Lassi A Liikkanen

•Exploring Problem Decomposition in Conceptual Design•

Master's thesis
in Cognitive Science

•Department of
Psychology

•Faculty of
Behavioural Sciences

•University of
Helsinki

•November 2005

HELSINGIN YLIOPISTO - HELSINGFORS UNIVERSITET - UNIVERSITY OF HELSINKI

Tiedekunta - Fakultet - Faculty Käyttäytymistieteellinen tiedekunta	Laitos - Institution – Department Psykologian laitos

Tekijä - Författare – Author
Lassi A. Liikkanen

Työn nimi - Arbetets titel
Ongelmien dekompositio konseptisuunnittelussa

Title
Exploring Problem Decomposition in Conceptual Design

Oppiaine - Läroämne – Subject
Kognitiotiede

Työn laji - Arbetets art – Level Pro gradu –tutkielma	Aika - Datum - Month and year Marraskuu 2005	Sivumäärä - Sidoantal - Number of pages 82 + liitteet

Tiivistelmä - Referat – Abstract

Tuotteita ja palveluita suunnittelevia aloja on nykyään lukuisia. Suunnittelun eri alueet ovat kehittyneet itsenäisesti, ja viime aikoina niistä on kiinnostuttu myös psykologiassa. Suunnittelukognitioksi kutsutaan tutkimussuuntaa, jossa suunnittelua tutkitaan kognitiivisen psykologian näkökulmasta. Suunnittelukognition tulokset ovat suunnittelun normatiivisten mallien ohella kiinteä osa alan tietämystä. Kaikilta osin psykologinen todellisuus ei kuitenkaan vastaa mallien olettamuksia, vaan ne eroavat toisistaan kuten optimaalinen ja tosiasiallisen päätöksenteko. Tämä tutkimus selvitti mahdollista eroavaisuutta ongelmien dekomposition suhteen.

Ongelmien dekompositio on menetelmä, joka sisältyy niin suunnittelun oppikirjoihin kuin ihmisen ongelmanratkaisun yleisiin malleihin. Dekomposition idea on helpottaa ongelmanratkaisua keskittämällä ongelmanratkaisijan huomio kerrallaan vain yhteen osaan ongelmasta. Dekomposition keskeisyydestä huolimatta dekomposition merkitys konseptisuunnittelussa on epäselvä ja niukasti tutkittu asia. Tässä työssä käytettiin protokolla-analyysia dekomposition tutkimiseksi. Aiemmissa kokeellisissa tutkimuksissa on esitetty dekomposition toimivan sekä implisiittisesti että eksplisiittisesti, mutta asiaa ei ole teoreettisesti perusteltu. Niinpä tässä tutkimuksessa käytiin ensin läpi viimeaikaisia suunnittelu- ja ongelmanratkaisuteorioita, joista koottiin kognitiivinen malli konseptisuunnittelun etsintävaiheesta. Tässä mallissa konseptisuunnittelun etsintävaihe nähdään tunnistus- ja dekompositio-skeemojen ohjaamana ongelmanratkaisuna.

Teoreettista vaihetta seurasi empiirinen koe dekomposition tutkimiseksi. Kokeeseen osallistuneet kuusitoista (N=16) kokenutta koneensuunnittelun opiskelijaa tuottivat konsepteja kahdesta vaihtoehtoisesta aiheesta. Yhtäaikaista ääneenajattelua ja protokolla-analyysia hyödynnettiin dekomposition tutkimiseksi. Tulokset osoittivat, että huolimatta dekomposition korostamisesta koulutuksessa, ainoastaan muutama (N=3) suunnittelija käytti menetelmää spontaanisti ja eksplisiittisesti esitetyn kaltaisissa tehtävissä, vaikka tehtävät muuten ratkaistiin järjestelmällisesti käyttäen ylhäältä-alas-ohjausstrategiaa. Eksplisiittisen dekomposition käyttäminen tutkituissa tapauksissa ei myöskään tuottanut odotettuja tuloksia Eksplisiittisen dekomposition sijaan suunnittelijat käyttivät ohjausstrategioista päätellen implisiittistä dekompositiota. Nämä tulokset tukevat aiemmin tehtyjä havaintoja, mutta myös korostavat dekomposition lisätutkimuksen tarvetta. Tulevaisuudessa pitäisi selvittää, voidaanko dekomposition käyttöä tehostaa, millainen kognitiivinen prosessi dekompositio on ja mitkä sen käytön vaikutukset työskentelylle ovat. Näiden kysymysten selvittämisen jälkeen voidaan nykyisiä tuloksia arvioida uudelleen.

Avainsanat – Nyckelord
ongelman ratkaisu, dekompositio, suunnittelu, konseptisuunnittelu, suunnittelukognitio

Keywords
problem solving, decomposition, design, conceptual design, design cognition

Säilytyspaikka – Förvaringsställe - Where deposited
Käyttäytymistieteellisen tiedekunnan kirjasto, Helsingin yliopisto

Muita tietoja - Övriga uppgifter - Additional information
Tämä työ valmisteltiin osana Suomen Akatemian rahoittamaa Navigo–tutkimusprojektia

HELSINGIN YLIOPISTO - HELSINGFORS UNIVERSITET - UNIVERSITY OF HELSINKI

Tiedekunta - Fakultet - Faculty Faculty of Behavioural Sciences	Laitos - Institution – Department Department of Psychology	
Tekijä - Författare – Author Lassi A. Liikkanen		
Työn nimi - Arbetets titel Ongelmien dekompositio konseptisuunnittelussa		
Title Exploring Problem Decomposition in Conceptual Design		
Oppiaine - Läroämne – Subject Cognitive Science		
Työn laji - Arbetets art - Level Master's Thesis	Aika - Datum - Month and year November, 2005	Sivumäärä - Sidoantal - Number of pages 82 + Appendixes

Tiivistelmä - Referat – Abstract

Design embraces several disciplines dedicated to the production of artifacts and services. These disciplines are quite independent and only recently has psychological interest focused on them. Nowadays, the psychological theories of design, also called design cognition literature, describe the design process from the information processing viewpoint. These models co-exist with the normative standards of how designs should be crafted. In many places there are concrete discrepancies between these two in a way that resembles the differences between the actual and ideal decision-making. This study aimed to explore the possible difference related to problem decomposition.

Decomposition is a standard component of human problem-solving models and is also included in the normative models of design. The idea of decomposition is to focus on a single aspect of the problem at a time. Despite its significance, the nature of decomposition in conceptual design is poorly understood and has only been preliminary investigated. This study addressed the status of decomposition in conceptual design of products using protocol analysis. Previous empirical investigations have argued that there are implicit and explicit decomposition, but have not provided a theoretical basis for these two. Therefore, the current research began by reviewing the problem solving and design literature and then composing a cognitive model of the solution search of conceptual design. The result is a synthetic view which describes recognition and decomposition as the basic schemata for conceptual design.

A psychological experiment was conducted to explore decomposition. In the test, sixteen (N=16) senior students of mechanical engineering created concepts for two alternative tasks. The concurrent think-aloud method and protocol analysis were used to study decomposition. The results showed that despite the emphasis on decomposition in the formal education, only few designers (N=3) used decomposition explicitly and spontaneously in the presented tasks, although the designers in general applied a top-down control strategy. Instead, inferring from the use of structured strategies, the designers always relied on implicit decomposition. These results confirm the initial observations found in the literature, but they also suggest that decomposition should be investigated further. In the future, the benefits and possibilities of explicit decomposition should be considered along with the cognitive mechanisms behind decomposition. After that, the current results could be reinterpreted.

Avainsanat – Nyckelord
ongelman ratkaisu, dekompositio, suunnittelu, kognitiivinen mallintaminen, suunnittelukognitio

Keywords
problem solving, decomposition, design, cognitive modelling, design cognition

Säilytyspaikka – Förvaringsställe - Where deposited
The Library of Behavioural Sciences, University of Helsinki

Muita tietoja - Övriga uppgifter - Additional information
This work was conducted as a part of Navigo research project, funded by the Academy of Finland

Acknowledgements

The foundation for this work was established back in 2003 when Pekka Sipilä together with Matti “Masa” Perttula, and Professor Kalevi Ekman received funding for conceptual design research project Navigo from the Academy of Finland. My employment in the project began as a research assistant in January 2005 with the aim to complete this thesis by the end of the same year. I have since been dominantly chuffed with the challenge I accepted. The road has been a steep and rough one, but this work hopefully demonstrates that at least some bypaths have been necessary.

All through the year, the inspiring discussions with and the delicate guidance from Masa have greatly facilitated this work. Two other persons also contributed to this effort. Designer Miikka Vanhamaa provided the example sketches and J. Matias Kivikangas carried out a part of data analysis. Sinikka Hiltunen kindly proofread the thesis and offered valuable comments. This work was written in several locations. Firstly, I wish to express gratitude for the research infrastructure provided by Helsinki University of Technology. I also spent a wonderful week writing at the biological station of Kilpisjärvi, made possible by a grant from University of Helsinki Master’s theses fund. And a small part was done in Saint-Petersburg on my stay with Antti and Maija.

The intellectual support from several sources within University of Helsinki has also been very helpful. Especially the guidance of Professor Christina Krause and the comments from cognitive science students Anna-Mari Rusanen, Markus Mattsson, and Otto Lappi were useful. Jarmo Herkman provided me with initial literature references, which were later complemented by Isto Maarttola. I must also mention the incidental communications with the researchers Sami Paavola, Pauli Brattico, Kai Hakkarainen, and Ilkka Toikka. Last but not least I’m indebted for all the volunteers that participated in the experiment, and of course, beloved Riikka, who was luckily mostly spared from my distress with the thesis, but still had the spirit to encourage me until the end.

My sincere thanks for everyone mentioned, this work would not have been completed without You!

