KMi Technical Report

KMi–TR–107

TOWARDS A FRAMEWORK FOR ACQUISITION OF DESIGN NOWLEDGE

Martin Dzbor, Zdenek Zdrahal

Knowledge Media Institute
The Open University

To appear in: 27th ASME DETC Design Automation Conference,
Pittsburgh, USA, September 2001

April 2001

Proceedings of DETC'01
ASME 2001 Design Engineering Technical Conferences
and Computers and Information in Engineering Conference
Pittsburgh, Pennsylvania, September 9-12, 2001

DETC2001/DAC–21049

TOWARDS A FRAMEWORK FOR ACQUISITION OF DESIGN KNOWLEDGE

Martin Dzbor and Zdenek Zdrahal
Knowledge Media Institute
The Open University
Milton Keynes MK7 6AA, United Kingdom
Tel. +44 1908 858524 Tel. +44 1908 654512
Email: M.Dzbor@open.ac.uk Email: Z.Zdrahal@open.ac.uk

ABSTRACT

Engineering design is a knowledge-intensive process driven by various design objectives. Design is an iterative process where the objectives evolve together with the solutions in order to deliver an artefact with the desired properties and functions. Many design theories developed so far suggest more or less efficient ways for finding a suitable solution to the given goals. However, they often leave open the issue of '*solution talkback*'. Discovery of new design objectives and amendment of the existing ones is as important as the development of design solutions. The biggest issue with solution talkback is the presence of tacit knowledge in addition to the explicit one. This paper draws on a theory that incorporates some typical features of design problems, and transfers theoretical findings about reflection on the design actions to a tool for acquisition of design knowledge. First, key terms are defined and theoretical framework is introduced. Afterwards we look at the means for capturing explicit and tacit design knowledge more in depth.

1 INTRODUCTION

According to Simon [1] design is an ill-structured problem; i.e. its initial specification is usually incomplete; the initial vagueness prevents the designers from constructing a precise problem solving space and setting clear criteria for determination of a (final) solution. Due to the initial uncertainty, the problem space for a design task is not existing objectively in advance but must be constructed on the fly. The sheer amount of possible combinations of the primitive design elements significantly contributes to the 'hit-and-miss' nature of the design problem solving. Dynamic features, such as trials, errors, dead ends and consecutively backtracks are much more typical for design than any algorithmic determinism or productivity.

In our view design is a sequential process [2] with features similar to those discussed by Iwasaki, Gero, Chandrasekaran, and others, who claim that design is results from an interplay between functional and structural objects [3,4,7]. Following tenets summarise the key points of our theory:

- a problem is specified as a set of design requirements and constraints referring to functions and properties of design elements known in a particular domain
- a solution is developed in terms of structural elements and relations among them assuring the proposed structure meets the desired functions and/or properties
- knowledge about the structural objects delivering certain functions and having certain properties can be expressed by sentences of the first order logic

The observations made in this paper draw upon results from an experimental study conducted in summer 2000. The participants in the study were two experienced designers who were solving tasks taken from the domain of large-scale systems controller design. We were looking at how they formulate the design problem step-by-step in multiple changing contexts, and how they justify and develop their decisions informally. One of the objectives of the study was to evaluate the feasibility of methodology and toolkit for capturing design knowledge in a semi-structured form; more details are in sections 5 and 6.

2 DESIGN KNOWLEDGE IS TACIT

Our theory of sequential design is based on the first order logic for the sake of simplicity, but also other sufficiently powerful representations may be used instead. The theory builds on top of basic logical axioms and deductive rules [5], and initially refines only one – a free formula $\beta(x)$ is usually understood as its universally quantified closure $\forall x: \beta(x)$. More precisely it is the universal closure applied to all objects in a certain *conceptualisation* of the real world. Thus a free formula is always understood *as* $\forall x: \beta(x) \mid conceptualisation=Conc_n$.

We define Γ as *all* design objectives that may be demanded from, or must not be violated by the designed artefact. Γ is a *tacit* entity that does not have a definite structure and in general is 'unstructurable'; however it objectively exists and has an immense influence on design [6]. Tacit knowledge is seen as something that is inherently present and used when tackling a problem, though we may not be able to express or explain it

explicitly. Original illustrative example from M. Polanyi[1] describes cyclists who are able to stay upright on a bike, ride it, and still cannot say how exactly they turn the handlebars so that they keep the balance and do not fall. They may attempt to deliver some explicit explanation but this will be insufficient for a novice cyclist to learn cycling. Thus, cycling involves both explicit and tacit knowledge. They complement each other, and typically, one type can be used to acquire the other one but it is rarely a simple transformation [2, 6]!

In the case of design problems, explicit knowledge consists of various algorithms, methods and methodologies, logical, mathematical or physical models. On the other hand, tacit knowledge prevails in the situations when designers talk about 'liking a solution'; they are not able to express what exactly is causing their attitude only that there is a tacit feeling of a hidden flaw. The logical framework as detailed in [2] suggests two new operators formalising the position of tacit knowledge. First, $\Gamma \textcircled{\models} \gamma$ replaces the logical consistency ($\models$) and means that sentence γ is '*consistent*' with tacit knowledge Γ. In other words it is 'acceptable' with respect to tacit knowledge.

Similarly, operator denoting logical inference ($\vdash$) assumes an explicit logical theory and does not account for influences outside of logic. Corresponding 'tacit' operator $\Gamma \textcircled{\vdash} \gamma$ says that sentence γ is 'inferred' using the tacit experience. In line with [6], tacit 'inference' is an attempt of a designer to formulate his or her tacit feeling in explicit terms. Such a formulation is not strictly logical but nevertheless very useful in creative design.

Tacit and explicit design specifications are also related. We distinguish an *explicit design specification*, which consists of *explicit* goals (G) and constraints (C), from the unstructured *tacit* 'specification'. Tacit expectations from the designed may artefact help to generate new explicit goals and/or constraints (Eq. 1). It is obvious that a solution s that is tacitly consistent is also explicitly (i.e. logically) consistent with the explicit problem specification $G \cup C$ (Eq. 2) but not vice-versa!

$$\Gamma \textcircled{\models} G \cup C; \qquad \text{Eq. 1}$$

$$(\Gamma \textcircled{\models} G \cup C) \wedge (\Gamma \textcircled{\models} s) \Rightarrow (G \models s) \wedge (s \cup C \not\vdash \bot) \qquad \text{Eq. 2}$$

Let us denote the current explicit goals and constraints in the design step i as G_i and C_i, and the solution corresponding to the explicit specification as s_i. A combination of design elements is a solution in the given step if it satisfies the current problem specification; i.e. all explicit goals and constraints:

$$\forall s_i \in E^*: (G_i \models s_i) \wedge (s_i \cup C_i \not\vdash \bot) \Leftrightarrow \textit{maybe-solution}(s_i) \qquad \text{Eq. 3}$$

Note that the explicit (logical) consistency (Eq. 3) does not guarantee that a solution is also consistent 'tacitly'. Eq. 2 cannot be simply reversed to imply tacit consistency from the explicit one. Designer's 'tacit' satisfaction with the solution s_i ($\Gamma \textcircled{\models} s_i$) may thus serve as the missing stop condition of the design process. The explicit goals and constraints in step i would then act as a *sufficient design specification* and s_i as an *acceptable solution*. Nevertheless, 'sufficiency' and 'acceptability' are concepts very distant from the 'optimality', and design is more about the 'acceptability' than any 'optimality'.

