Proceedings of DETC'02:
ASME 2002 Design Engineering Technical Conferences
and Computers and Information in Engineering Conference
Montreal, Canada, September 29 – October 2, 2002

DETC2002/DTM–34009

MODELLING DESIGN INTERPRETATION AS INTERACTION OF PROBLEM FRAMING AND PROBLEM SOLVING

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

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

ABSTRACT

Design is usually considered a reflective and ill-structured process. This paper presents a new, sequential model of such a process. Design is modelled as an interplay of two conceptually distinct activities – an explicit specification of a problem and a solution to it. The approach is novel in defining an operation of framing, i.e. *interpretation* of a given problem using certain conceptual commitments. So far, the interpretation of design problems enjoyed less rigorous investigation as the aspect of problem solving in both design theory and methodology. In this paper we model three reasoning patterns for (re-)interpreting design problems. These patterns are complemented by an operational framework based on abstracted similarity, and illustrated by extracts from experimental studies.

1 INTRODUCTION

Much effort has been put toward computational models of engineering design, and there are numerous theories accounting for problem solving aspect of this extraordinary activity. However, many existing models of design task focus solely on finding a suitable solution to a given design specification. The issues with the interpretation of problems are left aside, and empirical studies often report various intuitive decisions made by the designers at this stage [1]. This paper presents and models design as an interaction of two knowledge-level actions – problem solving as well as problem specification.

We may define engineering design is an iterative transformation of the initial incomplete requirements to an acceptable design solution [2]. Partial and candidate solutions are developed to validate the correctness of the explicitly articulated requirements, which may be modified to address the outstanding inadequacies. In turn, the modified requirements further refine and develop the solutions, thus revealing a principle of the iterative co-evolution.

It has been widely accepted that design is an ill-structured task [3], which requires a significant effort to understand the 'structure' of the problem. Nonetheless, what does it mean to 'give a problem its structure'? Is it possible to model such an operation as 'structuring' (or as we refer to it – *framing*) using more formal means instead of referring solely to a designer's 'intuition' and 'insight'?

The investigation of problem interpretation and its place in a design process is important for improving our understanding of this activity. Empirically, it is known that designers are rarely given a detailed specification of the problem [4]. In line with the empirical observations, we argue that a specification is subject to the same evolution as a design solution. Moreover, a set of statements expressing a desired state may be proclaimed '*a problem specification*' only at the end of design; i.e. once a designer *accepts* the proposed artefact as a design solution. The idea of co-evolving design solution and specification is not a new one [5]. However, the empirical findings lack a formal theoretical foundation.

Framing as a distinctive design phase appears in works of Schön [4], who observed that practitioners know how they may achieve their goals, and shape (*frame*) the design situation to reflect this experiential and tacit knowledge. In the next sections, we look at several interesting patterns of the operation of 'framing'. This paper focuses on an abstract, conceptual level that is illustrated by more specific, operational models.

2 BACKGROUND

This paper discusses a sequence of decisions a designer performed in one of the experimental sessions we conducted. Our observations show the presence of two complementary knowledge-level actions. First, a specification or interpretation of a design problem in selected explicit conceptual terms. Second, a satisfaction of such an explicit specification by a design solution. In certain cases, the proposed solution was not acceptable and had to be replaced by an alternative. However, this 'replacement' was accompanied by a *problem re-interpretation* to account for the issues the designer became aware of only after that initial attempt to generate a solution.

Our experimental background consisted of 24 experiments with two design practitioners. They were solving tasks from a domain of controllers for large-scale systems. We illustrate our findings on a conceptual design of control strategies for a paper-smoothing plant and an active shock absorber. Initial findings from the experiment and the information on experimental tools appeared also in [6, 7].

The designer's task was to design a structure and control strategy for a plant that takes raw wrinkled paper as an input, and delivers a smooth paper with an even thickness as the end result. In a design process, we wanted to observe the reflective behaviour and problem re-formulation trying to complete a rather vaguely specified problem. We wanted to confirm our hypothesis that behind an 'insightful interpretation and refinement' of a problem specification there are certain re-occurring decision *patterns*. Milestones of the investigated experiment are listed below, together with the illustrative sketches of an assembly, as scanned from a designer's notebook. We shall refer to some of the listed reasoning steps in the next sections:

1. an initial principle for smoothing featured a pair of rolling drums with a paper passing through a gap between them;
2. this layout was enhanced, when a designer joined several pairs into a linear sequence to achieve better quality;
3. next, he proposed dampening the raw paper before entering the rolling drums, and drying it afterwards to achieve more acceptable performance (see Figure 1)
   → *introduction of an additional requirement refining the current conceptual frame restores the acceptability*;

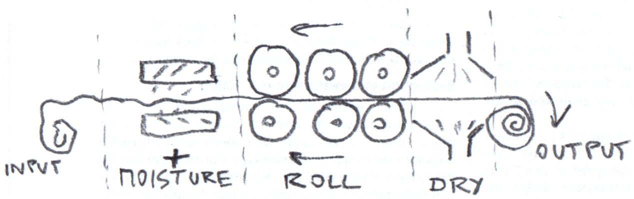

Figure 1. Linear layout of drums, pre- and post-processing units

4. another reflective turn occurred when designer found out that smoothing depended on the pressure of drums, and that it might damage certain types of paper; an alternative was to reduce the pressure and increase the size of the plant
   → *contradictory requirements are spotted and attended*;
5. to address the contradictory interpretations of a problem, various layouts of the drums were considered (linear vs. alternate) – see modifications in Figure 2
   → *re-interpretation of a conceptual term in the current design frame leads to an alternative solution*;

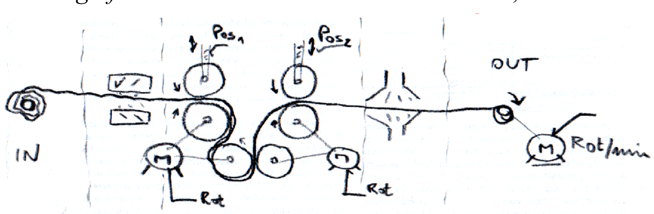

Figure 2. 'Zigzag' layout = better quality, smaller dimensions

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

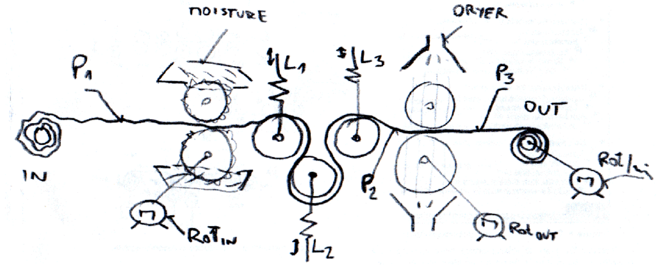

Figure 3. Design solution in a re-interpreted design frame

In our opinion, the observed modifications cannot be attributed purely to the search for the 'right' solution. The milestones described above feature a process of *exploration* [2] that in addition to the solution construction involves also a structuring of the design space and the *interpretation* of a design problem. This exploration may show many aspects of intuitive and tacit reasoning but some patterns are explicable in terms of evolving conceptual frames and solution acceptability. Let us detail these arguments below.

