Exploring problem decomposition in conceptual design among novice designers

Lassi A. Liikkanen, Helsinki Institute for Information Technology, P.O. Box 9800, FI-02015 TKK, Finland and Cognitive Science Unit, Department of Psychology, University of Helsinki, P.O. Box 9, FI-00014 UH, Finland
Matti Perttula, Mechanical Engineering, Helsinki University of Technology, P.O. Box 4100, FI-02015 TKK, Finland

Conceptual product design is commonly described as problem solving. In the present study we attempt to expand this view. Focusing on the solution search phase, we analyse explicit and implicit problem decomposition techniques and integrate them into a descriptive cognitive model. To evaluate the prevalence of decomposition modes empirically, we provide results from a verbal protocol analysis study involving 16 senior students of mechanical engineering. Data indicated that the subjects apply top–down control strategies coupled to implicit decomposition. Explicit decomposition was used seldom and without obvious benefits. We relate these results to the model that considers implicit decomposition as an integral part of the problem interpretation process and discuss the role of decomposition in a structured idea generation process.

Keywords: design cognition, cognitive strategies, conceptual design, problem decomposition, protocol analysis

Divide et impera is the rationale of problem decomposition, one of the most essential problem-solving techniques in the psychological literature (Newell and Simon, 1972; Anderson, 1983). Decomposition has found its way to models of decision making (Gettys et al., 1987) and penetrated several normative theories of the design process, for example in mechanical (Pahl and Beitz, 1984) and software engineering (Horning and Randell, 1973; Sommerville, 2001). The pattern language (Alexander et al., 1977) is an example of a decomposition-based methodology from architecture.

Despite the emphasis that design domains place on problem decomposition, its psychological underpinnings have not been a topic of extensive research. The benefits of applying decomposition have been studied mostly in quite generic, non-design contexts (Dennis et al., 1996, 1999; Coskun et al., 2000). In design, the most recent, dedicated investigation of decomposition was carried out by

Corresponding author:
Lassi A. Liikkanen
lassi.liikkanen@hiit.fi

www.elsevier.com/locate/destud
0142-694X $ - see front matter *Design Studies* **30** (2009) 38–59
doi:10.1016/j.destud.2008.07.003

Ho (2001) who identified two modes of decomposition, implicit and explicit. Explicit decomposition referred to the deliberate analysis of a function structure at the beginning of the design process. This corresponds to a typical textbook description of the method. In implicit decomposition, the function structure was not openly sought or revealed. However, an analysis of the design process demonstrated the existence of the structure, indicating that the problem had been decomposed implicitly. In this paper we call this proposition the dual-mode view of decomposition.

This paper examines problem decomposition in the solution search phase of conceptual product design, particularly related to the design of physical products. During solution search designers generate and evaluate concept candidates, making this stage crucial in determining the final product (Ulrich and Eppinger, 2003). Idea generation (IG) is a promising area for investigating problem decomposition, as it is a phase where repeated synthesis–analysis cycles are assumed to take place. The method we apply is verbal protocol analysis (Ericsson and Simon, 1984), which has been demonstrated effective for IG research (see Cross et al., 1996).

Our study of decomposition starts with a review of related theoretical and empirical work. This analysis leads to the formulation of a descriptive, cognitive model of design IG in product design, which treats problem decomposition as a specific problem-solving technique. In particular, we propose that explicit decomposition is a tool for verifying the usually spontaneously created initial implicit decomposition of the problem and that implicit decomposition is an inextricable part of the problem interpretation phase. Using two novice designer groups and two IG design tasks, we study empirically how the dual-mode view of decomposition is manifested in design IG.

The results showed that the designers mainly relied on implicit decomposition and the explicit decomposition was not essential for the designers, in an obvious contradiction to the educational design literature. Explicit decomposition was also clearly unrelated to structured control strategies that were constantly used in IG. Additionally we discovered that the designers were very solution-centred, but capable of constantly evaluating and discarding their ideas, even if instructed otherwise.

1 *Design cognition*

The current investigation approaches decomposition from the perspective of design cognition (Eastman, 2001). This paradigm describes design as information processing, employing explanatory concepts such as problem spaces, goal hierarchies, control strategies, and search methods (Akin, 1986; Chandrasekaran, 1990). In this framework, design is considered as ill-structured problem solving (Simon, 1973) because design problems define the problem in a fuzzy manner, with regards to initial state, goal state, and operators. This leads to

some special characteristics. For instance, as the goal state is ill-defined, designers must adopt stopping rules to terminate the problem-solving process at some point (Goel and Pirolli, 1992) so as to satisfy, not to optimize the solution (Simon, 1996).

Each problem domain is also characterized by the kind of problem-solving methods (or operators) used. When analysing the scope and reliability of these methods, a distinction between weak and strong methods is commonly made. Weak methods refer to domain-independent procedures that do not guarantee finding a solution whereas strong methods are based on domain-specific knowledge and thus more reliable and efficient with familiar problems. Research on design (Smith and Browne, 1993) and on other areas of ill-structured problem solving (e.g. scientific discovery in Klahr and Simon, 1999) has revealed that these problems are commonly solved using a combination of both weak and strong methods.

1.1 Decomposition in design problem solving

In cognitive science, decomposition is described as a weak method that designers use to cope with the complexity of ill-structured problems (Simon, 1996). It belongs to a family of problem reduction techniques (Volkema, 1983). The basis of this technique is to break problems into more transparent and less-complex parts that can be solved separately. These parts, called subproblems or subgoals, are therefore vital landmarks in the problem space. While decomposition in principle is a generic tool for defeating all problems, in practice the subproblems tend to enclose different kinds of interdependencies and become somewhat resistant to it. Hence design problem is called 'nearly decomposable' (Simon, 1996). To meet this challenge, designers must deliberately ignore interdependencies and treat the subproblems as 'leaky modules' (Goel, 1995).