[1] Term 'tacit knowledge' was introduced by Michael Polanyi in 1963; the example with cycling is borrowed from [6].

3 REASONING MODES IN DESIGN

Design is ill structured not only in respect to the problem space but also with regard to the reasoning strategies used to navigate in such a space. An overview of different reasoning modes in design is summarised in [7]. This section briefly summarises the positions of various logical operations in reasoning on a knowledge level, and highlights major issues of such reasoning. More details are published in the complementary paper that focuses on the theoretical aspect [2].

Step I: Logical abduction ($G, A \Rightarrow G$) $\vdash A$

Abduction is reasoning when some outcomes are desired and we are interested in finding means for achieving the desired objectives. The basic principle of abduction in design can be illustrated as follows. Assume there is a set of desired design goals $F(x) \subseteq G$. In abduction, the designer looks for an artefact $A(x)$ that implies the desired functionality $F(x)$ consistently with the remaining design goals and constraints (i.e. $G \not\models A(x)$ and $A(x) \cup C \not\vdash \bot$). Presence of a formula of the type $A(x) \vdash F(x)$ in the logical theory concludes that artefact $A(x)$ has sufficient means for the satisfaction of desired functionality $F(x)$, and as such may be considered a (partial) design solution (see Eq. 3).

Abduction takes the explicit design goals and investigates how these may be achieved. It discovers the sufficient conditions for achieving a desired effect; however, it must be noted that a solution that is 'sufficient now' does not have to be acceptable in a longer run! Abduction makes a tentative commitment that can be whenever abandoned for a better one. This is an approach addressing especially well the exploratory nature of the design and on-the-fly construction of a problem space.

Step II: Logical deduction ($A, A \Rightarrow D$) $\vdash D$

After an artefact $A(x)$ was found that satisfied certain explicit goals G designers may be interested in knowing the additional implications of using artefact $A(x)$ as a (partial) solution to the design problem. In other words, they may want to deduce the consequences of the solution discovered by the abduction. Such a knowledge may be acquired by deploying rules of the type $A(x) \Rightarrow D(x)$, or in general terms $A(x) \vdash D(x)$. The main purpose of deduction according to the deduction theorem [5] is to generate new knowledge about the existing elements.

The logical consistency between $A(x)$ and design goals G does not change during deduction but in addition to it, artefact $A(x)$ is now consistent also with some additional functions $D(x)$ that were not among the initial ones (i.e. $D(x) \not\subset G$). However, very little can be deduced about the consistency between the consequence $D(x)$ and the remaining design goals G. From this point, the uncovered consequence of the current solution in the space of functions and properties may or may not be acceptable with regard to the tacit design knowledge. And here begins the interesting and 'ill structured' part of a reflective design.

Step III: Evaluation of (tacit) consistency (A, D) $\textcircled{\vdash} D \vee \neg D$

There are formal means already available for assessing the *logical* consistency between the discovered consequence $D(x)$ and the existing explicit problem specification $G \cup C$. Such an assessment would be purely logical and could be performed by some truth maintenance system [8]. We skip this logical assessment for a while and direct our attention to the assessment

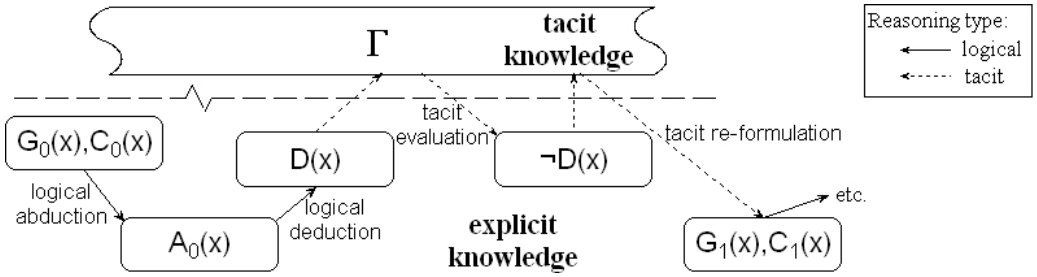

Figure 1. Interplay between explicit and tacit modes of reasoning in design

of 'tacit consistency' between the deduced consequence *D(x)* and our tacit design expectations. As we mentioned earlier, we may be able to use the tacit knowledge but not say how exactly we do it. The designer may thus tacitly appreciate the uncovered consequence, and may find that:

a. *D(x)* is an irrelevant consequence and does not have any influence on the subsequent design.
b. *D(x)* is a relevant and desirable feature referring to an implicit design requirement not mentioned in the original specification. It is consistent with tacit expectations and may be added to the current explicit design specification.
c. *D(x)* is an undesirable feature of a tentative solution and should be avoided in the further design. Thus its opposite ¬*D(x)* is intuitively noted as the necessary condition of a solution acceptability. However, this intuitive inclusion makes the logical theory inconsistent!

The former two situations do not bring any strikingly new knowledge into design. On the contrary, the last one not only uncovers a new 'goal' ¬*D(x)* but also introduces inconsistency into the logical theory (¬*D(x)* is demanded and *D(x)* is provable). We understand this situation as tacitly inconsistent because we arrived at it through the tacit appreciation of the current solution and specification. Before we can proceed any further, the (tacit) inconsistency must be removed because eventually $\Gamma \Subset A(x)$ must hold for any acceptable solution[2].

4 DESIGN TASKS ARE EVOLVING

The cause of the (tacit) inconsistency in the design theory may have its roots in any of the 'logical' reasoning steps depicted in Figure 1 and it may be removed in any of them. New objectives, constraints and obviously solutions could be formulated in order to remove the undesired inconsistency[3]. We say that the design problem is *re-formulated*. The available re-formulations may be logical or may include a tacit element. Below we discuss briefly a logical fix for the *abduction*, and two tacit fixes – for the *evaluation* and the *conceptualisation*.

4.1 Re-formulation of logical abduction

Assuming that the formulation of the requirements $F(x) \subseteq G$ is correct apart from their possible incompleteness or vagueness, and the design theory contains several rules with *F(x)* as an implicant, the abduction may have applied a wrong rule that eventually led to a tacit inconsistency. The unsuitability of that particular rule was discovered tacitly later in the design. There was no knowledge available during the abduction phase that would have discriminated between several alternative rules.

