KMi Technical Report

KMi–TR–108

TOWARDS LOGICAL FRAMEWORK FOR SEQUENTIAL DESIGN

Martin Dzbor, Zdenek Zdrahal

Knowledge Media Institute
The Open University

To appear in: 13th ASME DETC Conference on Design Theory and Methodology, 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/DTM–21710

TOWARDS LOGICAL FRAMEWORK FOR SEQUENTIAL DESIGN

Martin Dzbor
Knowledge Media Institute
The Open University
Milton Keynes MK7 6AA, UK
Tel. +44 1908 858524
Email: M.Dzbor@open.ac.uk

Zdenek Zdrahal
Knowledge Media Institute
The Open University
Milton Keynes MK7 6AA, UK
Tel. +44 1908 654512
Email: Z.Zdrahal@open.ac.uk

ABSTRACT

Engineering design is usually seen as a knowledge-intensive process that driven by certain objectives eventually delivers an artefact having the desired properties or functions. Design is inherently iterative and the design goals evolve together with the solutions. Many current design theories present more or less efficient ways for finding a suitable solution to the given goals. However, they often leave open the question of the '*solution talkback*'. Under 'solution talkback' we understand the reasoning process that is able to infer what formal amendments to the initial design specification need to be made in order to produce a feasible solution. Modified explicit design specification would in turn enable designers to refine the solutions to their design problems. This paper suggests an early-stage theory that incorporates some typical features of design problems, and defines a reasoning framework for the reflection on the actions in design. First, the key terms are defined that are elaborated later with the focus on generation of new design goals through the reflection on the partial design solutions.

1 INTRODUCTION

Simon [1] includes engineering design among ill-structured problems; i.e. the initial specification of a design problem is usually incomplete, the initial vagueness prevents the designers from constructing a precise problem solving space and setting the clear criteria to determine a (final) solution. Due to the initial uncertainty, the design space for a particular task does not objectively exist in advance but must be constructed on the fly. The sheer amount of possible combinations of 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 more typical for design than any algorithmic determinism or productivity.

We approach design as a sequential process with the view similar to that of Gero [2], Iwasaki, Chandrasekaran [3], and others, who claim that design is an interplay between the functional and structural objects. We can summarise our particular variation of this viewpoint as follows:

- a design problem can be specified as a set of requirements and constraints that refer to functions and properties of the elements known in a particular domain;
- a design solution is developed in terms of structural elements and relations among them assuring the proposed structure meets certain 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 theoretical implications presented in this paper draw upon observations made during a recently conducted experimental study, where as the participants were two designers with some experience in their field. The tasks they were solving were taken from the domain of large-scale systems controller design and we expected the production of a rough control strategy as the main output. We were looking at how the designers formulate the design problem using multiple changing contexts, as well as how they justify and support their decisions. A separate section is devoted to the experimental work and findings.

2 TACIT CONSISTENCY IN DESIGN

We base our proposals on the first order logic for the sake of simplicity, but also other sufficiently powerful expressing mechanisms may be used instead. Initially, the basic logical axioms and rules of deduction [4] are adopted, refining only one – a free formula $\beta(x)$ is usually understood as its universally quantified closure $\forall x: \beta(x)$. More precisely it should be said that the universal closure applies to all objects in a certain *conceptualisation* of the world. Thus the formula will be always understood as $\forall x: \beta(x) \mid conceptualisation = Conc_n$.

We define Γ as *all* design goals that may be demanded from, or must not be violated by the designed artefact. Γ is a *tacit* entity that is not only unstructured but in general 'unstructurable'; however it objectively exists and has an immense influence on the design [5]. Tacit knowledge is understood as something that is inherently present and used when tackling a problem, though one may not be able to express or explain it explicitly. Original example from M. Polanyi illustrating tacit knowledge[1] describes cyclists who are able to cycle and stay upright on a bike and still being unable to say how exactly they turn the handlebars so that they can 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. Both, explicit and tacit knowledge is involved in cycling. They complement each other and typically, one can be used to acquire the other one. However, we should note that it is never a direct transformation of one type to the other one [5]!

In the case of design, the explicit knowledge contains various algorithms, design methods and methodologies, logical, mathematical, physical models or rules. On the other hand, tacit knowledge is perceived in the situations when designers talk about 'liking a solution'; they are not able to express what exactly is causing their attitude but there is a tacit feeling of hidden flaw. Only the solution consistent with all explicit rules as well as tacit knowledge can be considered as acceptable. Since the operators of logical 'consistency' and 'inference' assume an explicit logical theory and do not consider the influences outside of logic, we suggest corresponding 'tacit' operators with similar roles as the traditional ones ($\models$, $\vdash$) but with reference to tacit knowledge. Thus $\boldsymbol{\Gamma \circleddash \gamma}$ means that sentence γ is '*consistent*' with tacit knowledge Γ; it is 'acceptable'. In line with [5] explicit knowledge may in certain cases be generated from tacit feelings; this 'tacit inference' would be below denoted as $\boldsymbol{\Gamma \ominus \gamma}$.

We also distinguish an *explicit design specification*, which consists of *explicit* requirements (R) and constraints (C), from the unstructured *tacit* 'specification'. As already stated, tacit and explicit design knowledge are closely related; e.g. tacit knowledge may help to generate new explicit requirements or constraints (Eq. 1). In addition we stipulate also the following: if solution s is consistent tacitly then it is guaranteed to be also explicitly (i.e. logically) consistent with the current explicit problem specification $R \cup C$ (Eq. 2).

$$\Gamma, s \ominus R \cup C; \qquad \textbf{Eq. 1}$$

$$(\Gamma, s \ominus R \cup C) \wedge (\Gamma \circleddash s) \Rightarrow (R \models s) \wedge (s \cup C \nvdash \bot) \qquad \textbf{Eq. 2}$$

Let us denote the current explicit requirements and constraints in the design step i as R_i and C_i; and the solution corresponding to the explicit specification as s_i. Let E is a set of available design elements. A combination of design elements $s_i \in E^*$ is a design solution in the particular design step when it satisfies the current explicit problem specification; i.e. all explicit requirements and constraints:

$$\forall s_i \in E^*: (R_i \models s_i) \wedge (s_i \cup C_i \nvdash \bot) \Leftrightarrow maybe\text{-}solution(s_i) \qquad \textbf{Eq. 3}$$