24th of November 2005, In Helsinki

Lassi A. Liikkanen

TABLE OF CONTENTS

LIST OF APPENDIXES

1. Preface

Design is an act of organizing reality. Designers create artifacts that fill in our daily lives, making up the environments we live in and the tools we use in everyday activities. Design as an applied science is important in shaping our society and has been so long before its current status as a discipline. The design process affords many intriguing research questions. How do artifacts come into being? Why do some artifacts appear so special that we call them creative? More specific riddles can be laid, but who should answer them? Psychology investigates mechanisms underlying human behavior; does it have answers for design? The psychological discipline has existed for a relatively short period of time, but it has already provided plenty of information about human behavior in various domains. Design is definitely one of those. The biggest steps in the psychology of design have been taken during the last three decades. The progress has happened mainly under the flag of cognitivism, a new psychological paradigm born in the 1950's. Cognitivism and the more recently surfaced cognitive science have provided psychology with methods and concepts that have made it possible to explore and explain new kinds of psychological phenomena. This has also occurred for design where concepts of information processing have been adopted (Simon, 1996). Although some particular areas, such as conceptual product design, are yet to have their own theory.

The history of the psychological research of design from the problem-solving perspective is a relatively short one, but has already caused a paradigm shift in design research. A growing number of theories about and models of human designing are cognitively oriented and often referred to as the *design cognition* literature, which is based on the symbolic paradigm of cognitive science. Still, it seems that the design cognition approach has not yet been able to fulfill its promise - to provide a detailed account of design as cognitive activity - as the most of research publications still discuss their results in the future tense rather than the present, i.e. focusing more on what should be done than on what has been achieved thus far. Optimistically interpreted, this means that there is still a plenty of room for investigation to be done, and indeed a lot of energy is currently invested into design studies worldwide.

This thesis describes a research effort within the paradigm of symbolic cognitivism. I have chosen to investigate problem decomposition, one of the classical components of the problem-solving architecture, in the environment of conceptual design of products, using classical methods to derive new results. The work is organized in four chapters. The introduction presents a brief overview of cognitive science and the methods and concepts of design, concentrating on the particular theories of problem solving. A reader who is already familiar with the basics of cognitive science and problem solving can therefore skip the first sections and step straight into section 2.4 of Introduction, in which the links between this work and the heterogeneous creativity research are discussed. The synthesis continues in section 2.5, which introduces a cognitive model of design idea generation and the study question. The experimental chapter describes how this question was evaluated empirically; it is followed by the results. The work is finished off with a chapter on conclusions and speculations as to why the results turned out the way they did. This is by nature multidisciplinary and the review is written in a verbose style considering the engineer audience, who might not be familiar with the theories of cognitive science and cognitive scientists who know little about design. The variant of protocol analysis developed for this investigation is also reported in detail. With this preface, it is my hope that the reader will find decomposition a topic of justified interest. In the text, I have chosen to refer to a designer as “she” or “her” regardless of the hypothetical designer’s sex.

2. Introduction

> If cognitive science does not exist then it is necessary to invent it.
> (Johnson-Laird, 1980)

This thesis is an empirical study of internal solution search in conceptual design of products inquiring the role of problem decomposition in it. The method chosen for the study, protocol analysis, is one of the classical methods of cognitive science. However, as the quotation from Johnson-Laird notes, it may still be uncertain what this discipline is. In order to present a thesis in the context of a discipline, it is advisable to define the discipline and the paradigm within. This is particularly clarifying if there exists as a wide range of opinions on the essence of the field as there is in cognitive science, which has become generally known only after the 1970's.

Concerning cognitive science

Cognition is defined in the Oxford English Dictionary as

> The action or faculty of knowing taken in its widest sense, including sensation, perception, conception, etc., as distinguished from feeling and volition; also, more specifically, the action of cognizing an object in perception proper.
> Oxford English Online, version 2

Cognitive science is hence a wide-ranging domain, not *a* discipline but possibly several: cognitive sciences. The development began in the 1950's when the information processing approach started to invade the psychological discipline. This new view was greatly influenced by the invention of digital computing and it borrowed many concepts and working models from the computer science of that time. Cognitive science added a computational aspect to psychology, posing precision requirements of computer science for psychological theories. This approach suggests that human behavior is controlled by mechanisms that are more complex than those proposed by the behaviorists. Instead of regarding thinking as a mere mediation of behavior, cognitivism appreciates the multitude of explanations, usually building on the concept of cognitive structures (Mayer, 1992).

The majority of the theoretical work presented here takes the unitary (Anderson, 1983), computational approach (Marr, 1982) to cognition, which is compatible with the physical symbol system hypothesis advocated by Newell (e.g. Newell, 1990; Newell & Simon, 1972).The discussion about whether this approach is adequate for explaining human behavior is essential. Goel, using design as a example target, has considered this issue at length and his primary conclusion is that physical symbols systems are sufficient, but restricted means (Goel, 1995). The alternative, connectionist approaches to design and creative processes are rare and will not be considered in this work.

This work addresses the human cognition underlying creative design work, particularly conceptual design of products. The main motive comes from the fact that conceptual design is still poorly understood as a psychological activity (see section 2.5), but it still affords the investigation of decomposition well. In order to achieve a linkage between two separate disciplines, the theoretical part reviews some concepts of general cognition, goes through specific aspects of creative cognition, and finally grapples with some issues of design practice and cognition. The objective of the theoretical work is to provide the reader with an overall understanding of conceptual design as cognitive activity and describe how a process called problem decomposition is related to conceptual design. Therefore, this introduction will first introduce a cognitive model of conceptual design, and then ask the question of how a component of the model, decomposition, can be empirically examined with the methods of cognitive science. The experimental results will question the role of decomposition in actual design, revealing the need for additional investigations.

Exclusions

Many interesting but not strongly related things must be omitted from the theoretical work. The missed bypaths are mentioned in the appropriate locations along the way, but it must be now stated that individual differences and abnormalities, and developmental issues are completely dismissed. This study examines the expert designers of today and the future, who have been highly educated in their own field, not the neophytes in design. The design process as a whole, and especially conceptual design, includes a lot of decision-making which is another excluded subject. And of course, the human behavior, including design, is not a static, but flexible process that steadily improves. However, learning presents a different research domain that is ruled out of this work.

Cognitive structures

Cognitive science is about explaining psychological phenomena by using concepts based on human information processing models. The traditional model is the man-computer metaphor that sees a human being as equivalent to a universal computer or Turing machine (Turing, 1950). This model, influenced by the idea of digital computer, includes the following components: memory systems, central processing, perceptual, sensory, and motor systems. Each component has been extensively studied along with the major functions such as language, reasoning, decision making, and problem solving. This work is concerned with the central components and will omit the perceptual and motor functions (Eysenck & Keane, 2000).

Cognitive architectures

The key question in cognitive science concerns the cognitive structures; what are the central functions of cognition and how they are organized. Several propositions exist. Some of these propositions are models in a more abstract sense, for instance the models of creative process, while others are more formal and are realized as computational, symbolic models. In the last three decades, a couple of very detailed computational models of cognitive structures have been introduced. These models are known as the cognitive architectures, the best examples of which are SOAR and ACT[1] (Anderson, 1990; Newell, 1990). These architectures are important in proving that computational systems can carry out many complex activities with an effort and accuracy comparable to that of humans. Explicit architectures also facilitate the evaluation of underlying assumptions by simulating complex interactions afforded by the theories; even though they have some inherent limitations (see Lewis, 2001). I feel that all cognitive theories should be implemented in some architecture or a computational model in order to assess their feasibility and I fully agree with Anderson who stated (Anderson, 1980 in Russell and Norvig, 2003) "a cognitive theory should be like a computer program."

Memory and attention

Memory is one of the most essential human cognitive capacities. It would be impossible to imagine a culture such as ours without the memory resources of a human. There exists a great variety of different psychological characterizations of memory. Associationism is an old way of describing the nature of memory, which has been adopted by psychologists. Currently there exist several psychological memory models that could be called associative. Before introducing these fairly recent applications of the old principles, I will take a glance at the first half of the 20th century.

The idea of memory stores is an elemental part of the computer metaphor, as the architecture of a digital computer includes transient and persistent memory stores in addition to a central information processing unit. The quest for identifying corresponding memory systems in humans can be considered to be one of the first steps in the cognitive revolution that began in 1956 (Bechtel *et*

---

[1] The Adaptive Character of Thought model (ACT) has had several revisions. The current model is called ACT-R 6. See http://act-r.psy.cmu.edu/.

al., 2001). But actually, the idea of memory stores with different temporal characteristics was suggested before that era by Hebb (1949). He proposed a distinction between long- and short-term memories that was adopted by cognitive scientists. Short-term memory (STM) has a limited capacity, but it is rapidly accessible. STM contents decay fast, therefore it must be connected to LTM that allows finite amounts of information to be stored for elongated period of times. Long-term memory can be accessed relatively fast, but storing new information takes time. To compensate for the slowness of writing to LTM, an external memory (EM) can be utilized. EM refers to all objects in the world, including notebooks, other people, product catalogs, Internet, and so forth. Thus, using EM memory requires both perceptive and motor actions (Newell & Simon, 1972).

More detailed memory theories subsist. For a while it was believed that short-term memory would be a unitary cognitive component and its capacity could be stated in content-independent chunks. Miller infamously stated that this volume was “seven plus minus two” units (Miller, 1956)[2]. This view was replaced a few decades later by a multi-component theory of STM (Baddeley, 2003)and the estimates of the capacity have been reduced (Cowan, 2001). The best known model is Baddeley’s working memory model. It is composed of four independent STM components processing different types of information: a phonological loop, a visuospatial sketchpad, and an episodic buffer. These buffers are coupled to a common central executive and to corresponding long-term stores (Baddeley, 2003). The specialization of short-term memory structures leads to further questions about long-term memory, such as how information is stored, what is this information like, and how it is accessed. A part of this discussion goes under the name of knowledge representation. Several propositions have been made about it, but only some necessary for the

---

[2] The ‘magical number seven’ has since become accepted as a psychological fact in applied sciences, although this it has been shown that this estimate was not correct

current work will be pointed out see (e.g. Eysenck & Keane, 2000; Mayer, 1992).