3 PROBLEM INTERPRETATION AND FRAMING

Let us begin with a definition of a related term known in cognitive science – a 'design perspective'. *A perspective is a point of view, which implies that certain design goals exist, certain bodies of design knowledge are relevant, and certain solution forms are preferred* [8]. The term 'design perspective' is used for expressing designer's intentions. We may understand a design perspective as a kind of vocabulary of concepts a designer decides to use in the problem solving phase of design.

We agree with the empirical findings that framing is an important operation that precedes the problem solving, and complements it [4, 9]. However, what are the implications of 'framing' on the knowledge level? What is actually happening with a designer's knowledge during the problem framing? Can a 'woolly' operation of problem (re-)interpretation (*framing*) be expressed in a formal language? These are the challenges tackled in the rest of this document.

3.1 Definition of basic concepts

It seems redundant to say that designers do design; i.e. they tackle design problems. Is there really a redundancy? We do not believe so. Let us denote the problem a designer tackles as $\mathfrak{DP}$. The design problem $\mathfrak{DP}$ is usually too vague to be tackled directly, and must be *interpreted* before any attempt on solution. Therefore, a designer associates this design problem $\mathfrak{DP}$ with an explicit specification $\boldsymbol{S}$, and tackles the problem *specified* (read 'interpreted') in this way. In other words, he or she tries to articulate a solution *satisfying* this explicit problem specification. However, a solution satisfying $\boldsymbol{S}$ must not necessarily be a solution to the design problem $\mathfrak{DP}$.

In terms of logical theories, this specification is a kind of *circumscription* [10] declaring that only the statements from the explicit specification $\boldsymbol{S}$ are needed for interpreting (and later solving) the problem. Relation defined in (1) expresses an assertion of such a circumscriptive reasoning step.

$$\exists\, S \subseteq \mathcal{S}^{*}\!: specifies_{\Phi}\,(\, S, \mathfrak{DP}\,) \qquad \textbf{(1)}$$

However, such an assertion can be only made within certain conceptual boundaries – a *conceptual design frame*. 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 $\mathcal{T}^{*}$ and $\mathcal{S}^{*}$ for a formal notation of these circumscribed knowledge spaces.

Space $\mathcal{T}^{*}$ is 'a closure' constituted by the selected conceptualisation $\mathcal{T}$ and an appropriate domain theory ***DT***. Domain theory ***DT*** is a generic, problem-independent knowledge, possibly applicable to the different problems. For example, physics is a generic domain theory applicable to a design of an elevator as well as a spacecraft. A generic domain theory ***DT*** is 'instantiated' for a particular conceptual base $\mathcal{T}$, thus creating a usable theory for solving a particular problem. Let us therefore, refer to closure $\mathcal{T}^{*}$ as a ***problem solving theory***.

Similarly, space $\mathcal{S}^{*}$ is a hypothetical instantiation of the generic problem specification statements ($\mathcal{S}$) that can be articulated for the chosen conceptualisation of the problem $\mathcal{T}$ and the selected domain theory ***DT***. The knowledge spaces $\mathcal{T}^{*}$ and $\mathcal{S}^{*}$ may be defined as closures and instantiations of finite sets of conceptual primitives but they may not be enumerated in explicit terms or completely formalised!

A design frame $\boldsymbol{\Phi}$ is then a pair $\langle \mathcal{T}, \mathcal{S} \rangle$. Let us detail the terms used in the presented definition:

a) Conceptualisation $\mathcal{T}$ is a vocabulary of basic concepts, for which a designer decides they are available for expressing statements about a particular design problem. A conceptual base may include a terminology defining functional and structural objects, and/or behavioural mappings between them [11].

b) An applicable domain theory ***DT*** is a shared, generic vocabulary defining the background [12], on which any conceptualisation is applied. An abstract domain theory is typically interpreted for a specific conceptual base to derive a useful problem solving theory.

c) A space of conceptually relevant problem specifications $\mathcal{S}^{*}$ complements the problem solving theories $\mathcal{T}^{*}$. It provides a vocabulary for expressing the desires or intentions of a designer in a particular problem [8]. It is a set of relations articulated using the elements of a particular problem solving theory ($\mathcal{T}^{*}$).

Design frames, as defined above, do not exist objectively! They are highly volatile, and are constructed on the fly from the information about a particular design problem that is available to a designer. Typically, using a customer's initial design brief a designer identifies similar design situations he or she is familiar with. In these familiar terms, he or she then expresses the desired and intended states for the particular problem $\mathfrak{DP}$; i.e. the problem $\mathfrak{DP}$ is explicitly specified as $\boldsymbol{S} \subseteq \mathcal{S}^{*}$.

After the initial specification, the conceptual design continues with an attempt to formulate a minimal sub-set $\boldsymbol{T} \subseteq \mathcal{T}^{*}$ that *satisfies* a 'given' problem specification. ($\boldsymbol{S} \subseteq \mathcal{S}^{*}$). This explicit problem specification thus includes all such statements that express the desires about a design problem $\mathfrak{DP}$, to which a designer made an explicit commitment. This 'satisfaction' of an explicit specification can be seen as shrinking of the vast space of a problem solving theory $\mathcal{T}^{*}$ into a chunk that can be manipulated with. This manageable chunk corresponds to a 'solution model' [13, 14]. Due to its generative nature, it seems better to call this chunk a 'problem solving model'.

Formally, a ***problem solving model*** is a minimal sub-set of the problem solving theory that sufficiently *satisfies* the explicit problem specification. Relation '*satisfies*' is a binary association of a problem-solving model ***T*** and the current explicit problem specification ***S***; the conceptual frame $\boldsymbol{\Phi}$ serves as a circumscribing contextual parameter. See also formula in (2).

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

From an operational point, one may distinguish design requirements ***R*** from constraints ***C***, and assert that a problem specification is a union of the two – i.e. $\boldsymbol{S} = \boldsymbol{R} \cup \boldsymbol{C}$ [12]. Requirements are those statements demanding the explicit presence of a particular feature, whereas constraints are conditions that must not be explicitly violated by any candidate solution. The conceptual duality of requirements and constraints is discussed e.g. in [6, 12]. In this paper, we propose only one possible definition of relation '*satisfies*' – see (3). Other conceptual backgrounds may lead to other operational definitions.

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

Symbols used in (3) have their usual meanings [15]; '$\models$' stands for a semantic entailment, '$\vdash$' is a proof-logical implication, and '$\bot$' is a contradiction. Accordingly, sub-theory ***T*** is a problem solving model satisfying an explicit specification ***S***, if it is <u>complete</u> in respect to the required features ($\forall r \in R\!: T \models r$), and <u>admissible</u> in respect to the constraining conditions ($\neg\exists c \in C\!: T, c \vdash \bot$). Any 'satisfactory' solution must have a potential to deliver all required features without contradicting any of the explicit constraints.