In Eq. 3 we abbreviated the consistency conditions when there is no explicit mention about the logical theory in which these conditions hold. Nevertheless, all statements are implicitly referring to the given underlying theory T that expresses design and domain knowledge; e.g. in a form of elementary properties and dependencies between the structural and functional objects. For the brevity, the properties are denoted by predicates (e.g. $D(x)$) and the dependencies by the operator of logical deduction (e.g. $A(x) \wedge \neg P(x) \vdash D(x)$):

- *is_a (X, 'object)* $\wedge$ *property (X, 'weight)* … concept denoted as 'X' belongs among objects in our world and as such has some 'weight' as a descriptive property;
- *apply ('pressure, Material, Tool)* $\vdash$ *modify (shape (Material))* … whenever a 'Tool' is used to apply pressure on any 'Material' it will result in a change in the shape of the 'Material'

It must be noted that Eq. 2 cannot be reversed so that the explicit (logical) consistency (Eq. 3) implied that a solution was also consistent 'tacitly'. Thus Eq. 2 drawing on designer's 'tacit' satisfaction with the solution s_i ($\Gamma \circleddash s_i$) is a sufficient stop condition for the design process. The explicit design requirements and constraints in step i would then act as a *sufficient problem specification* and s_i as an *acceptable solution*.

3 REASONING SEQUENCE IN DESIGN

Design is ill structured not only with regard to the problem space but also with regard to the reasoning strategy used to navigate in such a space. Different authors pay attention to different types of reasoning that may be observed in design; concise but very rich summary of the reasoning strategies in design is presented in [6]. This section explores what logical operations can be associated with some typical design activities that operate on the explicit knowledge level.

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

Abduction is a form of hypothetical reasoning when some feature is observed/desired and we are interested in finding a sufficient means to achieve the desired feature. Abduction for the purposes of design may take some of the explicit requirements $F(x) \subseteq R$ and investigate how these may be achieved. Assume that a requirement $F(x)$ is given that specifies some fact about the desired function or property. The designer looks for an artefact $A(x)$ that implies functionality $F(x)$ consistently with the logical theory (i.e. $R \models A(x)$ and $A(x) \cup C \nvdash \bot$). Discovery of such an artefact and presence of sentence $A(x) \vdash F(x)$ in the logical theory are sufficient to conclude that artefact $A(x)$ not only implies but also satisfies given goal $F(x)$ (see completeness theorem e.g. in [4]: $\alpha(x) \vdash \beta(x)$ iff $\alpha(x) \models \beta(x)$), and as such may be a partial design solution (it complies with Eq. 3).

Abduction is a specific form of non-monotonic reasoning, and it seems to be a suitable strategy for the exploratory design because it makes only a tentative choice for the exploration of a particular area in the design space. This is a valid perspective because the designers' attempts to find a structure for the desired function contain that tentative element of non-monotonic theories. Typical propositions in design are often in a form: *'…let us try alternative X; it satisfies my needs provided nothing else is found later that may violate this suitability.'*

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

Once artefact $A(x)$ was found that satisfies certain explicit goals $F(x)$ designers may be interested in knowing the additional implications of having artefact $A(x)$ as a solution to the design problem. They may want to deduce the consequences of the solution discovered via abduction. Typically, this knowledge can be acquired by the deployment of the rules $A(x) \Rightarrow D(x)$ (or generally $A(x) \vdash D(x)$). From the design point of view, deduction generally does not create new structures. According to the

[1] Term 'tacit knowledge' was introduced by Michael Polanyi in 1963; we borrowed the example with cycling from [5].

deduction theorem [4], it only generates new knowledge about the existing structures. In this sense, deduction is more analytical than abduction – hence, many qualitative modelling tools deploy logical deduction as a basic principle for generating new data about the existing objects.

The logical consistency found between artefact *A(x)* and desired goals *F(x)* does not change during deduction. But in addition, artefact *A(x)* is also consistent with the deduced function *D(x)* that was not among the initial requirements ($D(x) \not\subset Gi$); i.e. $A(x) \models D(x)$ holds, where *D(x)* is an uncovered consequence of the current solution in the space of functions and properties that may or may not be acceptable with regard to the tacit design knowledge Γ. Tacit consistency needs to be assessed by the designer, and here begins the interesting and 'ill structured' part of reflective design to which we pay more attention below.

Step III: Evaluation of (tacit) consistency $(A, D) \ominus D \vee \neg D$

In any given logical theory it is possible to assess logical consistency between the discovered consequence *D(x)* and the explicit problem specification $R \cup C$. This would be a purely logical check that can be performed by some truth maintenance system. We skip this logical assessment at the moment and focus on the assessment of the 'tacit consistency' of the deduced consequence *D(x)* with 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 in our case tacitly appreciates the uncovered consequence, and may find that:

a. *D(x)* is an irrelevant feature with respect to tacit expectations and will not influence the subsequent design;
b. *D(x)* is a relevant and desirable feature referring to an implicit design requirement not mentioned in the initial specification. Since it is consistent with tacit expectations, it may extend current design specification $R \cup C \cup D(x)$.
c. *D(x)* is an undesirable feature of the tentative decision and as such must be avoided in the further design. Thus its negation $\neg D(x)$ is intuitively noted as the necessary condition for 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 discovers a new 'goal' $\neg D(x)$ but also introduces inconsistency into the logical theory ($\neg D(x)$ is demanded and *D(x)* is provable). Before we can proceed any further, the (tacit) inconsistency must be removed because eventually $\Gamma \ominus A(x)$[2] must hold for any acceptable solution. We understand this situation as tacitly inconsistent because we arrived at it through the tacit appreciation of the current solution and specification.

In order to avoid purely theoretical development of the framework for the sequential design, the operations mentioned in this section shall be illustrated using an example taken from one of the conducted experimental sessions (see section 5).

