Interactions between specification and solution in design

Martin Dzbor and **Zdenek Zdrahal** §

Abstract. This paper introduces a new model of a design process. This 'sequential model' takes into account a reflective nature of designing, and it is based on the interplay between two conceptually distinct knowledge sources – an explicit specification of a design problem and a solution to it. The approach is novel in the former investigated aspect that is presented as a semi-formal operation of framing, i.e. interpretation of a given problem using certain conceptual primitives. We argue that the aspect of interpreting design problems lacks a similar rigorous investigation as the aspect of problem solving received in both design theory and methodology. This paper discusses two patterns of modelled design decisions.

1. INTRODUCTION

While many efforts have been made toward computational models of certain parts of engineering design, there is a large portion of design knowledge that is rather intuitive. It must be noted that many existing models of design task deal with finding a suitable solution to a given design problem. They leave aside the issues connected with the explicit formulation of a problem. This paper describes selected findings in modelling both, the process of problem solving, as well as problem articulation.

Typically, *design task* occurs when an agent decides to change the status of the surrounding world [1]. It is a goal-oriented process leading from initial objectives to an artefact realising the change. Usually, design is an ill-structured task [2], i.e. a solution may not be found until significant effort to understand the 'structure' of the problem has been made. Nonetheless, what does it mean to 'give a problem its structure'? Is it possible to model such a structuring or *framing* using formal language instead of 'intuition', 'insight' or 'experience' [3]?

We believe the questions, as those above, are extremely important for improving our understanding of design as a process. The primary reason is that designers are rarely presented with a detailed specification of design problem [4]. A specification of a design problem is built from the initial vague descriptions. We argue that a specification is subject to the same evolution as a design solution. Moreover, we believe that a set of statements regarding a desired state may be proclaimed 'a specification' only at the end of design; i.e. once a designer is satisfied with a proposed artefact. An idea of 'co-evolving' design solution and specification is not a new one [5]. However, there is limited formal account of this important phenomenon in the literature.

Operation of design framing appears, for instance, in works of Schön [4], who mentions that professional designers simply 'know' what to do to achieve their goals; their past experience helps to tackle the current problem. In this paper, we articulate several interesting patterns that this peculiar operation of 'framing' involves. We devote more space to a more abstract, conceptual level that is only illustrated by more specific, operational models.

In the proposed approach, the engineering design is understood as an iterative transformation of initial incomplete requirements to an acceptable specification of design problem and its solution. Proposed (partial) solutions influence the requirements. In turn, the modified requirements refine the solutions, thus revealing a principle of the co-evolution of mutually complementary concepts.

In order to investigate the relationship between the problem specification and solution development, we conducted a set of 24 experiments with design practitioners. They were solving tasks from a domain of controllers for large-scale systems. We illustrate our findings on one of the sessions – a design of a controller for a paper-smoothing plant. For additional details on experiment and the experimental tools, see also [6, 7].

Designer's task was to suggest a layout, structure and control strategy for a plant that takes raw, wrinkled paper on the input, and delivers a smooth paper with an even thickness at the other end. In a design process, we were particularly keen to observe the reflective behaviour and different occasions of problem re-formulation. Main milestones of the design process are shown below, together with sketches of the assembly for the illustrative purposes. We shall refer to the selected reasoning steps in the next section.

1. an initial principle for smoothing contained a pair of rolling drums with a paper passing through a gap between;
2. this layout was enhanced, when a designer proposed dampening the raw paper before entering the rolling drums, and drying it afterwards to achieve acceptable performance
   → *introduction of an additional assumption restricting the scope of artefact acceptability* (see Figure 1);

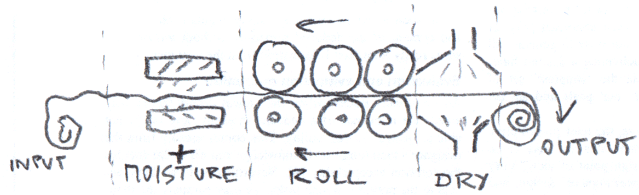

Figure 1. Linear sequence of drums, pre- and post-processing

3. another reflective turn occurred when designer found out that smoothing depends on the pressure of the drums, which may damage certain types of paper; an alternative is to reduce the pressure and increase the size of the plant
   → *contradictory requirements are spotted and attended*;
4. various layouts of the drums were considered (linear alternate)
   → *re-interpretation of a conceptual term in the current design frame leads to an alternative solution* (see Figure 2);