Step IV/A: Alternative abduction $(G, D, B \Rightarrow G) \vdash B$

After the results of the abduction had been tacitly appreciated, the missing condition ¬*D(x)* has been made explicit, and it may serve as a discriminator of 'wrong', potentially inconsistent abductive rules. We define that artefact *B(x)* is a functionally alternative artefact to *A(x)* when the following holds:

$$\neg(B(x) \equiv A(x)) \wedge (B(x) \vdash G(x)) \wedge (\neg(B(x) \vdash D(x))) \qquad \textbf{Eq. 4}$$

In other words, *B(x)* is an alternative to *A(x)*, when it is different from *A(x)* and does not imply the negative feature *D(x)*. Note that it does not mean that *B(x)* must imply the opposite – $B(x) \vdash \neg D(x)$! From the strategic point of view, alternative abduction is basically the old, traditional backtracking from an inconsistent state to the last consistent state. In plain words the formula in Eq. 4 is formal version of sentence '*When a solution is not working as expected, go back and find a different one.*'

4.2 Re-formulation of tacit evaluation

More interesting situation may happen when designers do not want to give up the current solution and backtrack. Often they try to conjecture a condition that when satisfied would restore the consistency of the design albeit in a restricted form. This approach is based on the fact that we rarely can describe any system in its entire complexity. Therefore, some new assumptions can be identified that would allow only some of the existing theorems to enter the reasoning.

Step IV/B: Solution restriction $(A, D) \Subset (P \Rightarrow A')$

From the previous steps we have an artefact *A(x)* as a potential design solution, and among its consequences is also *D(x)* that is from some reasons not desirable. Designer wants to avoid this potentially dangerous consequence and still build on the current solution *A(x)*. The deduction $A(x) \vdash D(x)$ may be restricted by a condition *P(x)*, e.g. according to the following schema:

$$(P(x) \Rightarrow (A(x) \wedge P(x))) \nvdash D(x) \qquad \textbf{Eq. 5}$$

In plain language Eq. 5 reads: '*If condition P(x) was satisfied then it could restrict current solution A(x) so as the undesired consequence D(x) is no more observed.*' The purpose of such a conjecture is to move from a universal deduction to the assumption-based one. An assumption is basically a tentatively accepted condition upon which deductions can be made simi-

[2] The stop condition of every design is the designer's tacit satisfaction with the solution and the artefacts used; see discussion in section 2.
[3] Also Schön [13] talks about unexpected surprises with the current solutions and modifications of the problem solving frames.

larly as with other 'proper' axioms. The advantage shows when an assumption later contradicts some other assumptions or theorems. It can be simply cancelled, and the consistency is restored once again.

As we explain in [2] an assumption can be added in a form of a new requirement or a new constraint. These two forms differ on a conceptual level rather than formal; they may look similarly but are used differently. A conjecture in Eq. 5 can be seen as a constraint conditioning the appearance of a certain feature as a consequence of some premise. A requirement can be easily made from such a constraint, when demanding that the premise must be always satisfied. Thus a constraint is not violated also when the premise is not valid; however, the requirement would be satisfied only if the premise is satisfied and simultaneously the consequence holds.

Tacit fixation through the evaluation is definitely a form of non-monotonic reasoning because it introduces new formulae to different contexts of the logical theory. From our experience it is a useful and significant strategy for coping with vagueness of the design problems.

4.3 Tacit re-conceptualisation (Step V)

Both fixes described above – the logical one in the abduction and tacit one in the evaluation, focus on the amendment of a design solution. The designer either restricts the known solution, or backtracks to find an alternative one. These fixes typically introduce a new condition that discriminates between plausible and inconsistent alternatives or sub-parts. In the unfortunate case, when there are no alternatives or restrictions available in the domain theory, it may suggest that the theory itself is incomplete and is not powerful enough to describe all the objects we would otherwise need in order to resolve the deadlock. Therefore, we shall now look at a mechanism that would not only modify the usage of available knowledge about the current conceptual objects but as well allow for a generation of new concepts for the logical theory of design.

Assuming the 'less rigorous' abduction was reasonable, we are forced to admit that a flaw may have occurred in the more rigorous deduction. However, the deduction simply has to produce the same result whenever it is given the same initial conceptual axiomatisation of the world. If the deduction rules are impenetrable then a flaw must lie in the actual conceptualisation. In other words, our conceptual apparatus is not powerful enough to represent our world, and we have to modify the current apparatus that allowed tacit contradictions.

What we desire, is a new set of conceptual objects (let us denote them as x') for which $A(x') \vdash G(x')$, and $A(x') \models G(x')$ holds but the sentence $A(x') \vdash D(x')$ looses its original meaning; for the new objects is irrelevant. The questions that appear with such a desire include the following:

- What possible conceptualisation shall be chosen?
- What objects constitute the 're-conceptualised' world?
- How can be appreciated suitability of an object for the re-conceptualisation when its details are not known in the current conceptual world? etc.

Though we are not able to directly name the new conceptual objects, we can set boundaries where to find them. Also we may set the conditions they should satisfy. From tacit knowledge we know at least some properties the new conceptual objects shall exhibit (e.g. given requirements G) or avoid (e.g. undesired feature $D(x)$). Since knowledge used for the discovery of new concepts is tacit, the descriptive properties would be only partial and incomplete.

Now we need to change the conceptual domain of x; instead of $\forall x \in Conc_1$ we want to use $\forall x \in Conc_2$. Since we cannot make a statement *about* a theory involving re-conceptualisation of its axioms *within* the theory, we must refer to another, complementary reasoning strategy. We may refer to *analogy* as another such reasoning mechanism that works on a different basis than a purely logical approach. Analogous reasoning often relies on the available tacit knowledge and may use 'tacit' similarity between two cases to derive some explicit conclusion [9].

Reasoning by analogy uses the known features of the current solution as a tacit matching pattern for the retrieval of the base analogs. Among previous design cases may be also an equivalent situation with respect to the undesired feature $D(x)$. Having found a set of equivalent situations according to some feature, we may look at conditions $\delta(x)$ typically appearing in connection with feature $D(x)$ or its opposite $\neg D(x)$. In other words, from the tacit similarity we are moving towards formulation of some explicit feature descriptions.

The discovered analogous objects or features were probably irrelevant at the first sight, and the designer was originally unaware of them. However, some of the discovered 'typical' features may serve as a missing link that will shift the conceptual base and eventually remove the inconsistency. A designer may want to adapt an existing object so that it complies with the identified conditions. Useful methods for the modification of the design objects include among others also combinations, mutations and various transfers as described in [10].