In spite of the central role of decomposition in all problem solving, there are evident gaps in the cognitive theory of decomposition. This shows in the way authors apply various terms when referring to this process, for instance, design researchers use labels such as structuring (Goel and Pirolli, 1992; Restrepo and Christiaans, 2004), analysis (Schön, 1995), break-down (Pahl and Beitz, 1984), splitting (Jones, 1970), partitioning (Akin, 1986), and decomposition (Ulrich and Eppinger, 2003). Among psychologists terms such as factorization (Newell and Simon, 1972) and subgoaling (Laird et al., 1986) are employed. It must be noted that several of the concepts listed above denote an extensive phase of problem constraint induction and deduction, whereas we define decomposition in a more limited sense to refer to the processes producing subgoals (cf. Newell and Simon, 1972).

Most information about decomposition is derived from applied contexts as only a very limited number of studies can be found within cognitive

psychology (e.g. Egan and Greeno, 1974). An interesting design study of decomposition was carried out by Ho (2001), who demonstrated a difference in decomposition techniques (called 'strategies' by the author) between an expert and a novice designer in a product design task. Ho observed that the expert relied on *explicit* problem decomposition, whereas the novice did not utilize such a procedure. However, the emergent subgoal structure led the author to infer that some *implicit* mode of decomposition had been utilised. We labelled this finding as the dual-mode view of decomposition.

The cognitive basis of decomposition has been extensively considered by Akin in *The Psychology of Architectural Design* (1986). His account primarily concerns what kind of knowledge underlies decomposition. Knowledge is mandatory for problem decomposition, because in order to divide a problem, something must be known about it. As Lloyd and Scott (1994) put it, '[the] specific experience of the problem type enables designers to perceive new problems through old solutions, and…to structure and decompose a design problem'. Akin considers two specific types of human knowledge: representational and procedural (Akin, 1986). His treatment concentrates on procedural knowledge, but the issue of how it enables decomposition or interacts with representational knowledge is left open (ibid., p. 170).

Representational knowledge is organized as hierarchical pieces of knowledge called chunks (Akin, 1986). This enables decomposition because the hierarchical tree structure can be used to establish a problem space. Adapting this idea to product design, decomposition is a feasible technique since the problems (and solutions) can be represented as hierarchies of function structures (see Smith and Browne, 1993). Hence, we consider problem decomposition as a domain-independent process (a procedural schema) that operates on hierarchical, domain-specific knowledge.

The dual-mode view of decomposition by Ho (2001) can be seen as a complementary perspective to the chunk-based decomposition theory of Akin. Instead of specifying what kind of knowledge is required, it considers when and how decomposition is applied by the designers. Akin (1986) reported that the architects he studied mostly relied on a weak method called generate-and-test, but also acknowledged that there was a need for problem structuring to occur on a higher level. Ho (2001) extended this by demonstrating that indeed there is a structure on the higher level and this structure is brought about by implicit decomposition. Data from Ho's experiment additionally suggested an expertise effect, which might reflect the generic psychological phenomenon referenced by Akin about the transformation of declarative (representational) knowledge into procedural (know-about into know-how). In this paper, we focus on the implicit–explicit distinction, which seems more important for understanding practical design. Next, we analyze some previous empirical studies relevant for decomposition from the dual-mode perspective.

The problem-solving process has been investigated in several areas of design. A study comparing two groups of architecture and science students showed that designers concentrate their efforts on producing solutions instead of analysing the problem (Lawson, 1979). This implies a lack of explicit decomposition. In contrast, the investigations regarding software engineers (Jeffries et al., 1981; Ball and Ormerod, 1995) have all demonstrated seemingly explicit decomposition and that experienced programmers produce more useful and elaborated problem structures than the novice do. Also, a study considering the problem-solving strategies of experienced electrical engineers found them relying heavily on decomposition as they developed solutions using systematic breadth-first control strategies (Ball et al., 1997).

Finally, an unpublished study reported by Ball (Ullman et al., 1986 in Ball et al., 1994) revealed that the senior students of mechanical engineering use structured problem-solving strategies, but not explicit decomposition. Using the logic applied by Ho (2001), this suggests that some implicit decomposition method was applied in the process. Although the evidence is not abundant, these empirical findings support the dual-mode view by demonstrating that experience has an effect on problem decomposition and that this effect might be domain specific. However, the major shortcoming in referenced studies is that a cognitive account on how decomposition methods operate in detail is nowhere to be found (see Ball et al., 1997; Ho, 2001).

1.2 *Idea generation as problem solving*

We have presented problem decomposition as a design problem-solving technique, which operates in an implicit or explicit manner. To describe the role of problem decomposition in the conceptual design process and prepare for empirical work, we will consider how the modes of decomposition work in co-operation with other cognitive functions. This end is achieved with a cognitive model (see Figure 1) which is described in the following paragraphs. The model is heuristic and descriptive in nature and principally intended to support the later application of verbal protocol analysis. The aim is not to provide a parsimonious account of all cognitive activities relevant to design (cf. Akin, 1986), but to align decomposition into a meaningful framework. Although the focus is on the processes involved in IG, the model is compatible with other macro-level process models of design (e.g. Kruger and Cross, 2006).

In the model, we consider 'an idea' as a description of a device that can satisfy the task requirements. The idea is a combination of solution principles, which could be called subsolutions as they satisfy subgoals. On the top level (stage 0 in Figure 1), the model includes strategies to control the processing of different subgoals (control strategies resemble Akin's (1986) global search methods). They are necessary because several subgoals must be satisfied during design IG. Control strategies are usually classified into breadth-first and depth-first, reflecting the order and the level of abstraction in subgoal

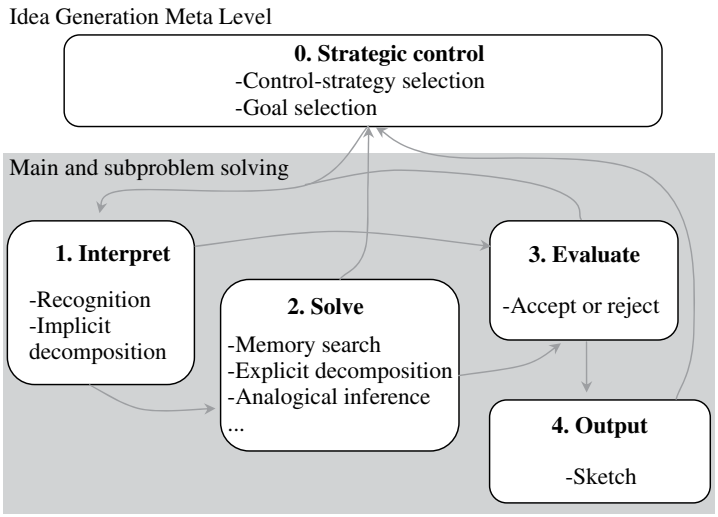


Figure 1 A heuristic model of the information-processing stages (0–4) involved in design idea generation

processing (Ball and Ormerod, 1995). Both can be labelled as structured, top–down approaches. The breadth-first strategy requires that each subproblem is solved at a certain level of detail before getting into a more detailed level, whereas the depth-first strategy means that each subgoal is independently solved to a considerable level of detail. Sometimes these modes unite in apparently opportunistic designs (Ball and Ormerod, 1995).