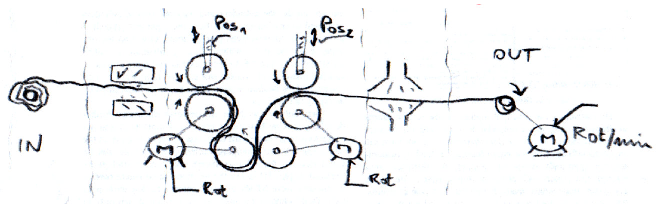

Figure 2. 'Zigzag' sequence = smaller dimensions, better quality

5. eventually, a principle of rolling was given up and replaced by 'abrasion' (this is accompanied by a re-design of a solution)
   → *shift in a design perspective (frame), seeing smoothing as an instance of a different physical principle* (see Figure 3);
6. final artefact (design solution) consists of: (a) pairs of drums to unwind the raw paper and maintain the tension before the output coil; (b) rolling drums 'merged' with dampening mechanism; (c) from each pair only one drum remains; the drums are positioned in a 'zigzag' manner (see Figure 3, below)

In our opinion, the observed modifications cannot be attributed purely to a search for the 'right' solution. The milestones described above feature also a process of <u>*exploration*</u> [8] that involves also a construction of a design space and interpretation of a design problem. However, we do not agree with a prevailing view that such an

§ Knowledge Media Institute, Open University, Milton Keynes, MK7 6AA, UK; Email: {M.Dzbor, Z.Zdrahal}@open.ac.uk

exploration cannot be expressed in a formal language. It may show many aspects of intuitive and tacit reasoning but at least certain patterns seem to be explicable in terms of evolving conceptual frames and solution acceptability.

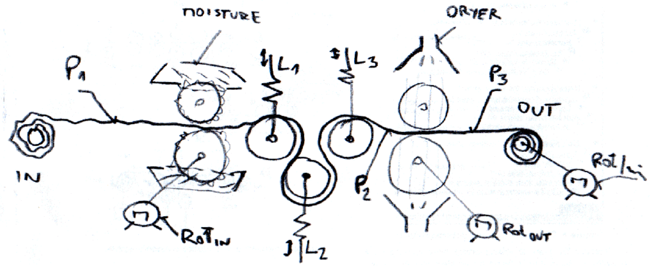

Figure 3. Design solution in a re-interpreted design frame

2. WHAT IS A DESIGN FRAME?

In the available literature, Nakakoji et al. address the issue of 'design problem framing', and formally define a 'design perspective' in the following terms [9]. *It is a* <u>*point of view*</u>*, which implies that certain design goals exist, certain bodies of design knowledge are relevant, and certain solution forms are preferred.* The authors use the term 'design perspective' mainly for expressing designer's intentions. In this context, a design perspective can be seen as a kind of vocabulary of concepts used in the problem solving phase. We believe that a definition of design frame may be based on the above-given position.

The first major gap in the existing research on design is an interpretation and a formal clarification of terms 'frame' and 'framing'. We agree with the framing is indeed an important operation that precedes the problem solving, and complements it [4, 10, 11]. However, what are the implications of 'framing' on the knowledge level? What is actually happening with designer's knowledge during the problem framing? Can this 'woolly' operation be expressed in a formal or semi-formal language? These are some of the challenges we tackle in the further text.

2.1 Essential definitions

It seems to be redundant to say that designers do design; i.e. they address <u>design problems</u>. Is there really a redundancy? We believe that what designers do, is trying to <u>satisfy the explict problem specification</u> (say $\boldsymbol{S}$). In other words, they attempt to specify a given design problem (say $\mathfrak{DP}$) in terms of explicit 'statements' ($\boldsymbol{S}$) from a conceptual space of allowed problem specifications (say $\mathcal{S}$). Symbol $\mathcal{S}$ stands for a conceptual set, from which the explicit interpretation $\boldsymbol{S}$ is instantiated. In order to solve design problem $\mathfrak{DP}$, a designer associates it with a suitable, <u>explicit specification</u> space $\boldsymbol{S}$, and tackles the problem specified (read '*interpreted*') in this way. In terms of logical theories, we may say that designer *circumscribes* [12] a design problem by declaring that only the statements from the explicit specification $\boldsymbol{S}$ are needed for interpreting (and later solving) the problem[1]. This 'operation' can be represented by an assertion of a relation defined in (1), below.

