is to focus first on sets of requirements expected to have the largest impact on the design of the artefact.

A completely different strategy will most often be employed for the creation of the design object description: which factors to consider when is determined by the strategy for design object description manipulation. Partial design object descriptions are extended during design on the basis of additional knowledge and integration of sub-solutions. The design strategy with respect to the manipulation of design object descriptions determines how this is approached. An example of a design strategy for design object description manipulation is to follow the problem-solving method generate-and-test.

Figure 1 shows the flows of information between requirement qualification set manipulation (RQSM), design object description manipulation (DODM) and design process co-ordination (DPC), which is responsible for the co-ordination of the overall design strategy (see [16]). The figure also shows the flows of information that represent the input and output of a design process. The functionality of the three main components of the generic model, requirement-qualification-set-manipulation (RQSM), design-object-description-manipulation (DODM) and design-process-coordination (DPC), is described below.

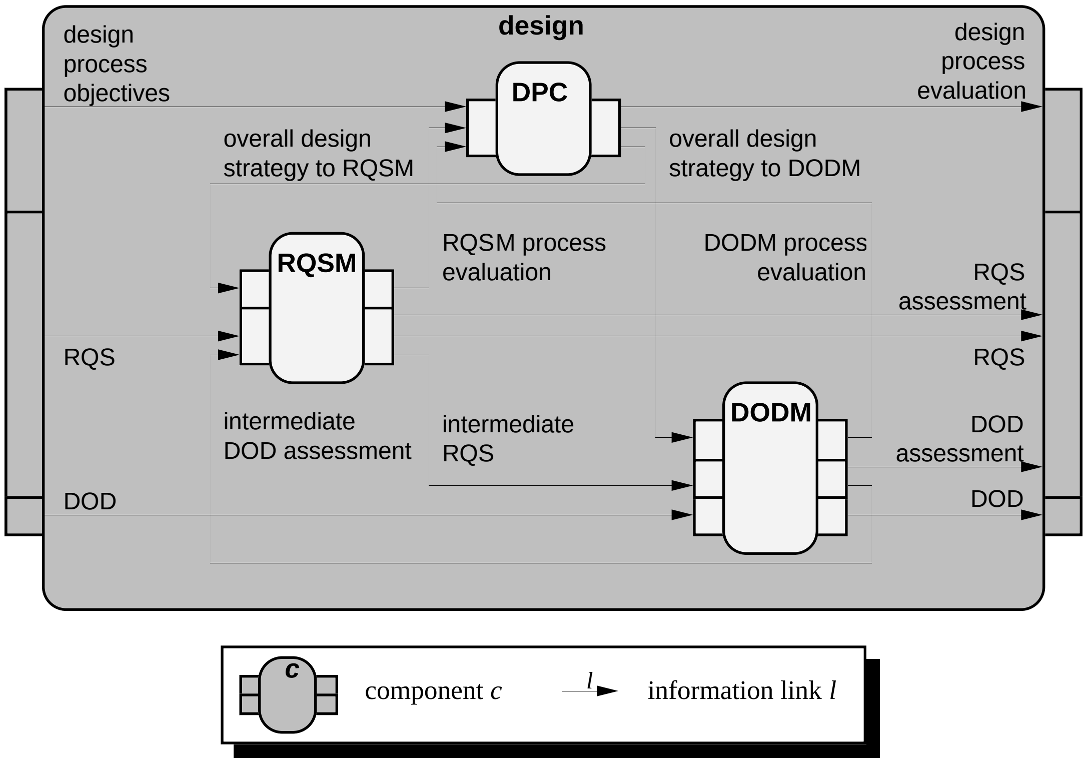

Figure 1. Top-level of the generic model of design.

3.1. Requirement Qualification Set Manipulation

To choose the most relevant subset of requirements, given a current set of requirements and their qualifications, entails consideration of the relevance, importance, and strength of the individual requirements and the relations between requirements. For example, hard requirements must, by definition, hold for the final design object description but are not necessarily imposed continually during design. For instance, an architect may decide to pay attention to the customer's requirements first before taking building regulations into account. The strategy chosen for the determination of the set of requirements to be considered are based on knowledge of preferences between requirements.

Explicit ranking criteria between preferred sets of requirements are sometimes available, but often also other types of strategic knowledge are required. One global strategy is to make a distinction between the sources of a requirement: requirements based on customer preferences may be given higher priority than requirements based on default assumptions, which in turn may be given higher priority than requirements which were the deductive consequence of previous requirements (i.e. the approach described by [21]).

3.2. Design Object Description Manipulation

Creating a design object description on the basis of the requirements imposed involves a dynamically determined strategy. A possible strategy is to focus on a given number of related elements of the design object (for example, a specific view; e.g., the electrical wiring of a building), on the basis of a set of related requirements and using the domain knowledge available to adapt the (partial) design object description, resulting in a new (partial) design object description. This process may be repeated for another focus on the design object, and the resulting design object descriptions assessed and compared.

3.3. Co-ordination of the Design Process

The co-ordination of the design process includes determination (in a dynamic manner) of the overall design strategy, monitoring the progress of the design process, deciding whether to continue or not and if so, where to continue. Thus, design process co-ordination determines the course of a design process and decides when to interrupt or stop the process.