At the beginning of the actual problem-solving process designer must interpret the problem (stage 1), that is, produce its mental representation, the problem space. Interpretation, alike the following stages, is applicable on all problem-solving levels, iteratively from the main problem to various subproblem levels, creating nested, hierarchical problem spaces. Interpretation is achieved by attempting to match the current problem by its surface-level and conceptual similarities with the existing solution models (Federico, 1995).

Two important cognitive actions are involved in the interpretation process: recognition and implicit decomposition. Recognition is a strong problem-solving method that can lead to the immediate discovery of a solution. It is usually described as spontaneous (implicit) and effortless activity, characteristic of expert performance, and dependent on domain knowledge (Gobet and Simon, 1998; Lawson, 2004; Stein, 2004). The chances of successfully applying recognition increase as the similarity between the current problem and known solutions increases. If the current problem and the known models match poorly, then the interpretation process may be extended by analogical reasoning mechanisms, which allow transferring knowledge across domains (Klahr and Simon, 1999).

In the current model, recognition is intimately coupled with implicit decomposition. When task-relevant knowledge is retrieved during a recognition attempt,

it seems possible to use the recovered knowledge for deducing a function structure of the solution (effectively, decompose the problem). This implies a difference between experts and novices in problem decomposition because experts can recognise and recall more complex solutions owing to their huge knowledge base, organized in bigger chunks (Akin, 1986). For the same reason, experts might be able to find an alternative function structure later on by restarting decomposition explicitly (in stage 2). In contrast, novice designers may only retrieve incomplete or unfit matches that still provide enough information for implicit decomposition. Similar expertise effect on analytical skills has been observed, for instance, in classifying physics problems (Chi et al., 1981).

If interpretation does not provide a direct solution via recognition, it may provide a function structure through implicit decomposition. The existence of a problem structure allows the designer to start solving individual subproblems (stage 2). According to Laird et al. (1986), the search begins with the selection of the highest priority subgoal. Assuming that the solution cannot be achieved by recognition, the solution must be sought using a variety of other problem-solving methods. These include, but are not limited to, memory search mechanisms, analogical reasoning, explicit decomposition, and heuristics of IG (see Akin, 1986; Klahr and Simon, 1999). Explicit decomposition at a lower abstraction level must be preceded by higher level decomposition (implicit or explicit), because we cannot expect the designer to decompose an unknown subproblem. This relates to the restriction presented in the educational literature that decomposition should precede IG, although there is no other psychological reason to expect that. On the other hand, decomposition may be completely skipped if there are no matching solutions, or several recognizable solutions are available. In the former case, decomposition is impossible, in the latter, it is unnecessary for solving the problem.

Eventually, all generated solutions must be evaluated for their compatibility with the problem in question (stage 3). Only if the solution satisfies the adopted stopping rules (Goel and Pirolli, 1992) it will be produced on paper. For example, designers seeking novel ideas can intentionally discard a solution for being too ordinary or trivial. This should be especially important when dealing with recognised solutions that are by definition routine answers. If the idea qualifies, then the cycle ends to the output of that idea (stage 4). This application of stopping rules controls the iteration of the IG process. Due to working memory capacity limitations, it may be necessary to document each subsolution after generation (Bilda and Gero, 2007). For this reason, an additional recomposition procedure seems neither necessary nor feasible in IG as the complete solution is always a combination of solutions, generated in a serial order after resolving all interdependencies in a linear fashion (but see Ho, 2001 for a different view).

To sum up, we outlined a descriptive model of design idea generation, which considers problem decomposition as a knowledge-sensitive weak method. The model describes two modes of decomposition: implicit decomposition is a cognitive activity following an automatic recognition attempt and explicit is a complementary activity ideal for experts. Generally our model of IG bears resemblance to many dual-process models found in cognitive science (e.g. in Smith and DeCoster, 2000; Feldman Barrett et al., 2004) in which behaviour is regulated by two kinds of processes, automatic and controlled. This introduces many interesting associations and problems, but a full treatment of these parallels here is unnecessary and exceeds the scope of this paper.

2 Empirical study

After outlining a model of design IG, we next present a re-analysis of a large dataset (originally reported in Perttula and Liikkanen, 2006) to examine the relationship of implicit and explicit decomposition in conceptual product design. The main interest is in how novice designers use explicit and implicit decomposition in the design IG process and whether the process can be described by our IG model. To this end we employ verbal protocol analysis, which is a widely used method among design researchers (Gero and McNeill, 1998; Cross, 2004). This method is required to investigate the control strategies that can reveal implicit decomposition. IG involves several iterations of the concept on a coarse, easily describable level and therefore it should encourage the participants to apply explicit decomposition. We desired to keep the design event as intact and designer-driven as possible and consequently decided not to urge or imply the subjects to apply decomposition by any manipulation.

2.1 Participants

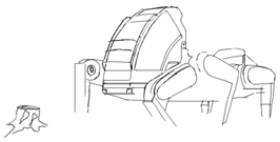
Sixteen senior students of mechanical engineering at the Helsinki University of Technology volunteered in the experiment. Experienced design students who had participated on two courses introducing conceptual design process and IG methods were recruited. On average, these students were completing their Master's level studies and possessed some practical design experience ($M = 0.8$ years, $SD = 0.7$ years), so they can be considered as advanced beginners. Except for one, all participants were male with the mean age of 26.8 years ($SD = 2.4$ years). They completed the assignment individually and signed informed consent to be videotaped during the task.