However, as we defined it, the explicit problem specification ***S*** is only an interpretation of a design problem $\mathfrak{DP}$ used for problem solving. It is only 'a model' of the actual problem $\mathfrak{DP}$. Consequently, the existence of a model ***T***, for which relation '*satisfies*' holds, is a <u>necessary but not sufficient</u> condition of declaring ***T*** a 'design solution'! In addition to a satisfaction of the explicit specification, the candidate solution ***T*** must be also '*acceptable*' as a design solution for the problem $\mathfrak{DP}$. The problem is that relation '$acceptable_{\mathfrak{DP}}(T)$' often cannot be defined in the explicit terms of the languages of $\mathcal{S}$ or $\mathcal{T}$ and in advance. It is appreciated subjectively and tacitly.

Nevertheless, it may be defined as a residual category; see also formula (4). What does it mean that a relation is residual? We argue it corresponds to the position in [16] claiming that certain tacit decisions cannot be stripped of their contextual background. It may be difficult to define exact conditions of 'acceptability', but one may proclaim a certain problem solving model acceptable or not, when s/he sees it. Note the change of a parameter representing the contextual dependency in (4):

$$\mathit{satisfies}_{\Phi}\,(T, S) \wedge \neg \mathit{acceptable}_{\mathfrak{DP}}\,(T) \Rightarrow \neg \mathit{specifies}_{\Phi}\,(S, \mathscr{DP}) \quad \textbf{(4)}$$

A corollary of formula (4) asserts that if an otherwise valid problem solving model is not accepted by a designer as a design solution, it may point to an incorrect (i.e. incomplete) interpretation of the actual design problem $\mathfrak{DP}$. The explicit interpretation in terms of statements $\boldsymbol{S}$ does not reflect the real design problem $\mathfrak{DP}$, and may be subject to amendment. Such an amendment may feature a refinement of the incomplete conceptual frame or complete *re-framing* of a design problem. These 're-interpretations' are dealt with next.

3.2 Sequence of conceptual decisions in design

In the following paragraphs, we introduce a recursive model of framing in design using the terminology of conceptual frames. The model is defined as a sequence of decisions driven by the returned values of predicates '*satisfies*' and '*specifies*', as introduced in section 3.1. The sequence is running across several mutually dependent, conceptual levels that are numbered from 0 to 5. The model clearly shows the interplay of two types of knowledge-level actions represented by predicates '*specifies*' and '*satisfies*'. The former is trying to complete a problem specification, whereas the latter attempts to devise an artefact satisfying such a specification.

We define a few auxiliary predicates for a conceptual representation of the earlier-mentioned re-interpretations, in order to read the model intuitively as a sequence of decisions followed by actions. The simplest form of problem re-interpretation is *frame refinement* depicted in Figure 4 as a decision on level 3. It attempts 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 chosen conceptual language $\mathfrak{T}$), design continues with a refined (i.e. more complete) specification $\boldsymbol{S'}$. This extension or refinement occurs *within* a known conceptual frame $\boldsymbol{\Phi}$, and is defined in (5).

$$\begin{gathered}\mathit{can\text{-}refine\text{-}frame}_{\Phi}\,(S) \Leftrightarrow \\ \exists s \in \mathscr{S}^*,\, S \subseteq \mathscr{S}^*\!:\, S' = S \cup \{s\} \;\wedge\; \mathit{specifies}_{\Phi}\,(S', \mathscr{DP})\end{gathered} \quad \textbf{(5)}$$

The auxiliary definition (6) shows a more complex decision that tries to re-interpret the explicit problem specification in a new frame. Thus, there is still a refinement of the current problem specification; however, the new statements are introduced from a new design frame rather than the existing one. Schema (6) differs from (5) in the fact that a re-interpretation can occur only when changing the conceptual frame (i.e. the currently available conceptual categories from set $\mathfrak{S}$). This re-interpretation technique relies on a decision, whether a 'tacit' non-acceptance may be resolved by committing to the terms borrowed from a new design frame (new view on the problem).

$$\begin{gathered}\mathit{can\text{-}reframe\text{-}spec}\,(S, \Phi) \Leftrightarrow \\ \exists \Phi_{NEW} = \langle \mathscr{T}, \mathscr{S}_{NEW} \rangle\!:\, S' \subseteq \mathscr{S}^*_{NEW} \wedge\; \mathit{specifies}_{\Phi NEW}\,(S', \mathscr{DP})\end{gathered} \quad \textbf{(6)}$$

The most radical form of re-interpretation is defined in (7). In this particular case, the conceptual foundation of the current frame $\boldsymbol{\Phi}$ (i.e. $\mathfrak{T}$, the set of explicitly selected conceptual objects) is not acceptable for problem solving. It is not possible to find an admissible solution using the current interpretation ($\mathfrak{T}$), and therefore it must be modified. A new frame ($\boldsymbol{\Phi_{NEW}}$) is articulated so that the current explicit specification (i.e. $\boldsymbol{S} \subseteq \mathfrak{S}^*_{NEW}$) becomes a consistent and admissible specification of the design problem (albeit in a modified design frame, modified context, modified conceptual vocabulary).

$$\begin{gathered}\mathit{can\text{-}reframe\text{-}concept}\,(\Phi) \Leftrightarrow \\ \exists \Phi_{NEW} = \langle \mathscr{T}_{NEW}, \mathscr{S} \rangle\!:\, T' \subseteq \mathscr{T}^*_{NEW} \wedge S \subseteq \mathscr{S}^*_{NEW} \wedge \mathit{specifies}_{\Phi NEW}\,(S, \mathscr{DP})\end{gathered} \quad \textbf{(7)}$$

Figure 4 and the presented theoretical model exhibit empirically observed interplay of knowledge-level actions. The interaction of two complementary knowledge spaces is observable as an exchange of information and control during a design process. The model of framing presented below obeys a few simple rules (or modelling assumptions):

1) By 'requirements' and 'constraints', we mean hard, strict demands that *must not be relaxed*.
2) Sentence "*Prove that* **λ** *holds*" shows a *recursive step* that always goes back to level 1. It is an attempt to address an outstanding issue by amending available knowledge spaces. It is 'an order' to an agent to "*evaluate predicate* **λ** *with the new arguments provided.*"
3) Monotonic *refinement* of a problem specification only 'fine tunes' the existing problem solving model ($\boldsymbol{T}$), and

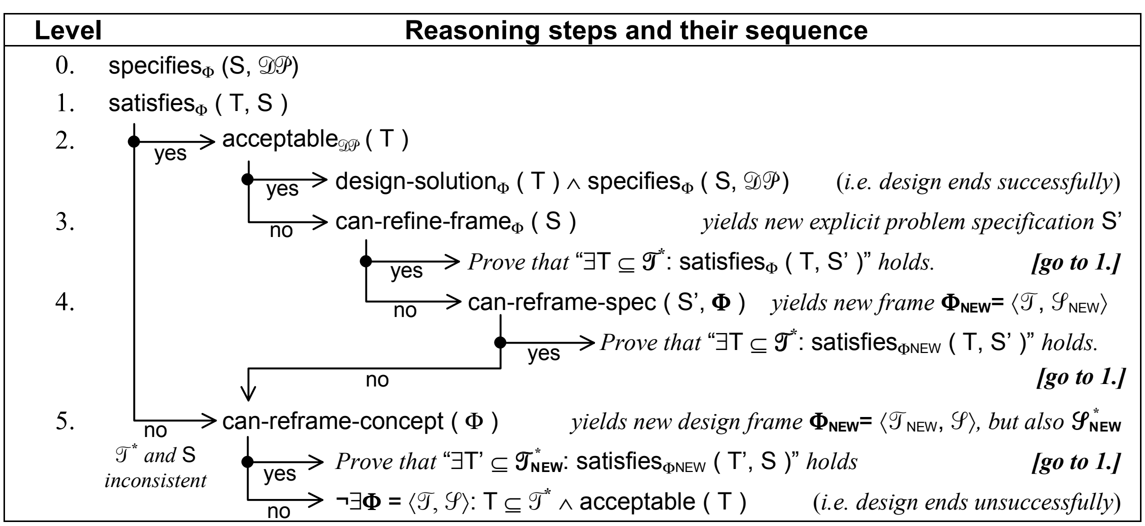

Figure 4. Modelling design process as interaction of conceptual framing and problem solving

reduces the number of derivable alternatives. It is affordable only if an admissible candidate solution already exists for the current conceptual frame.