Design process co-ordination can provide guidance to the design process in very different ways. For example, it may prescribe precisely what has to be done during the manipulation of requirement qualification sets or design object descriptions. A less dictatorial strategy is to merely describe the desired results of the manipulation processes and to suggest ways to achieve those results.

4. STRATEGIC KNOWLEDGE

To support designers, a design support system needs to provide at least two of the three levels of interaction distinguished in Section 2: object level interaction and strategic level interaction. Designers need to be able to influence a design process by providing both facts and strategic considerations (e.g., preferences and objectives). In fact, analysis of design tasks in a number of domains of application has shown that strategic level interaction refers to a broad spectrum of interaction. Different levels of strategic interaction can be distinguished; these levels of interaction correspond to the different levels of reasoning, and the corresponding knowledge, modelled in the generic model of design described above.

In this model, four levels of knowledge are distinguished. At the lowest level, not visible in Figure 2, the object level reasoning and knowledge is defined. At the next three higher levels, strategic reasoning and strategic design knowledge are specified. These levels are meta-levels with respect to each other; each of the three meta-levels has a semantics that is based, in part, on the processes at the lower level. Figure 2 shows the three meta-levels and the information types within the interfaces of the three main components of the generic model of design.

The *object level* (hidden within component DODM and therefore not visible in Figure 2) includes facts expressing properties of a given design object and domain knowledge on the type of objects to be designed and their environment. The *first meta-level* includes knowledge of requirements and their qualifications, and meta-descriptions of the DODs. The *second meta-level* includes knowledge with which to reason about DODs, RQSs and their modifications and about the relations between DODs and RQSs. The *third meta-level* includes knowledge with which to reason about overall design strategies: overall design strategies for the entire design process and overall strategies for RQS manipulation and DOD manipulation, respectively.

In the following sections, specifications of strategic knowledge and reasoning during the design of a house, given client requirements, environmental requirements and designer requirements, are presented. This example is based on a practical case of design in the

Netherlands. Strategic knowledge for the overall design process (at meta-level 3) is presented in Section 5, more specific strategic knowledge for requirement qualification set manipulation (at meta-level 2) in Section 6 and strategic knowledge for design object description manipulation (at meta-level 2) in Section 7.

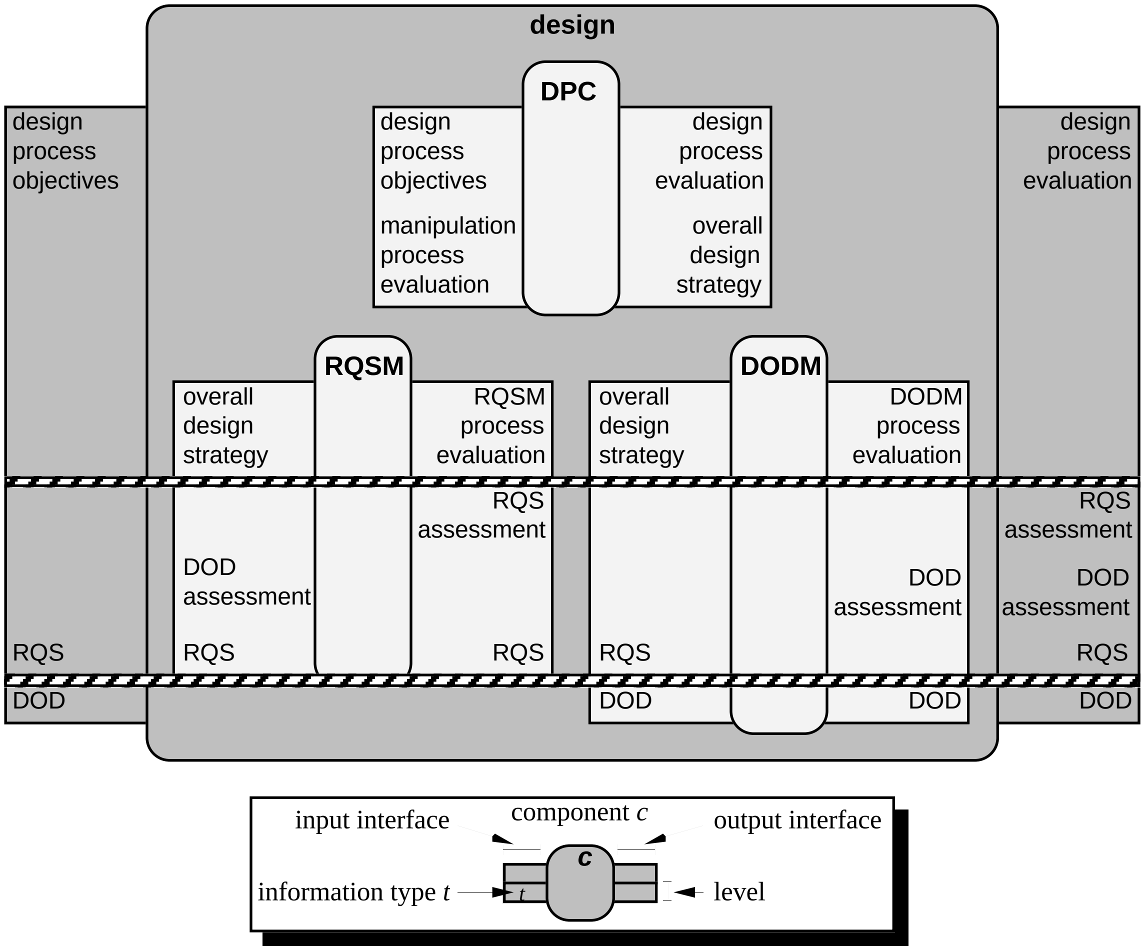

Figure 2. Levels of strategic knowledge in the generic task model of design.