4 RE-FORMULATION IN DESIGN

The cause of the (tacit) inconsistency in the design theory may have its roots in any of the 'logical' reasoning steps discussed in the previous section and depicted in Figure 1, and it may be also removed in any of them. To achieve the consistency, some new requirements, constraints and possibly solutions need to be formulated. It may be said that the design task is *re-formulated* in order to fix the undesired inconsistency[3]. Some of the available 'fixes' are purely logical, while others may contain strong tacit element. Below we discuss more in depth a logical fix for the *abduction*, and two tacit fixes – first, for the *evaluation* and finally for the logical *conceptualisation* of the world.

4.1 Fixing logical abduction

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

Assuming that the formulated requirement *F(x)* is indeed desirable but possibly incomplete or vague, and the design theory contains several hypotheses with *F(x)* as a consequence, the abduction may have chosen a wrong rule that eventually led to tacit inconsistency. The unsuitability of that particular rule was discovered tacitly at the later stage of the design. There was no knowledge available during the abduction phase that would have discriminated between multiple alternative rules.

Once the results and consequences of the abduction were tacitly appreciated, the missing condition $\neg D(x)$ was defined explicitly, and it may discriminate 'wrong', potentially inconsistent abductive rules. Formally we say that *B(x)* is a functionally alternative artefact to *A(x)* when the following holds:

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

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 absence of the harmful feature *D(x)*; $B(x) \vdash \neg D(x)$ would not be a valid conclusion! Nevertheless, the reasoning strategy applied to abduction involves the traditional backtracking from an inconsistent state to the last consistent state.

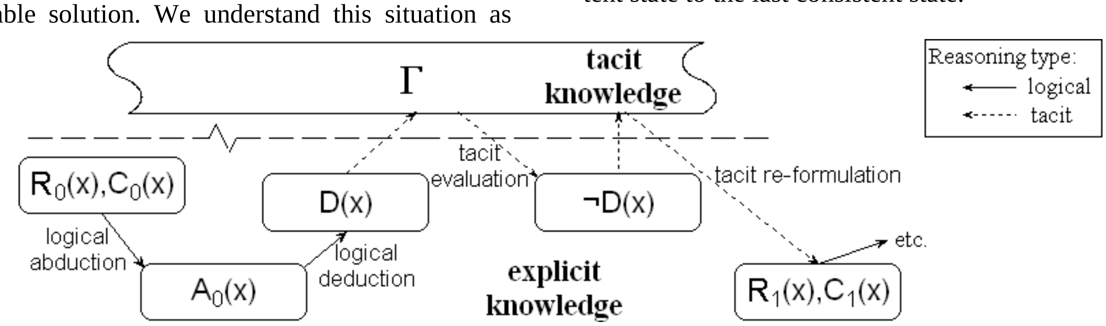

Figure 1. Interplay between explicit and tacit reasoning in design

[2] The stop condition of every design is the designer's tacit satisfaction with the solution and all discovered artefacts (see section 2).

[3] Also Schön [13] talks about unexpected surprises with the current solutions and modifications of the problem solving frames.

4.2 Fixing tacit evaluation

The case described in the previous section is basically a formalised version of the statement '*When you do not like the solution, go back and choose a different one.*' More interesting case in design is when the designers do not like the current solution but instead of giving it up and backtracking they investigate the reasons for such a state. Their motivation is to conjecture some condition under which the consistency of the design would be restored. Such an approach for the restriction of the design space basically corresponds to the term that Simon calls 'bounded rationality' [1]. In simple words, it is based upon the commonly accepted truth that we develop models of the systems because we are not able to attend to the real world in its entire complexity. In design, a set of a few fundamental assumptions can be chosen that allow only certain combinations of theorems to enter the reasoning process.

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

In the previous sections artefact $A(x)$ was found as a potential design solution, and among its various consequences also $D(x)$ was deduced that raised some worries about desirability. The designer would probably want to avoid this ambiguous feature $D(x)$ in the further design and still keep the current artefact $A(x)$ as a solution model. The original deduction chain of the type $A(x) \vdash D(x)$ may be enhanced with a conjecture $P(x)$ as given in the following schema:

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

The reading of the schema is as follows: '*Assuming condition P(x) is satisfied then it restricts the artefact A(x) so as the undesired consequence D(x) is no more observable.*' Schema from Eq. 5 can be also reversed to read: '*If a condition existed upon which satisfaction the occurrence of consequence D(x) depended, it could be possible to restrict the design theory and artefact A(x) by assuming the complement of condition P(x).*'

$$\neg P(x) \Rightarrow \neg \{ (P(x) \Rightarrow (A(x) \wedge P(x))) \vdash D(x) \} \qquad \textbf{Eq. 6}$$

The main purpose of the conjectures in Eq. 5 and Eq. 6 is to move from the generic deduction to the assumption-based one. Assumption is a tentatively accepted theorem upon which the deduction can be based similarly as with general theorems. However, in case when an assumption shows to contradict another theorem, it can be simply cancelled without making the entire logical theory inconsistent.

An assumption can be added to the current design specification either in a form of a new requirement or a constraint. These two forms differ on a conceptual level rather than structural; they are used in a different fashion. Conditions $P(x)$/ $\neg P(x)$ would be considered as *requirements* if they are specifically demanded and their formal record is a schema having two premises:

1. $P(x)$ / $\neg P(x)$
2. Eq. 5 / Eq. 6

In case that a conjectured condition is only a *constraint*, it is not explicitly demanded but it must not be violated. For example taking Eq. 6 as the base for a constraint, whenever condition $P(x)$ holds, the entire scheme stipulated by Eq. 6 is not defined and thus not violated. Unlike requirement that is always given in its explicit form, constraint is given in implicit form; i.e. Eq. 6 by itself can serve as a constraint without any further premises but it needs explicit demand for $P(x)$ if it is to be a requirement. Similarly, Eq. 5 is a constraint, and explicit demand of $P(x)$ makes from it a new design requirement.

Tacit fixation of the current design through restriction of a solution is considered to be a significant strategy for reasoning in design. Clearly, tacit fixation introduces new formulae to the logical theory and therefore is a non-monotonic reasoning step. Its primary purpose is to fix a potential inconsistency created in the previous, monotonic reasoning steps.