$$specifies_{\Phi}(S, \mathfrak{DP}) \qquad \textbf{(1)}$$

However, such assertion can be made only within certain conceptual boundaries – a <u>conceptual design frame</u>. We define design frame $\boldsymbol{\Phi}$ as a pair of two circumscribed knowledge spaces that are constructed on top of the allowed problem specifications $\mathcal{S}$ and the relevant problem conceptualisation $\mathcal{T}$. Thus, 'framing a design problem' means articulating a set of conceptual objects $\mathcal{T}$ that may be used for doing the design, as specifiable by the concepts from $\mathcal{S}$ (relevant problem specifications). Let us use symbols $\mathcal{S}^*$ and $\mathcal{T}^*$ for a formal notation of the circumscribed knowledge spaces constructed on top of both conceptual entities ($\mathcal{S}$ and $\mathcal{T}$).

We define $\mathcal{T}^*$ as 'a closure' constituted by the selected conceptualisation $\mathcal{T}$ and an appropriate domain theory $\boldsymbol{DT}_{\mathcal{T}}$. Domain theory $\boldsymbol{DT}_{\mathcal{T}}$ is a problem-independent knowledge, possibly applicable to many different problems. Consider, for example, physics. It is a domain theory applicable for a design of an elevator as well as a spacecraft. However, for different problems, different parts of the domain are used. I.e., a generic domain theory $\boldsymbol{DT}$ must be 'instantiated' for a particular conceptual base $\mathcal{T}$, in order to obtain a usable theory for solving the problem. Let us therefore, refer to closure $\mathcal{T}^*$ as a ***problem solving theory***. Similarly, we define closure $\mathcal{S}^*$ as an instantiation of the potentially relevant problem specification statements ($\mathcal{S}$) in the chosen conceptualisation ($\mathcal{T}$). Finally, we express design frame $\boldsymbol{\Phi}$ formally, as follows: $\boldsymbol{\Phi} = \langle \mathcal{T}^*, \mathcal{S}^* \rangle$. Let us strictly interpret the terms used in the presented definition:

a) A space of problem conceptualisations $\mathcal{T}$ may be seen as an ontology, a vocabulary of basic concepts, for which designer decides they are available for expressing a statement about a particular design problem. A conceptual base may include, e.g. terminology for the definition of functional and structural objects. It may include problem-specific mappings between the functions and structures, e.g. in form of behaviours [13].
b) An applicable domain theory $\boldsymbol{DT}_{\mathcal{T}}$ may be a shared ontology, a generic vocabulary defining the background [14], against which any conceptualisation is applied. Domain theory per se is too generic and abstract to be of any direct use in problem solving. In order to derive a useful problem solving theory, it must be interpreted in a specific conceptualisation!
c) A space of relevant problem specifications $\mathcal{S}^*$ is complementing the problem solving theories $\mathcal{T}^*$. Its principal purpose is to provide a vocabulary for expressing the desires or intentions of a designer in a particular problem [9]. It can be seen as a set of relations that can be formulated using the elements of a particular problem solving theory.

Design frames, as defined above, do not exist 'per se'! They are highly volatile, and are constructed (and re-constructed) on the fly using the information that is believed to be relevant to a particular design problem. Typically, a designer uses customer's initial problem specification ($\boldsymbol{S}$) to identify similarly looking design situations, he or she is familiar with. Thus, a design frame may indeed be seen as a meta-relation of similarity of the current case with a past design case or a set of cases. This is a desirable feature, because it corresponds with Schön's theoretical prediction [4].

We argue that the conceptual design begins with an attempt to formulate a minimal sub-set $\boldsymbol{T} \subseteq \mathcal{T}^*$ that <u>satisfies</u> a given problem specification. A 'given problem specification' is hereby understood as set $\boldsymbol{S} \subseteq \mathcal{S}^*$, and defined as a formulation of the <u>explicit design requirements and/or constraints</u>. Symbol $\boldsymbol{S}$ denotes all such statements that serve to specify desires about design problem $\mathfrak{DP}$, to which a designer made a specific and explicit commitment. In other words, designer tries to shrink the vast space provided by a problem solving theory ($\mathcal{T}^*$) into a manageable size that can be manipulated with. This manageable chunk corresponds to 'solution model' – a term appearing in [15, 16]. Or perhaps, due to its generative nature, it would be better to call it a 'problem solving model'.