The uncovered feature δ acts as a *necessary extension* of the current state of design, and has impact on both, the conceptual objects and connections among them. The whole strategy for the resolution of inconsistency instead of removing it in the 'original' conceptual world, draws an analogy to a 'parallel' conceptualisation using the problematic statement as a match. After finding solutions in the 'parallel' worlds and their generalisation, we transfer the resulting knowledge to the 'original world' and re-formulate the current conceptual objects.

5 DESIGN ACQUISITION TOOLKIT

In order to validate the theory as proposed in sections 2 to 4 a prototype of the design support tool was developed to address the issues mentioned in the proposed methodology [2]. This section presents the basic architecture of the tool that has potential to evolve so as to support designers at work. Before immersing into details, it should be noted that we did not attend to the questions of efficient control flow and implementation. On the contrary, the aim was to identify the flow of data and decisions during a design process supported by a computational tool. Consequently, we expected to find a suitable form of knowledge-based support for the early design.

The fundamental theory of a reflective sequential design as briefly presented in earlier sections, assumes the presence of

two complementary types of knowledge – explicit and tacit. Both of them are important for the design, especially if one is interested in the innovative or creative design as we are. The biggest and most significant issue is that these two types of knowledge are located in the opposite extremes of a hypothetical 'axis of knowledge structure'.

On one hand, explicit design knowledge is often encoded as formal models, rules, laws and similarly; whereas tacit knowledge is much more subjective, experiential and resists most attempts of its formalisation. Therefore, from the very beginning we were aware that two diametrically different capturing mechanisms were needed to address different needs. First, the tool for the acquisition of formal design knowledge is introduced and its purpose is illustrated in examples and figures in section 5.1. Later on, the tool for capturing the informal and tacit reasoning processes is proposed in section 5.2.

5.1 Capturing the structure of design task

The main purpose of the formal design-capturing tool is to record the evolution of explicit design knowledge. The concepts this tool is working with include the formulation of requirements, constraints and solution. The tool is able to record the evolution of the specification, addition of new design objectives, modification or cancellation of the existing ones. Chaotic recording of the current requirements and constraints is useful; nevertheless a more significant contribution of the tool is the possibility to record not only the standalone statements but also structure and group them in the multiple design contexts.

Design context is a powerful concept that in plain language can be defined by a set of currently attended explicit design requirements, constraints and various tentative assumptions. Context represents a consistent and sufficient subset of all possible design objectives. The internal consistency of a single context does not prevent two independent contexts to be mutually contradictory. Thus another important contribution of the formal acquisition tool is the ability to keep open several potential ways forward simultaneously. Such a contribution is in accordance with the demands of some cognitive studies of design [10] that argue the need for the exploration of the design space in several parallel directions.

In the tool as depicted in Figure 2 we can see a sample design context from the design of paper mill controller that is described more in depth in [2]. The snapshot was made after finishing the design when the tool was used in a 'reproductive' mode – that is why we can see all developed solutions in the top right panel and all different requirements in the left panel. To this particular context belong first four requirements that are marked with a check (see left hand panel). The actual formulation of a particular design objective is presented in the bottom right corner. The information in this panel may, for instance stipulate the conceptual position of the respective objective; i.e. whether it is a requirement or a constraint.

Some of the requirements identified during the design were proposed once and kept without any major re-formulations (see e.g. numbers 1 or 2). Other requirements were proposed at some stage of the design process and later were subject to some extension/modification in line with the theory (see e.g. number 6 and its two modifications 6.1 and 6.2). The tool is thus able to distinguish different origin of the statements also visually – the statements made from the domain theory or tacit experience are located on a higher level than those modified and refined.

5.2 Acquisition of tacit and informal knowledge

The tool for capturing formal decisions in design in their explicit form is helpful for getting rough information about the evolution of a particular design task. However, it does not say why the decision was made, what other alternatives were considered or what reasons influenced the particular choice. The answers to these and similar questions are tacit and less structured, and as such require different approach.

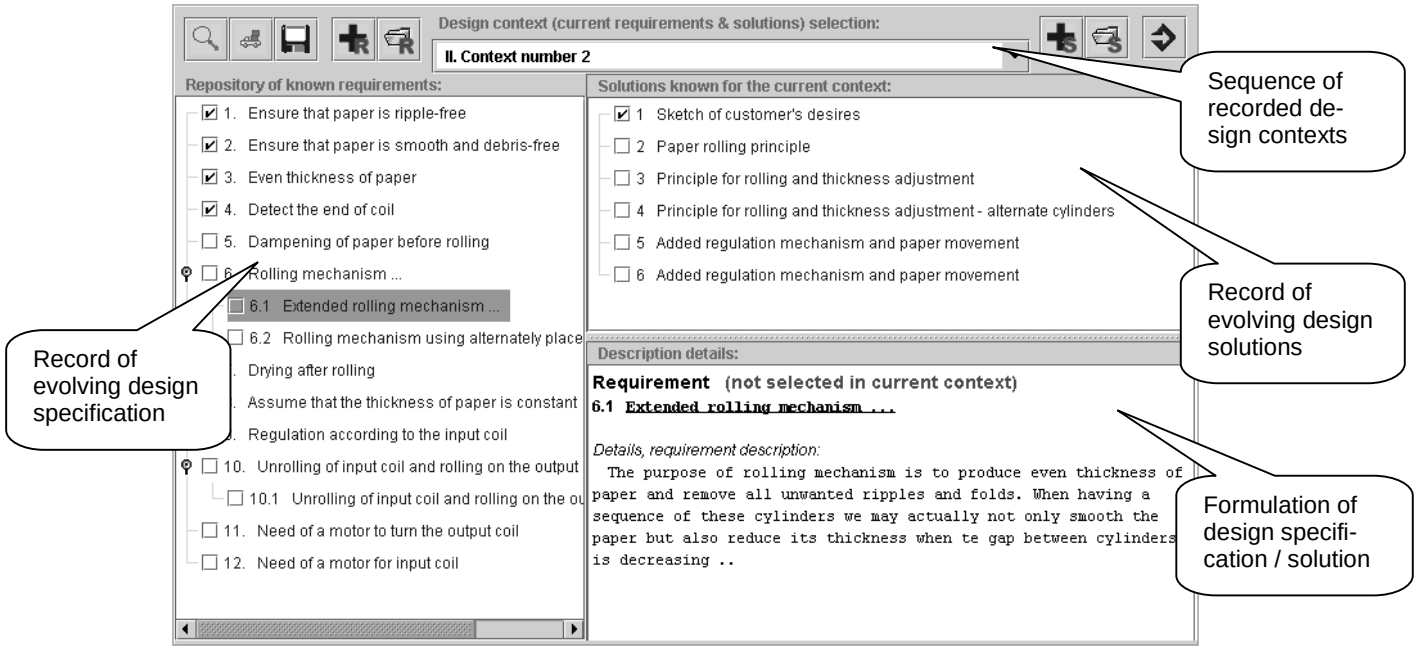

Figure 2. A tool for acquisition of design knowledge

Due to a different nature of tacit design knowledge entirely different strategy was chosen for capturing the informal reasoning in design. The acquisition of informal and tacit knowledge has a form of a hypothetical 'debate', in which designer's ideas and counter-proposals are appearing in the *threads* along some interesting feature, requirement or observation.