2.2 Experimental design

Idea generation was examined using two independent design tasks, *Plant* and *Forest*, illustrated in Figure 2. Two tasks, requiring different kinds of background knowledge, were used to increase the reliability of the findings. This resulted in a between-groups design of eight subjects in two groups.

The IG session was observed by the experimenter and captured on video. Upon starting the design session, the subjects were given a general brief which

The Forest Task
Design a machine to demolish trunks from a forest. The purpose is to remove the trunks efficiently on site. The machine moves on legs, so this feature can be omitted. Create as many different concepts as possible.



The Plant Task
Design an automatic watering device for home plants. The device should sustain a plant for at least one month, delivering one decilitre of water a week. Create as many different concepts as possible.

Figure 2 The design briefs for Plant and the Forest tasks (translated originals)

instructed them to avoid elaborating the sketches more than necessary and that they would not receive additional guidance during the task. Before starting the task, the subjects practiced thinking aloud with a 15-puzzle (see Ericsson and Simon, 1980). Next, the participants received the design brief. After reading it, the task began and the subjects were given 20 min to generate ideas and sketch them on separate sheets. The subjects were instructed to sketch one concept per sheet and include short textual annotations. After completing the assignment, documented ideas were inspected by the experimenter and the subject was asked to clarify the sketches if the function principles could not be otherwise identified. The setup is illustrated in Supplementary data.

2.3 Dependent variables

This experiment intended to study the modes of problem decomposition. To analyse data, operational criteria relevant to our model (Section 1.1) were devised instead of adopting any generic coding scheme (e.g. Gero and McNeill, 1998). The proposed model of decomposition assumes that during or after the decomposition process, the subgoals should emerge. Therefore to qualify as explicit decomposition, it was required that the designer in some way stated that he is decomposing the problem or presented the essential subfunctions in succession. For example, the former would be evident in statements such as 'I will now analyse this device' or 'let's see what parts we need to build this'. The latter criteria included the essential subgoals, which we defined as the subfunctions needed for constructing an operational device. Single code (called DEC) was used to identify protocol segments related to explicit decomposition. The relative timing and successfulness of decomposition were also evaluated for each idea.

To detect implicit decomposition, we adapted the indirect method previously utilised by Ho (2001). As it is impossible to detect implicit actions directly from a protocol, the top–down control strategies must be analyzed and their existence taken as an indicator of implicit decomposition. This relies on the

logic that the control strategies are inferred from the order and the abstraction level in subgoal processing (see Akin, 1986; Ball et al., 1997). Because the subgoals are considered to be the products of the problem decomposition process, it follows that if the designer is clearly following a structured goal-by-goal strategy, then implicit decomposition must have taken place.

Control strategies were determined for each idea by tracing the problem-solving activity in the problem space. This was done with problem decomposition schemes (PDSs), which correspond to the problem description graphs used to visualize problem solving (Akin, 1986). The PDSs were initially used for coding the protocols, later on the designers' productive steps were fitted to either a breadth or a depth-first control strategy (see Section 2.4). However, if the fitting failed, the concept was labelled as a mixed-strategy product (cf. Ball and Ormerod, 1995).

The protocols were also inspected for recognition-based problem solving. Uncovering recognition poses many of the same problems as implicit decomposition does. The operational definition given for recognition required the subject to indicate that he or she was very familiar with the solution in the first segments of a new idea: to know its name or otherwise prove acquaintance with the particular concept. Although not a bullet-proof method, it was considered necessary to try to exclude the recognized ideas from the analyses of control strategies. Also, the productivity of designers was examined in terms of the number of documented ideas (sketch sheets). Finally, we kept a record for events where designers verbalised ideas that they never documented on paper.

2.4 Processing protocols

The verbal protocols were coded in several steps as shown in Table 1. First, the audio track was transcribed to create a verbal protocol of the session (phase 1). The initial segmentation followed the natural train of speech. With the aid of the video recording, all sketches were numbered in the order of appearance and the segments referring to them were identified (phase 2).

In the third phase were created the task-specific problem decomposition schemes (PDSs), which described the function structures of the designed devices. This was done using the protocols and documented ideas bottom-up to abstract a general scheme that would cover the majority of solutions (see Van someren et al., 1994). For current purposes, three abstraction levels were considered necessary: *main function*, *subfunction*, and *solution principle* levels. The resulting schemes are provided in Appendix A. The main use of PDS was to enable tracking down control strategies, but the schemes also allowed determining the most essential subfunctions for each task through the subfunctions interdependencies.

Table 1 Ordered phases of the protocol coding process

Phase	Description
1.	Transcription and segmentation
2.	Association of segments and documented ideas
3.	Establishment of problem decomposition schemes
4.	Coding decomposition codes
5.	Coding verbalized PDS codes
6.	Coding non-verbalized PDS codes
7.	Verification of explicit PDS coding (repeated twice)
8.	Identification of control strategies and recognition

In the following phase (4), the segments that demonstrated explicit decomposition were coded with a positive decomposition code. Next, all segments that made progress to the current idea were coded with a *verbalized PDS* code (phase 5). In addition, the tone of reference was identified as problem or solution-oriented. Repetitive or non-informative segments that did not contribute to design were not coded. An example of a segment PDS coded for main function could be: 'So I should design a moving machine', of a subfunction phrase 'It has some removal mechanism', and of a solution principle 'It then burns the trunk'. All these examples were solution-oriented, which was an important distinction especially in the case of subfunctions, which were rarely articulated as problems such as 'I need a mechanism for removing the trunk', but given as generalised solutions for the subgoal. This is an important detail, because they are considered here as equally important pointers to the function structure, although they appear very different in the protocols. The segments that included a reference to the main function, subfunction or a solution principle, but were not associated with any documented design detail, were labelled distinctively (see the PDS code postfix '/0' in Table 2).