Formally, a ***problem solving model*** is a minimal sub-set of the problem solving theory that sufficiently <u>*satisfies*</u> the explicit problem specification. Relation '*satisfies*' is binary (with $\boldsymbol{\Phi}$ as a contextual parameter), because it associates a problem solving model $\boldsymbol{T}$ with current explicit problem specification $\boldsymbol{S}$, and this happens within an underlying conceptual frame $\boldsymbol{\Phi}$. Formal definition of a problem solving model is given in (2).

$$T \subseteq \mathcal{T}^*: satisfies_{\Phi}(T, S) \wedge (\neg \exists\, Y \subset T: satisfies_{\Phi}(Y, S)) \qquad \textbf{(2)}$$

[1] This is indeed a circumscription whose purpose is to 'close' designer's understanding of an incompletely defined (i.e. ill-structured) problem.

From an operational point, it is possible to distinguish design requirements $\boldsymbol{R}$ from design constraints $\boldsymbol{C}$, and assert that a problem specification is a union of the two – i.e. $\boldsymbol{S} = \boldsymbol{R} \cup \boldsymbol{C}$ [14]. In this context, requirements are those statements demanding the explicit presence of a particular feature, whereas constraints are conditions that must not be explicitly violated by a design solution. More on the conceptual distinction between requirements and constraints appears also in [6]. In this paper, we only present a simplified definition of relation '*satisfies*' in (3) below:

$$satisfies_{\Phi}(T, S) \Leftrightarrow \{(S = R \cup C) \Rightarrow T \models R \wedge \neg(T, C \vdash \bot)\} \qquad (3)$$

In (3), the symbols used have their usual meanings [17]. Symbol '$\models$' stands for a semantic entailment, '$\vdash$' is a proof-logical implication, and '$\bot$' is an 'empty' formula (a contradiction). Accordingly, theory $\boldsymbol{T}$ is a problem solving model in respect to a given explicit problem specification $\boldsymbol{S}$ and a design frame $\boldsymbol{\Phi}$, if it is complete in respect to the required features ($\forall r \in R: T \models r$), and admissible in respect to constraining conditions ($\neg \exists c \in C: T, c \vdash \bot$).' In other words, a candidate solution must have a potential to deliver all required features without contradicting the constraints.

However, the explicit problem specification is only an interpretation of a design problem $\mathfrak{DP}$, which is used in problem solving. It is not the same as problem $\mathfrak{DP}$. We argue that the existence of a problem solving model $\boldsymbol{T}$, for which relation '*satisfies*' holds, is a necessary but not sufficient condition of declaring it a 'design solution'! In addition to a satisfaction of an explicit specification, discovered problem solving model $\boldsymbol{T}$ must be also '*acceptable*' as a design solution! Often, such a relation as '*acceptable (T)*' cannot be defined formally. Acceptability may be appreciated subjectively and tacitly, but may not be expressed in the languages of $\mathcal{S}$ or $\mathcal{T}$.

Nevertheless, it may be defined as a residual category. Formula (4) may help understand this weird, residual relation of *problem solving model acceptability*'. What does it mean that a relation is residual? We argue that it means the same, as the statement made in [18] saying that certain tacit decisions cannot be stripped of their contextual background. It may be difficult to define exact conditions of 'acceptability', but every designer may proclaim a certain problem solving model acceptable or not, when s/he sees it. Tacit decision on acceptability makes sense only in a particular context, such as frame $\boldsymbol{\Phi}$ in (4). In the formal sentence (4), the used symbols correspond to the definitions made earlier in this section.

$$satisfies_{\Phi}(T, S) \wedge \neg acceptable_{\Phi}(T) \Rightarrow \neg specifies_{\Phi}(S, \mathfrak{DP}) \qquad (4)$$

We interpret formula (4) that whenever an otherwise valid problem solving model is not accepted by a designer as a design solution, it may point to an incorrect (~ incomplete) specification of the actual design problem. The explicit interpretation of a design problem $\mathfrak{DP}$ in terms of statements $\boldsymbol{S}$, does not reflect the real design problem $\mathfrak{DP}$, and as such it may be desirable to amend it. Such an amendment however, features a design extension, re-formulation, or re-framing that constitute the remainder of this paper.

2.2 Sequence of conceptual decisions in design

In the following paragraphs, we propose a sequential model of a design process on the level of conceptual frames. The building bricks of such a model are the conceptual entities identified in the previous section. The model is defined as a sequence of decisions driven by the validity (a.k.a. returned values) of predicates '*satisfies*' and/or '*specifies*', as defined in section 2.1. The sequence is running across several mutually dependent, conceptual levels – each numbered from 'I' to 'V'. The model shows design as interplay of two distinct knowledge-level actions represented by semi-formal predicates ('*specifies*' and '*satisfies*'). The former action is amending problem specification, the latter attempts to solve the specification.