First, there are specialized computational models of memory retrieval and recognition, such as SAM and MINERVA 2. They use an associative structure to explain memory storage and retrieval (see a review in Raaijmakers & Shiffrin, 1992). The basic idea is that memory search is rather simple, probabilistic process, controlled by few variables (Raaijmakers & Shiffrin, 1981). This can be elaborated by saying that LTM is built on associations. However, it is more informative to refer to semantic organization as a guiding principle of associations. It has been proposed that LTM is organized in semantic hierarchies, so that concepts such as animals, mammals, dogs, and cats are associated together in a (prototypical) semantic network of nodes and links (Eysenck & Keane, 2000). Concepts and their properties are connected together with cue validity, a property comparable to associative strength. However, there is experimental evidence that have displayed several effects that are compatible with the theory and confirmed it to a certain extent, although some theorists heavily disagree with this type of "semantic decomposition" (see Keane, 1997 and Lakoff, 1986).

It is also believed that there is more than just one type of memory, just as there are STM modules, and that these types of information are stored in different formats (the format of knowledge –question). The most fundamental difference is between propositional and visual information, which are very likely two separate systems (Eysenck & Keane, 2000). There is possibly a dedicated format for each sensory channel (tactile, olfactory, etc.). Additionally, types such as explicit or declarative, procedural, episodic knowledge have been suggested. Declarative knowledge is usually propositional and semantic, for example "Switzerland has no coast line", whereas procedural knowledge is about guiding behavior through *if-then* type of rules such as "if I go to Switzerland, I don't take my diving equipment with me". Knowledge types are very essential for determining other cognitive structures, because the differences in knowledge must show in the processes that handle the knowledge. For instance, a control structure for problem-solving might be based on procedural knowledge, while these structures would in turn process declarative knowledge (Anderson, 1983; Eysenck & Keane, 2000).

Going from the small bits of knowledge to bigger compositions such as large mechanical devices, schemata are often mentioned. They are used to refer to extensive, connected chunks of information, and can be used to guide perception and acquisition of new information. Today they have been somewhat replaced by a more recent concept of mental representations, but these two concepts are different. The major difference is that schemata contain procedural knowledge whereas mental models are declarative, although conceptualizations vary between theorists. To avoid unnecessary confusion after this point, only schema will be used from this point on. Extensive discussion on schemata, mental models, scripts, and frames can be found elsewhere (Boden, 2004; Eysenck & Keane, 2000; Finke *et al.*, 1992).

Declarative or semantic knowledge can be defined more precisely. The theory of long-term working memory (LT-WM) has proposed that expert's memory consists of memory chunks that grouped together. It is assumed that a "higher-level" chunk contains "pointers" (references) to "lower-level" chunks (Ericsson & Kintsch, 1995). This sort of formulation resembles the prototypical approach discussed above. For example, the Tequila Sunrise is a mixture of alcohol, grenadine, ice, and orange juice. The concept of Tequila Sunrise consists of certain components, but the concept has at least its own name to distinguish it from other concepts that may have similar constituents. This work will use this simple model of knowledge representation as a working hypothesis in later chapters.

Memory research is still in progress. In spite of the recent development, the original, quite abstract information-processing model based on the computer metaphor still seems acceptable. It can be accepted in principle by authors that hold otherwise incompatible views[3]. It is therefore also used in current study.

Attention

While memory is undeniably the one most essential factors of human cognition, the existence of a central executive in Baddeley's memory model reminds us that memory is not a sufficient concept for explaining all cognitive activity. Attention is another component that has an important role. Attention is usually seen as a process which selects what information we become aware of. This can mean both focusing on and ruling out thoughts and sensory information (Eysenck & Keane, 2000). Attention, like STM, is limited and in normal circumstances, only one thing can be attended at a time. This makes the conscious human information processing serial. Failures to maintain attention, e.g. in tests of vigilance, can lead to significant trouble in cognitive activities. Considering narrow attention together with the limitations of short-term memory, two centrals limits to human problem solving have been introduced.

---

[3] See Cowan (2001) and commentaries in the special issue of Behavioral and Brain Sciences, Volume 24, issue 1

Problem-solving research

> It is not my aim to surprise or shock you … But the simplest way I can summarize is to say that there are now in the world machines that think, that learn and that create.… The range of problems they can handle will be coextensive with the range to which human mind has been applied.
> (Simon & Newell, 1958)

Problem solving is one research area of cognitive science. It partially continues the exploration began in the early 20th century by the Gestalt school (Duncker, 1945). After the 1960's, the information processing approach (cognitive science and artificial intelligence) has taken over the problem-solving research. The research initially advanced in leaps, but after the fifties and sixties, the progress has been more subtle. On the other hand, the cognitive revolution has only recently inspired many near and distant disciplines, e.g. design theory. The purpose of this section is to review the relevant parts of the problem-solving literature and show how conceptual design can be treated as a problem-solving activity, as has been claimed by the design theorists (e.g. Pahl & Beitz, 1984)

Problem solving is an activity that focuses on a problem, however, defining what a problem is, can be a problematic. This is because problems are not "absolute", but defined in a context. Objects and situations are not problematic in nature, but in a context. For psychological research, this property can be described along these lines:

> A person is confronted with a *problem* when he wants something and *does not know immediately* what series of actions he can perform to get it. The desired object may be tangible or abstract. It may be specific or quite general. It may be a physical

> object or a set of symbols.
> (Newell & Simon, 1972) emphasis added

Similar description can be found from other authors, (for instance, Reitman, 1965). Common to all is the idea of having a goal as a result of an interpretation of the problem situation.

Problem types

The abstract definition of a problem requires more precision. In cognitive science, problems can be categorized to *well-* and *ill- structured* problems (WSP and ISP). Well-structured problems are those that a problem-solving system can unambiguously test to see whether a proposed solution is an answer or not. This test determines whether the *goal state* definite or not. But an *initial state* must also be defined. In games, such as chess, it is clear how the pieces are arranged, so the initial state is well-defined. In real life problems, initial state is usually incompletely or inaccurately specified, so the initial state is ill-defined. Thus, to be a WSP, it is necessary that both the initial and the goal state of the problem are well-defined (Newell & Simon, 1972; Reitman, 1965). This requirement can be extended by requiring the legal problem-solving operators to be defined, as legal moves in chess are (Newell, 1969). Sometimes terms “well- and ill-defined problems” are used instead of “well- and ill-structured” problems, but I prefer the latter option, which better captures the both ends of the problem situation.

Given these criteria, is it possible to determine whether a problem is a WSP? Simon (1973) found that the strict determination of WSP:s lead to a situation where no problems studied so far could not be labeled as well-structured. Thus, real problems seem to resist classification. This is comparable to writing one’s Master’s thesis that is undeniably an ISP. In spite of being ill-structured, the thesis is eventually evaluated and graded, which shows that this ISP indeed has *some* constraints regarding the goal state. It is therefore more realistic to think that problems lie somewhere on a continuum between well and ill structured end points than to think that there are only WSP:s and ISP:s (Reitman, 1965; Simon, 1973).

One possibility that provides a more flexible and psychologically sound classification is to define problem types as radial categories. A radial category consists of a set of properties that make up a prototypical member of a conceptual category. Category membership is evaluated by the extent that an instance matches the prototypical properties (Lakoff, 1986). This approach has been applied to separate design and non-design problems. Goel et al. (Goel & Pirolli, 1992) included twelve items to the list of design problem features. This makes design, including conceptual design, one type of problem solving, so that the aforementioned problem dichotomies can be combined to produce a presentation, such as Figure 1.

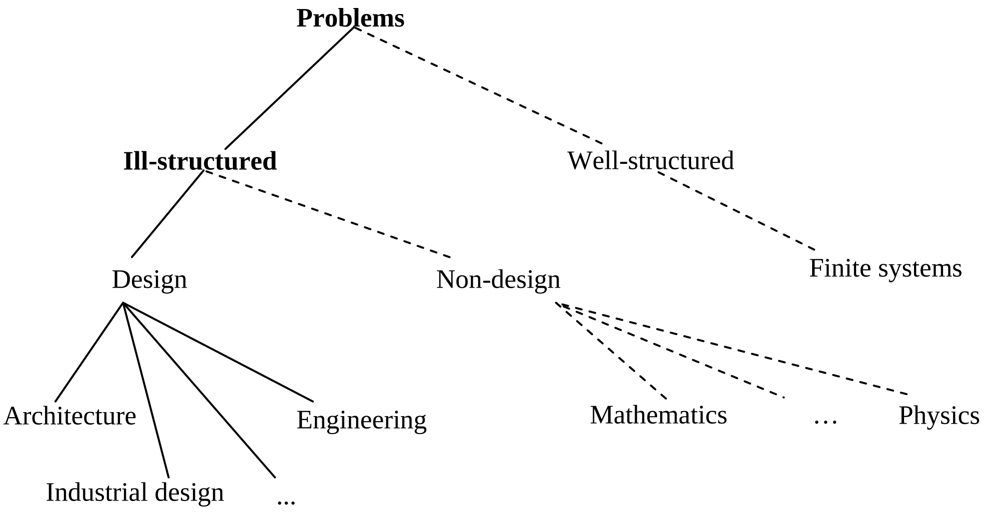


Figure 1. Tree model of the problem types highlighting some prototypical design and non-design disciplines based on Simon (1973) and Goel (1995).

Finally, an observation about the relation of problem solving and memory must be mentioned. Simon has stressed the need for knowledge in solving ISPs. He claimed that information from LTM, and possibly from EM, must be used comprehensively when solving ISP:s (Simon, 1973). This is due to inadequate information provided by an ISP, in terms of the initial state, goal state, and problem solving operations, which must be complemented by the problem solver. This implies that ill-structured problem solving is also *reproductive* (cf. Duncker, 1945), it makes much use of what we already know or can easily adapt. Now that problems have been introduced as the objects of this study, it is time to see how they can be solved according cognitive scientists.