Let us look at the conceptual patterns of re-interpretation in the context of our experiment. The patterns discussed in section 4 are implied corollaries of the proposed model of framing. Note that these are *conceptual schemas*, not *problem solving methods*. They are abstract models of certain types of reasoning that may be observed in design, and we clarify them using the experiment introduced in section 2.

4 FRAME RE-INTERPRETATION SCHEMAS

Each schema can be represented as a branch leading to a 'leaf' in the decision sequence depicted in Figure 4. The conceptually most straightforward branch leads directly to a design solution; thus generating a problem solving model that is both admissible and acceptable with no need for re-interpretations. A set of problem solving techniques, which may operationally implement the relation '*satisfies (T, S)*' was discussed in an earlier paper [6]. Therefore, we will skip this scenario, and focus our attention on the re-interpretative patterns.

Figure 4 contains three such techniques for a problem re-interpretation, which can be combined in order to achieve more robust reasoning. First features a monotonic refinement of the problem specification within a particular frame. In (5), it is formally defined as relation '*can-refine-frame*'. The second schema shifts a conceptual frame and extends the problem specification in a new frame; it corresponds to formula '*can-reframe-spec*' in (6). Conceptually most disruptive is an amendment of a conceptual base of the current frame articulating new conceptual primitives to work (and think) with. It is defined as relation '*can-reframe-concept*' and appears in (7).

4.1 Uncovering implicit expectations

Consider the following situation observed in a design of paper-smoothing plant as introduced in section 2. It is verbally described in milestones 2 and 3 in the list, and it corresponds to the specification *refinement within a frame*. As we mentioned, at a certain stage, a designer considered pairs of rolling drums[1] that applied pressure on the raw paper, reduced its thickness and smoothed its surface. When exploring the consequences of this trivial principle, the designer observed that the effectiveness of smoothing depended on the pressure and 'active' surface of the drums. The higher the pressure (bigger active surface), the better quality was expected. However, paper was a relatively fragile material regarding the tension, and certain types of paper could easily tear, if certain limits were exceeded.

The designer did not *accept* this candidate solution, despite its admissibility in respect to the explicitly desired features. The paper was smoothed and thickness was reduced, exactly as desired in the explicit specification $\boldsymbol{S}$. However, a designer articulated an additional condition that was not mentioned in the initial problem specification. In addition to the existing statements, he also demanded that paper remained whole (i.e. not torn or damaged). In a justification to this introduction of new knowledge, he maintained that it was "*so obvious that it was rarely emphasised explicitly*".

Let us look at this apparently straightforward situation from a perspective of conceptual frames. We see that the conceptual base $\mathfrak{T}$ for interpreting a problem remained unchanged. The newly added condition only monotonically refined the explicit specification $\boldsymbol{S}$. However, this monotonic refinement strongly influenced otherwise non-monotonic problem solving theory $\mathfrak{T}^*$, and the admissibility of the current candidate solution ($\boldsymbol{T}$). The articulation of a new condition rendered the 'old' problem solving model inconsistent – the new condition demanding undamaged paper could be sometimes violated by the 'old' candidate solution. In this case, the inconsistency was rectified by an introduction of pre- and post-processing units to the plant (shown in Figure 1 as 'moisture' and 'dry'). These units softened paper before rolling, so that lower pressure was needed, paper was shaped easier, and the danger of tearing eliminated.

Let us focus on the articulation, however. From a knowledge-level viewpoint, the new condition can be seen as an attempt to *align* an explicit conceptual frame with the implicit expectations. Such expectations usually remain 'hidden'. These expectations may not have been expressed originally, but they tacitly influenced a designer's decision on the solution acceptability. Reflecting on the 'hidden' (perhaps empirical) expectations brought to the designer's attention an overlooked or forgotten feature, and led to an explicit articulation of a new statement addressing this 'forgetfulness'. The 'aligned condition' may act as a falling domino brick that triggers further conceptual amendments (see also section 4.3).

4.2 Specification refinement with re-framing

The situation in section 4.1 required only minor clarifications of the existing problem specification. It was modelled with the same conceptual primitives; only a specific commitment emerged to an 'untold' expectation. However, we also observed a more complex type of a conceptual 'turnaround'. As visible in Figure 4, this pattern typically follows the schema described in the previous section. While the previous schema can be verbally expressed as "*I have forgotten to say that λ must have value x*", this pattern had slightly different reading. It was justified in a more 'tentative language' conjecturing a possible cause of the solution unacceptability. Its verbal description often goes along the following line. '*I can't tackle the problem specified in this way. <u>What if</u> I looked at the problem from another end and assumed λ?*'

As an illustration, consider an episode from a design of an active shock absorber for a vehicle. The initial frame of this problem was built on the concept of 'spring'. The designer became soon aware of the toughness of an absorber that needed to be controlled and modified. His interpretation of the 'active absorption' required a simple and fast modification to the toughness. However, with a spring, this could only be achieved by changing its material or structural properties (such as diameter or length). Therefore, a pneumatic piston-based absorber was introduced as a functional alternative (see Figure 5B).

Simultaneously, a set of rules adjusting the pressure in response to the measured disturbances was defined. The essence of the control strategy was in softening the absorber when the

[1] In line with Figure 4, this would be $\boldsymbol{T} \subseteq \mathfrak{T}^*$ – an admissible solution.

disturbance was measured. When the designer recalled the original frame from which the piston was introduced, he was not satisfied with this strategy. He demanded a 'more active' behaviour of the absorber; especially when he introduced the concept of a reversible action/control. Using his knowledge of pneumatic devices and their behaviour, he realised that not only a rough road may activate the absorbers but also the smooth surface. If a car with soft suspension enters a smooth surface, it may start vibrating and his current control strategy would amplify the vibration rather than eliminate it.