A non-verbalized PDS code was used to take into account the features of a design idea that had been sketched, but not verbalised (phase 6). The non-verbalized

Table 2 Translated extract from the protocol of the subject number 13

Time	Concept no.	DEC	Verbalized PDS	Non-verbalized PDS	Segment
0:19:20	14	0			How about electricity − could electricity
0:19:31	14	0	48/0		It would likely cut or
0:19:33	14	0	37		Burn it somehow
0:19:37	14	0			If the trunk is wet enough it would conduct
0:19:40	14	0			Electricity, so er…
0:19:42	14	0	(37)		Electricity burns it down,
0:19:43	14	0		70	A proper lightning

While developing the 14th idea, the subject first introduces a solution principle that is not implemented (PDS code 48), then adds an electric burner for an onsite demolition subfunction (PDS code 37) and mentions it twice. Finally he sketches the burner as being attached to the machine body by a crane, but does not mention it and thus commits a non-verbalized design action (PDS code 70).

PDS code was attached to the nearest segment that had not been PDS or decomposition coded. The resulting coding scheme is illustrated in Table 2 as a protocol extract. The coding was carried out twice by the author and resulting codes were inspected for consistency (phase 7). Some portions of the protocols were cross-checked by the co-author and any differences were discussed and resolved.

After the PDS coding, control strategies were determined for each documented idea (phase 8). The strategy identification was based on the PDS codes related to each concept. It was also required that the idea included all the essential subfunctions. In a typical breadth-first occurrence, the subject would first introduce a couple of subfunctions in a row and then describe the solution principles for them. In terms of PDS coding, this might be: 3, 4, 31 and 41 as numbers less than 10 correspond to subfunctions and greater numbers refer to solution principles (see Appendix A). In contrast, a depth-first strategy would be indicated by presenting one or several solution principles for a single subgoal before considering the next subgoal. The search patterns that failed to match either criterion were classified as mixed-strategy products. Some designs identified as being produced by recognition were not analysed. Eventually, the strategic orientation of each subject was determined by the mode class of the control strategies applied by him or her.

3 *Results*

All subjects were able to produce several alternative concepts during the experiment ($M = 9.8$, SD = 3.6 concepts). More ideas were generated in the *Forest* than in the *Plant* task (10.6 vs. 8.9 concepts, respectively), but the difference was not statistically significant ($T(7) = 1.10, p > 0.10$). Subjects verbalized a considerable amount of information while working ($M = 869.9$, SD = 378.6 words per protocol) and the data from all designers were included in the analysis.

Designers talked mostly about their ongoing efforts. For each subject, on average, 32 segments (SD = 8.8) were PDS coded of all 212 segments (SD = 72) transcribed. The ratio of PDS segments from all segments is small (15%) because the segments containing repetition and elaboration were discarded. The participant #7 was less talkative and produced the majority of his designs in non-verbalized form, although his designs were otherwise similar to other designers' ideas. Also, four other subjects produced 15–20% of their PDS coded segments in non-verbalized form.

The coded segments consisted almost completely of solution-oriented statements. Designers seldom stated explicitly what problem, or subproblem, they were considering at the time, as only 4.0% of all segments were problem-oriented. Several designers verbalized a solution that they never sketched, and in the case of subject number 13 this behaviour was recurrent.

3.1 Control strategies and recognition

The majority of the design ideas were produced using an identifiable, structured, top–down control strategy (see Table 3). In the *Plant* task the dominant strategy over all concepts was the breadth-first (71% of 71 concepts), but in the *Forest* task the proportions of mixed and depth-first strategies (38% vs. 36% of 85 concepts) were almost equal. The patterns of subgoal selection were also analyzed during the inspection of solution patterns. It appeared that the development of each concept began from an apparently random subgoal, and there was no identifiable pattern of subgoal selection across ideas.

To gain insights into the nature of mixed-strategy designs, a qualitative analysis was performed per design. This inspection revealed that three designers (#9, #15 and #16) classified as mixed-strategy oriented, applied an 'opportunistic depth-first' strategy. That is, they neglected one of the essential subgoals and developed only one or two principles in a concept, which excluded them from the breadth-first category. Additionally, one mixed-strategy oriented designer (subject #13) clearly used a breadth-first strategy, but constantly omitted the same essential subfunction from his designs. Thus, he applied an incomplete breadth-first strategy. Six instances of recognition were detected, all in the *Plant* task, where a common solution was the

Table 3 Summaries of the control strategies used by the subjects in different tasks

Subject	Recognition (%)	Depth-first (%)	Breadth-first (%)	Mixed (%)
1	0	**75**	19	6
2	0	11	**67**	22
3	0	**90**	0	10
4	0	0	**78**	22
5	0	0	**100**	0
6	0	8	**92**	0
7	0	**69**	0	31
8	14	0	**86**	0
9	0	10	30	**60**
10	0	0	**91**	9
11	0	**46**	23	31
12	17	17	**67**	0
13	0	0	14	**86**
14	13	13	**75**	0
15	0	0	20	**80**
16	25	13	13	**50**
Forest *M*	0	36	26	**38**
Plant *M*	9	8	**71**	13
Overall *M*	4	22	**48**	25
Strategic orientation of subjects				
Forest ($n = 8$)	0	50	12	38
Plant ($n = 8$)	0	0	88	12

The bold typeface indicates the strategic orientation of each subject. The odd-numbered subjects comprise the Forest group, even-numbered the Plant group.

adaptation of a medical drip bottle (1st concept of subject #12, and 4th concept subject #16). This was deemed as implicit analogical reasoning.

3.2 Problem decomposition

The majority of the subjects did not use explicit problem decomposition. Only two designers stated during the task that they were analysing the problem (subjects #10 and #11), and in the case of subject number 13 explicit decomposition could be inferred from the serial verbalisation of the essential subgoals. Two of these three designers had been working with the Forest task, one with Plant. Among the remaining 13 subjects, no traces of explicit decomposition were found. Evaluation of these three designers in terms of timing and completeness showed that only one designer (#10) started his session by applying decomposition, and only another one (#13) verbalised all subgoals. The use of decomposition seemed to affect neither designers' productivity measured by the number of ideas produced nor their selection of a control strategy (see Table 4), although the productivity of these designers was above the average.