4.3 Tacit re-conceptualisation

Both fixes described above – a simple logical one in abduction and restrictive, tacit one in evaluation, focus on the amendment of a design solution. The designer either restricts the scope of the validity of a known solution, or backtracks to find an alternative one. In the unfortunate case, when there are no alternatives available or restrictive conditions found in the domain theory, it may suggest that the theory itself is incomplete and is not powerful enough to describe all the objects that are needed in order to resolve the deadlock. In this section a mechanism is sketched that is able to not only modify the available knowledge but as well allow for a generation of new conceptual objects for the logical theory.

As traditional logic [4] is arguing, the deduction simply has to produce the same result whenever it is given the same initial conceptual axiomatisation of the world. Deduction rules are in general very simple, and do not bring additional ambiguities to the reasoning. Nevertheless, the underlying conceptual apparatus may not be powerful enough to represent the real world, and it needs to be modified to prevent logical contradictions.

What we want, 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 this statement would be irrelevant. The questions that appear in this context include the following:

- Which 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. From the tacit knowledge we may know at least some properties the new conceptual objects shall exhibit (e.g. given requirements *R*) or avoid (e.g. undesired consequence $D(x)$). Since knowledge used for the discovery of new conceptual entities is tacit, the descriptive properties are only partial and incomplete.

Now the conceptual domain of *x* needs to be changed from $\forall x \in Conc_1$ to $\forall x \in Conc_2$. Since a modification of a theory involving re-conceptualisation of its own axioms cannot be made *within* the theory, some complementary reasoning strategy needs to be introduced. For instance, *analogy* is one such reasoning mechanism that works on a different basis than a pure logic. Reasoning by analogy often uses available tacit knowledge, and attempts to use the 'tacit' similarity between the base and target cases to derive some explicit conclusion [7].

Similarly as in the tacit re-formulation discussed in section 4.2, a procedure for the conflict resolution by analogy uses the known predicates to discover tacit matching pattern for the retrieval of the base analog. A repository of the previous design cases may contain an equivalent situation with respect to the undesired feature *D(x)*. Having found a set of situations ES_D that are equivalent according to *D* as defined by Eq. 7, features δ_1, δ_2 ,… appearing in the previous case and blocking the occurrence of undesirable effect *D* may be collected. The purpose is to use 'tacit' similarity for the discovery of some explicit features that may be beneficial in the current case.

$$ES_D(s) = \{\beta \in E^*: s \approx_D \beta\}; \text{ where}$$
$$\alpha \approx_D \beta \text{ iff } \exists\, \delta \in T: \delta(\beta) \vdash (D(\beta) \vee \neg D(\beta)) \qquad \textbf{Eq. 7}$$

Once the designer found some features in the previous cases that may be possibly relevant also to the current task, the consequences of such a choice could be explored on two complementary levels. First, it is possible to learn more about the base case and the context, in which the suggestion of the relevance for a particular feature originated. E.g., designer can explore how the previous design case evolved, and what are the justifications and explanations for using the particular feature.

Second level of the exploration focuses on the current task, and basically aims at finding the place of the identified feature within the current design solution. This level is largely similar to the exploration of the sufficient assumptions defining design contexts, in which the design is consistent – see section 4.2.
The feature discovered by analogy was probably found irrelevant at the first sight, and the designer was originally unaware of it. Alternatively, some earlier assumptions may have rejected this feature without raising any suspicion about its potential usefulness. However, the discovery in the previous case may serve as a trigger that shifts the conceptual base and via introduction of the missing concept removes the inconsistency. There are various ways how the designer may adapt an existing design so that it acquires a discovered feature; combination or mutation are possible methods, as mentioned e.g. in [8, 9].

The uncovered feature δ acts as a *necessary extension* to the current design. The whole strategy for the resolution of inconsistency draws on analogy with a 'parallel' conceptualisation from a previous design, instead of removing it in the 'current' conceptual world. After finding a solution in a 'parallel' world and its adaptation, the result is a knowledge transfer to the 'current' world and consequent design re-formulation.

5 DETAILS OF THE EXPERIMENTAL STUDY

In order to understand the deeper processes that underlie design some 25 experiments were carried out. Each covered a single design task that was vaguely defined (the specificity of initial problem specifications varied however). All tasks were from the domain of controller design and we expected the production of a rough sketch of device and a control algorithm as outputs. Designers used a design knowledge-capturing tool developed in house to record the design specification and context evolution. In addition to capturing explicit design knowledge, we were also capturing the less formal justifications of the design decisions made. Further, pencil and notebook were used for sketching and relevant technical literature for reference.

To illustrate the reasoning described in sections 3 and 4, imagine a designer is asked to develop a strategy for rewinding raw paper from one roll to another one, and simultaneously smoothing and polishing it. Some interesting bits taken from this problem are expressed in a formal language throughout this section, and also illustrated with the sketches of partial design solutions taken from the real experiment.

The designer began with the clarification of customer's demands regarding the functionality of the plant. From the design brief he identified the main goal and formulated it in the customer's language:

desired (property ('paper, 'smooth)) ∧
∧ *desired (property ('paper, 'thickness))* ∧
∧ *value (property ('thickness, VT))*

The requirements as set by the customer were recorded in the design capture tool and justified by the customer's consent. Using generic design knowledge the designer was able to translate the customer's requirements into engineering terms. Suppose, the formal expression of relation between smoothness and more general features is given by the following formulae:

observed-on (property (Material, 'smooth),
part-of (Material, 'surface))

type-of (property (Material, 'smooth),
property (Material, 'structural-prop))

The purpose of the abduction of generalised features showed when the designer inferred the need for some mechanism that would modify the paper surface. The rule for abduction of some hypothetical mechanism from the absence of a desired property on the material *M* in place *P* could be encoded in the following abstract form:

¬*observed-on (property (M, V), P)* ∧
∧ *type-of (property (M, V), property(M,'structural-prop))* ∧
∧ ∃*D: principle ('modification (property(M,'structural-prop), P))*
⇒ *observed-on (property (M, V), P)*

This formula enabled to infer an engineering specification of the task, namely the need for designing a mechanism that is acting on paper and changing its structural properties. The designer continued the started path and looked for the known artefacts delivering such a principle as identified above. Among generic models of the devices he found a principle of rolling using a pressure of a drum on a fixed surface that delivered the desired behaviour. A sketch of such a generic device is shown in Figure 2A, and formal description of its behaviour is given in the simplified ontological definition in Table 1.