Cognitive approach

Cognitive research on problem solving began more as a subject of computer science than psychological interest because the researchers were eager to build an artificial intelligence (AI) that would equal human thought[4]. Therefore, it was natural that the cognitive structures would be modeled by a computer right from the start. To be able to achieve this goal, information about the human style of problem solving had to be acquired. First comprehensive model from that background was the Problem space theory. (Newell & Simon, 1972; Russell & Norvig, 2003)

Problem space theory

Probably the most influential framework of problem solving is the Problem space theory. It was implemented in a model called General Problem Solver (GPS) that was the slickest application of the heuristics models and which was referred to at the opening of this section. The theory states that problem solving is essentially about representing the problem in an effective way and making changes to a representation to transform it to a solution. The model is built on the concepts of *problem states*, *problem space* and *problem solving operations*. The problem (or solution) states refer to definite solution positions, in terms of the problem constituents. E.g. a chess position is a problem state in chess. The problem space is a network of all possible problem states connected by nodes

---

[4] An actual discipline called Artificial Intelligence emerged much later, in the late 1980's according to Russell and Norvig (2003).

presenting legal problem-solving operations. The initial state denotes the state at beginning of the problem-solving process. The goal state(s) is located on the other end of the space and additional rules may be needed to identify the goal state, as in chess. Connections between states depend on the restrictions problem makes on the solution, in the case of chess, the rules of the game. Problem solving occurs by applying problem-solving operations (also termed operators or actions), which make moves from a problem state towards another state using the legal connections. In brief, problem solving is a process of making moves within in a problem space. The actual problem is how to select the right moves to achieve the goal state (Newell & Simon, 1972). An example of a problem space for a simple task called the Tower of Hanoi is displayed in Figure 2.

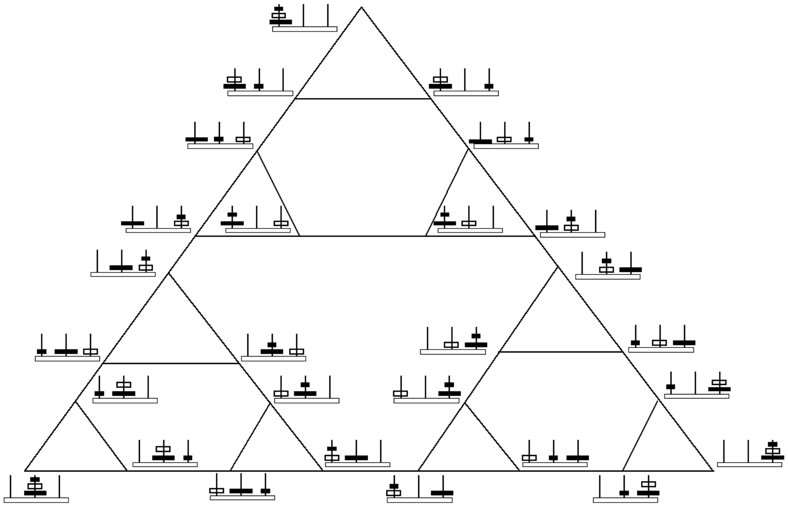


Figure 2. Visualization of one problem space related to the Tower of Hanoi puzzle. The puzzle consists of three pegs and three blocks. Objective is to move all blocks, one by one, from the left peg to the right peg. Initial state is located on the top, goal state in the bottom right corner. Note that even a simple game has more than twenty seven *possible states* that are shown in the figure.

The efficiency of the solution search in a problem space can be measured. For instance, in the illustrated problem space of the Tower of Hanoi puzzle, there are several paths in the problem space that all lead to the goal state(s). It is evident from the pictorial description that only some routes are shorter than others, and thus, favorable. Roughly speaking, if it is possible to determine one or several paths that lead to a goal state with the smallest number of moves, then the solution can be *optimized* in relation to this particular parameter. However, optimization is not possible in all problem environments. In solving ISPs, such as design problems, it is impossible to optimize a solution and the problem solving becomes a task satisficing task constraints rather than optimizing a solution. (Simon, 1996)

Some problems may not be represented as a single problem space. Problem solvers decompose problems to make them more manageable, creating new subproblems. The decomposition of a problem leads to a set of subproblems that each have their own, perhaps more limited, problem spaces. New subgoals maybe added with decomposed at all times during the process. Decomposition is therefore a crucial procedure for solving problems of more than a trivial size and it is included in most problem-solving theories (Anderson, 1983; Newell & Simon, 1972). The application of decomposition assumes that the subproblems can be solved independently (Simon, 1996). Violations to this assumption along with a more detailed discussion of decomposition will be presented in the section 2.5.

Search methods

The most important part of problem solving is the selection of problem-solving operators. There are two fundamentally different options for doing this: the generate-and-test method and constrained search methods. The former method generates all possible moves blindly applying all possible operators and the latter class uses knowledge to pick some moves over the others. Their details will be discussed next. Note that these methods are not specific to the Problem space theory, but they are a part of the more extensive AI literature, which contains a plethora of methods, many distant from psychology (see Russell & Norvig, 2003).

The conception about the nature of search methods has changed throughout the years. In the first AI applications, problem solving was seen as a blind search going through all options without any *a priori* selection of operators. The nature of the problem and the number of available operators determined the complexity of the task and the bigger the space, the harder it was to find the goal state. This problem was visible even in well-structured problems, for instance in chess, where the uninformed search became impractical due to combinatorial explosion (Russell & Norvig, 2003). This gave a good reason to suspect that other problem-solving methods are needed as generate-and-test was neither psychologically plausible.

There are several ways that the search process can be constrained by knowledge. Traditionally these are called strong and weak methods. Strong methods are procedures that lead to a correct answer, such as mathematical functions. Strong methods are typically domain-dependent and their activation requires recognizing the fit between the problem situation and the method (Russell & Norvig, 2003). One strong method is recognition, which means that complete solutions may be recognized and retrieved in entirety from the knowledge base. Weak methods, sometimes called strategies, were an essential part of the GPS. They do not guarantee a success and can even bypass the optimum solution. They either work on an abstract, syntactic, level or apply some general heuristics as means to problem solving and are thus domain-independent. Weak methods in psychological problem-solving models include means-ends analysis (MEA) and hill climbing (Newell & Simon, 1972).

Weak and strong methods can apply domain-dependent rules[5] heuristics (rules of thumb) relevant to the particular problem. Heuristics are simple, and therefore, the problem solver must possess lots of them for being effective. The more specific the heuristic is, the more likely it is applicable only to a single situation (Newell & Simon, 1972). The options presented this far have been summarized in Figure 3.

---

[5] Concepts of weak methods and heuristics are sometimes used interchangeably. In this work it is assumed that weak methods are general, domain-independent and heuristics specific, domain-dependent methods. The use of the term heuristics also varies from author to author, see e.g. Boden (2004)and Simon and Newell (1958).

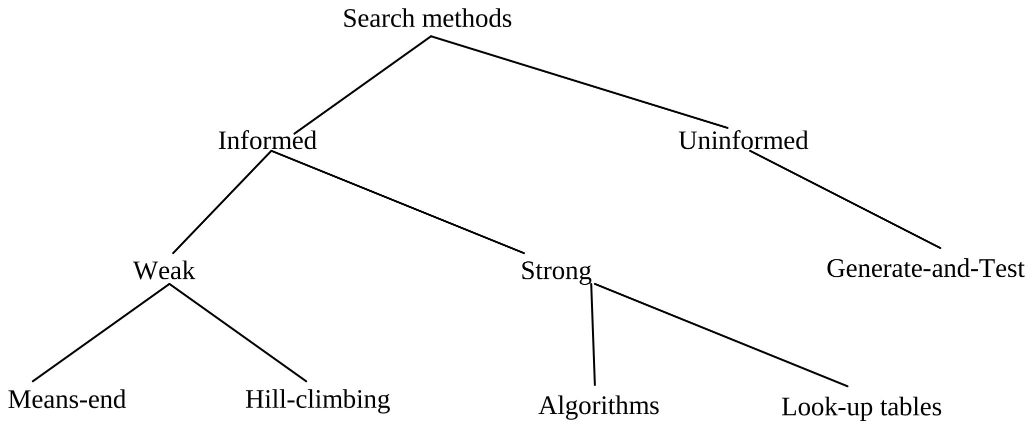


Figure 3. Classification of general problem solving methods with some examples from problem-solving and AI literature.

While the selection of problem-solving operators is an important part, problem solving is also constrained by the limitations of general cognitive structures, which were considered earlier. Short-term memory is capacity limited and may thus impede the solving of more complex problems. Long-term memory has a very large capacity, but has problems with retrieval and storage. External memory is thus a valuable aid for the problem solver, as new solutions may be found from the environment and external memory can be used to overcome limitations of STM and LTM.

The Problem space theory has been implemented as a computational model and its operation has been compared with human performance. This has been done on few well-structured problems, e.g. theorem proving, Moore-Anderson, and cryptarithmetic tasks. The results were inspiring. The GPS model could solve several types of problems and seemed to produce the solutions in a similar manner as human problem solvers (Newell & Simon, 1972). However, the GPS had some limitations. The greatest concern has been the limited validity of laboratory experiments that used simple problems. One aspect is the acquisition and administration of the knowledge needed for the problem solving. In the examples problems used in the experiments, all information was provided in terms of legal operators and well-defined initial and goal states, whereas problems in real life are different. It seems that the Problem space theory at best describes only well-structured problem solving (Eysenck & Keane, 2000). This problem was well perceived at the time when the model was introduced by other

AI researchers and the model was accused of unwarranted promises, starting from its name (see McDermott, 1976). In this work I have chosen to include the Problem space theory as a reference point, especially because the AI research has since the 1970's diverged from psychology and cognitive science (Russell & Norvig, 2003), but the concepts are still widely used.

Expertise

The GPS and similar efforts considered only a limited class of problems. However, there has been an interest to understand more complex, ill-structured problem solving from the information-processing perspective. As a result, the expertise research that focuses on more difficult and realistic problem environments (cf. Voss & Post, 1988) has emerged. It had already been proposed that ill structured problems, in particular, are knowledge-sensitive (Simon, 1973) and it followed that the expertise research focused on the knowledge structures that make sophisticated problem solving possible. This also shifted the focus of AI to the knowledge-based or expert systems.