For instance, the thread from which a callout is drawn in Figure 3 is addressing the issue that a material is better shaped when treated before the application of pressure. Its first refinement introduces some moisturiser for damping the paper. Another follow-up brings forward the need for drying the paper so that it can be stored on the output roll in a desired form.

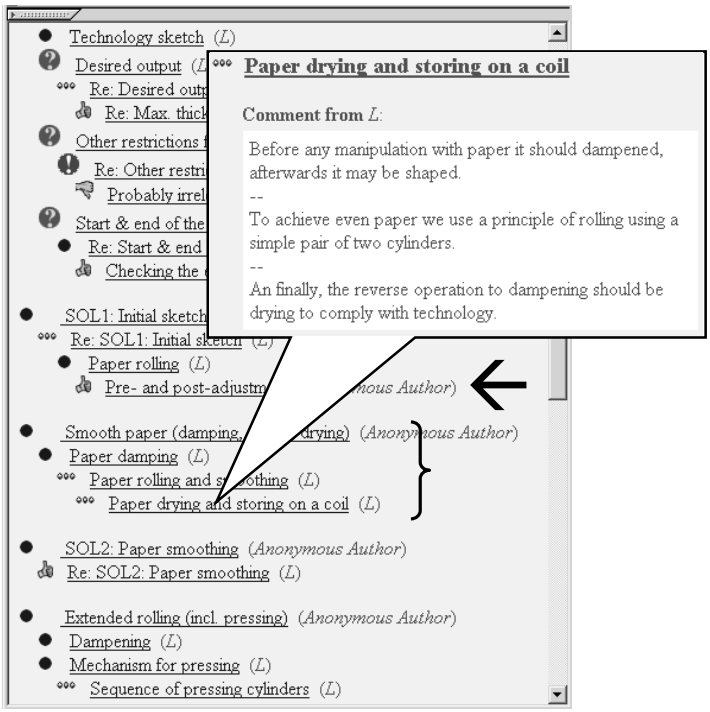

Figure 3. Sample discussion about early design

As visible in Figure 3, the designer has a range of symbols for expressing the 'tacit meaning' of a particular thread development. For example, the actual thread about dampening the paper (marked by '***') is in fact the result of a positive idea introduced in the previous thread (see the record on which the arrow is pointing). Designer marked this idea with the 'thumb-up' icon to emphasise his tacit feeling about the suitability and potential benefit of accepting this tentative proposal.

Typically, a successful conclusion of one 'discussion thread' may serve as a base for developing another thread that refines the proposed positive or negative observation to deeper details. The structure of the informal design knowledge is less rigid and seems to be less transparent. Nevertheless, when viewed together with the formal tool, the structure rapidly emerges. The tacit 'debate' is simply complementing the formal records in respect to the origin of a particular requirement or solution. Also, the debate addresses the positive or negative features that are resulting from the designer's tacit evaluation of the current design state. In other words, most of the 'tacit' inferences are introduced exactly in this tool.

6 ACQUISITION TOOL IN EXPERIMENTS

In order to understand the deeper processes that underlie design we carried out a set of 25 experiments. Each covered a single design task that was defined very loosely. All tasks were from the domain of controller design and we expected the production of a rough layout sketch and a control algorithm as outputs. Designer could use the support tool that was introduced in section 5 to assist with the design formulation and context management. In addition to the formal acquisition tool a tailored discussion environment was available to capture the justifications of design decisions. Other 'design' tools included pencil and notebook for sketching and relevant technical literature.

Assume, a customer gives the designer a task to design a control strategy for smoothing paper and simultaneously rewinding it from an input roll with raw paper onto an output roll with smooth paper without any folds. Since the sequence of design decisions and their translation into a formal language are available in the complementary paper [2], in this paper we shall attend only to the selected parts of the design and the usage of the acquisition tool. Special emphasis shall be given to the description of some situations, in which we observed the tacit reflection, and to the role the tool played in the tacit discovery of a new requirement or constraint.

After the clarification of customer's demands, the designer set a few basic requirements that enabled him to retrieve the rolling drums as a suitable prototype for paper smoothing. The prototype depicted in Figure 4A triggered a couple of interesting enquiries about prototype suitability. For example, when looking into details of the previous design cases the designer discovered a concept that typically appeared in connection with rolling – thermal pre-processing. Thermal treatment softened the material that was less likely to break or damage. The idea of thermal treatment sounded as useful; however, it could not be applied directly. Therefore, the designer transferred this idea and came up with moisturising the paper before rolling, and drying it afterwards.

$$\neg \textit{observed-on (property ('paper, 'moist), on ('input-roll)) } \Rightarrow$$
$$\Rightarrow \textit{before (apply ('pressure, part ('paper, 'surface), 'drum)),}$$
$$\textit{desired (property ('paper, 'moist), on ('input-roll)))}$$

$$\textit{observed-on (property ('paper, 'wet), on ('output-roll)) } \Rightarrow$$
$$\Rightarrow \textit{after (apply ('pressure, part ('paper, 'surface), 'drum)),}$$
$$\neg \textit{desired (property ('paper, 'moist), on ('output-roll)))}$$