Let us define a few predicates representing earlier-mentioned re-interpretations. These definitions help interpret the model, and read it intuitively as a sequence of *decisions* followed by *actions*. The simplest form of re-interpretation of a design problem is attempting to explicate a statement that is believed to refine the current specification. If such a statement can be articulated using the current frame $\boldsymbol{\Phi}$ (and the language of the chosen conceptualisation $\mathcal{T}$), design continues with an action shown in Figure 4, level III. A simple extension within a known conceptual frame is defined in (5).

$$\begin{gathered} can\text{-}extend_{\Phi}(\mathrm{S}) \Leftrightarrow \\ \exists s \in \mathcal{S}^*: \mathrm{S} \subseteq \mathcal{S}^* \wedge \mathrm{S'} = \mathrm{S} \cup \{s\} \wedge specifies_{\Phi}(\mathrm{S'}, \mathfrak{DP}) \end{gathered} \qquad (5)$$

Slightly more complex decision features an operation attempting to re-interpret the current design specification. It corresponds to an articulation of new conceptual objects for a problem specification. Schema (6) differs from the one in (5) in the fact that such a re-interpretation can be done only when changing the conceptual frame (i.e. the currently used conceptual categories from set $\mathcal{S}$). Decision, whether a 'tacit' non-acceptance may be resolved by re-formulating the current interpretation of a design problem in terms of a newly discovered perspective, is shown below.

$$\begin{gathered} can\text{-}reformulate(\mathrm{S}, \Phi) \Leftrightarrow \\ \exists \Phi_N = \langle \mathcal{T}^*, \mathcal{S}^*_N \rangle: \mathrm{S''} \subseteq \mathcal{S}^*_N \wedge specifies_{\Phi N}(\mathrm{S''}, \mathfrak{DP}) \end{gathered} \qquad (6)$$

Finally, the most radical form of interpretation amendment is defined in (7). In this particular situation, the conceptual foundation of the current perspective is changed (i.e. $\mathcal{T}$ – set of conceptual objects lying at the very basis of a conceptual frame $\boldsymbol{\Phi}$). However, this change is not performed totally 'intuitively'. A new frame ($\boldsymbol{\Phi}_N$) is articulated re-using a part of the current interpretation (i.e. a conceptual base for a problem specification $\mathcal{S}$). Thus, similar concepts

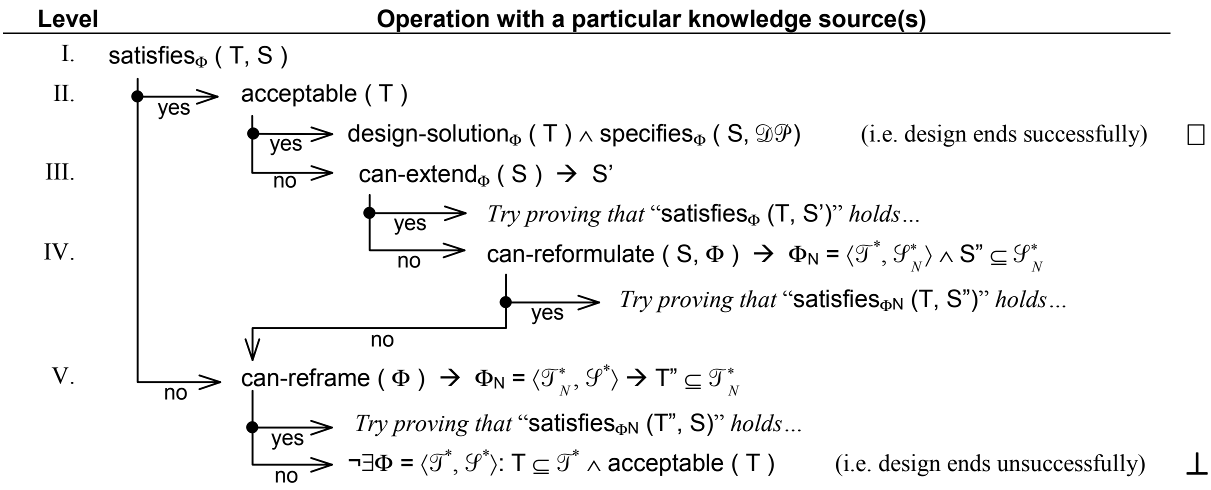

Figure 4. Sequential model of a design process on the level of conceptual frames

specify a design problem, albeit they are interpreted in a modified design frame – a different context.