Expertise is defined as a domain-specific skill acquired by a substantial amount of training. It results in a knowledge base that quantitatively and qualitatively differs from a database of a novice (Ericsson & Smith, 1991a). Where does this difference actually originate and how is expertise embedded in the cognitive structures? First suggestion was that the difference is due to a more effective chunking, not to bigger short-term memory capacity as such. It seems that expertise allows bigger information structures to be used as chunks, where as novices rely on more simple chunks. In chess, this could mean using complete position board patterns as chunks, instead of single pieces. This claim has been proofed quite consistently over the years (Charness, 1991). However, chunking theory is not the only possible or even adequate explanation for experts' extraordinary performance. For example, recently it has been suggested that this skilled memory could be facilitated by long-term working memory (Ericsson & Kintsch, 1995; Eysenck & Keane, 2000).

Whichever theory we choose, it seems that expertise leads to sophisticated LTM structures. It has been proposed that expertise affects both schematic (procedural) and declarative knowledge bases of a person (Mayer, 1992). The different types of knowledge make it possible for the expert to handle many types of problems effectively, e.g. they do not blindly select procedures that are uncertain or loading and normally rely on a database of declarative knowledge. If the solution is not directly recognized, then the experts can proceed in their problem solving by using procedural knowledge to decompose (analyze) the ill structured problem into a set of well-structured problems, which can then be solved with declarative knowledge (Chi *et al.*, 1988). This work also makes the assumption, that the reuse of existing knowledge is the key concept for understanding design.

This far expertise has been considered from a general perspective. However, several domains of ill structured problem solving have been investigated, not only chess. These include specific cases of observing challenging decision-making processes (Voss & Post, 1988), scientific discovery (Klahr & Simon, 1999), and design (Smith & Browne, 1993). Each area has received notable attention, but only design problem solving will be considered in detail in the section 2.5 see also (Chi et al., 1988; Ericsson & Smith, 1991b; Mayer, 1992).

Analogical thinking

Analogical thinking is a weak problem-solving method that supports knowledge-based problem solving (Klahr & Simon, 1999). If a solution to a particular problem is missing, a solution to another problem can possibly be used as a model. This is the idea of analogical problem solving. Analogies do not solely serve the purpose of problem solving as they are used in everyday conversations as explanations and arguments (Thagard *et al.*, 1990).

Analogical inference is a process that is commonly accepted to have two stages: retrieval and mapping. Retrieval refers simply to the activation of certain knowledge structures, say solution to a similar problem and the unsolved problem at hand. Mapping then connects these source and target structures together, if possible. Of these phases, mapping is the most essential. At least three types of constraints affect the mapping process: structural, semantic, and pragmatic. Several, highly sophisticated computational models of analogical mapping exist, but they are not discussed here (see Hummel & Holyoak, 1997 for details).

What makes analogies especially interesting for design problem solving is their connection to creativity. For example, in the history of scientific discovery, several remarkable breakthroughs seem to originate from a visual analogy (Klahr & Simon, 1999; Weisberg, 1986). An interface to analogical thinking should therefore be included in a cognitive model of conceptual design, because it is likely that design can benefit from analogical thinking in many ways. Next section will introduce the creativity research to see if it has any other connections with problem-solving or conceptual design.

Creativity research

> “Creativity has long been a topic of interest to educators, artists, and historians of science. Until recently, however, it has not been a subject of serious study among cognitive scientists and experimental psychologists. It has been largely regarded as unresearchable, for two reasons”
> --- its unscientific connotations and experimentally unapproachable nature
> (Finke et al., 1992)

The previous section presented some ideas about problem solving, but avoided treating it as creative behavior. However, it is common to see terms such as innovation, creativity, and so forth coupled with conceptual design. It would thus be a mistake not to address this connection in a review that considers this activity. Thus, the cognitive creativity research will be familiarized briefly. This presentation is only a fragment of the creativity research (see Mayer, 1999; Mumford, 2003 for recent reviews).

Psychological theories of creativity

As quoted in the opening of this section, the creativity may appear as a too challenging target for scientific research. For that reason, in order to investigate it, it has become favorable to use a model of the creative process (Weisberg, 1986) and decompose the process to independent parts, which are then studied separately with psychological methods. There are some psychological theories of creativity that have been experimentally tested (Runco & Sakamoto, 1999). The first two suggestions, RAT (Mednick, 1962) and SIAM (Nijstad, 2000), are comparable to the knowledge-based methods discussed in the problem-solving section. More specifically, they assume that new ideas are based on novel combinations of existing knowledge and generated by the common cognitive structures. They also emphasize the role of LTM associations as the controllers of creative behavior. Their contribution is the idea that simple properties of a cognitive structure can explain differences in a complex behavior.

GenePlore model

There has been recently an effort called Creative cognition, which has studied creativity with the methods of cognitive science. Its main product is a model of creativity called the GenePlore model. GenePlore describes creativity as a process with two phases: generative and explorative. The generative phase is described as being the source of mental representations of an idea. Several types of mental representation are possible, but the GenePlore mainly relies on special “preinventive structures”, which are an abstract class made of several different types of mental representations. Exploration is a process that interprets and examines preinventive structures to see how they fit the posed problem. Exploration can lead to a new cycle of preinventive structure generation, until the task constraints are satisfied and the final creative structure has been generated. The model includes several processes and structures that have been presented in the psychological literature, for instance, analogical reasoning, categorical reduction, and conceptual synthesis. Another finding within the Creative cognition framework is a phenomenon called “structured imagination”, which refers to structural tendencies found in creative production experiments. The idea of structured imagination is that previously acquired knowledge tends to guide imagination in a predictable way. (Finke et al., 1992)

The main lesson from GenePlore is that creative work proceeds in discreet phases and cycles. Several different processes and types of representation may contribute to a final product. Therefore, the forecoming model of design cognition should be compatible with the GenePlore framework and in a similar fashion, connectable to a multitude of cognitive processes that support creative work.

Design research

> The engineer’s main task is to apply his scientific knowledge to the solution of technical problems and then to optimize that solution within the given material, technological and economical constraints. (Pahl & Beitz, 1984)

Design is an act that engages a designer, or a design team, and requires a considerable effort in time and money. The above quotation gives a purpose for investigating design as engineering is built on knowledge. The definition could be extended by saying that design is about the development of artifacts and services for *a purpose*. This would better capture the idea of design as a profession (see Ulrich & Eppinger, 2003). However, there is no single design discipline, but instead several (see Figure 1) and major disciplines break up into specialties such as environmental architecture or mechanical engineering (Schön, 1995). Some believe that design in its varieties can be studied as a single subject (Goel, 1995). However, the different design disciplines are somewhat remote and this thesis focuses particularly on product design that is a central activity in both mechanical engineering and industrial design.

Propositions from the other fields of design will be considered, but the cognitive model of conceptual design under preparation will be about conceptual design of products and not on generic design.

Design process

The development of products is commonly portrayed as a design process. The design process can be conceptualized in many ways. One of the most common lines of attack is to present it with cascading, waterfall models that display the development process from the start towards the end. They take the perspective of either an organization or of a developer. One developer-centered formulation is presented in Figure 4 and an alternative, organization-centered can be found from a textbook by Ulrich & Eppinger (2003, figure 1-4).

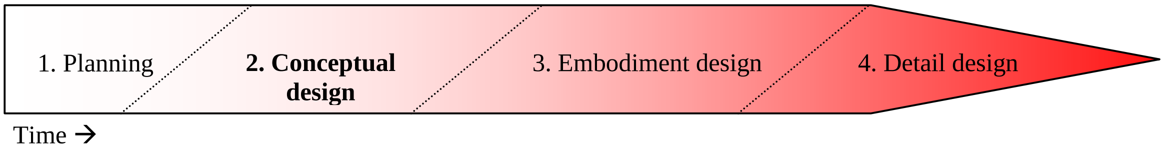


Figure 4. The phases of product design process according to Pahl and Beitz (1984). Phases are not discreet, but overlapping possibly to even greater extent than what is shown in the figure.

Of the several phases, the current interest is in the conceptual design phase, sometimes called concept design or concept development. It has the greatest effect on the cost and quality of the final product. It generates product concepts that are sort of prototype products, which are to be used as models in the following phases of the design process. A product concept is an unrefined idea that lacks details of construction materials, exact dimensions, price, and so forth. The usual presentation format for concepts is visual, possibly accompanied with corrective text passages. (Ulrich & Eppinger, 2003)

Concept design is not an indivisible chunk; it can also be analyzed into individual phases. This is illustrated in Figure 5. From the designer’s point of view, the work begins from identifying essential problems based on a general specification of the product. As the problems have been clarified, the designer(s) should decompose the problem to establish a function structure for the product

(Pahl & Beitz, 1984). Next, solution principles are sought for each subfunction included in the function structure. The search for solution principles requires that the designer can create novel solutions based on her previous knowledge and external sources. This emphasizes the fact that the search has both an internal and an external component (Ulrich & Eppinger, 2003). After generating the principles, they are combined and the combination the product concept is evaluated (Pahl & Beitz, 1984).

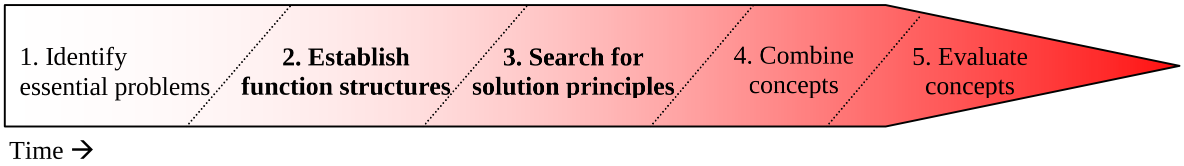


Figure 5. The phases of conceptual design process aligned in time as presented by Pahl and Beitz (1984).

Although the main objective is to find *a* product concept, several ideas should be explored to discover all alternative possibilities. The invention of a good solution is essential for a successful design, as it has been shown that initially selected concepts are rarely abandoned, instead they are refined and modified even if they prove problematic (Cross, 2004). A general description of the design process has been now provided. A more detailed account can be found from the textbooks (e.g. Pahl & Beitz, 1984; Ulrich & Eppinger, 2003). The following subsection will consider how an internal search could be realized in a cognitive structure. For this purpose, a framework of design as problem solving will be presented.