Table 1. Ontological definition of a 'rolling drum' prototype

Rolling-drum (RD)	
structure-type:	'composite
descriptive-property:	['weight, 'length, 'diameter…]
has-functionality:	['modify (part (M, 'surface)), …]
has-parts:	['drum, 'spindle, 'drive, …]
has-connections:	[]
has-behaviour: [apply ('pressure, part-of (M, 'surface), 'drum) ⇒ ⇒ apply ('modify, property (M, P), part-of (M, 'surface))]	

Retrieved prototype (Figure 2A) is trivial, and must be further developed into a partial solution for the given design task. Refinement can be done via formulating a set of assumptions when the proposed artefact can be considered as a (partial) design solution. As mentioned in section 4.2, the purpose of the assumptions is to restrict the vast design space to a manageable size. In the previous applications of rolling the designer discovered that material shaped by pressure is usually subject to some heat treatment or similar method. From the previous justifications of such treatment it was possible to infer that the purpose was to loose bonds in material, and prevent its damage.

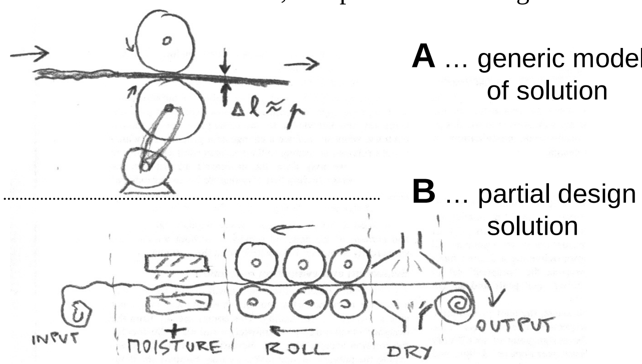

Figure 2. Step from solution model to partial solution

Designer accepted the need for pre-processing, and instead of heat decided to use water to partly 'dissolve' paper and simplify its shaping. Described knowledge transfer occurred, as predicted by the theory in section 4.2, using the tacit knowledge. Records of the underlying reasoning process and tacit analogy were captured as design justifications in the design support tool we made available to designers. As a result of such a tacit inference through analogy, behaviour defined in Table 1 was conditioned by an assumption of modifying a damp paper:

observed-on (property ('paper, 'moist), 'paper) ⇒
(apply ('pressure, part-of (M, 'surface), 'drum) ⇒
⇒ apply ('modify, property (M, P), part (M, 'surface)))

In this particular case, designer assigned the identified assumption the role of a requirement; i.e. he explicitly demanded a mechanism that would increase the humidity of the paper. In other words, he formulated a new requirement that extended the current specification of the design task. The new requirement was also recorded in the design capture tool. Since the tools for the acquisition of design decisions and formulation of design justifications and additional tacit knowledge are detailed in [10], in this paper we skip further details about means for capturing of design decision.

As a direct consequence of the assumed requirement a need arose to relate the rolling mechanism and moisturiser. Since, the current design solution did not address this issue neither had means how to formulate such a relation; the designer referred again to the analogy. He explored in the literature the paper production procedure, and discovered a sequence of actions ranging from shaping the cellulose solution, to removal of excessive water by pressure and heat. He decided to replicate this procedure in his design and introduced causal and temporal relationships between the different sub-processes. For instance, before rolling the paper, it would pass through moisturiser, and afterwards through a dryer:

¬observed-on (property ('paper, 'moist), on ('input-roll)) ⇒
⇒ before (apply ('pressure, part-of ('paper, 'surface), 'drum)),
desired (property ('paper, 'moist), on ('input-roll)))

observed-on (property ('paper, 'wet), on ('output-roll)) ⇒
⇒ after (apply ('pressure, part-of ('paper, 'surface), 'drum)),
¬desired (property ('paper, 'moist), on ('output-roll)))

Prototypic model of a solution was thus extended with pre- and post-processing units in order to comply with assumed requirements. A sketch of partial design solution for smoothing the paper that incorporates the above reasoning is depicted in Figure 2B. Actually, this sketch includes also another transferred assumption about multiple application of pressure in order to achieve better results.

Through tacit reflection on the current solution as proposed in Figure 2B, designer discovered several potential improvements. In order to illustrate the re-formulation of the current position in design, we present one of them. Namely, the issue was that the linear structure was not very efficient with respect to the 'active surface'. Designer calculated that only very small part of the drums was actually acting on paper, and smaller 'active surface' meant higher pressure of the drums. Higher pressure however, increased the danger of damaging the paper.

Therefore, the designer tacitly proposed the requirement of increasing the active surface for rolling. Although the current design did not provide him with sufficient means for achieving such a goal, he decided to keep the current solution and avoid backtracking. According to the theoretical prediction given in section 4.3, he looked at the task from slightly different perspective and eventually discovered the missing feature. In other words, designer appreciated several other features of the solution he did not focus on so far, and drawing on the knowledge of previous cases he explored the possibilities of modifying the layout of the drum assembly.

As a result of such a 'creative leap' [9] he proposed to change the layout from linear to alternate as shown in the sketch A (see Figure 3). Having modified the layout of the drums in the assembly, the designer expressed his tacit satisfaction with the solution, and decided to attend to deeper details. Further development included the proposal for control strategy, identification of measured variables, external parameters and controlled variables. This refinement was rather straight, and we skip it in order to have a look at the next stage, in which the designer tacitly appreciated the commitments he made in the design of a controller. For instance, he looked at the relation between the control precision and overall complexity of the control strategy.