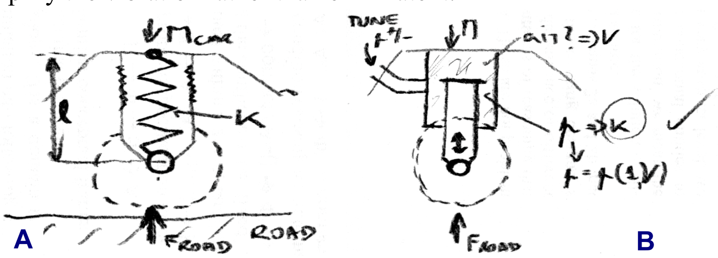

Figure 5. Sketches of a spring (A) and pneumatic piston (B)

Therefore, the designer had to clarify his interpretation of the concept 'active suspension', and with that goal he borrowed a vocabulary from other previously encountered design frames. Namely, he conjectured that a simple 'prediction' based on the 'historical' measurements and 'trends' might help to address the new needs. However, these conjectured requirements were not expressible in the original frame. They required the designer to allow for 'historical' data to be used for a prediction. Furthermore, the designer had to introduce *hypotheses* and *assumptions*. Although he was still working in the same domain of pneumatic devices, for the explicit problem specification, he borrowed a few terms from different domains.

Using such a tentative and speculative language he was able to move beyond the restricted perspective of the 'old' frame, and re-interpret the requirement of 'active suspension'. In order to modify his specification he had to bring in a different conceptual frame, a new vocabulary. The terms 'trends' or 'prediction' were not forgotten as it was the case in section 4.1. They might stem from some tacit expectations but to make them explicit the designer needed a new specification of the problem – partly, in a new conceptual frame.

This refinement of a problem specification with re-framing is clearly a *non-monotonous* modification. The tentative assumptions are part of the explicit problem specification until they are proved incorrect, then they may be easily retracted. The tentative refinements serve as a kind of *exploratory probe* into the designer's tacit understanding of the design situation. If a solution can be found for such a conjectured specification, the probe is confirmed and upheld. In other words, a new conceptual frame is vaguely recognised as potentially useful *before* any refinement to the specification is explicitly articulated.

The schema in section 4.1 worked solely with strict requirements and constraints. Schema in this section brings in speculative, tentatively assumed conditions. The schema in this section acts as a kind of 'circumscribing condition' that allows a designer to use a particular problem solving theory $\mathcal{T}^*$ consistently. Without such an assumption the proposed candidate solution may work in some cases and fail in others.

Described pattern enables the designer to focus on certain aspects of a design problem while disregarding the other. It makes it possible to test the available design space. Assumptions and speculations in the problem specification would typically narrow down also the solution space. In the same time, these speculations are challenges for a designer. Can design be concluded without these auxiliary beliefs? What do I need to make this speculation 'natural'? This was also the case in the described experiment. Eventually, the designer proposed a merger of several contexts that he generated during his exploration of the term 'active suspension'. However, this is a consequence of working with assumptions rather than a reason.

4.3 Conceptual re-framing in contradictory theory

In the previous two sections, we looked at how a problem interpretation may change in the space of problem specifications. Consider another type of re-interpretation that was observed in the design of paper-smoothing plant and other experiments. The scenario in this section addresses milestones 3 and 4 from the list in section 2. Suppose Figure 1 depicts a candidate solution at a certain stage. This solution satisfied the explicit requirements, and complied with the designer's experience from similar cases (e.g. metal sheet rolling). However, a conflict emerged, when the designer looked at the efficiency and controllability of the overall process. As stated in section 4.1, higher pressure or larger active surface of rolling were the remedies to a low quality of paper. The increases in pressure were tackled earlier, and it was resolved to add additional processing steps to soften the paper, rather than increase the pressure.

Adding more pairs of rolling drums to the sequence would increase the active surface of rolling. However, the plant could not grow forever, because it would pose difficulties for control and maintenance. It was clear that designing an assembly with fewer drums was desirable in order to keep the controllers simple. However, fewer drums negatively influenced quality, and increased danger of damaging paper. Thus, the designer entered a 'magical circle' of mutually contradicting requirements.

He resolved a contradiction by shifting the conceptual foundation for a problem interpretation. When it was impractical to satisfy the constraints in one dimension, he brought in another one. Instead of arranging the rolling components in 'one dimension' (i.e. linearly laid-out drums), he articulated a concept of '*two-dimensional layout*'. The origins of this new conceptual entity come from a slightly cumbersome requirement in the language of the existing frame of '*two-dimensional squeezing*'. Thanks to problem re-framing, he introduced an alternate (zig-zag) layout of drums that exhibited a larger effective surface of rolling. Subsequently, fewer drums were needed, and the size- as well as pressure-related constraints could be met simultaneously. Thus, the conditions that were mutually contradictory in one conceptual frame became harmless and consistent in a re-interpreted frame. New concepts can be clearly seen in a re-designed plant (see Figure 2).

A similar reasoning pattern of articulating new concepts to tackle an outstanding issue was repeated in milestone 5 in section 2. In this step, the designer re-framed the problem more radically. He re-visited his original interpretation of rolling as a basic principle for smoothing. Instead focusing on the application of pressure during smoothing, he became aware that in a zigzag layout, one drum from each pair was using larger surface than the other one. Hence, he considered removing a 'redundant' drum from each rolling pair and replacing the principle of pressing with a principle of abrasion. The components of the plant were similar, but their conceptual roles and functionality were re-interpreted in the light of a new 'insight'. Eventually the changes led to a design shown in Figure 3.

Unlike in sections 4.1 and 4.2, where only a problem specification was refined and extended, this operation went conceptually much deeper. It all began with a contradictory problem solving theory $\mathcal{T}^*$, in which certain pairs of constraints were violated (i.e. $\exists c \in C: \mathcal{T}^*, c \vdash \perp$). In line with our initial assertions, none of the violated conditions could be 'retracted'. The designer was forced to re-visit the underlying domain theory, as interpreted in conceptual terms $\mathcal{T}$. Having defined new conceptual primitives (e.g. '2D layout' or 'abrasion'), he actually changed his conceptual vocabulary for interpreting and solving design problem $\mathfrak{DP}$.

Is it possible that a domain theory as a generic problem-independent source of knowledge may contain contradictory sentences? Isn't this a violation of its generic character? The answer is both – yes and no. It is true that domain theory is generic. However, as we stated at the beginning, in order to use it, it must be *interpreted* in the light of certain conceptual commitments. In other words, the conceptual instantiation of a consistent generic domain theory can become contradictory when certain explicit conceptual commitments are made.

It is the combination of the explicit commitments to particular concepts that 'activates' mutually incompatible instances of a sound, generic domain theory. The idea of 'domain contradictions' has already appeared in the literature drawing on the extensive empirical studies of design and inventive problem solving [13]. Therefore, the presented model has a sound practical base, which we theoretically extended by relating the conflicts to a relativistic 'design frame'.

A new conceptual base for interpreting the problem ($\mathcal{T}_{NEW}$) triggers articulation of a new conceptual frame $\boldsymbol{\Phi}_{NEW}$. In the context of new frame, the conflicting constraints 'lose their edge'. A problem solving theory with contradictions can be interpreted differently, thus regaining its consistency. Design process may continue – at least, until another, explicated 'hidden' expectation in one of the next steps invalidates the current perspective. This schema for rectifying contradictions gives a theoretical, knowledge-level background to the mentioned, *empirically* observed resolution of physical contradictions in the inventive problems as reported by Altshuller [13].