5. STRATEGIC KNOWLEDGE FOR THE OVERALL DESIGN PROCESS

In the generic task model of design, the component design-process-coordination receives information about the design process itself: design process objectives (including requirements on the design process), status information about the requirement qualification set manipulation process, and status information about the design object description manipulation process. It determines an overall strategy for the design process. In this section, examples of strategic knowledge defined for this purpose are described and specified. The following example is used.

Example

The amount of time available for a design process is assumed to affect the overall strategy employed. If more than 100 hours are (still) available, some creative freedom is possible; if less time is available, a more practical approach is needed. The amount of time allocated to the design process (a requirement on the design process), and the amount of time still available are used to determine the overall design strategy.

5.1. Strategic Reasoning about the Overall Design Process

Part of the knowledge used to specify the overall strategy is the following:

if is_objective(max_processing_time(T1: Time_Stamp))
 and is_time_currently_spent(T2: Time_Stamp)
then is_time_currently_left(T1: Time_Stamp - T2: Time_Stamp);

if is_time_currently_left(T: Time_Stamp)
 and T: Time_Stamp ≤ 100:00:00
then is_possible_design_strategy(to_be_practical);

if is_time_currently_left(T: Time_Stamp)
 and T: Time_Stamp > 100:00:00
then is_possible_design_strategy(to_be_explorative);

if is_possible_design_strategy(S: Overall_Design_Strategy)
 and not is_rejected_design_strategy(S: Overall_Design_Strategy)
then is_best_design_strategy(S: Overall_Design_Strategy);

if is_best_design_strategy(S: Overall_Design_Strategy)
then is_current_design_strategy(S: Overall_Design_Strategy);

Depending on the input provided, this knowledge base determines the current overall design strategy. If, for example, the input includes:

is_objective(max_processing_time(200:00:00))
is_time_currently_spent(20:00:00)
not is_rejected_design_strategy(to_be_practical)
not is_rejected_design_strategy(to_be_explorative)

then the output includes:

is_current_design_strategy(to_be_explorative).

If the input, however, includes a different objective:

is_objective(max_processing_time(100:00:00))

then the output includes:

is_current_design_strategy(to_be_practical).

Note that the information and knowledge on which this reasoning process is based, is positioned at meta-level 3 in Figure 2. The implications of the choice of either being practical or being explorative for the manipulation of requirement qualification sets, and for manipulation of design object descriptions, are determined by additional strategic knowledge, discussed in Sections 5.2 and 5.3.

5.2. Reasoning about the Overall Strategy for Manipulation of Requirement Qualification Sets

An implication of being practical for the manipulation of requirement qualification sets could be to ignore a client's soft requirements and only take the client's other requirements into account. An implication of being explorative could be to re-negotiate requirements. This exemplary strategic knowledge, which can be used within the component RQSM at meta-level 3 (see Figure 2), is specified as follows:

```
is_supported_by_RQSM_strategy(to_be_practical,
  to_exclude_requirements_with_qualification(qlf(client, soft)));
is_supported_by_RQSM_strategy(to_be_explorative, to_renegotiate_requirements);

if  is_current_design_strategy(S1: Overall_Design_Strategy)
  and  is_supported_by_RQSM_strategy(S1: Overall_Design_Strategy, S2: RQSM_Strategy)
  and not  is_rejected_RQSM_strategy(S2: RQSM_Strategy)
then  is_best_RQSM_strategy(S2: RQSM_Strategy);

if  is_best_RQSM_strategy(S: RQSM_Strategy)
then  is_current_RQSM_strategy(S: RQSM_Strategy);
```

If the overall design strategy is to be practical, then the output of strategic reasoning with the above specified strategic knowledge includes:

```
is_current_RQSM_strategy(to_exclude_requirements_with_qualification(qlf(client, soft))).
```

Otherwise, if the overall design strategy is to be explorative, then the output includes:

```
is_current_RQSM_strategy(to_renegotiate_requirements).
```

5.3. Reasoning about the Overall Strategy for Manipulation of Design Object Descriptions

The implications of being practical or being explorative for the manipulation of the design object description are also determined by additional strategic knowledge. An implication of being practical could be to use an existing design. An implication of being explorative could be to generate a new design from scratch. This exemplary strategic knowledge, which can be used within the component DODM at meta-level 3 (see Figure 2), is specified as follows:

```
is_supported_by_DODM_strategy(to_be_practical, to_try_reusing_an_earlier_design);
is_supported_by_DODM_strategy(to_be_explorative, to_generate_a_design_from_scratch);

if  is_current_design_strategy(S1: Overall_Design_Strategy)
  and  is_supported_by_DODM_strategy(S1: Overall_Design_Strategy, S2: DODM_Strategy)
  and not  is_rejected_DODM_strategy(S2: DODM_Strategy)
then  is_best_DODM_strategy(S2: DODM_Strategy);

if  is_best_DODM_strategy(S: DODM_Strategy)
then  is_current_DODM_strategy(S: DODM_Strategy);
```