Design as problem solving

Both psychologists and designers regard design as problem-solving (Goel, 1994; Pahl & Beitz, 1984; Simon, 1996).This section presents a cognitive problem-solving model of the internal search phase in conceptual design. For this purpose, the relevant texts will be referenced. The description will remain on a textual level, even though some authors have proposed an algorithmic presentation format (see Ball *et al.*, 1994; Jeffries *et al.*, 1981). In this study, design problem solving will be discussed from the perspective of “generic design" (Goel, 1995), to include cognitive theories about several design disciplines.

The problem-solving view of design has been developed by independent authors working in different design domains. Thus, no coherent and detailed design theory exists yet, although several approximations have been presented (Akin, 1986; Simon, 1996; Smith & Browne, 1993). During the last few decades, design research has been balancing between artificial intelligence and psychological models of design (Chandrasekaran, 1990), partially at the expense of the latter. This means that there has been more interest in developing computer-aided design (CAD) systems[6] and designing AI to facilitate the design work than in understanding the cognitive processes of human designers.

Background

Most cognitive design theories are based on experimental work. However, all researchers that use an experimental psychological method must exclude some variables in order to control their experiments. Often one such variable is the effect of group work (social context) on idea generation. Group work during concept development is considered to be a normal practice in industrial environments. Group dynamics are one of the primary areas of interest in social psychology whereas cognitive psychologists, and this work, concentrate more on cognitive processes of an individual. Even though group experiments might be ecologically more valid, many researchers make use of individual subjects (Potter & Balthazard, 2004). This can be justified by arguing that "…ultimately, the foundation of any group's productivity is the thinking of its individual members" (Potter & Balthazard, 2004). Also, it has been shown that designers who work alone or in pairs produce are more productive than groups (Diehl &

---

[6] The term CAD is used in this work in a broad sense to refer to all computer systems that can be used in design.

Stroebe, 1987), so it is assumed that experimenting with individual subjects is a sufficient method.

The problem-solving theories that will be discussed concern the cognitive *processes*, but they must consider *knowledge* representation as well. It is impossible to design a nuclear plant, an oil tanker, or a computer processor without a massive amount of knowledge. The psychological literature contains several ideas on knowledge representation (see section 2.2). Cognitive design research has not yet made significant advances in this area compared to the work involving CAD and design AI. However, a few propositions exist.

It has been proposed that design knowledge is stored as design prototypes, i.e. schemata, which contain assorted, but mostly declarative, knowledge (Gero, 1990). It should be noted that the term schema is also used to describe the control structures that guide design problem solving and consist of procedural knowledge. It is must hence be assumed that schemata of a varying abstraction level exist to support different types of problem-solving tasks (Ball et al., 1994). Possibly the most extensive discussion on the subject was provided by Akin (Akin, 1986), who considered several knowledge types in design context. This work will follow the approach taken by Ball and associates, and will use schemata and declarative knowledge as components of a design cognition model. The model of declarative knowledge is the one adapted from LT-WM theory (see section 2.2). According to this model, complete designs or product concepts are blends of some particular subsolutions, but have their own identity, such as the Tequila Sunrise.

Problem interpretation

The design process begins with the interpretation of the problem. This part is sometimes called establishing a problem space, forming a mental problem representation of the problem. This process is poorly defined in the psychological models of design. It is still an absolute necessity for all problem-solving models. This is because problem-solving methods require a problem representation, which presents the main problem decomposed into manageable subproblems (Smith & Browne, 1993) (Ball & Ormerod, 1995). So for now it must only be assumed that the designer somehow interprets the task assignment and understands what the main objective is. This process should definitely be clarified in the future.

Recognition

Problems can be solved immediately after the interpretation of the problem by recognizing a suitable solution. The problem-solving literature has established the concept of a (direct) recognition method, which refers to the instant retrieval of a solution after the problem interpretation (Ericsson & Simon, 1984; Klahr & Simon, 1999; Newell & Simon, 1972). This means that recognition is a very simple memory-based operation that can solve problems. Recognition is used often in design, as will be argued later on. Recognition works by matching a problem to the solution database and succeeds if an adequate match is found. Details of this mechanism are not considered here (see Anderson, 1983; Klahr & Simon, 1999)[7].

The application of recognition is restricted; it works well only for constrained problems that have a good match between the problem and the solution. Other methods are therefore required for resolving the problem. Control structures and solution search methods will thus be considered. However, the main problem must be divided first as the control structures work on several subgoals. Decomposition is therefore needed.

Decomposition

Decomposition is a weak method that also designers use to cope with the complexity of ill structured problems (Simon, 1996). Decomposition is present in the design literature in many ways. The conceptual design itself is a product of decomposing the design process to the more manageable subproblems of conceptual development and detail design. Several decomposition methods are also a part of the normative design theory. The educational literature puts most emphasis on *functional decomposition* that provides a basis for other design theories and methods (Pahl & Beitz, 1984) (Ulrich & Eppinger, 2003). Functional decomposition begins with a “black box” (simplified) representation of the main function, which is then divided into several subfunctions. There is usually no single right decomposition for a device. Two additional methods are also mentioned: *decomposition by sequence of user actions* can be used with simple technical functions that involve a considerable amount of user interaction. *Decomposition by customer needs* is applicable when user requirements are critical for the product (Ulrich & Eppinger, 2003). No empirical studies were found concerning the prevalence or effectiveness of these methods. However, these methods should have an impact on designers because the methods are a part of their formal education. From a psychological perspective, all techniques seem equal in terms of output - one problem is transformed into a group of others through a deliberate act.

Despite the remarkable presence of decomposition in design it is a fairly neglected topic in the psychology of design. This gap has also been noted in the literature (Akin, 1986; Ho, 2001), and the existence of this gap may reflect one “assumption of rationality” of design sciences (cf. Schön, 1995). In comparison, in computer sciences, and in the development of CAD tools, decomposition is

---

[7] The term recognition is also used in cognitive psychology to signify a particular type of task in cognitive memory research (see Raaijmakers & Shiffrin, 1992). In this work, they are considered to be unrelated, although this may be debatable.

still a timely topic[8]. However, decomposition or a functionally similar process is mentioned in most psychological conceptualizations of design. Authors use labels such as analysis (Schön, 1995), factorization (Newell & Simon, 1972), partitioning (Gero & McNeill, 1998), structuring (Goel & Pirolli, 1992), successive refinement (Anderson, 1983), chunking (Akin, 1986), break-down (Pahl & Beitz, 1984), or occasionally decomposition (Smith & Browne, 1993) when referring to this procedure. Sometimes it is considered to be a part of problem interpretation and not a separate process (Simon, 1996). This shows that the concept clearly is ill-defined.

What is known about decomposition is that it is a domain-independent tool for splitting problems into a set of more manageable ones. The rationale is that decomposition reduces problem complexity by limiting the problem space, i.e. ruling out some portions of the whole, so there is smaller space to start with (Simon, 1996). Decomposition makes the assumption that the problem can be solved in independent or semi-independent parts, i.e. it is decomposable. This may not be always the case. For example, you can complete a jigsaw by starting from the edges, but you cannot first read a book and then borrow it and return it to the library in one go. Even though this be an optimal solution (it would save you a visit), the order of the processes is fixed and thus cannot be decomposed. Simon talks about this property when referring to “nearly decomposable” design problems. Another formulation of the same property is to describe subproblems as “leaky modules.” It states that subproblems are loosely connected, so that the interconnectivity can be ignored when solving a single subproblem (Goel, 1995).

---

[8] In a database search of the ISI Web of Knowledge conducted on 12.8.2005, over fifty relevant articles were found with the keywords “sub-problem”, “decomposition” and “partitioning” in engineering and computer science journals published from 2004-2005, but none in psychological journals.

How does decomposition operate? Thus far, it has been assumed that decomposition is based on universal, global methods (Newell & Simon, 1972; Laird *et al.*, 1986). However, some see decomposition more as the flipside of formerly learned solutions. When learning solutions, problem subgoal structures might also be acquired (Egan & Greeno, 1974). The studies of software design show that the experienced programmers produce more explicit, successful, and elaborated problem decompositions than the novice (Jeffries et al., 1981) (Ball & Ormerod, 1995). This demonstrates that experience has an effect on decomposition and that the effect is probably task-specific, even though the actual balance is unknown. Because decomposition seems to be a tool that can be learned and applied intentionally, this method could be called explicit decomposition. The adjective explicit is used to indicate that usually weak methods are automatic, non-conscious actions. However, it has been proposed that there could also operate an implicit form of problem decomposition (Ho, 2001). The psychological mechanisms of decomposition are currently unknown. One possibility could be that decomposition is based on a partial recognition (retrieval) of a solution that does not solve the problem at hand, but provides an adequate functional structure. Alternatively the decomposition could be a product of inference based on the same information, which resembles the function of AI expert systems.

As a result of decomposition, the problem solver possesses some subproblems that are referred to as a *goal stack* or *tree*, a *fundamental list*, or *a problem agenda* in cognitive models and a *functional structure* in design literature (Anderson, 1987; Ball et al., 1994; Egan & Greeno, 1974; Ho, 2001; Pahl & Beitz, 1984). The functional structure and the other propositions are very similar. Issues such as subgoal priorities and subgoal representation are considered in computational models, but are uninteresting for the current work (see Egan & Greeno, 1974). More important is the fact that subgoals, along with the main goal, must be maintained throughout the problem-solving process (Ball & Ormerod, 1995). The upkeep of the goal stack can pose a problem for the problem solver. Because it is assumed that the goal stack is stored either in STM or in LTM, problem solving can be hindered due to their limitations. STM capacity is also required for actual problem-solving operations and its overload may result in forgetting some subgoals. The outcome will be same if LTM storage or recall fails (Anderson, 1983; Egan & Greeno, 1974).

Control structures

Control structures administer how subproblems are solved. It is assumed that the control structures are very general cognitive processes or schemata and they apply strategies, which are also schemata, for performing different cognitive functions. Control strategies can be characterized by several criteria. First, there is a basic distinction between the top-down and the bottom-up strategies. The top-down method begins with a problem (an initial state), whereas the bottom-up approach begins with a solution (a goal state). These modes are sometimes called the goal-driven (working-forward) and the data-driven (working-backward) search modes (Anderson *et al.*, 1981; Ball & Ormerod, 1995). It is generally agreed that design is a top-down controlled task, as it would be inconceivable to begin the search from a solution which is initially inexistent (Smith & Browne, 1993). Secondly, it has been considered whether solutions are developed using a depth-first or a breadth-first control strategy (Akin, 1986; Ball & Ormerod, 1995). The operation of these styles is depicted in Figure 6.