4 Discussion

In this paper we have developed the theory of problem decomposition in conceptual product design and empirically reproduced some previous findings (Ho, 2001). We proposed implicit decomposition as an instinctive technique for interpreting familiar problems. The results showed that the working model of design IG was useful in exploring decomposition through the design protocols. For instance, the processes of recognition and analogical inference were utilized in the task that concerned a familiar problem environment. Some subjects also came up with design ideas that were never sketched, in spite of the

Table 4 The modes of decomposition related to the strategic orientation and productivity of each designer

Subject no.	Mode of decomposition	Strategic orientation	Number of ideas produced
10	Explicit	Breadth-first	11
11	Explicit	Depth-first	13
13	Explicit	Mixed	14
1	Implicit	Depth-first	16
2	Implicit	Breadth-first	9
3	Implicit	Depth-first	10
4	Implicit	Breadth-first	9
5	Implicit	Breadth-first	3
6	Implicit	Breadth-first	13
7	Implicit	Depth-first	13
8	Implicit	Breadth-first	7
9	Implicit	Mixed	10
12	Implicit	Breadth-first	6
14	Implicit	Breadth-first	8
15	Implicit	Mixed	5
16	Implicit	Mixed	8

explicit instruction to document all ideas. This demonstrates how stubbornly evaluation mechanisms operate even in no-feedback circumstances.

Among other noticeable features of the data, the designers talked about solutions, not about the problems. This has been a repeated finding in the design literature, which describes designers as solution-oriented (Lawson, 1979; Restrepo and Christiaans, 2004; but see Kruger and Cross, 2006) Also, sometimes design details were merely drawn, without any spoken reference. While talking aloud might improve with practice, this once again (see Akin, 1986) emphasizes the need to consider all available information sources in order to gain a realistic conception of the design behaviour. In all, the findings highlight the individual differences in IG and verbalisation, as idea rejections and non-verbalized details were an issue only with some designers.

4.1 Design under control

In the protocol analysis we could associate all designs with a control strategy. This indicated that the IG process proceeded as a structured search process. This was expected (cf. Ball et al., 1994) although the two tasks clearly called for different control strategies. In the *Plant* task, subjects used a breadth-first strategy, which was anticipated as the task guidelines did not encourage using a depth-first strategy. In the *Forest* task, a considerable proportion of depth-first and mixed strategies was encountered. The qualitative analysis of mixed strategies showed that designers used variants of the standard top–down strategies (incomplete breadth-first and opportunistic depth-first strategies), not truly opportunistic approaches (Ball and Ormerod, 1995). The difference in strategy selection between the tasks is likely attributable to the dissimilar function structures (see Appendix A). The function structure for the Forest task was more complicated and therefore, subjects may have incidentally omitted subfunctions and adapted a depth-first strategy, possibly due to problems with retrieving subgoals from long-term memory.

The properties of human memory might explain another uncovered aspect of opportunism. There appeared to be no fixed order of attending to subgoals, regardless of the control strategy applied. Although this could be explained by proposing that the subgoals are not prioritised in the first place (but see Laird et al., 1986), a more informative explanation is memory search. Searching memory for a solution to fulfil a subgoal can produce several solutions in a short time (see Liikkanen and Perttula, in press). This can lead to a situation where a designer possesses several solutions for the same subgoal and chooses one of them as a starting point for the next idea just because that is 'the easiest thing' to do. This hypothesis of memory-based generativity, to be trialled in future, as it might help understanding opportunism in subgoal selection and in the use of an atypical depth-first control strategy.

4.2 Explicit and implicit decomposition

The goal of this paper was to advance the theory of problem decomposition in design cognition. Our main thesis, embedded in a cognitive model of IG, was that implicit decomposition should be viewed as an inevitable, automatic part of the problem interpretation process. Consider a metaphor; sometimes you have the feeling that you are seeing something wrong ('it can't be'), forcing you to correct your perception by having another look. We suggest that explicit decomposition is a similar corrective and elaborative process that experts can utilise to challenge the implicit decomposition to 'see things differently'. Interpreted this way, decomposition becomes similar with the old concept of problem restructuring, associated with creative problem solving (Duncker, 1945).

This study replicated the previous finding that novice designers do not use explicit decomposition in conceptual product design (Ullman et al., 1986 in Ball et al., 1994; Ho, 2001). There neither appeared to be a clear relation between the mode of decomposition and the control strategy used. This implies that these are two independent issues as assumed in our model. We found that only a single subject presented a complete function structure of the problem while all designers could generate plenty of ideas. This shows that explicit decomposition was neither necessary nor beneficial for IG in the kind of design tasks presented, unlike one would expect based on the educational literature.

Considering the suggested model of decomposition, likely the prime reason for why the novice designers did not use explicit decomposition is that they did not possess a large enough base of relevant knowledge. This might be connected to the recent finding that novice designers do not employ limited commitment mode (an extension of the breadth-first control strategy) as often as experts do (Kim et al., 2007). It seems plausible that the large domain-specific knowledge base of an expert designer is what makes explicit decomposition practical. This is probably also the reason why this method has emerged from the literature often grounded on best-practices. However, this implies that decomposition may have limited applicability in the design of radically new kind of products, because designers cannot possess an accurately fitting model for a novel artefact. The value of decomposition might also be bound to the design domain. Compared to electrical and software engineering (Jeffries et al., 1981; Ball and Ormerod, 1995; Ball et al., 1997) product design appeared here as a more holistic and solution-centred domain, at least in the initial stages of design – if this observation is not an artefact of the research methodology.

There are alternative interpretations for the lack of explicit decomposition. First, the task assignment was to produce several alternative designs quickly and without elaboration. Also, the subjects were not instructed to apply problem decomposition. Thirdly, it is common that the design experiments carried

out in a laboratory fall short on ecological validity because of limited time and commitment. It is possible that these factors may have guided the subjects to favour more straight-forward methods and strategies over analytical ones, considered to take time away from generating ideas (cf. Kruger and Cross, 2006).

Fourthly, the subjects were on average able to generate almost 10 concepts in 20 min; hence the tasks were relatively easy. It is possible that in a more complex and difficult task, explicit decomposition might be spontaneously used. The final option is that subjects were not familiar enough with explicit decomposition in order to utilise it effectively as it was not controlled. These options should be considered in upcoming studies.