If the overall design strategy is to be practical, then the output of strategic reasoning with the above-specified strategic knowledge includes:

```
is_current_DODM_strategy(to_try_reusing_an_earlier_design).
```

Otherwise, if the overall design strategy is to be explorative, then the output of strategic reasoning with the above-specified strategic knowledge includes:

```
is_current_DODM_strategy(to_generate_a_design_from_scratch).
```

5.4. Evaluation of the Overall Design Strategy

After the overall design strategy has been determined and has provided input for the manipulation of requirement qualification sets and for the manipulation of design object descriptions, the resulting process is evaluated, and, if required, modified. In this specific example, this includes knowledge to determine whether the allocated amount of time has been used or not, as well as knowledge to determine whether the current design strategy has proved to be successful. This exemplary strategic knowledge, which can be used within the component DPC, is specified as follows:

if is_objective(max_processing_time(T1: Time_Stamp))
and is_time_currently_left(T2: Time_Stamp)
and T2: Time_Stamp ≥ 00:00:00
and is_objective(is_RQS_to_be_used(S1: RQS_Name))
and is_result_of_RQS_modification_to(S2: RQS_Name, S1: RQS_Name)
and is_solution_for(D: DOD_Name, S2: RQS_Name)
then is_fulfilled(max_processing_time(T1: Time_Stamp));

if is_objective(O: Design_Objective)
and is_fulfilled(O: Design_Objective)
and is_current_design_strategy(S: Overall_Design_Strategy)
then is_successfully_handled_by(O: Design_Objective, S: Overall_Design_Strategy);

6. STRATEGIC DECISIONS FOR THE MANIPULATION OF REQUIREMENT QUALIFICATION SETS

Strategic decisions related to the manipulation (determination of foci and modifications) of requirement qualification sets are modelled and specified explicitly. In this section an example is used to illustrate a few of the types of strategic knowledge involved in reasoning about the modification of requirement qualification sets. How this knowledge is used depends on the overall design strategy determined by design process co-ordination. The example shows a case, in which strategic knowledge is required for the selection of one of the alternatives generated for modification of the current RQS.

Example

A house is to be built on a plot with a road to its west. During preliminary design of this house one of the aspects considered is the volume of the house. The client specifies his/her needs and desires with respect to floor space and cost. In interaction with the architect, this is translated into a requirement for a volume of between 255 and 265 m^3. (Dutch architects use requirements for cubic meters, rather than square meters, as a basis for their designs.) Another aspect is the position of the front door. Given the location of the house, the client indicates a preference for the front door to face west (to provide easy access for guests). The last aspect for which the client provides input, concerns the outer walls: the outer walls are not to be built from synthetic material.

The architect, knowing that the prevailing wind comes from the west, would prefer the front door to face south. The two options for the position of the front door are related to two different criteria: the criterion of accessibility (initially put forward by the client) and the criterion of protection against out-door conditions (put forward during the design process by the architect). According to the first criterion, the best option would be for the front door to face west. According to the second criterion, the best option would be for the front door to face south. To make a strategic decision, knowledge is needed about which optimisation criterion is preferred. The architect has a preference for the criterion of protection. For the material of the outer walls the architect takes two criteria into account: durability and aesthetic value. The material brick scores best on durability, wood scores best on aesthetic value. The architect has a preference for durability above aesthetic value.

In this example, the requirement initially imposed by the environment is the following:

is_qualified_requirement(QR00, qlf(environment, hard), road_west)

The requirements initially put forward by the client are expressed as follows (where criteria are modelled as qualifications of the empty requirement tuple [ ]):

```
is_qualified_requirement(QR01, qlf(client, hard), volume_is_between_255_and_265m3)
is_qualified_requirement(QR02, qlf(client, hard), not(synthetic_outer_walls))
is_qualified_requirement(QR03, qlf(client, soft), front_door_west)
is_qualified_requirement(QR04, qlf(client, accessibility_optimality), [ ])
```

The requirements put forward by the designer during the design process are expressed as:

```
is_qualified_requirement(QR05, qlf(designer, soft), front_door_south)
is_qualified_requirement(QR06, qlf(designer, protection_optimality), [ ])
is_qualified_requirement(QR07, qlf(designer, accessibility_optimality), [ ])
is_qualified_requirement(QR08, qlf(designer, preferred_over),
  [protection_optimality, accessibility_optimality])
is_qualified_requirement(QR09, qlf(designer, durability_optimality), [ ])
is_qualified_requirement(QR10, qlf(designer, aesthetic_value_optimality), [ ])
is_qualified_requirement(QR11, qlf(designer, preferred_over),
  [durability_optimality, aesthetic_value_optimality])
```

The strategic knowledge needed to implement the two design strategies for the manipulation of requirement qualification sets described above in Section 5, namely to ignore the client's soft requirements, respectively to re-negotiate and extend the client's set of requirements, are described below in more detail.

6.1. A Practical Approach to Requirement Qualification Set Manipulation

The strategic knowledge specified in Section 5 for a time-constrained design process, translated the overall strategy to be practical into the strategy for the manipulation of requirement qualification sets to ignore all soft requirements put forward by a client. All environmental requirements and designer requirements are considered. The strategic knowledge needed to implement this approach is straight-forward.