$$\begin{gathered} \textit{can-reframe}\ (\,\Phi\,)\ \Leftrightarrow \\ \exists \Phi_N = \langle \mathcal{T}^*_N, \mathcal{S}^* \rangle : \mathrm{T}'' \subseteq \mathcal{T}^*_N \wedge \mathrm{S} \subseteq \mathcal{S}^* \wedge \textit{specifies}_{\Phi N}\,(\,S, \mathcal{DP}\,) \end{gathered} \tag{7}$$

The breakdown depicted in Figure 4, is not an exhaustive combination of different states that can be achieved with and without the modification of a conceptual frame. Nevertheless, the proposed model is exhibiting the empirically observed interplay of knowledge-level actions. Oscillation between complementary knowledge sources is observable in an exchange of information and control during a design process. In a construction of a model, we obey a few simple rules:

1) When using terms 'requirements' and 'constraints', we always mean hard, strict demands that *must not be relaxed …*
2) Monotonic extension of a problem specification corresponds to a designer's attempt to 'fine tune' a problem solving model, to *reduce* the number of derivable alternatives. Problem specification can be refined, only if a valid problem solving model exists for the current conceptual frame…
3) Sentences "*Try proving that* **λ** *holds…*" represent <u>*a recursive step*</u> returning to level I. of s sequential model, and a designer's attempt to address the unresolved problem by amending one or another available knowledge source. It is 'an order' to an agent to "*evaluate predicate* **λ** *with the new arguments provided.*"

Let us describe two of the patterns of reasoning that are explainable directly from the proposed sequential model. Note that these are not problem solving schemas. They are proposed as abstract models of certain types of reasoning that may be observed in design. Due to limited space of this paper, we discuss only two patterns in details. First, it is a non-monotonic introduction of new domain knowledge from an external source in form of new assumptions. The second is an example of conceptual re-framing.

2.3 Frame alignment and design assumptions

Consider the following situation that was observed during design of a paper-smoothing plant. This account refines point 2 in the list of milestones (section 1), and in the sequential model is represented as a branch leading to predicate '$\textit{can-extend}_{\Phi}(S)$'. At certain stage, designer considered a sequence of rolling drum pairs[2] that would apply pressure on the raw paper, thus reducing its thickness and smoothing its surface. When deriving consequences of this simple approach, he observed that the effectiveness of both operations depended on the actual pressure and 'active' surface of drums. The higher pressure (bigger active surface), the better quality could be achieved. Nonetheless, paper was a relatively fragile material with certain limits in respect to pressure and tension, and it could easily tear, if these limits were exceeded.

Designer decided to discard a simple sequence of rolling drums as an unacceptable alternative, despite the completeness and admissibility of the candidate in respect to the explicit specification. The paper was smoothed and thickness was reduced, exactly as desired. However, designer assumed another condition that was never mentioned in customer's initial specification. In addition to the pair of existing statements, he demanded that paper remained whole (i.e. not torn or otherwise damaged). In a justifying record of this introduction of new knowledge, he maintained that it was "*such an obvious condition that nobody bothered to emphasise it explicitly*".

When we attend to this apparently straightforward situation, we may note that the conceptual base ($\mathcal{T}$) for a problem interpretation remained unchanged. Addition of a new assumption monotonically extended and refined the <u>explicit specification</u> (***S***)! However, this monotonic extension had strong implications on the otherwise non-monotonic problem solving theory ($\mathcal{T}^*$) and candidate solution (***T***). Such an articulation rendered the current problem solving theory inconsistent – the new condition was obviously violated by the 'old' candidate solution. In this case, the extension led to an introduction of pre- and post-processing units to the plant (shown in Figure 1 as 'moisture' and 'dry'). These units softened the paper before rolling, so that lower pressure was needed, and the danger of tearing was reduced.

What happened in this situation from a knowledge-level point of view, can be seen as an attempt to align an explicit conceptual frame with an 'internal', implicit one. The internal frame may contain additional assumptions and expectations, which may tacitly influence designer's decision on solution acceptability. Mostly, these expectations remain 'hidden'; however, when an admissible candidate solution is judged as unacceptable, they may become extremely useful. Reflecting on the 'hidden' (perhaps empirical) expectations may lead to an explicit articulation of a new statement. With a new statement, the existing problem solving theory may become inconsistent, and may need to be conceptually amended. However, the actual addition of the new concepts) is already a topic covered by a different reasoning schema detailed in section 2.4.