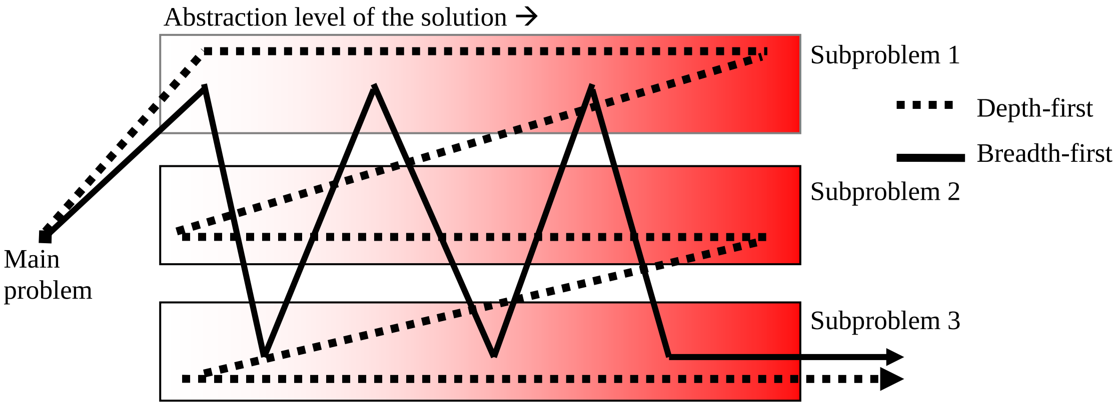


Figure 6. Two alternative control strategies for searching solutions. The figure shows a design problem consisting of three subproblems. The arrows present how a depth-first and a breadth-first strategy would proceed to solve the problem.

The breadth-first strategy portrays the normative model of design in which all the subproblems are solved at one level of abstraction before getting into a more detailed level. This supposedly most effectively avoids the problem of interconnectivity. The alternative schema is the depth-first search in which individual subproblems are deciphered to a detailed level before considering other subproblems.

The question of how and when these strategies are actually applied has been empirically studied. The results from different design disciplines indicate that expert designers adopt controlled, top-down and breadth-first strategies whereas novices use more unprincipled, depth-first approaches. However, many investigations have shown that designers usually employ both strategies with some opportunistic deviations (Ball & Ormerod, 1995; Ho, 2001). In explaining related findings, Goel describes a limited-least-commitment strategy (LCM), which remarks that subsolutions may be developed to any level of detail and then be put on hold. LCM is thus similar to the depth-first strategy, but used by expert designers (Goel, 1995). This finding increases the uncertainty related to the notion of “generic design” as control strategies seem to differ between disciplines.

In brief, the main purpose of control structures is to select which subgoal should be solved next and this is achieved with a control strategy, which is realized as a schema in human problem solvers. Actually, the control structures do not have to initiate a solution search for each subproblem - instead they may iterate the problem further. As decomposition is not limited to any detail level, the designer can become very specific on every nut and bolt on an oil tanker, as new subproblems can be decomposed over and over. To avoid infinite recursion during the process, a stopping-rule is required (Goel, 1995; Simon, 1996). After the solution search has generated some results, a decision-making process is required to assess the suitability of alternative designs. After having now stated that there is a need for such function I will deliberately ignore it, as it poses one more issue that cannot be considered in appropriate depth.

Solution search

The general solution search methods of problem solving discussed in section 2.3 can be applied to design problems with some modifications. For instance, the weak and strong methods are called analytical and knowledge-based strategies in design (Smith & Browne, 1993). They are both informed methods, and the uninformed generate-and-test (GAT) method has, to the author's best knowledge, not been discussed in the design context. The reason is clear. Design problems are ill structured and they lack information about the operators are needed to generate solutions. Gaining information about the operators is an essential part of solving an ISP. Therefore, GAT is not possible with design problems. When the generate-and-test method is mentioned in the context of design problem solving, it simply refers to using existing knowledge to generate solutions (Akin, 1986).

The weak methods in design have not been comprehensively investigated (cf. Smith & Browne, 1993). In the remotely related studies of scientific discovery, analogical mapping is considered to be one significant weak method that combines the analytical and knowledge-based method (Klahr & Simon, 1999). Preliminary studies show that analogical mapping is also used by designers (Ball *et al.*, 2004). Strong, knowledge-based methods have received more attention.

Looking up answers from LTM requires a memory search that can be characterized with some general memory model, such as SAM or MINERVA (Raaijmakers & Shiffrin, 1992) or a specialized one such as SIAM or CuPRIG (Nijstad, 2000; Perttula & Liikkanen, 2005). The related studies of creative cognition have successfully adopted some features of general memory models to explain retrieval (Finke et al., 1992). It is also possible that solutions are sought from EM instead of internal memory. This issue has been widely investigated and it has been shown that resorting to EM may have both positive and negative effects on a creative task such as conceptual design (Jansson & Smith, 1991; Nijstad *et al.*, 2002). Referring to earlier discussion about decomposition, explicit decomposition should also be considered as a domain-dependent strong method. Other strong procedural methods include idea generations methods such as Brainstorming, the Gallery method, and TRIZ (Ulrich & Eppinger, 2003). These idea generation methods can work in different ways. They can usually be interpreted as ways to influence the memory search or provide new methods for manipulating retrieved ideas. This means that there are more like meta-level methods than direct methods.

Cognitive model of conceptual design

The given details about design problem solving together with the general problem-solving theory and the psychological conception of creativity form a working model of internal search in conceptual design. The model presented in Figure 7 begins with problem interpretation, immediately followed by a recognition attempt. If recognition fails, problem is implicitly decomposed and the process of solution search begins. The control strategies control this search process and apply weak and strong methods to modify and solve subproblems in a recursive manner. Typically subsolutions are found using recognition or memory search. In addition to design specific weak methods, domain-independent weak methods are also available, like the aforementioned analogical mapping and explicit decomposition, and possibly others, such as conceptual combination (Costello & Keane, 2000) or idea generation methods (Shah *et al.*, 2000). Using explicit decomposition will likely lead to a new cycle of solution search utilizing a new subgoal. Note that an evaluation process and stopping-rules are mandatory for controlling this recursion.

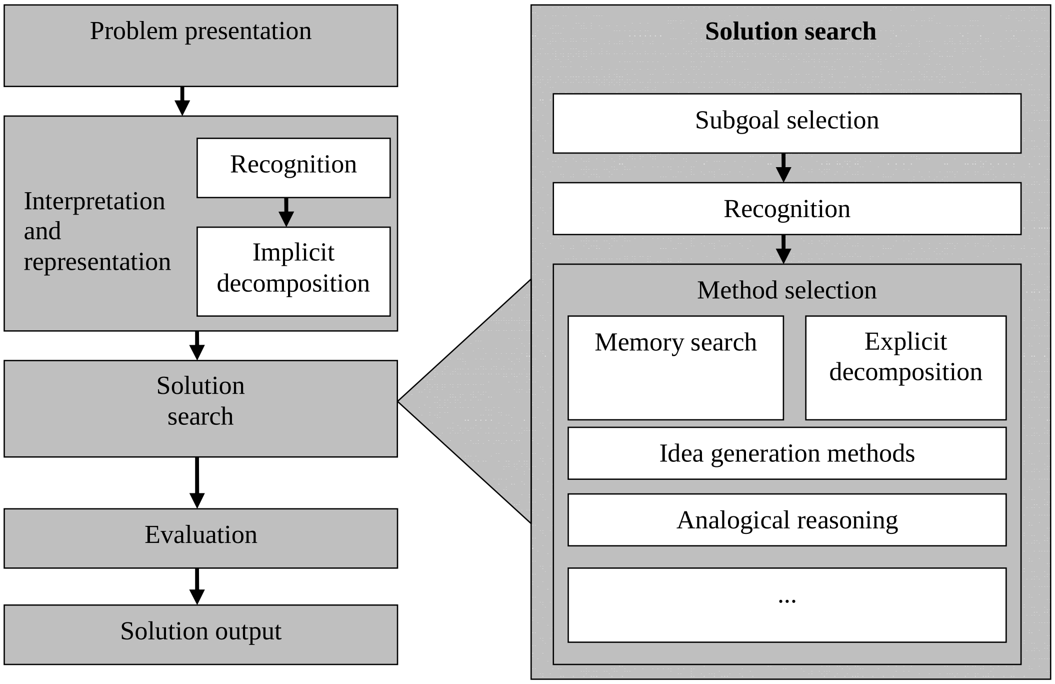


Figure 7. A cognitive model of conceptual design. The main process is shown in the left column beginning from the top and details of solution search are on the right pane.

Finally, after the produced idea has been evaluated, this process can lead to output of an idea. Output in this context means that a sketch is drawn or a description of a solution is written. It is likely that STM capacity limitations may force this output after some cycles of solution search because STM capacity is limited and it is impossible to store the intermediate results in LTM. Some parts of the process maybe automatic, for instance recognition and implicit decomposition, but most processes, such as the solution search methods, require conscious effort and should therefore be available for verbalization. This modular connectivity is similar to the GenePlore model (subsection 2.4). As the model has now been presented, it is time to consider how decomposition as a module in the model, could be empirically tested.

Grounds for empirical study of decomposition

This work addresses the role of problem decomposition in the solution search phase of the conceptual design. But how can decomposition be studied? The majority of studies that examine design cognition have relied upon think-aloud methods (Cross, 2004). The think-aloud method, together with video-assisted observation, provides the most complete picture of cognitive activities during a demanding cognitive task (van Someren *et al.*, 1994). Thus, these methods will also be used in this study. Protocol analysis, a sophisticated form of the think-aloud method, requires a cognitive model of the activity under observation (Ericsson & Simon, 1984). The abstract model was already introduced, but to apply the method, assumptions about how decomposition works and how can it be found from a verbal protocol must be clarified. These suppositions are derived from two sources: the normative design literature and psychological studies.