From the set of initial qualified requirements imposed by the client on the design process, the subset of soft requirements is determined and explicitly marked as rejected. When the current requirement qualification set manipulation strategy includes:

```
is_current_RQSM_strategy(to_exclude_requirements_with_qualification(qlf(client, soft)))
```

then the following knowledge suffices to exclude soft client requirements:

```
if to_exclude_requirements_with_qualification(qlf(S: Source, Q: Qualification))
  and  is_current_qualified_requirement(QR: Qualified_Requirement,
          qlf(S: Source, Q: Qualification), T: Requirement_Tuple)
then  is_rejected_qualified_requirement(QR: Qualified_Requirement);
```

When the current requirement qualification set manipulation strategy also includes:

```
to_include_requirements_with_source(environment)
to_include_requirements_with_source(designer)
```

then the following knowledge can be used to select all environmental requirements and all designer requirements (with application of the Closed World Assumption on the predicate is_rejected_qualified_requirement):

```
if to_include_requirements_with_source(S: Source)
  and  is_current_qualified_requirement(QR: Qualified_Requirement,
```

```
            qlf(S: Source, Q: Qualification), T: Requirement_Tuple)
then   is_potentially_selected_qualified_requirement(QR: Qualified_Requirement);

if  is_potentially_selected_qualified_requirement(QR: Qualified_Requirement)
  and not  is_rejected_qualified_requirement(QR: Qualified_Requirement)
then   is_selected_qualified_requirement(QR: Qualified_Requirement);
```

Qualified requirements are marked as ‘potentially selected’ first, as at the same time (and for other reasons) they may also have been marked as rejected. This knowledge is used for reasoning at meta-level 2 in Figure 2 within the component RQSM, the level at which knowledge about strategic decisions on specific choices between qualified requirements is specified.

6.2. An Explorative Approach to Requirement Qualification Set Manipulation

The strategic knowledge specified in Section 5 for a non time-constrained design process, translated the overall design strategy to be explorative into the strategy for the manipulation of requirement qualification sets to re-negotiate (qualified) requirements. Since the client and the designer do not agree in their preference about the position of the front door, these requirements are good candidates for re-negotiation.

Reasoning about the choice of requirements at a given point in a design process requires knowledge specified at meta-level 2. For example, the strategic knowledge that if an overall best option (considering all relevant criteria) exists it is to be selected, can be specified as follows (with a Closed World Assumption on the predicate is_disqualified_criterion and by defining the sort Criterion as a subsort of the sort Qualification):

```
if  to_renegotiate_requirements
  and  is_current_qualified_requirement( QR: Qualified_Requirement, C: Criterion, [ ])
then  is_potentially_relevant_criterion(C: Criterion);

if  is_potentially_relevant_criterion(C: Criterion)
  and not  is_disqualified_criterion(C: Criterion)
then  is_relevant_criterion(C: Criterion);

if  to_renegotiate_requirements
  and  is_qualified_requirement( QR: Qualified_Requirement, Q: Qualification,
           T: Requirement_Tuple)
  and  T: Requirement_Tuple ≠  [ ]
then  is_potentially_selected_qualified_requirement(QR: Qualified_Requirement);

if  is_potentially_selected_qualified_requirement(QR1: Qualified_Requirement)
  and  is_potentially_selected_qualified_requirement(QR2: Qualified_Requirement)
  and  is_relevant_criterion(C: Criterion)
  and  entails_qualified_requirement_selection_ranking(C: Criterion,
           [QR1: Qualified_Requirement, QR2: Qualified_Requirement])
then   is_rejected_qualified_requirement(QR2: Qualified_Requirement);
```

Here the knowledge about comparison of qualified requirements for a given criterion is assumed to be provided by an RQS assessment sub-task. Application of the Closed World Assumption to the predicate is_rejected_qualified_requirement provides a set of non-rejected candidates for ‘overall best option’. If this set is not empty, an overall best option can be selected from this set (by means of the knowledge specified at the end of Section 6.1). If an overall best option exists, then by strategic reasoning with the knowledge specified above, an option will be determined that is optimal with respect to all relevant criteria. Note that in this case no knowledge on preferences between criteria is required.

If no optimal option exists (i.e., if each option is beaten by another option on at least one relevant criterion), then only a pareto-optimal option can be selected (i.e., an option for which no other option exists which beats it for at least one criterion and which is not beaten on all other criteria). Strategic knowledge to determine such a pareto-optimal option can take into account preferences between criteria, as is specified below:

```
if  is_potentially_selected_qualified_requirement(QR1: Qualified_Requirement)
  and  is_potentially_selected_qualified_requirement(QR2: Qualified_Requirement)
  and  is_potentially_relevant_criterion(C1: Criterion)
  and  is_potentially_relevant_criterion(C2: Criterion)
  and  is_current_qualified_requirement(
         QR: Qualified_Requirement, preferred_over, [C1: Criterion, C2: Criterion])
  and  entails_qualified_requirement_selection_ranking(C1: Criterion,
         [QR1: Qualified_Requirement, QR2: Qualified_Requirement])
  and  entails_qualified_requirement_selection_ranking(C2: Criterion,
         [QR2: Qualified_Requirement, QR1: Qualified_Requirement])
then  is_disqualified_comparison_criterion_for(C2: Criterion,
         QR1: Qualified_Requirement, QR2: Qualified_Requirement);
```