2.4 Contradictory theory & conceptual re-framing

Consider another type of re-interpretation that was observed repeatedly in the design of paper-smoothing plant, as well as other experiments. The following situation refines points 3, 4 in the list of milestones (see section 1). A sequence of rolling drums with pre- and post-processing units, as shown in Figure 1, depicts a candidate solution at a certain stage. This solution had no apparent weakness; it even complied with designer's experience from similar problems (e.g. metal sheet rolling). However, a hidden weakness appeared when a designer took into account efficiency and economy of the overall operation. As already mentioned in section 2.3, higher pressure or larger active surface of rolling could remedy a low quality of product. The increases in pressure were tackled earlier, and it was resolved to add the additional processing steps to soften the paper, rather than increase the pressure.

Active surface of paper could be increased easily – by adding more pairs of rolling drums to the sequence. Nonetheless, the sequence could not grow forever, because a larger size meant a more difficult maintenance. It was clear that trying to design an assembly with fewer drums was desirable in order to simplify maintenance. However, fewer drums rapidly decreased the quality or increased the danger of damaging paper. Thus, designer found himself in a 'magical circle' of mutually contradicting requirements.

He resolved a threat of deadlock by shifting the conceptual foundation. Instead of squeezing or expanding the layout of the rolling assembly in 'one dimension' (i.e. linearly laid-out pairs of drums), he articulated concepts 'two-dimensional layout' and 'two-dimensional squeezing'. When it was impossible to go beyond the constraints in one dimension, he brought in another dimension. The result of such a shift was an introduction of an alternate (zigzag) layout of drums that featured larger effective surface acting on the paper. Thus, fewer pairs were needed, and the size- as well as pressure-related constraints could be managed – simultaneously! New concept is clearly visible in re-designed assembly in Figure 2.

A similar reasoning step introducing new concepts to tackle an outstanding problem was repeated in milestone #5 (see section 1). In that step, designer made a more radical conceptual re-framing. He re-visited his interpretation of the basic, underlying principle of rolling. Instead focusing on pressure application during rolling, he became aware that in a zigzag layout one drum was using much larger surface than the other one in a pair. Hence, he removed a 'redundant' drum from each rolling pair, thus replacing the principle of pressing by a principle of abrasion. The components of the plant

[2] Let us mark it by symbol ***T*** – a candidate solution.

remained the same, but their roles and functions were re-interpreted, eventually leading to a design depicted in Figure 3.

Unlike in section 2.3, where only a problem specification was monotonically refined, this operation went far deeper. It all began with a contradictory problem solving theory ($\mathfrak{T}^*$), in which some constraints were violated (i.e. $\exists c \in C: \mathfrak{T}^*, c \vdash \bot$). Since none of the violated conditions could be 'retracted', designer was forced to revisit the existing domain theory, as interpreted in conceptual terms $\mathfrak{T}$. Having defined new conceptual primitives (e.g. '2D layout' or 'abrasion'), he actually changed his conceptual vocabulary for interpreting and solving design problem $\mathfrak{DP}$.

A new conceptual base ($\mathfrak{T}_N$) triggered articulation of a new conceptual frame $\boldsymbol{\Phi}_N$, and in the context of new frame, the conflicting constraints 'lost their edge'. The 're-conceptualised' problem solving theory regained its consistency, and design could continue – at least, until other, explicated 'hidden' expectations in the next steps invalidated the current perspective… We believe that this schema gives a knowledge-level, theoretical background to a similar, *empirically* observed resolution of physical contradictions in the inventive problems by referring to less-usual concepts [15]!

3. Discussion

The schemas proposed above are defined on an abstract level of concepts and conceptual design frames. They model selected patterns, which conceptually underpin a designer's decisions on the level of his or her knowledge of design problem. It is obviously interesting to investigate, what is going on, when a designer calculates relation '$satisfies_{\Phi}(T, S)$', or looks for a new conceptual frame to resolve an explicit conflict. Although operational models of the conceptual operations are beyond the scope of this paper, let us discuss a few remarks in that direction.