5 *Conclusions*

In this paper, we attached a dual-stage view of problem decomposition into a model of design IG and used the model to describe verbal protocol data from 16 designers. It appeared that explicit decomposition was an unnecessary step in the IG process as designers successfully implicitly decomposed the problem. This fits our model which proposes implicit decomposition as a part of the problem interpretation process. This result also reflects the nature of design as an implicitly controlled psychological phenomenon, as was already discussed by Lawson (1979) and summarised by Lloyd and Scott (1994): 'The general education of a designer may promote certain attitudes to design problems but is unlikely to be a large factor when designing'. This is exactly the kind of discrepancy also revealed in the present study. We believe that there can be differences in the use of explicit decomposition across design domains; some domains may be inherently more holistic, some more analytic.

Many psychological features of problem decomposition remain unsettled. The interplay of a generic problem-solving technique and domain-specific knowledge presents open question. We speculated whether implicit decomposition can occur at multiple hierarchical levels, but could not address the issue because our problem environments had a restricted number of abstraction levels. For design, maybe the ultimate question is can we expect benefits for design IG to be gained from the application of explicit decomposition. There is sound evidence supporting the use of explicit decomposition and a depth-first strategy in idea generation in non-design contexts (Dennis et al., 1996, 1999; Coskun et al., 2000) and we did obtain some hints that decomposition might facilitate productivity (cf. problem-driven designers in Kruger and Cross, 2006). However, this issue should be investigated within design IG to understand its value for design work. These investigations must discriminate decomposition, control strategies, and their interaction to assess them reliably. Only after further experiments would we dare to say that explicit problem decomposition is a redundant activity in the product design IG process.

Acknowledgements

This work was conducted as a part of a research project funded by the Academy of Finland. We thank Miikka Vanhamaa for providing the task brief illustrations, J. Matias Kivikangas for transcribing the protocols, and all the people whose comments helped us to improve the manuscript, especially the two anonymous reviewers.

Appendix A. Problem decomposition schemes (function structure analyses)

PDS for the Forest *task*

Main function	0. Trunk disposal	
Solution type	1. Integrated devices 2. External devices	
Subfunctions	3. Destruction	4. Removal
Principles	30. Burning on site or in a tank 31. Detonation or shooting 32. Decomposition into soil 33. Organisms or animals 35. Water pressure 35. Chemical 36. Ultrasound 37. Electricity 38. Bacterial	40. Hook 41. Root destroyer 42. Cork screw 43. Loading shovel or plow 44. Harvester feet 45. Lever arm 46. Hydraulic grap 47. Cable and pulling 48. Water cutting 49. Vibrator 410. Hoover 411. Wedge 412. Hydraulic tentacle
Subfunctions	5. Dispersal	6. Recovery
Principles	50. Saw 51. Press 52. Circular saw 53. Drum chamber 54. Sprocket wheels 55. Gnawing, cutting, grinding, sand down 56. Drilling 57. Screw mechanism 58. Smash against ground 59. Axe or blade	60. Bag or cart 61. Suction 62. Hydraulic arm or shovel 63. Cart or platform 64. Screw mechanism 65. Arm or jib crane
Subfunctions	7. Attachment to machine	8. Mediating mechanisms
Principles	70. Jib crane 71. In front 72. Below 73. Behind 74. In a foot 75. Above	80. Applicator 81. Metering device or syringe 82. Transportation vehicles

PDS for the Plant *task*

Main function	0. Plant watering device	
Solution type	1. Devices attached to pot 2. Independent devices	
Subfunctions	3. Water supply	4. Regulation
Principles	30. Separate tank 31. Water pipe 32. Rain wall 33. Condensation water 34. Ice 35. Bound water 36. Tank in soil 37. Tank in pot or in Plant board 38. Waterin can 39. Steamer 310. Closed system	40. Mechanical timer 41. Valve or choke 42. Humidity sensor 43. Pipe, hose, or string size 44. Mail man 45. Weight sensor 46. Surface permeability 47. Phone, PC, or electric timer 48. Tank decomposition 49. Controlled vaporization 410. Controlled by transfer SubF 411. Controlled by energy SubF 412. Liquid pressure difference 413. Liquid buffering
Subfunctions	5. Transfer	6. Mediator
Principles	50. Hose, pipe, or gutter 51. Capillar hose, stick, or string 52. Pouring system 53. Gravity; syphon or drain 54. Sprinkler 55. Pump, water gun, or pressure 56. Syphon 57. Diffusion 58. Absorption 59. Scooping or sprinkling	60. Plant mover 61. Soil changer 62. Robot 63. Rotating ground 64. Household air 65. Plant air 66. Humidity collector
Subfunctions	7. Energy source	
Principles	70. Mains 71. Sun 72. Battery 73. Wind 74. Rubber band drive 75. Candle or camping cooker 76. Explosive 77. Gravity	

Appendix B. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.destud.2008.07.003.

References

Akin, Ö (1986) *Psychology of architectural design* Pion, London, UK

Alexander, C, Ishikawa, S and Silverstein, M (1977) *A pattern language* Oxford University Press, New York

Anderson, J R (1983) *The architecture of cognition* Harvard University Press, Cambridge, MA

Ball, L J, Evans, J S B T and Dennis, I (1994) Cognitive processes in engineering design: a longitudinal study, *Ergonomics* Vol 37 pp 1753–1786

Ball, L J, Evans, J S B T, Dennis, I and Ormerod, T C (1997) Problem-solving strategies and expertise in engineering design, *Thinking and Reasoning* Vol 3 pp 247–270

Ball, L J and Ormerod, T C (1995) Structured and opportunistic processing in design: a critical discussion, *International Journal of Human–Computer Studies* Vol 43 pp 131–151

Barrett, L F, Tugade, M M and Engle, R W (2004) Individual differences in working memory capacity and dual-process theories of the mind, *Psychological Bulletin* Vol 130 pp 553–573

Bilda, Z and Gero, J S (2007) The impact of working memory limitations on the design process during conceptualization, *Design Studies* Vol 28 pp 343–367

Chandrasekaran, B (1990) Design problem solving: a task analysis, *AI Magazine* Vol 11 pp 59–71

Chi, M T H, Feltovich, P J and Glaser, R (1981) Categorization and representation of physics problems by experts and novices, *Cognitive Science* Vol 5 pp 121–152

Coskun, H, Paulus, P B, Brown, V and Sherwood, J J (2000) Cognitive stimulation and problem presentation in idea-generating groups, *Group Dynamics: Theory Research and Practice* Vol 4 No 4 pp 307–329