Note that the sort Criterion is also defined as a subsort of the sort Requirement. To yield a ranking of qualified requirements, the following knowledge can be used (with application of a Closed World Assumption on the predicate is_disqualified_comparison_criterion_for and the predicate entails_qualified_requirement_selection_ranking):

```
if  is_potentially_selected_qualified_requirement(QR1: Qualified_Requirement)
  and  is_potentially_selected_qualified_requirement(QR2: Qualified_Requirement)
  and not  is_disqualified_comparison_criterion_for(C: Criterion,
              QR2: Qualified_Requirement, QR1: Qualified_Requirement)
  and not  entails_qualified_requirement_selection_ranking(C: Criterion,
              [QR2: Qualified_Requirement, QR1: Qualified_Requirement])
then is_as_least_as_good_as(QR1: Qualified_Requirement, QR2: Qualified_Requirement);
```

To remove qualified requirements that are ‘worse’ than others, the following knowledge can be used (with application of a Closed World Assumption on is_as_least_as_good_as):

```
if  is_potentially_selected_qualified_requirement(QR1: Qualified_Requirement)
  and  is_potentially_selected_qualified_requirement(QR2: Qualified_Requirement)
  and not  is_as_least_as_good_as(QR1: Qualified_Requirement,
             QR2: Qualified_Requirement)
then  is_rejected_qualified_requirement(QR1: Qualified_Requirement);
```

If no overall best solution exists, strategic reasoning with this knowledge (and the knowledge specified at the end of Section 6.1) determines as a solution a requirement that is assessed to be best for a most preferred criterion, given a choice between a number of soft requirements and a preference relation between the relevant criteria.

In the above example, initially three client requirements are considered: one on the volume of the house, one on the material for the outer walls, and one on the position of the front door. The environmental requirement on the position of the plot and the designer’s requirements as specified in Section 6.1 are also considered. The architect can re-negotiate the client’s preference on the basis of these criteria. In the example of the position of the front door, the preference of protection over accessibility is agreed, so the output of this negotiation process is the architect’s preference, namely that the front door faces south.

Reasoning at meta-level 2 determines that the architect’s requirement with respect to the position of the front door is selected:

is_selected_qualified_requirement(QR05)

and that the client's requirement with respect to the position of the front door is dropped. All other client requirements are selected.

7. STRATEGIC DECISIONS FOR THE MANIPULATION OF DESIGN OBJECT DESCRIPTIONS

Strategic decisions related to the modification (determination of foci and modifications) of design object descriptions are also modelled and specified explicitly. In this section the same example used above in Section 6 is used to illustrate a few of the types of strategic knowledge involved in reasoning about the modification of design object descriptions. How this knowledge is used depends on the overall design strategy determined by design process co-ordination.

The strategic knowledge needed to implement the two possible design strategies for the manipulation of design object descriptions described above in Section 5, namely to re-use an existing design, if possible, respectively to design from scratch (depending on the design process objectives), are described below in more detail.

7.1. A Practical Approach to Design Object Description Manipulation

The strategic knowledge specified in Section 5 for a time-constrained design process, translated the overall design strategy for the entire design process to be practical into the need to re-use an existing design, if possible, for the manipulation of design object descriptions. This design strategy, together with the requirements (derived using a strategy ignoring the client's soft requirements; see Section 6), are input for the DODM process. The strategic knowledge needed to implement this strategy could rely on case-based reasoning, using all requirements to index retrieval. In this example, however, for reasons of explanation, retrieval from the library of existing design object descriptions is based on the environment requirements only. On the basis of the environment requirement

is_qualified_requirement(QR00, qlf(environment, hard), road_west)

an existing design is retrieved from the library that indeed has the right orientation for a plot with an adjacent road directly to its west. The following knowledge at meta-level 2 is used to specify this choice:

```
if  to_try_reusing_an_earlier_design
  and  is_current_qualified_requirement(QR: Qualified_Requirement,
           qlf(environment, hard), [R: Requirement])
  and  exists(DOD1: DOD)
  and  satisfies(DOD1: DOD, R: Requirement)
then  candidate_DOD_to_be_retrieved(DOD1: DOD);
```

Existing designs are examined, one-by-one, to determine whether or not they fulfil the other requirements. A number of existing designs are found for houses with a volume of approximately 260 m^3, but not one for a house with a volume of approximately 260 m^3 and non-synthetic outer walls. The following possible modification options are generated for the first design of a house found with a volume of between 255 and 265 m^3:

is_potentially_selected_DOD_option(is_made_of(outer_walls, wood))
is_potentially_selected_DOD_option(is_made_of(outer_walls, brick))

The choice of material depends on the preference relation between the criterion of maximal durability and the criterion of aesthetic value. In this example this choice is made using the architect's preference for durability, specified in Section 6, and relevant strategic knowledge. Strategic knowledge that if an overall best option exists (i.e., best in all relevant criteria) it is to be selected, is specified using knowledge (and a closed world assumption on some of the predicates) very similar to the knowledge depicted in Section 6.2.

Strategic knowledge also includes knowledge that specifies that if not one of the options is the overall best option then the acceptable options are to be determined (based on preferences between criteria). This is also specified by knowledge very similar to the knowledge depicted in Section 6.2.