5 OPERATIONALISATION OF THE MODEL

A computational model implementing the relation '*satisfies*' was presented in [6]. In principle, it used techniques based on abductions and deductions from an explicit problem solving theory. The main assumption of this model was that at any given time, the problem specification is 'frozen', and a solution is sought to satisfy it. Abduction works in the incomplete domains, and is a form of exploratory and speculative reasoning [17]. Therefore, it is prone to arrive at different alternative conclusions under varying circumstances.

Deduction on the other hand, is a sound method of reasoning. However, if we allow assumptions ('tentative axioms') be part of a problem solving theory, the soundness may disappear. A theory with assumptions is non-monotonic and deductions in such a theory depend on a context [18]. A context is equivalent to our term 'design frame', and is based on the explicitly articulated problem specification. Let us look at how we can model the actual operation of 'framing' and the selection of 'the right' context.

5.1 Design frames and similarity of problems

Several researchers have emphasised the ability of the designers to 'see and tackle the new problems as the previous ones' [4, 5]. Uncertain design situation are often shaped so that they reflect the past experiences. Reasoning by analogy drawing on such experiences, is a mechanism complementing logical theories (whether monotonic or not). It attempts to re-use the familiar conceptual vocabularies from the past problems appreciating certain explicit or 'tacit' similarity of two cases.

Articulating a design frame means committing oneself to a particular, usually familiar conceptual vocabulary. This familiar vocabulary is a referential framework in which the designer interprets and solves the vaguely defined problem. Shift of a design frame or *problem re-framing* corresponds to an introduction of new, less common concepts that may trigger the designer's awareness of the unattended features or behaviours. According to [19], it is possible to distinguish several types of similarity between two design situations. Let us look at two of these types suitable for design.

First, a *literal similarity* looks at the low-level, often structural objects and direct relations among them. It typically accounts for 'in-domain' similarities, when objects 'looking alike may as well act alike' [19]. Second, *abstracted analogy* maps the higher-level relations among the objects in the similar design situations (e.g. functions or behaviours), without paying much attention to the low-level details. Often, this type is observable in 'cross-domain' similarities.

Literal similarity seems to be a suitable method of reasoning at an advanced stage of a design. It may help discriminate between different functional alternatives that may be abduced, or choose a suitable context for the deduction [6]. However, it does not seem to suit the ill-structured conceptual phase very well. On the contrary, abstracted analogy seems to be destined for the conceptual, ill-structured phase of design.

5.2 Abstracted analogy in design

If things act similarly in certain aspects, they may have also other commonalties, both in structure and functionality. This verbal description of an abstracted analogy shows its potential for the conceptual design. In the conceptual design, the available knowledge relates to the desired functions ('how things should act'), and possibly the abstract models ('prototypes'). Abstracted analogy may discover certain matches between the required functionality in the current, ill-structured problem (so called *target*), and the functionality of a well-defined past case (so called *base*). Having found a base case that 'acts similarly' as desired for the target one, the designer may re-use the well-structured knowledge of the base case in the target case.

We shall not repeat the extensive work on case-based design [20] or design by analogy [21, 22]. Instead, we look at how the analogy may help in articulating the sets and relations that were abstractly defined in sections 3 and 4. Let us assume that one relation among concepts in the domain theory we use is *typology*, in which one concept is *a type of* another, more generic term; e.g. smoothness is a special type of a structural property of any particular object *Obj*:

type-of (property (Obj, 'smooth), property (Obj, 'structural))

Now suppose, there is a design problem $\mathfrak{D}\mathfrak{P}_B$ that a designer tackled in the past; this may serve as a base case. This problem $\mathfrak{D}\mathfrak{P}_B$ is *specified* by a set of statements $\boldsymbol{\Theta}$ that explicitly and sufficiently specifies that case. Let us denote the current (ill-structured target) problem as $\mathfrak{D}\mathfrak{P}_T$. Assume that from the design brief a designer articulated an incomplete set of requirements $\boldsymbol{R} \subseteq \boldsymbol{S}$ to specify this target case. The specifications $\boldsymbol{R}$ and $\boldsymbol{\Theta}$ may be similar if they share the same statements. Since such a direct match is rare, let us define a 'remoter' similarity:

$$abstracted(\gamma, \delta) \Leftrightarrow \\ (\exists \delta\text{: } type\text{-}of(\gamma, \delta)) \vee (\exists \delta, X\text{: } type\text{-}of(\gamma, X) \wedge abstracted(X, \delta)) \quad \textbf{(8)}$$

In line with (8), statement $\boldsymbol{\gamma}$ is abstractly similar to statement $\boldsymbol{\delta}$, iff it is a direct descendant of $\boldsymbol{\delta}$ in the domain typology or a 'recursive descendant' of $\boldsymbol{\delta}$ in the same typology. Once we can appreciate abstracted similarity of two statements, we can define abstracted analogy of two problem specifications:

$$analogous(r, \varphi) \Leftrightarrow \\ specifies(\Theta, \mathfrak{D}\mathfrak{P}_B) \wedge (\varphi \subseteq \Theta) \wedge specifies(S, \mathfrak{D}\mathfrak{P}_T) \wedge (r \subseteq R) \wedge \\ \wedge \{ abstracted(r, \varphi) \vee instance\text{-}of(r, \varphi) \vee \\ \vee (\exists M\text{: } abstracted(r, M) \wedge analogous(M, \varphi)) \} \quad \textbf{(9)}$$

The first condition in (9); i.e. '*abstracted (r, φ)*', says that the requirement $\boldsymbol{r} \in \boldsymbol{R}$ is similar to a requirement $\boldsymbol{\varphi}$ from a base case, if it is an abstraction of $\boldsymbol{\varphi}$. For instance, 'rolling' is a specific type of 'motion'. The second condition; i.e. '*instance-of (r, φ)*', covers the situation, when the required function $\boldsymbol{r} \in \boldsymbol{R}$ is not a hierarchical type of a particular functionality $\boldsymbol{\varphi}$, but rather it is an instance of such a 'prototypic feature or functionality'. An example of such a relation can be a requirement for 'rolling a paper', as an instantiated form of more generic statement of 'rolling a material'.

The last condition '*abstracted (r, M) ∧ analogous (M, φ)*' is the most complex appreciation of similarity between design requirements. It recursively combines the previous conditions, and relates the required and the base concepts through multiple nested levels of 'type-of' and 'instance-of' relations. In our opinion, a formula like this might have been used in the discovery that specially arranged pairs of rolling drums behave as abrasive rather than rolling surfaces. Accordingly, 'rolling' would be an abstract type of 'abrasion'; see the experimental details in section 2 and the transition from Figure 2 to Figure 3.