Cross, N (2004) Expertise in design: an overview, *Design Studies* Vol 25 pp 427–441

Cross, N, Christiaans, H and Dorst, K (1996) *Analysing design activity* Wiley, New York

Dennis, A R, Aronson, J E, Heninger, W G and Walker, E D (1999) Structuring time and task in electronic brainstorming, *MIS Quarterly* Vol 23 No 1 pp 95–108

Dennis, A R, Valacich, J S, Connolly, T and Wynne, B E (1996) Process structuring in electronic brainstorming, *Information Systems Research* Vol 7 No 2 pp 268–277

Duncker, K (1945) *On problem-solving* American Psychological Association Inc., Washington, DC

Eastman, C (2001) New directions in design cognition: studies of representation and recall in **C Eastman**, **W Newstetter and M McCracken** (eds) *Design knowing and learning: cognition in design education* Elsevier, Oxford, UK

Egan, D E and Greeno, J G (1974) Theories of rule induction: knowledge acquired in concept learning, serial pattern learning, and problem solving in **L W Gregg** (ed) *Knowledge and cognition* Lawrence Erlbaum Associates, Hillsdale, NJ

Ericsson, K A and Simon, H A (1980) Verbal reports as data, *Psychological Review* Vol 87 pp 215–251

Ericsson, K A and Simon, H A (1984) *Protocol analysis* MIT Press, Cambridge, MA

Federico, P-A (1995) Expert and novice recognition of similar situations, *Human Factors* Vol 37 pp 105–122

Gero, J S and McNeill, T (1998) An approach to the analysis of design protocols, *Design Studies* Vol 19

Gettys, C F, Pliske, R M, Manning, C and Casey, J T (1987) An evaluation of human act generation performance, *Organizational Behavior and Human Decision Processes* Vol 39 No 1 pp 23–51
Gobet, F and Simon, H A (1998) Pattern recognition makes search possible: comments on holding, *Psychological Research* Vol 61 pp 204–208
Goel, V (1995) *Sketches of thought* MIT Press, Cambridge, MA
Goel, V and Pirolli, P (1992) The structure of design problem spaces, *Cognitive Science* Vol 16 pp 395–429
Ho, C-H (2001) Some phenomena of problem decomposition strategy for design thinking: differences between novices and experts, *Design Studies* Vol 22 pp 27–45
Horning, J J and Randell, B (1973) Process structuring, *ACM Computing Surveys* Vol 5 No 1 pp 5–30
Jeffries, R, Turner, A A, Polson, P G and Atwood, M E (1981) The processes involved in designing software in **J R Anderson** (ed) *Cognitive skills and their acquisition* Lawrence Erlbaum Associates, Hillsdale, NJ
Jones, J C (1970) *Design methods. Seeds of human futures* Wiley-Interscience, London
Kim, M H, Kim, Y S, Lee, H S and Park, J A (2007) An underlying cognitive aspect of design creativity: limited commitment mode control strategy, *Design Studies* Vol 28 No 4 pp 585–604
Klahr, D and Simon, H A (1999) Studies of scientific discovery: complementary approaches and convergent findings, *Psychological Bulletin* Vol 125 pp 524–543
Kruger, C and Cross, N (2006) Solution driven versus problem driven design: strategies and outcomes, *Design Studies* Vol 27 pp 527–548
Laird, J, Rosenbloom, P and Newell, A (1986) *Universal subgoaling and chunking: the automatic generation and learning of goal hierarchies* Kluwer, Boston
Lawson, B R (1979) Cognitive strategies in architectural design, *Ergonomics* Vol 22 pp 59–68
Lawson, B (2004) Schemata, gambits and precedent: some factors in design expertise, *Design Studies* Vol 25 pp 443
Liikkanen, L A and Perttula M (in press) Inspiring design idea generation: Insights from a memory-search perspective, *Journal of Engineering Design*, doi: 10.1080/09544820802353297.
Lloyd, P and Scott, P (1994) Discovering the design problem, *Design Studies* Vol 15 pp 125–140
Newell, A and Simon, H A (1972) *Human problem solving* Prentice-Hall, Englewood Cliffs, NJ
Pahl, G and Beitz, W (1984) *Engineering design* The Design Council, London, UK
Perttula, M K and Liikkanen, L A (2006) Exposure effects in design idea generation: unconscious conformity or a product of sampling probability? in **M Jonsson and R Unnporsson** (eds) *Proceedings of NordDesign 2006*, Reykjavik, Iceland
Restrepo, J and Christiaans, H (2004) Problem structuring and information access in design, *Journal of Design Research* Vol 4
Schön, D A (1995) *The reflective practitioner: how professionals think in action. Reprise edition*, Aldershot, Arena, UK
Simon, H A (1973) The structure of Ill structured problems, *Artificial Intelligence* Vol 4 pp 181–201
Simon, H A (1996) *The sciences of the artificial* (3rd edn) MIT Press, Cambridge, MA
Smith, G F and Browne, G J (1993) Conceptual foundations of design problem solving, *IEEE Transactions on Systems, Man and Cybernetics* Vol 23 pp 1209–1219

Smith, E R and DeCoster, J (2000) Dual-process models in social and cognitive psychology: conceptual integration and links to underlying memory systems, *Personality and Social Psychology Review* Vol 4 pp 108–131
Sommerville, I (2001) *Software engineering* (6th edn) Addison-Wesley, Harlow
Stein, B (2004) Engineers don't search in **W Lenski** (ed) *Lecture notes in computer science. Logic versus approximation* Springer-Verlag, Berlin, Germany
Ullman, D G, Stauffer, L A and Dietterich, T G (1986) *Preliminary results of an experimental study on mechanical design process* Oregon State University, Corvallis, Oregon
Ulrich, K T and Eppinger, S D (2003) *Product design and development* (3rd edn) McGraw-Hill, Boston
Van someren, M W, Barnard, Y F and Sandberg, J A C (1994) *The think aloud method. A practical guide to modelling cognitive processes* Academic Press Limited, London, UK
Volkema, R J (1983) Problem formulation in planning and design, *Management Science* Vol 29 No 6 pp 639–652