In this case the choice for durability results in the choice for the option with brick outer walls. Note that in this variant of the design process, no requirements are imposed on the position of the front door. The position of the front door is determined by the position of the front door in the retrieved design; the door may, for example, face south.

7.2. An Explorative Approach to Design Object Description Manipulation

The strategic knowledge specified in Section 5, for a non time-constrained design process, translated the overall design strategy to be explorative into the overall DODM strategy to design from scratch for the manipulation of design object descriptions. The requirements derived using the strategy to re-negotiate requirements in Section 6, are assumed to be imposed on this process.

During preliminary design a decision has to be made whether a bungalow or a two-storey building is preferred. Given the requirement that the total volume be between 255 and 265 m^3, the choice is between a bungalow with a floor area of 100 m^2 and a two-storey house with a floor area of 50 m^2 (in both cases assuming a floor height of 2.60 m).

The relative importance of the two criteria involved, average floor space/room and insulation value, is crucial. If the first criterion is more important, the best option is the bungalow (because in a bungalow no space needs to be reserved for a staircase). If the second criterion is more important, the best option is the two-storey house (because the sum of the outer wall area and the roof area of the two-storey house is less than that of the bungalow). To obtain information about these optimisation criteria and the preference between them, the RQS manipulation process becomes active. The following (soft) designer requirements are selected:

```
is_current_qualified_requirement(QR12, qlf(designer, room_area_optimality),  [ ]))
is_current_qualified_requirement (QR13, qlf(designer, insulation_optimality,  [ ]))
is_current_qualified_requirement (QR14, qlf(designer, preferred_over,
  [insulation_optimality, room_area_optimality]))
```

The strategic DOD modification knowledge determines, in this case, that a bungalow is preferred.

The only requirement with respect to material choice, is the requirement that the outer walls be made from non-synthetic material. Given the strategic knowledge on material choice and the architect's preference for durability (see Section 6), the choice for brick outer walls, is made.

The required position of the front door, namely facing south, is one of the main factors involved in the process of determining the position of the building on the plot.

8. DISCUSSION

Design entails strategic reasoning at different levels, for example, to determine requirements on the design process itself (design process objectives), to determine preferences between options and criteria, to assume values for specific attributes and to choose between design options. In addition to object-level interaction, three (meta-)levels of strategic reasoning are distinguished in this paper. As strategic interaction with the designer may be desirable at each of these levels, design support systems need to be designed to support such interaction.

At the highest level, design process objectives are determined, as are the implications for the overall design strategy for the entire design process, the manipulation of requirement qualification sets, and the manipulation of design object descriptions. Requirements on the design process, such as time constraints imposed by a client, may be acquired in interaction with the designer.

At the next level down, strategic reasoning about requirements determines which requirements are to be considered, following a given overall RQS manipulation strategy (determined one level higher). Interaction with the designer is possible on, for example, preference relations between requirements, inconsistent requirements, and a preferred focus.

The same strategic level determines which aspects of a design object are to be considered, following a given overall DOD manipulation strategy (determined one level higher) and a set of requirements and their qualifications. Interaction with the designer may include selection of a most preferred design or a preferred focus.

Note that each level of reasoning influences the lower levels of reasoning. The highest level determines the overall design process strategy and the implications for the overall strategy for RQS manipulation and DOD manipulation. The next level down determines which (modifications of) requirements and qualifications, and which (modifications of) design object descriptions, are to be considered, given the overall RQS manipulation strategy and the overall DOD manipulation strategy.

The corresponding meta-levels of strategic knowledge are discussed and illustrated in this paper for the generic model of design. In the literature on meta-level architectures such as [22, 23, 24, 25], meta-levels in models of design tasks are not addressed. The compositional approach DESIRE provides a means to explicitly model and specify such knowledge, as well as the strategic reasoning involved. Examples of strategic knowledge at each of these levels are specified, illustrating the flexibility that meta-representations provide.

REFERENCES

[1] Brazier, F.M.T., van Langen, P.H.G., and Treur, J., Strategic knowledge in compositional design models. *In:* J.S. Gero and F. Sudweeks, Eds, *Artificial Intelligence in Design '98*, Dordrecht, The Netherlands, Kluwer Academic Publishers, 1998, pp. 129-147.

[2] Ohsuga, S., Framework of knowledge-based systems: multiple meta-level architecture for representing problems and problem solving processes. *Knowledge Based Systems*, 1990, **3**(4), 204–214.

[3] Oshuga, S., Strategic knowledge makes knowledge based systems truly intelligent. *In* L. Candy and K. Hori, Eds, *Proc. First Int. Workshop on Strategic Knowledge and Concept Formation*, Loughborough University, UK, 1997.

[4] Hori, K., Where is, what is and how can we use strategic knowledge. *In* L. Candy and K. Hori, Eds, *Proc. First Int. Workshop on Strategic Knowledge and Concept Formation*, Loughborough University, UK, 1997.

[5] Sumi, Y., Supporting the acquisition and modelling of requirements in software design. *In:* L. Candy and K. Hori, Eds, *Proc. First Int. Workshop on Strategic Knowledge and Concept Formation*, Loughborough University, UK, 1997.