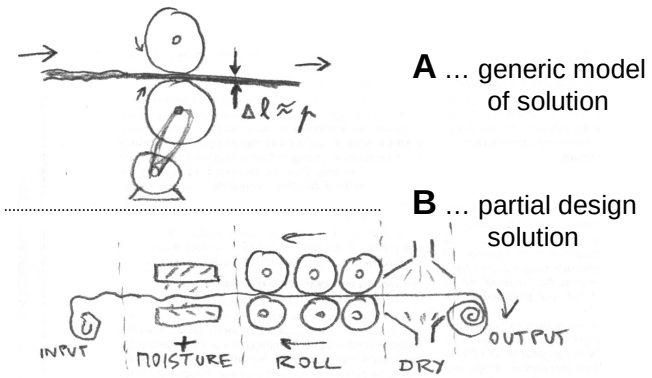

Figure 4. Acquisition of prototype extension requirements

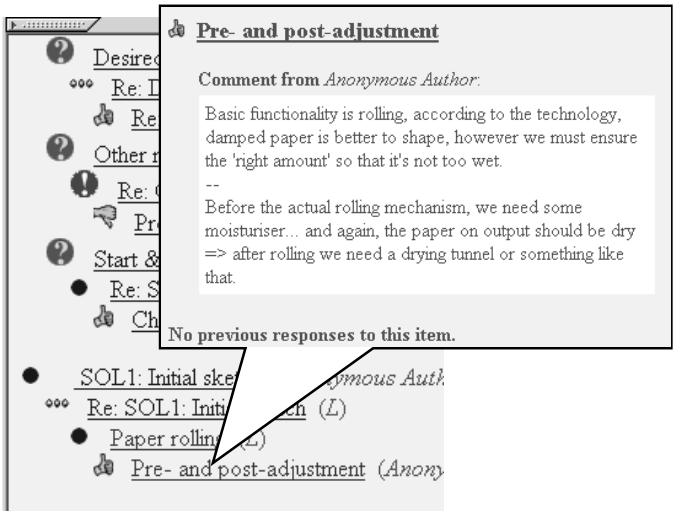

Figure 5. Introduction of a tacit extension of design

Prototypic model was thus extended with the pre- and post-processing units to comply with the discovered requirements. A sketch of a design solution with moisturiser and dryer is shown in Figure 4B, and in Figure 5 is a cut-out from the tacit introduction of the extension, in which the conclusion to introduce two additional units to the assembly first time appeared.

A similar reflection occurred when the designer attempted to improve the efficiency of the assembly. The 'active surface' of the sequence of rolling drums was very low in comparison to the size of the drums, and only very little could have been done in the current prototype. However, when the designer changed his perspective and looked at the issue from a different angle, he noticed a feature that was not explicitly defined. He tried to modify the layout of the drums from the linear to the alternate and such a perspective brought the desired effect – the 'active surface' of the sequence of drums rapidly improved. Figure 6 shows the sequence of tacit ideas recorded in the acquisition tool and leading to the introduction of alternate layout.

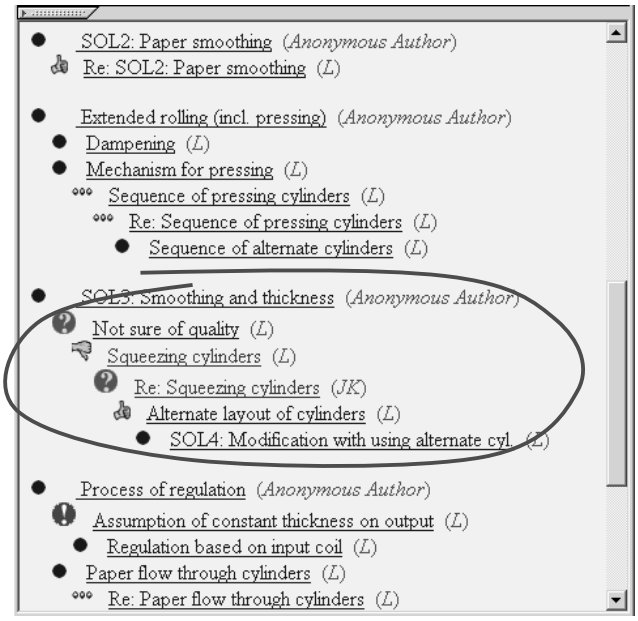

Figure 6. Tacit introduction of alternate layout of drums

By restricting the potential layouts of the rolling drums to alternate, the designer resolved the issue of efficiency, and was able to pursuit the undertaken direction in the early design. We skip the straightforward part of the design refinement, and return back to the details in the moment when designer tacitly re-formulated the basic principle of rolling.

Designer re-framed the problem and perceived the principle of 'rolling' more abstractly – an object moving along the surface and keeping certain pressure against another one. This left him with a single drum instead of a pair. Such an arrangement enabled higher precision in controlling the pressure between the drum and paper than it was in the case of controlling the gap between the drums. He iterated to a solution with single drums attached to 'springs'. Moisturiser and dryer were moved 'inside' the cylinders, and only one pair of cylinders remained to unwind the paper from the input roll (see sketch in Figure 8).

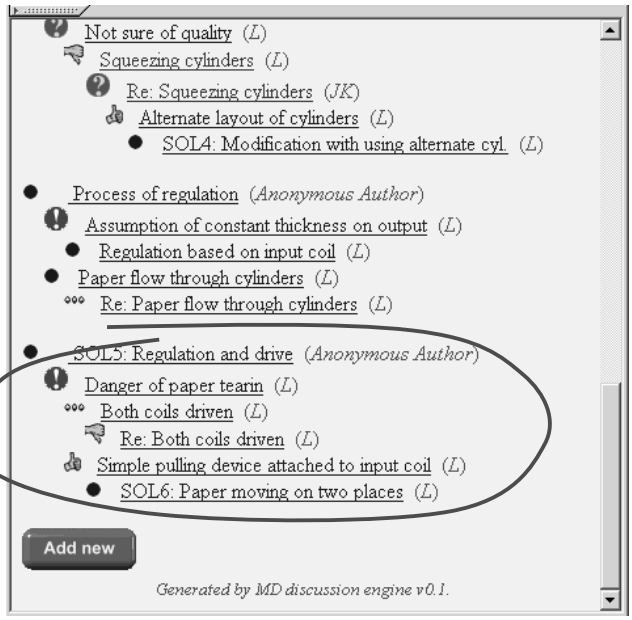

Figure 7. Major modification of final design

It was only at this stage, when the designer elaborated further details (e.g. regarding motor drives) and assumptions (e.g. desired thickness of paper given at the beginning and not changing during operation) so that he could develop the solution sketch and control strategy as required. Figure 8 shows the solution sketch with basic control parameters; for details see [2]. We leave the scenario at this stage having almost complete solution. This excerpt sufficiently illustrates the re-formulation of the design task and its importance for innovative design.

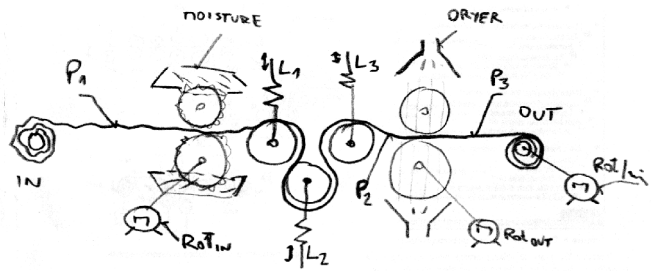

Figure 8. Final design solution

7 ACQUISITION TOOL EXTENSION FOR KNOWLEDGE-INTENSIVE DESIGN SUPPORT

In this section we look at the possible extensions of the acquisition tool that could provide also some form of knowledge-intensive design support in addition to knowledge acquisition. Knowledge acquisition would thus become only a part of the overall functionality of a 'intelligent' design support tool. The ideas presented in this section are in a form of a hypothetical discussion; i.e. the techniques discussed below are not yet implemented in the support tool.