Having found at least one analogy among design requirements, it is possible to define an analogy between problem specifications. In formula (10), $\boldsymbol{R} \subseteq \boldsymbol{S}$ stands for the demanded features in the target problem, and $\boldsymbol{\Theta}$ is an explicit specification of the base case. Problem specifications $\boldsymbol{S}$ and $\boldsymbol{\Theta}$ are analogous, if the set of identified analogues $\boldsymbol{AN}_{\boldsymbol{R \approx \Theta}}$ is not empty (11). If problem specifications coincide according to condition (11), the respective design cases (base $\mathfrak{D}\mathfrak{P}_B$ and target $\mathfrak{D}\mathfrak{P}_T$) can also be considered abstractly analogous (12).

$$AN_{R \approx \Theta} = \{ \langle r, \varphi \rangle : r \subseteq R \wedge \varphi \subseteq \Theta \wedge analogous(r, \varphi) \} \quad \textbf{(10)}$$

$$analogous\ (S, \Theta) \Leftrightarrow \\ (\exists r \subseteq R \subseteq S,\ \varphi \subseteq \Theta\text{: } analogous(r, \varphi)) \Leftrightarrow AN_{R \approx \Theta} \neq \varnothing \quad \textbf{(11)}$$

$$analogous\ (\mathfrak{D}\mathfrak{P}_T, \mathfrak{D}\mathfrak{P}_B) \Leftrightarrow \\ specifies(S, \mathfrak{D}\mathfrak{P}_T) \wedge specifies(\Theta, \mathfrak{D}\mathfrak{P}_B) \wedge analogous(S, \Theta) \quad \textbf{(12)}$$

5.3 Re-using past solutions and specifications

Let us now look briefly at knowledge re-use; i.e. how the concepts from a base case $\mathfrak{D}\mathfrak{P}_B$ can be applied in the construction of a conceptual frame for the target case $\mathfrak{D}\mathfrak{P}_T$. For instance, an explicitly articulated constraint or structural concept in the base case may also be relevant for the target case, thus refining and completing the current explicit specification $\boldsymbol{S}$. In section 3, several different *conceptual entities* were defined theoretically. We shall now put some flesh around those theoretical bones. In general, it is possible to re-use two types of concepts:

- statements about functions and properties, and
- statements denoting design elements, structural objects

Intuitively, the first category aims to refine the current explicit specification $\boldsymbol{S_T}$ and the space of applicable specifications $\mathcal{T}_T^*$ of the problem $\mathfrak{D}\mathfrak{P}_T$. On the other hand, the second type of re-use may help to articulate the conceptual set $\mathcal{T}_T$ and deciding on an applicable domain theory $\boldsymbol{DT_T}$, and eventually $\mathcal{T}_T^*$.

First, we look at the past design structures, from which the following knowledge can be extracted from the base case $\mathfrak{D}\mathfrak{P}_B$:

- $\boldsymbol{\pi}$ … primitive concepts from the past solution $\boldsymbol{T_B} \subseteq \mathcal{T}_B^*$,
- $\boldsymbol{\psi}$ … abstract object types with appropriate constraints,
- $\boldsymbol{\rho}$ … relations among the concepts from $\boldsymbol{\pi}$ and $\boldsymbol{\psi}$

If we represent this re-use as a graph, set $\boldsymbol{\rho}$ contains its arcs. Set $\boldsymbol{\pi}$ makes up the nodes of the original problem $\mathfrak{D}\mathfrak{P}_B$, and its members may also appear in a target problem $\mathfrak{D}\mathfrak{P}_T$. The set $\boldsymbol{\psi}$ denotes *regions of relevance*; i.e. conceptual boundaries of the applicable structural or functional elements. This representation of knowledge re-use on different levels of conceptual abstraction is schematically sketched in Figure 6.

Knowledge re-use sequence in the figure shows three types of transfer from the base problem $\mathfrak{D}\mathfrak{P}_B$ (far left). We start with the relations defined among the previously used elements; (e.g. '*connected*', '*before*', '*contracts*', '*expands*', etc.) This may seem of a little informative value but these concepts are often

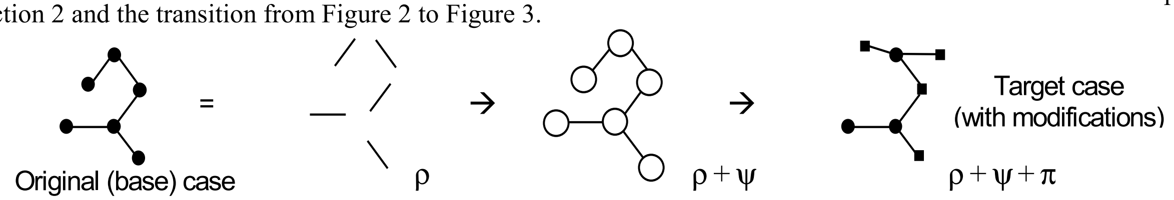

Figure 6. Knowledge transfer on three different levels: relations – object types – primitive elements

essential elements of the conceptualisation set $\mathcal{T}_T$. These concepts imply certain orderings of the elements, and influence also the choice of the domain theory – i.e. knowledge needed to implement these particular relationships.

By abstracting from the previously used structural elements we may acquire knowledge of applicable object types. These refine the context, in which particular relations occur. For instance, the following two 'types' may be articulated for a suspension device. One end of the device is *connected* to '*a movable part*' and another one to a static, '*suspended body*'. These abstract 'types' further frame the target, ill-structured problem $\mathcal{DP}_T$. The following formulae define how it is possible to acquire the conceptual knowledge from the base case $\mathcal{DP}_B$. All sets in (13), (14), and (15) may help in constructing the conceptual set $\mathcal{T}_T$ of a design frame for the target problem $\mathcal{DP}_T$.

$$\pi = \{ s \in T_B \subseteq \mathcal{T}_B^*: \neg\exists x \in \mathcal{T}_B^*: \textit{part-of}_{\mathcal{DP}B}(x, s) \} \quad \textbf{(13)}$$

$$\psi = \{ obj \in \mathcal{T}_B^*: \forall s \in \pi: \textit{abstracted}_{\mathcal{DP}B}(s, obj) \} \quad \textbf{(14)}$$

$$\rho = \{ \varphi \in \mathcal{T}_B^*: \pi, \mathcal{T}_B^* \vdash \varphi \wedge (\exists rel \in S_B: \textit{constraint}(rel(\varphi)) \vee \vee \textit{assumption}(rel(\varphi)) \} \quad \textbf{(15)}$$

In the previous paragraphs we briefly described how the problem conceptualisation $\mathcal{T}_T$ may evolve from the familiar structures and relations borrowed from a past problem $\mathcal{DP}_B$. In a similar manner, a designer may re-use the statements about the previous problem to refine or extend the specification of the target problem $\mathcal{DP}_T$. The primary source of knowledge for the refinement of problem specification S_T is set ρ (15). This set aims to elicit those properties or behaviours that appear in the constraints of the previous problem $\mathcal{DP}_B$.

These 'additional' constraints are important because they are often implicit in the specifications of the problems of a particular class. As we showed in section 4, these implicit constraints and assumptions are typically responsible for the tacit appreciation of the unacceptability of the partial solutions. The relation '$\textit{acceptable}_{\mathcal{DP}T}(T_T)$' was defined only superficially, and we have not mentioned its full engineering meaning. This is rectified now; we argue that the constraints from the past case $\mathcal{DP}_B$ play their role in this 'tacit' assessment of solution acceptability.