[6] Brazier, F.M.T., Dunin-Keplicz, B., Jennings, N.R. and Treur, J., Formal specification of multi-agent systems: a real-world case. *In* V. Lesser, Ed., *Proc. First Int. Conference on Multi-Agent Systems (ICMAS'95)*, Cambridge, MA, MIT Press, 1995, 25–32. *Extended version in* M. Huhns and M. Singh, Eds, *Int. J. Cooperative Information Systems*, Special Issue on Formal Methods in Cooperative Information Systems: Multi-Agent Systems, 1997, **6**, 67–94.

[7] Brazier, F.M.T., Treur, J., Wijngaards, N.J.E. and Willems, M., Temporal semantics of compositional task models and problem solving methods. *Data and Knowledge Engineering*. In press, 1998.

[8] Brazier, F.M.T., Jonker, C.M., and Treur, J., Principles of compositional multi-agent system development. *In*: J. Cuena, Ed., *Proc. of the IFIP'98 Conference on Information Technology and Knowledge Systems, IT&KNOWS'98*, Chapman and Hall. In press, 1998.

[9] Ford, K.M., Bradshaw, J.M., Adams-Webber, J.R. and Agnew, N.M., Knowledge acquisition as a constructive modeling activity. *In* K.M. Ford and J.M. Bradshaw, Eds, *Int. J. Intelligent Systems*, Special Issue on Knowledge Acquisition as Modeling, 1993, **8**(1), 9–23.

[10] Brazier, F.M.T., Treur, J. and Wijngaards, N.J.E., Modelling interaction with experts: the role of a shared task model. *In* W. Wahlster, Ed., *Proc. Twelfth European Conference on AI (ECAI'96)* ,Chichester, UK, John Wiley and Sons, 1996, 241–245.

[11] Akin, Ö. How do architects design? *In* J.-C. Latombe, Ed., *Artificial Intelligence and Pattern Recognition in Computer Aided Design*, Amsterdam, The Netherlands, North-Holland, 1978, 65–119.

[12] Brown, D.C. and Chandrasekaran, B., *Design Problem Solving: Knowledge Structures and Control Strategies*. Research Notes in Artificial Intelligence, London, UK, Pitman, 1989.

[13] Pahl, G. and Beitz, W., *Engineering design*. New York, NY, Springer-Verlag, 1984.

[14] Schön, D.A., *The Reflective Practitioner: How Professionals Think in Action*. London, UK, Temple Smith, 1983.

[15] Smithers, T., Corne, D. and Ross, P., On computing exploration and solving design problems. *In* J.S. Gero and E. Tyugu, Eds, *Formal Design Methods for CAD*, Amsterdam, The Netherlands, Elsevier, 1994, 293–313.

[16] Brazier, F.M.T., van Langen, P.H.G., Ruttkay, Zs. and Treur, J., On formal specification of design tasks. *In* J.S. Gero and F. Sudweeks, Eds, *Artificial Intelligence in Design '94*, Dordrecht, The Netherlands, Kluwer Academic Publishers, 1994, 535–552.

[17] Brazier, F.M.T., van Langen, P.H.G. and Treur, J., Modelling conflict management in design: an explicit approach. *In* I.F.C. Smith, Ed., *AIEDAM*, Special Issue on Conflict Management in Design, 1995; **9**(4), 353–366.

[18] Brazier, F.M.T., Langen, P.H.G. van, Treur, J. and Wijngaards, N.J.E., Redesign and reuse in compositional knowledge-based systems. *In* P.M. Wognum and I.F.C. Smith, Eds, *Knowledge Based Systems*, Special Issue on Models and Techniques for Reuse of Designs,1996, **9**, 105–118.

[19] Brazier, F.M.T., Langen, P.H.G. van, Treur, J., Wijngaards, N.J.E. and Willems, M., Modelling an elevator design task in DESIRE: the VT example. *In* A.Th. Schreiber and W.P. Birmingham, Eds, *Int. J. Human-Computer Studies*, Special Issue on Sisyphus-VT, 1996, **44**, 469–520.

[20] Brazier, F.M.T., Langen, P.H.G. van and Treur, J., A compositional approach to modelling design rationale. *In* P.W.H. Chung and R. Bañares-Alcántara, Eds, *AIEDAM*, Special Issue on Representing and Using Design Rationale, 1997, **11**(2), 125–139.

[21] Haroud, D., Boulanger, S., Gelle, E. and Smith, I.F.C., Strategies for conflict management in preliminary engineering design. *In* Smith, I.F.C., Ed., *Proc. AID '94 Workshop on Conflict Management in Design*, 1994.

[22] Davis, R., Meta-rules: Reasoning about control. *Artificial Intelligence*,1980, **15**, pp. 179-222.

[23] Harmelen, F. van, Wielinga, B., Bredeweg, B., Schreiber, G., Karbach, W., Reinders, M., Vos, A., Akkermans, H., Bartsch-Sporl, B., and Vinkhuyzen, E., Knowledge-Level Reflection. In: B. Le Pape, and L. Steels, Eds., *Enhancing the Knowledge Engineering Process -- Contributions from ESPRIT*, Elsevier Science, Amsterdam, The Netherlands, 1992, pp. 175-204.

[24] Maes, P., Nardi, D., Eds, *Meta-level architectures and reflection*, Elsevier Science Publishers, 1988.

[25] Weyhrauch, R.W., Prolegomena to a theory of mechanized formal reasoning, *Artificial Intelligence*, 1980, **13**, pp. 133-170.