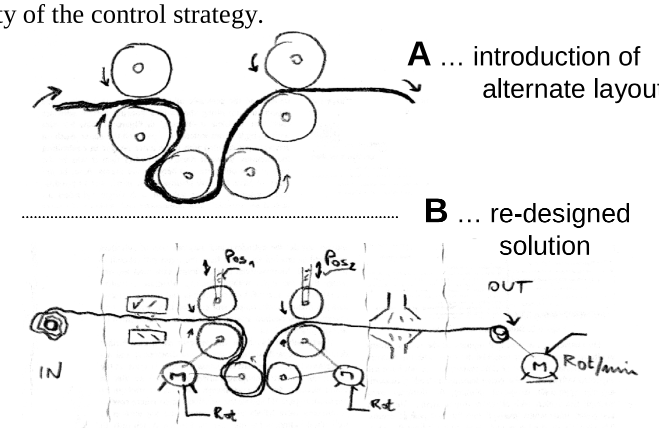

Figure 3. Innovative extensions of partial design

Sketch B in Figure 3 shows added mechanisms for measurement of selected variables and means for controlling the assembly. After reflection a tacit opinion was formulated about structure complexity and vulnerability to frequent disruptions. Designer expressed the doubts about the performance of the proposed artefact with respect to one requirement that was not taken into account yet. Tight winding of paper at the output assumed maintaining a certain tension that was rather difficult to realise in such a complex layout as Figure 3B suggested.

Yet again a shift in the perspective was observed when the designer realised that in the case of alternate layout 'active surfaces' are not evenly distributed among the drums. In each pair there was one drum doing most of the work, whereas the role of the other one was rather unclear. Designer thus conjectured a situation, in which only one drum would remain from each pair. The consequence of such a potential conjecture was the shift from the original application of pressure for smoothing the surface (Figure 2) through a combination of pressure and friction between paper and drum (Figure 3) to be completely replaced by the sole application of friction (Figure 4).

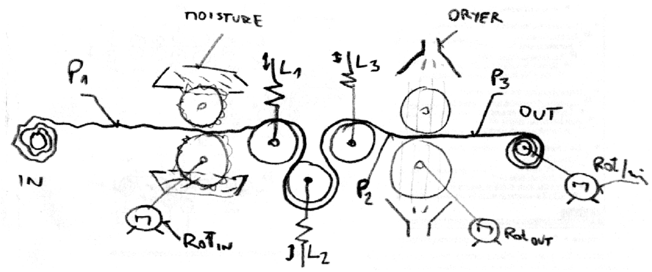

Figure 4. Creative re-formulation of ambiguous design

The shift of a perspective and the removal of selected drums brought also another very desirable feature – the complexity of the whole assembly decreased! Designer thus had more space for positioning the moisturiser and dryer so that the device was more compact and also seemed easier to operate because there were fewer measured and controlled variables. On the conceptual level the designed artefact satisfied the explicit customer's demands, as well as tacit designer's expectations. Simulation or prototype testing could tell us more about the actual performance of the device, but that is already the issue we are tackling in a different project. Basically, the 'final' product of the design is depicted in Figure 4.

As visible in Figure 4, the friction and pressure of each drum is regulated by some adjustable spring or piston ($L_1 \ldots L_3$). Current tension of paper is measured in several places ($P_1 \ldots P_3$) in order to obtain precise information. The actual humidity of paper is measured at input and output (H_1, H_2) to control the amount of water to be added in moisturiser. Moisturiser and dryer are moved 'inside' the drum assembly, and two pairs of drums remained; one to unwind the paper from the input roll (Rot_{IN}) and one to maintain the tension for tight winding at the output (Rot_{OUT}). The output roll is driven by a motor (*Rot/min*) whose performance is also controlled. Desired thickness of paper and maximal allowed values of tension for a particular type of paper are assumed as external parameters (DT, P_{MAX}).

We will leave the scenario at this stage having almost completed the solution. This excerpt sufficiently illustrates the reformulation of the design task and its importance.

6 KNOWLEDGE-INTENSIVE DESIGN SUPPORT

In this section we look briefly at what kinds of knowledge-intensive support could be provided to the different phases of the reasoning process in design, and what are the implications of our theoretical framework. Some of the mentioned means are well known in the design context, whereas some may need certain modifications to suit the nature of reasoning in design.

6.1 Abduction and abduction discrimination

Logical abduction looks for a sufficient feature explaining or implying a desired fact. Since many different explanations may exist for the same fact, it may be useful to restrict our attention to those least presumptive or least abnormal [11]. In design, it is not practical to generate all possible structures that may in theory deliver a desired function or property; thus the proposal of the 'simplest' structures and relations that satisfy our goals and do not make unnecessary commitments, sounds as correct.

However, as stated 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 there is a need to refine the 'least presumptive' design, and according to the theory in section 4.1, it can be achieved by formulating a condition that would discriminate between the alternative designs. Nevertheless, any additional condition introduces more presumptions and commitments to the design.

In order to find the approximately right level of specificity we can think about using the known structure of the design knowledge to decide whether to move further in the abduction. The heuristic may determine if the last abduced formula is already 'a structure' or only an intermediate principle. We may wish to perform the logical abduction until a full-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.

For example, in section 5 we inferred that a device able of modifying the structural property of a material would be sufficient to smoothen the paper. This discovery is however, too abstract to be of any direct use. Thus, it is desirable to continue with the abduction to refine generic notion of modification of a structural property to the pressure application. Pressure applied on a material is only one of many ways for modifying the surface, nevertheless, it is sufficient for our purposes. And one further step of the abduction already revealed a generic structure of 'rolling drums' that delivered the desired functionality.

At this point the abduction was stopped, although we could go further into details and abduce various types, shapes or arrangements of 'rolling drums'. But any further abduction would already make too big commitments to a particular solution, and we wanted to avoid too much commitment in such an early phase of design exploration. The selected structure of 'rolling drums' is rather abstract, but simultaneously it is sufficiently familiar and expressive to be used in the deduction.

6.2 Deduction filtering

The deduction may in general infer a huge number of theorems, and in design that can be undesirable. Fortunately, 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. Various heuristic rules may limit the set of theorems participating in the deduction. For instance, designers typically begin with models of the solutions that are instantiated through some design decision.