Why is a logically admissible design questioned? We already mentioned that one reason might be in unaligned explicit and internally used conceptual frames. We also referred to a term of 'tacit knowledge'. These features deserve further attention because they seem to be closely related. Recall a definition of design frame as an interpretation of a design problem using a familiar vocabulary from similar design cases tackled in the past. That 'woolly' term of internal frame draws on these familiar, past situations. A designer may perceive a similarity between the current and previous cases on different levels of abstraction. Sometimes, the analogy may be so abstract or so complex that it is hard to articulate it. While simpler analogies may be re-used in the interpretation of a new problem, the more abstract ones may remain 'hidden'.

These past experiences may be difficult to articulate formally and explicitly per se, as standalone relations of analogy. However, the essence of these relations may come forward in the context of a particular solution candidate. Examination of further consequences of a particular commitment may raise designer's awareness of inadequacy in the current approach. Such an origin may have design extension described in section 2.3. Past knowledge of pre-heating of metal slabs before rolling was contrasted with a lack of any similar operation in otherwise analogous problem. Designer went more in depth to investigate the reasons of pre-processing in the past case, and became aware of material flexibility. Eventually, a notion of improved flexibility was translated into existing context as a new assumption in respect to fragility of paper. An, 'intuitive' articulation of new condition loses its mystery in light of analogy-based discovery of similarity and subsequent knowledge transfer…

Similarly, re-interpretation of concepts in section 2.4 may seem confusing and sudden. It surely is sudden and unexplainable within the particular conceptual frame. However, designers use framing as a *temporary circumscription* of a vast, incomplete problem space. By articulating a set of conceptual primitives, they deliberately circumscribe the world for their problem solving. Nevertheless, they still may re-open that closure, and circumscribe the problem space in a slightly different manner. And here appears the main difference of the proposed sequential model in comparison to the existing research [1, 12, 13, 16]. Most other models begin with an assumption along the following lines: "*Given a problem specification, we can apply such and such problem solving method…*"

If a need arises in most other research models to update problem specification, it is referred to a designer and his 'deep immersion' to a problem domain [3]. In other words, most other work concerns with *solving* design problems. In this paper, we proposed a conceptual basis for *interpreting* design problems by solving them. In that aspect, theory reported in this paper extends rich empirical findings of Schön in the field of *reflection* on design actions [4]. Moreover, the 'problem interpretation through solving' and 'solution generation through re-interpretation' also address the exploratory, iterative nature of design argued at the beginning. Design problems are inherently open; they are closed (read 'circumscribed') only 'temporarily' to address the incompleteness and complexity…

References

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

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

[3] R.L. Dominowski and P. Dallob, *Insight and Problem Solving*. In *The Nature of Insight*, R.J. Sternberg & J.E. Davidson (Eds). MIT Press: USA, pp.33-62 (1995).

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

[5] S. Nidamarthi, A. Chakrabarti, and T.P. Bligh. *The significance of co-evolving requirements and solutions in the design process*. In *Proc. of 11th ICED*. Finland (1997).

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

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

[8] B. Logan and T. Smithers, *Creativity and Design as Exploration*. DAI research paper 597, University of Edinburgh. (1992).

[9] K. Nakakoji, T. Sumner, and B. Harstad. *Perspective-based critiquing: helping designers cope with conflicts among design intentions*. In *Proc. of AI in Design'94*. pp.449-466 (1994).

[10] G. Fischer. *Domain-Oriented Design Environments*. In *Proc. of 7th Knowledge-Based Software Engineering Conf.*, pp.204-213 (1992).

[11] M. Dzbor. *Design as a Problem of Requirements Explication*. In *Proc. of ECAI'2000 Workshop on Knowledge-Based Systems for Model-Based Engineering Design*. Berlin, Germany (2000).

[12] J. McCarthy, Circumscription – A Form of Nonmonotonic Reasoning. *Artificial Intelligence*, **13**:27-39, (1980).

[13] T. Bylander and B. Chandrasekaran. *Understanding Behavior Using Consolidation*. In *Proc. of 9th Intl. Joint Conference on AI (IJCAI'85)*. Los Angeles, California. pp.450-454 (1985).

[14] B. Wielinga, J.M. Akkermans, and A.T. Schreiber. *A Formal Analysis of Parametric Design Problem Solving*. In *Proc. of 9th Knowledge Acquisition for KBS Workshop*. Banff, Canada. pp.37:1-15 (1995).

[15] G.S. Altshuller, *Creativity as an Exact Science*. 1984, USA: Gordon & Breach Science Publishers.

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

[17] M. Levin, *Mathematical logic for computer scientists*, MAC project technical report TR-131, MIT: Cambridge, Mass. (1974).

[18] 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).