7.1 Knowledge support for design abduction

Logical abduction looks for a sufficient feature explaining or implying a desired fact. Since many different explanations may exist, it may be useful to restrict our attention to the least presumptive, least abnormal [11], or partial ones [12]. In design, we do not want to generate all possible structures and principles delivering a desired function or property; thus we propose the 'simplest' set of structures and relations that satisfy our goals and do not make unnecessary commitments.

However, as we mentioned above, the least presumptive structure from the logical perspective does not have to be the most suitable one from the design point of view. Very often we need to refine the 'least presumptive' design, and according to our theory we can do it by formulating a new requirement or constraint that would discriminate between the alternative designs. Any such condition, however, introduces more and more presumptions to the design.

In order to find the approximately right level of coarseness we may use the known structure of design knowledge to decide whether to move further in the abduction. The heuristic may determine if the abduced formula is 'a design structure' or only an intermediate principle. Typically we wish to perform the logical abduction until a fully-fledged, though possibly highly abstract structure is discovered. As soon as abduction arrives at a structure it can stop and propose the result to the designer for the tacit evaluation. Such approach would enable designers to skip through unnecessary refinements and evaluations.

In our example with paper smoothing, a device for structural modification of material resulting to the surface smoothing is the least presumptive abduction. However, it is too abstract for any direct use, and a possible way forward could be the inference of a device applying pressure on the material and thus modifying the surface. This inference is already more specific, and enables the designer to find a prototype of rolling drum as a possible structure delivering the desired effect. There are many different ways for modifying the surface of a material; the application of pressure (and rolling drums) is only one of many. It is thus an additional commitment but nevertheless one that cuts down the large space of possible designs.

7.2 Knowledge support for deduction in design

Deduction may in general infer a large number of consequences, which may not be desirable in design. However, not every inference is relevant to be presented to the designer and evaluated. Because of the creative nature of design we may wish to infer the broadest possible implications without overwhelming the designer with the irrelevant knowledge. Poole's *prediction of what is in all* [design] *extensions* [11] seems to be a plausible formalism regulating deduction in design. This heuristic is generating new formulae that are valid in all different design contexts restricted by different design assumptions. Such a deduction is often a generalisation of the actual inferences in multiple contexts. However, it still does not address all issues specific for design; namely, the problem with having as broad deductions as possible and avoiding the irrelevant ones is not resolved.

In order to avoid the deductions that are not bringing any particularly new information into design we stipulate that theorems are possibly irrelevant (i.e. not a new finding) if they further lead to the desired goal:

$$(A(x) \vdash D(x)) \wedge (D(x) \vdash E(x)) \wedge (E(x) \in G_i)$$

Theorem $D(x)$ is directly responsible for satisfying the goals G and although it is formally a derivation, it does not bring any new knowledge for the subsequent design. As such it may be removed from the set of so-called allowed theorems (theorems that are allowed to participate in deduction). In such a manner we may tacitly regulate the amount and content of the deduced sentences without withholding some really important and new information from the designer.

Another mechanism for the regulation of the amount of deductions can re-use the previous design cases and deduce only those theorems that have an analogy with the previous experience. Alternatively, it may be interesting to focus only on those theorems that are different in comparison to the previous cases. More about this approach is given in the next sub-section.

7.3 Knowledge support for tacit evaluation

As we already stated earlier, the logical design theory may assess the consistency of a deduced theorem $D(x)$ with respect to certain assumptions but is not able to 'judge' its desirability. This judgement comes usually from the designer's tacit (empirical) knowledge and his or her feel for the solution acceptability [6]. Nevertheless, if we have access to the previous design tasks, we may compare the theorem $D(x)$ deduced in the current problem with the deductions in the analogous cases. If we discover that $D(x)$ is 'typically' a negative feature, and usually appears with a particular requirement or constraint $P(x)$, then we may suggest a similar assessment also for the current problem. Obviously, the last word has the designer who must 'ratify' this potential analogy and/or adapt it for the current design problem.

In our example, with paper smoothing, this situation is illustrated by the discovery of thermal treatment of the rolled material. Rolling by pressure was in the past cases accompanied by heating of the material in order to reduce the danger of its damage. Such an observation seemed to be valid also for the current problem – thus the idea of *some* treatment was recorded. The designer later refined this abstract need to the treatment using water instead of heat. The whole reasoning chain was inspired by a condition abstracted from the previous cases and knowledge of the procedure for paper manufacturing.

This operation may be seen as an attempt to tacitly generate some means for avoiding the occurrence of a negative feature.

We agree at this point with Cook and Brown [6] who claim that new explicit rules are not equivalent to the explicated tacit understanding of the problem. On the contrary, they can be partially generated using the tacit (empirical) knowledge but still the explicit and tacit forms of knowledge complement one another rather than replace. From the logical point of view, we modified our knowledge about the current conceptual objects and their applicability in the current design.

7.4 Knowledge support for re-conceptualisation

Support for the re-conceptualisation is the most interesting one and it can be based on the support for the restriction of evaluating conditions. Only instead of restricting the known facts about the conceptual objects, we introduce new conceptual objects. And similarly as in the previous case, we can draw on the tacit knowledge we have about the artefact in question and the previous design cases. Using tacit knowledge we generate 'seeds' of an explicit description of new conceptual entities.

It seems reasonable to look for analogy on a higher conceptual level. Since the aim of re-conceptualisation is the identification of new concepts the designer was originally unaware of, the conceptual jump across the domains is likely to move the current state of the design from the 'local' deadlock. Knowledge of the abstract dependency between various objects may be used to trigger a non-traditional transfer. Suppose α is a known object; then theorem β is abstracted from α if it lies in the '*type-of*' hierarchy above α:

$$abstracted(\alpha,\beta) \Leftrightarrow$$
$$\Leftrightarrow (\exists\beta\text{: type-of}(\alpha,\beta)) \vee (\exists\beta,X\text{: type-of}(\alpha,X) \wedge abstracted(X,\beta))$$

Now it is possible to use not only the currently known objects (α) for the exploration, but also at the analogies that occur on the higher level of conceptual abstraction (β). Once a higher-level object β is identified, it can be used in the reasoning by analogy exactly in the same way as the original α. New and less traditional similarities may be discovered between the abstracted concept β and another concept in a different domain on the same or different level of abstraction – e.g. γ.

8 OVERVIEW OF RELATED RESEARCH

The tool described in this paper incorporates also most of the findings reported by Schön [13] who observed in the studies of the professional designers the oscillation between the solution development and reflection on it. Schön refers to the inconsistency of the current solution as 'a surprise', and claims that any such surprise may trigger a modification of the [conceptual] frame for the solution development. Modified frame allows designers to perceive the objects they were previously unaware of. Despite the complexity of this operation with regards to knowledge, we attempted in this paper to shift the reflection from an indescribable art to a computational heuristic. Some features of such a 'heuristic for reflection' in the design were suggested and discussed in sections 4 to 7.