What constraints or assumptions can be relevant for the current problem $\mathcal{DP}_T$ is determined by what properties, function, or behaviours they refer to. Thus, we can directly re-use those constraints that refer to the properties that appear in both conceptual sets $\mathcal{T}_B$ and $\mathcal{T}_T$. However, more interesting is to look at the constraints referring to the *abstractly similar* properties, because those may not be easy-to-spot. At the same time, these abstracted constraints and assumptions often reflect the designer's expectations, which may be tacit and inarticulate.

Note that the purpose of this heuristic is to spot potential hazards drawing on the similar conceptual commitments and similar design frames. This speculation about an analogy-based frame construction focuses on the principled exploration of the opportunities in design rather than the actual knowledge re-use. In other words, we looked more closely at using the analogy to answer the question 'What if?' instead of more common 'How to?' [21, 22].

6 RELEVANT RESEARCH AREAS

Research areas relevant to this paper include the knowledge-level modelling of design as an ill-structured and inherently incomplete problem, as well as analogy-based reasoning in design. Let us briefly discuss the difference of this research from the other approaches that appeared in the literature in the following sub-sections.

6.1 Knowledge-modelling relevance

From the perspective of knowledge-level modelling, we formally defined an abstract model of the interpretation of ill-structured design problems. The operation of problem framing is an important extension of the existing models of design tasks because it relates the 'insightful' decisions of a designer to the relativistic concept of a design frame. Since most approaches focus on the problem solving portion of design [2, 10, 14], they tend to assume a 'given' problem specification. We showed that the specification is 'given' only within a particular conceptual frame; when a frame changes, the problem specification and the conceptual basis of the problem solving theory may change as well.

We argued that the problem specification is never 'given'; it is open-ended [13], and subject to the same evolution as a design solution. One of the main purposes of design is then to recognise how the specification may be refined or extended so that it reflects the tackled design problem $\mathcal{DP}$ more closely. From this point of view, we defined a more formal model of the cognitive action of solution talkback similar to that in [4]. We also addressed the issues of tacit reflection, whose aim is to surface the designer's tacit and implicit intentions and expectations about the vaguely defined design problem [8, 9].

6.2 Design engineering relevance

Since our research was more theoretical and focused primarily on the knowledge modelling aspect of problem framing in design, this section presents our vision of how this model and research may evolve. The current model of framing in design is too abstract for a practical deployment; however, we started the research into its 'real-world' implementation. The aim is to devise a support tool that would use a rich base of real designs tackled by the design teams in a testing organisation.

Our research prototype structures the descriptions of the tackled problems in the terminology defined in section 3. Thus, we are able to associate the 'real' problem with the alternative interpretations – design frames. Each frame may make explicit commitments to a different problem specification – this is typically represented by various constraints or circumscriptions of the vague design problem. At this stage, we are already able to cross-reference the cases that exhibit similar patterns on the level of problem framing rather than structural or functional primitives.

Our next goal is to be able to make some prediction in respect to the possibly relevant extensions of the current conceptual frame. Furthermore, we aim to provide more support the construction of a conceptual basis for problem solving. All these methods of support are observed in many empirical studies, however, they are not accounted for in the majority of the existing support tools and methodologies. More information can be found on the URL http://clockwork.open.ac.uk.

7 SUMMARY AND DISCUSSION

The conceptual schemas proposed in this paper are models of the interpretative decisions in design. In this contribution, we focused on the abstract level of relevant conceptual sets and conceptual design frames. We argued that the operation of problem framing underpins a designer's interpretation of an ill-structured and incomplete design problem. Due to limited space, only a few potentially applicable models of the conceptual operation of *re-framing* were discussed.

Why can a logically admissible design be questioned? As stated, one reason might be that the explicit frame and the implicit expectations are not 'synchronised'. These *expectations* play an important role in the real-world design, and we tried to give some formal background to their application in design. In the words of Schön [4], a familiar approach to the ill-structured design situations may produce unintended results. However, the results are 'intended' or 'surprising' only in reference to a particular frame – a particular familiar vocabulary. Thus, any 'surprise' seems to be a manifestation of an inadequate interpretation (*frame*) of a design problem; see formula (4).

Briefly, we discussed also the potential sources of these implicit or tacit expectations, and discussed them in the context of reasoning by analogy. The investigation of the consequences of an explicit commitment to a particular concept or constraint may raise a designer's awareness of the features overlooked in the current approach. Such was the origin of the problem specification and solution extensions in our illustrative example of a paper smoothing plant in section 2. A concept of improving the flexibility of rolled material was translated into the current problem as an additional requirement. Thus, an 'intuitive' articulation of a new condition loses its mystery in the light of an analogy-based knowledge transfer from a 'familiar' experience.

The conceptual amendments defined in section 4 may seem confusing and sudden. They surely are sudden and hard-to-explain within a particular conceptual frame. However, when we allow stepping out of the current conceptual frame, the situation becomes better structured. Thus, we believe that it is possible to explain a designer's 'deep immersion' and 'insight' into a problem [1] in terms of an evolving conceptual frame of a particular vaguely defined problem. The proposed model of framing is a small step towards improved support tools and methodologies.

8 ACKNOWLEDGEMENTS

The authors wish to acknowledge the support received from the Open University and the European Commission in funding this research. Thanks to Phil Sainter for his comments on the earlier drafts of this paper.

9 REFERENCES

[1] R.L. Dominowski & P. Dallob, *Insight and Problem Solving*, in *The Nature of Insight*, R.J. Sternberg and J.E. Davidson, Editors. MIT Press: USA, 33-62, 1995.

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

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

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

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

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

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

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

[9] G. Fischer. *Domain-Oriented Design Environments*. In *7th Knowledge-Based Software Engineering Conference (KBSE'92)*. IEEE Computer Society. pp.204-213, 1992.

[10] J. McCarthy, Circumscription - A Form of Nonmonotonic Reasoning. *Artificial Intelligence*, **13**(1):27-39, 1980.

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

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

[13] G.S. Altshuller, *Creativity as an Exact Science*. (Studies in Cybernetics, vol. 5). 1984, USA: Gordon & Breach Science Publ.

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

[15] M. Levin, *Mathematical logic for computer scientists*. MAC project TR-131, MIT: Cambridge, Mass. 1974.

[16] S.D.N. Cook & J.S. Brown, Bridging Epistemologies: The generative dance between organizational knowledge and organizational knowing. *Organization Science*, **10**(4):381-400, 1999.

[17] D. Poole, A Methodology for Using a Default and Abductive Reasoning System. *Intl. Journal of Intelligent Systems*, **5**(5):521-548, 1990.

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

[19] B. Falkenhainer, K. Forbus & D. Gentner, The Structure Mapping Engine: Algorithm and Examples. *Artificial Intelligence*, **41**(1):1-63, 1989.

[20] I. Watson & S. Perera, Case-based design: A review and analysis of building design applications. *AI for Engineering, Design, Analysis and Manufacturing*, **11**(1):59-87, 1997.

[21] A.K. Goel, Design, Analogy, and Creativity. *IEEE Expert*, **12**(3):62-70, 1997.

[22] D.A. McAdams & K.L. Wood. *Quantitative Measures for Design by Analogy*. In *12th* ASME *Intl. Conf. on Design Theory and Methodology*. Baltimore, USA. DETC00/DTM-14562, 2000