Instantiation can be seen as a non-monotonic operation that introduces new theorems to the theory; therefore it seems to be valid to check whether the instance of a model complies with the behavioural predictions made by the model. The justification for such a heuristic is rather simple; models may predict behaviour on a higher level of abstraction, where certain constraints seem to be satisfied. Lower, more detailed levels may however, absorb the influences from multiple models, and thus exhibit slightly different behaviour and compliance with the predictions. It is then useful to prove by the deduction those features of the instances that are somehow predicted by the more abstract model.

Alternatively, it seems to be a useful strategy to check for the conflict between the model prediction and our desired state. For instance, a model of the solution does not violate certain constraint. Although it cannot be proven that model satisfies the constraint, it is sufficient not to violate in order label a solution as acceptable. However, compliance of a model with the constraint does not guarantee that its instance would be complying as well. A constraint that was not computable for the model may be easily computable for the instance – we may therefore focus on the deduction of the critical theorem that participate in the current constraints.

We may want to have as broad deductions as possible and in the same time avoid the irrelevant ones. E.g., we may stipulate that a deduced theorem *D(x)* is irrelevant deduction (i.e. not a new finding) if it further leads to a desired goal:

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

Theorem *D(x)* is formally a new deduction, but it is directly responsible for satisfying our goals and does not bring any new knowledge for the subsequent design. On the contrary, when a change of property *D* (which is not irrelevant according to the previous rule) is one of the deductions, it may be desirable to look into the details why the change occurred and whether it was desirable. Such a change may represent possibly relevant finding that may require designer's external judgement.

The filters mentioned in this section are only some means for the tacit regulation of the amount and content of deduced theorems without withholding really important information from the designer. From other literature, Poole's *prediction of consequences that are in all* [design] *extensions* [11] also seems to be a plausible technique for filtering deduction in design and 'guessing' what form of inference may be suitable.

6.3 Evaluation and design restriction

As already stated earlier, some heuristic incorporated into design theory may assess the relevance of a deduced theorem *D(x)* but it can hardly 'judge' its desirability. The satisfaction comes usually from the designer's tacit (empirical) knowledge and his or her feel for the solution acceptability [5]. Nevertheless, if a designer has experience from the previous design tasks, s/he may compare the consequence *D(x)* deduced in the current problem with the deductions in analogous previous cases. If *D(x)* is occurring 'typically' as a negative feature, and it appears jointly with some constraining condition *P(x)*, then it seems viable suggesting a similar constraint also for the current problem. Obviously, the last word has the designer who must tacitly 'ratify' this potential analogy discovered by heuristic.

Similarly, when a feature *E(x)* occurs in analogous cases as a direct consequence of decision *Q(x)*, then it may help to draw the designer's attention directly to formulae *E(x)* and *Q(x)* that act as sufficient conditions of design acceptability. This operation may be seen as an attempt to tacitly avoid the occurrence of a negative feature. We agree at this point with Cook and Brown [5] who claim that new explicit formulae are not equivalent to the explicated tacit understanding of a problem. On the contrary, explicit design decision can be partially generated using the tacit (empirical) knowledge but still the explicit and tacit forms of knowledge are separated. They complement one another rather than replace or map one to one.

6.4 Re-conceptualisation

Our knowledge about re-conceptualisation is the least definite of all above mentioned at the moment; however, this section explores some of the opportunities that may be beneficial in the support for this very difficult phase of design. In line with the theory in section 4.3 the support can be based on the selection of suitable evaluating conditions and criteria. Only instead of restricting the known facts (predicates) about the current conceptual objects, new concepts are introduced. And similarly as in section 6.3, the tacit knowledge about analogous artefacts may be extremely useful in the identification of potentially desired but currently not known conceptual objects. Using tacit knowledge some 'seeds' of an explicit description of new conceptual entities may be generated.

It seems to be a reasonable proposal to look for analogy on a higher conceptual level than it was the case when trying to filter deductions and assist with the evaluation. Since the aim of re-conceptualisation is the identification of new concepts the designer was not aware of initially, the conceptual jump across the given domain is more likely to bring the current state of the design from a 'local optimum'. Knowledge of the abstraction dependency between various entities may be deployed to discover a non-traditional transfer. Suppose the following rules; a theorem β is abstracted from α if it lies in the '*type-of*' hierarchy above α:

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

In addition to using currently known objects (α) for the exploration, it may be useful to look also at the analogies that occur on the higher level of conceptual abstraction (β). Once a higher-level object β is positively 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 on the same or different level of abstraction – e.g. γ.

If the discovered analogy is too abstract, it may be advisable to look closer at its direct descendants – using exactly the same relation as for abstracting from α. Analogy through a highly abstract entity corresponds to the concept of cross-domain transfer as mentioned already in earlier sections. Such transfer is more difficult than in-domain one; it requires much broader expertise than just within a single domain. Typically it would benefit also from strong analytical skills of designers.

7 CORRESPONDENCE TO OTHER RESEARCH

The formalism proposed in this paper is in accordance with the outcomes reported by Schön [13] who observed in numerous studies of the professionals the oscillation between the solution development and reflection on it. Schön refers to the inconsistencies in the current solution as 'surprises', and claims that any such surprise may trigger a modification of the [conceptual] frame that is used for the solution development. A modified frame allows designers to perceive the objects they were previously unaware of. Despite the complexity of this operation with regards to knowledge, this paper attempted 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 and 6.

In our endeavours we were also inspired by Altshuller's work [14]. This Russian analyst developed a table of possible physical contradictions that may be observed in designs of technical systems, as well as typical means for the removal of such contradictions. His approach can be seen as a generalised case-based reasoning from a large base of previous cases. However, his strongest assumption that the contradictions are always explicitly observable by a designer/inventor is exactly the one we cannot agree with. Altshuller basically deals with re-engineering of artefacts whose performance in some aspect must be amended; the actual observations of the artefacts are straightforward and the comparison with the desired state is possible. In our case, we focus the design of new artefacts (not necessarily inventions) with an incomplete set of desired features exists, without the actual device or technology that can be observed, simulated and/or evaluated using external means. We rely on logical reasoning from our theoretical knowledge and tacit reasoning from the experience for the derivation of a solution from the current set of explicit requirements and constraints. We feel however that some steps in our formalism may be perceived as generalised conclusions of Altshuller's theory.