Our work exhibits also methodological similarity with Altshuller [14] and his theory of inventive problems that was implemented in a prototype called 'Invention Machine' [15]. This theory is based on a table of typical physical contradictions that may be observed in designs of technical systems, and suggests typical means for the removal of such contradictions. Theory of inventive problems can be seen as a generalised case-based reasoning from a large base of previous cases. However, the assumption that the contradictions in the technical systems are always observable is not always satisfiable. It poses basically no problem in the re-engineering tasks where there is an artefact whose performance in some aspect must be amended; the actual observations of the artefact and the comparison with the desired state are straightforward. In our case, we focus on the design of new artefacts (not necessarily inventions) – we have only an incomplete set of desired features and none device or technology that can be observed, simulated and/or evaluated by external means. Thus we replace the observation by logical and tacit reasoning. We feel however, that our formalism may generalise the conclusions of Altshuller's theory.

Tomiyama [16] or Takeda et al. [17] also proposed similar tools for design support. Their general model of design included abduction, deduction and circumscription, and very precisely divided the reasoning in design into two levels – reasoning about design *actions* and about design *objects*. This distinction is less emphasised in our framework; however, we may claim that supportive actions for the discrimination or restriction include more the reasoning about design actions than design objects. On the contrary, re-conceptualisation in our framework is focused more on the design objects than actions.

Takeda's (et al.) theory has several gaps in respect with the incompleteness of design specification and knowledge modification. In our opinion, its circumscriptive mechanism removes the outstanding contradiction by referring to other known objects. But what if the designer does not know all necessary objects? The whole theory draws on a rather strong assumption that we know *all* objects in our logical theory that must be searched. As we stated earlier, we might have tacit feelings about 'all potential' design objects, but to work with, these objects must be explicitly described. Design is far from a problem of searching large repositories of all design objects; it is mainly about constructing explicit design spaces 'on-the-fly' using tacit experience.

Similar research was conducted in the development of Edinburgh Designer System [18,19] with a well-elaborated assumption-based mechanism for the management of multiple design contexts. Nevertheless, their knowledge-based support is restricted to the identification of appropriate assumptions and generation of various consequences of given assumptions. In the described approach we extended the assumption-based context management with some techniques suitable for the reflective design.

9 CONCLUSIONS

In this paper we showed design as a sequential process, in which abduction, deduction and different modifications take place. We introduced so-called *tacit inconsistency* as an important evaluation criterion in design. Also, we identified several different causes of tacit inconsistency in the partial design solutions. These partial solutions are consistent with respect to the current explicit specification, which however, does not

guarantee the consistency with all the tacit, implicit, untold requirements and/or constraints. It seems appropriate to suggest that such an inconsistency of a solution with the implicit design goals is the most significant vehicle that enables the designers to progress in their design tasks.

In the paper we also introduced a prototype of a design support tool that can assist the designers in uncovering the inconsistencies using tacit knowledge and reflection on the current state of the design problem. We showed that although the reflection is a tacit operation on the tacit knowledge, it can benefit from the structured acquisition of different ideas and design justifications. The process of reflecting and using the tacit knowledge has many significant features that can extend the existing methodology for acquisition of design knowledge.

In the current paper we discussed only selected issues regarding the acquisition of explicit and tacit design knowledge. We mentioned that tacit knowledge might serve as an activator for uncovering new explicit rules or objects. However, we did not attend to the fact that the tacit knowledge itself is subject to evolution and change as the design task progresses. This idea is shall be investigated in some later papers.

ACKNOWLEDGEMENTS

This work benefited from the criticism and feedback from John Domingue and Paul Mulholland from KMi. A thanks belongs also to the anonymous reviewers for their valuable comments on earlier drafts of this paper.

REFERENCES

[1] H.A. Simon, The structure of ill-structured problems. *Artificial Intelligence*, **4**:181-201, 1973.

[2] M. Dzbor and Z. Zdrahal, *Towards Logical Framework for Sequential Design*. In *Proc. of 13th Intl. Conf. on Design Theory and Methodology (part of ASME DETC)*, Pittsburgh, USA, DETC01/DTM-21710, 2001.

[3] Y. Iwasaki, *et al. How Things Are Intended to Work: Capturing Functional Knowledge in Device Design*. In *Proc. of 13th IJCAI*, pages 1516-1522, 1993.

[4] J.S. Gero, Design prototypes: A knowledge representation schema for design. *AI Magazine*, **11**(4):26-36, 1990.

[5] E. Mendelson, *Introduction to Mathematical Logic*. 2nd ed., D. Van Nostrand Co., USA, 1979.

[6] S.D.N. Cook and J.S. Brown, Bridging Epistemologies: The Generative Dance Between Organizational Knowledge and Organizational Knowing. *Organization Science*, **10**(4):381-400, 1999.

[7] B. Chandrasekaran, Design Problem Solving: A Task Analysis. *AI Magazine*, **11**(4):59-71, 1990.

[8] J. de Kleer, An Assumption-based TMS. *Artificial Intelligence*, **28**(2):127-162, 1986.

[9] I. Watson and S. Perera, Case-based design: A review and analysis of building design applications. *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing*, **11**:59-87, 1997.

[10] N. Cross, Descriptive models of creative design: application to an example. *Design Studies*, **18**:427-440, 1997.

[11] D. Poole, Explanation and prediction: an architecture for default and abductive reasoning. *Computational Intelligence*, **5**(2):97-110, 1989.

[12] J. de Kleer, A.K. Mackworth, and R. Reiter. *Characterizing Diagnoses*. In *Proc. of AAAI*, pages 324-330, 1990.

[13] D.A. Schön, *Reflective Practitioner - How professionals think in action*, Basic Books, Inc., USA, 1983.

[14] G.S. Altshuller, *Creativity as an Exact Science*. Studies in Cybernetics (F.H. George, ed.), Vol. 5, Gordon & Breach Science Publishers, USA, 1984.

[15] V.V. Sushkov, N.J.I. Mars, and P.M. Wognum, Introduction to TIPS: a theory for creative design. *Artificial Intelligence in Engineering*, **9**(2):177-189, 1995.

[16] T. Tomiyama, From general design theory to knowledge-intensive engineering. *AI for Engineering Design, Analysis, and Manufacturing*, **8**:319-333, 1994.

[17] H. Takeda, *Towards Multi-aspect Design Support Systems*. Report NAIST-IS-TR94006, Nara Institute of Science and Technology, Japan, 1994.

[18] M. Tang, A knowledge-based architecture for intelligent design support. *Knowledge Engineering Review*, **12**(4):387-406, 1997.

[19] T. Smithers, *et al.*, Design as intelligent behaviour: an AI in design research programme. *Artificial Intelligence in Engineering*, **5**(2):78-109, 1990.