Tomiyama et al. [15] propose a general model of design that includes abduction, deduction and circumscription. They very precisely divide 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 the hinted supportive heuristics for the discrimination or restriction include also knowledge about design actions in addition to design objects.

Although Tomiyama's (et al.) theory appreciates the incompleteness of problem specification, the need of knowledge modification and iteration, it has several gaps. In our opinion, its circumscriptive mechanism removes the outstanding contradiction only by referring to the other known objects. But what if the designers do not know *all* objects? Theory presented in [15] draws on a questionable assumption that one is able to know all objects in the logical theory that is utilised. However, as stated earlier, there are *tacit* feelings about 'all potential' design objects, but these objects must be *explicit* in order to use them in a design solution! Design by far is not a problem of searching large design spaces; it is mainly about constructing such explicit spaces 'on-the-fly' using tacit insight.

Design by pure analogy as described e.g. by Maher et al. [16] is closer to Altshuller's work. It suits well the problems of re-engineering when the aim is to improve certain features, and there are at least some clues about how the existing system works. The retrieval of a meaningful pair of analogs for the design of new artefacts is more difficult; often there is not enough information in the current task to identify a possible analogy, and inspecting all possible analogies is too exhaustive.

The proposed logic describing sequential design tasks belongs among the non-monotonic reasoning strategies because the satisfiability of certain predicates changes with the introduction of new objects and/or tacit evaluation. Unlike most other approaches tackling the issues of non-monotonic reasoning such as circumscription or default logic, we do not attempt to avoid derivation of a potential conflict. On the contrary, we try to resolve the conflict once it occurs by other (i.e. not necessarily logical) means. Circumscription basically corresponds to the assumption that the objects needed for a solution are only those explicitly mentioned and no more. Our framework sees such a resolution of a conflict as one possibility (see discrimination of alternatives in section 4.1). By circumscribing the conceptualisation we give up that creative element that is so typical for the design tasks; circumscription only restricts an existing solution – nothing new is ever suggested.

Similarly, the aim of default logic is to provide the means for the reasoning in cases with insufficient information available. What the defaults say is, very simplistically, the statement of a form: '*unless known otherwise and the fact is consistent with the theory, consider the fact as valid*'. This approach avoids the contradictions very successfully; however, it does not say anything about what to do if we have that missing information and want to maintain the validity of a respective 'fact'. In other words, how to remove the contradiction using other means than simple deactivation of a default. Nevertheless we acknowledge importance of default reasoning for instance, in choosing the subsequent actions, in finding typical features of objects, etc. However, similarly as with circumscription, defaults loose their power when a conflict is discovered tacitly.

8 CONCLUSIONS

In this paper we show how design can be seen as a sequential process, in which abduction, deduction and various reformulations take place. *Tacit inconsistency* is introduced as an important evaluation criterion in design. Also, several causes of tacit inconsistency are identified in the partial design solutions. These partial solutions are consistent with respect to the current *explicit* specification. Such a logical consistency however, does not guarantee the consistency with all tacit, implicit, untold requirements and/or constraints. It seems appropriate suggesting that an inconsistency of a solution with the tacit design goals is a significant vehicle for the designers to progress in their design tasks. This paper attempted to shed some light on the role of tacit reasoning in the engineering design.

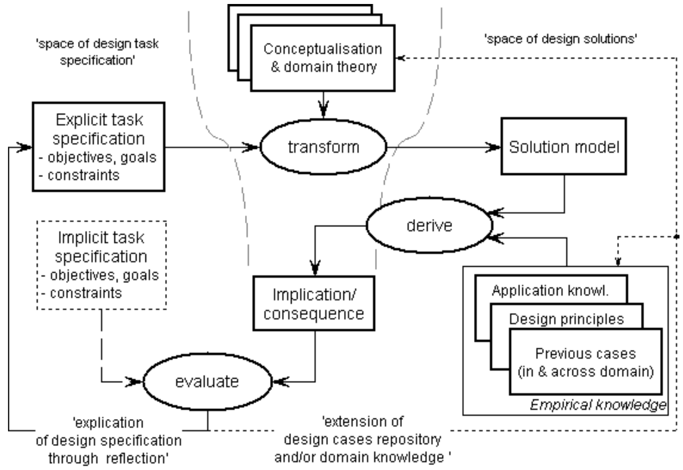

Figure 5. Design as a sequential process

The inconsistency as understood in this paper is uncovered thanks to the designer's tacit knowledge applied through the reflection on the current state of the design. We showed that although *reflection* is a tacit operation on the tacit knowledge, it does not have to be an indescribable art the expert designers are 'born with'. The process of reflecting and using the tacit knowledge has some significant patterns that can be implemented in an artificial design support system. Figure 5 depicts the place of various knowledge sources in the design sequence, and we believe that it represents the iterative and reflective nature of design rather well. The undertaken field study supports our confidence in this aspect.

In the current paper we discuss only the selected issues regarding the generation and modification of the explicit 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 hinted in Figure 5 and shall be investigated in some later papers. Another important aspect not covered in the current paper is the computational complexity of the heuristics proposed above. Efficient control of the design process is crucial for practical applications; however such questions are beyond the scope of this paper.

ACKNOWLEDGEMENTS

This work benefited significantly from the criticism and feedback from John Domingue and Paul Mulholland from KMi. Thanks to Dnyanesh, Lukas and the anonymous reviewers for valuable comments on earlier versions of this paper.

REFERENCES

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

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

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

[5] 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.

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

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

[8] L. Candy and E. Edmonds, Creative design of the Lotus bicycle: implications for knowledge support systems research. *Design Studies*, **17**:71-90, 1996.

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

[10] M. Dzbor and Z. Zdrahal. *Towards a Framework for Acquisition of Design Knowledge*. In *27th Design Automation Conference, (part of ASME DETC)*. Pittsburgh, USA. DETC01/DAC-21049, 2001.

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

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

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

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

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

[16] M.L. Maher, J. Poon, and S. Boulanger, *Formalising Design Exploration as Co-Evolution: A Combined Gene Approach*, in *Advances in Formal Design Methods for CAD*, (J.Gero & F.Sudweeks, eds) Chapman & Hall, 1-28, 1996