Based on the current review, there is only one similar study that has considered decomposition in conceptual design (Ho, 2001).The study conducted by Ho observed two designers in a conceptual design task. The author identified two decomposition strategies: an implicit strategy used by the novice and an explicit decomposition used by the expert. Implicit decomposition was inferred from the structured development of the concept and the lack of any explicit decomposition in the expert's behavior. These results are in line with the presented model of conceptual design. However, as only two designers with different levels of experience participated, the results must be considered preliminary.

There are a few corresponding design studies that have included decomposition in their framework, but have focused on how the control strategies and component processes operate. Empirical work concerning programmers has constantly demonstrated explicit decomposition (Ball & Ormerod, 1995). In an unpublished study (Ullman *et al.*, 1986 reported in Ball et al., 1994) senior students of mechanical engineering were found to use structured design methods(breadth-first control structure), which implies the use of decomposition. In a recent think-aloud study, researchers attempted to directly identify some protocol segments as decomposition among other cognitive processes. Their results showed that explicit decomposition could be detected in an electrical engineering task (Gero & McNeill, 1998).

There have been some decomposition studies in the domain of well-structured problem solving, but they have mainly investigated effects of altering decomposition by experimental manipulation, instead of addressing decomposition directly, and hence they are not helpful for the current investigation (Egan & Greeno, 1974; Eysenck & Keane, 2000). The effect of explicit, instructed decomposition on ill structured (non-design) problem solving has been shown to be positive on another problem domain (Dennis *et al.*, 1996), but these results do not reveal anything about the decomposition process. Therefore, the decomposition must be analyzed further.

Theoretical basis

This study is committed to investigate individual designer in a conceptual design task using verbal protocol analysis. To question the proposed method, it can be asked, is it compulsory to use a laborious method like protocol analysis or could decomposition be studied by other means? Consider a typical experimental setup. Concept sketches and brief descriptions are the usual output from conceptual design, usually several of them. Can decomposition be observed from these documents? If the concepts present a complete design, then it is impossible to make any conclusions about the process. Even if we have several alternative sketches, or the designer has structured the concept into discreet functional parts, comparable to a functional structure, this reveals nothing about the underlying process. The designer may have retrieved the concept as a whole and formed the functional structure as she sketched the constituent parts. The result of this recognition-based process will be indistinguishable from a decomposition-based process. Alternatively, the designer can deliberately decompose the sketches to follow the normative expectations of a proper design. Thus, it is unwarranted to study decomposition by inspecting the design documents and a more informative method, such as think-aloud, should be used.

The area of study influences the data. Conceptual design by definition demands preliminary and unelaborated results, product *concepts*. Designers must therefore adapt stopping-rules that will prevent them from getting too detailed in the work (Goel, 1995). This fact diminishes the difference we could hope to observe between depth- and breadth-first control strategies (e.g. Anderson et al., 1981). That is, if the solution contains only a few levels of abstraction, it will be hard to make a difference between these two modes. This fact requires paying special attention to even small changes in the abstraction level of solution development for distinguishing these modes.

What model of decomposition should be considered? Three alternative techniques from the design literature were introduced in the previous subsection, functional decomposition being the most prominent. The designer may have adopted one or more technique and apply it when necessary. While the alternative methods focus differently on the main problem, they all produce decompositions of some kind. In the current research, it is assumed that the type of decomposition is irrelevant. The cognitive models presented at the end of previous section described two essential steps in retrieving solutions: recognition and decomposition. The latter has an implicit and explicit form. Next, criteria how they can be detected from verbal protocols will be established.

Recognition and decomposition

Recognition is a method for generating complete solutions at once. If we allow the designer to take enough time, she should be able to retrieve and output an entire design by recognizing its fit. In that case, we would expect that she would do this in some ordered, piece by piece manner reflecting the internal knowledge structure. This is a direct result of the supposed memory organization: linking separate properties to a single concept that has a particular name or identity. It is also possible that this name, e.g. Tequila Sunrise, might be verbalized in the process. This makes up two ways to observe recognition: by explicitly naming a solution at the beginning and by the fast and determined development of a concept. For example, a subject in Ho's study (Ho, 2001) recognized a general solution and responded that the solution, at general level, was an answering machine.

For decomposition, the best demonstration of decomposition would be an explicit decomposition of the problem. On that occasion, the designer would be expected to provide a description of the relevant subgoals or metacognitive reflections of this process. The most essential subgoals should be presented in a row, without interfering solution propositions. A criterion could be that, if the designer cannot create a working solution with the presented subgoals, then something essential is missing. A metacognitive reflection in this case means that the designer would state that she is performing decomposition, using some phrase. As mentioned, explicit decomposition is not temporally restricted. The process may begin with the recall of full solutions, i.e. starting recognition-based, and continue by decomposition-based generation, and so this possibility must be also considered.

There are some possibilities to detect implicit decomposition. They are mostly based on the logic that if a solution is not produced by recognition or with the aid of explicit decomposition, then it must be based on implicit decomposition. As this criterion seems to encompass almost everything, it must be restrained by requiring that a top-down control strategy must have been used to produce the solution, so the design episode is a structured work based on a functional structure.

The difference between recognition- and decomposition-based processes could be possibly inferred by the time taken to produce results. As it necessarily takes more mental operations, and hence time, to decompose a problem, generate subsolutions, and recompose the solution than to retrieve a solution from LTM (see Mayer, 1992 on mental chronometry). A practical difficulty might rise from the fact that sketching requires considerable time that will compress the time effects of the cognitive activity. For that reason, this possibility will be disregarded in the current work.

There is a fourth possibility that might show in a conceptual design experiment. For a complex problem that has several subfunctions, it is possible that the designer completely misses the key idea of the task and takes a single subproblem as the general problem. While the following problem-solving steps may resemble problem solving based on a more correct interpretation, this option would probably shape the designed concepts remarkably.

The criteria needed for the protocol analysis have been now provided and are summarized in Table 1. There are several possible limitations in the current approach, but discussion about those issues is placed to the concluding chapter.

Table 1
Operational criteria for detecting recognition and two kinds of decomposition from verbal protocols

Recognition	
1.	Concept identity statements
2.	Temporally proximate development of complete concepts
3.	Fast, structured concept development
Explicit decomposition	
1.	Metacognitive statements about decompositoin
2.	Temporally proximate goal structure statements
Implicit decomposition	
1.	Distinguishable top-down global control strategy
2.	Unsystematic subgoal statements
3.	All non-recognition based concept development

Hypotheses

The approach taken in this study is explorative. This investigation attempts to confirm the findings made by Ho (Ho, 2001), but there are some additional assumptions that can be made, as the background model is more detailed and the characteristics of the sample are known a priori (details are provided in Methods chapter). The participants are fairly experienced designers, and thus it is expected that both recognition and decomposition will be used and that the tasks will be easy to interpret correctly. The use of recognition is possible due to the relatively large knowledge base of a highly educated adult and decomposition because of the formal education. It is also assumed that the chances to use recognition will make the task subjectively easier.

1) Subjects will comprehend the task in whole, no misinterpretation based designs

2) A small number of concepts will be produced by recognition

3) Amount of recognition correlates negatively with the perceived task difficulty

Studies with practicing software designers have shown that control strategies change from opportunistic, bottom-up to breadth-first, top-down strategies as the designer gains more experience (Ball & Ormerod, 1995) and that mechanical engineers apply structured methods in the early stages of product development (Ullman et al., 1986 in Ball et al., 1994). It is therefore predicted that subjects of this study should design more in a top-down manner and would explicitly decompose the problem as they have learned and as has been previously demonstrated (Ho, 2001). In the rest of the cases, it is expected that implicit decomposition will be successful. In other words, it is anticipated that some sort of decomposition will always occur.

4) More breadth-first than depth-first approach

5) Explicit decomposition in the majority of subjects

6) Implicit decomposition in all subjects

3. Experimental method

Sixteen subjects participated in a think-aloud experiment consisting of two twenty-minute design tasks. The experimental setup was designed to provide data for several research questions, of which only decomposition is considered in this work. Despite this fact, the experiment is still described in full detail and the reader is only reminded that many features of the experiment are not essential for studying decomposition. All the participants will be referred in the text as “she” or “her” regardless of the participant’s sex.

Participants

Senior students of mechanical engineering, who should be familiar with decomposition, were recruited for the study. The final sample was selected on the basis of ecological validity and convenience. The volunteers were sixteen ($N = 16$) predominantly male ($N_{male} = 15$) student at Helsinki University of Technology. The mean age of the sample was 26.8 years ($SD = 2.4$ years). All the participants had completed design related studies having a mean of 137 credits ($SD = 42$) of 180 required for Master’s degree and possessed some design working experience ($M = 0.8$ years, $SD = 0.7$ years). Everyone spoke Finnish fluently, even though three bilingual subjects were included in the sample. In exchange for their time, the participants were awarded a movie ticket. Expected rewards have been shown to significantly boost the productivity (e.g. Nijstad *et al.*, 2003), and therefore, to motivate the subjects, they were informed that the best designer in each task would be awarded with an extra movie ticket. Before the experiment, all participants gave a written consent prior to their participation and were briefed regarding the purposes of the study. They were also informed about the ethical guidelines of research material usage.

Procedure

This study used a factorial within-subjects design. Each participant completed two conceptual design tasks, called Plant and Forest. These tasks had one independent manipulation, the existence of the examples. This variable consisted of an examples and no-examples condition. The order of the tasks and manipulations was counter-balanced between subjects. This resulted in four different procedures, each shared by four subjects, although in this setup, one subject provided data for two conditions. Only the data from the no-examples condition is included in the current study. The study was sequenced to ten phases, which are listed in Table 2 and described in the following subsections.

Table 2
The order and durations of the briefs and tasks

#	Phase	Time in minutes
1.	Introduction	5
2.	First questionnaire	2-3
3.	Talk-aloud example and practice task	3-4
4.	General brief	1
5.	Written brief for Task #1	1
6.	Task #1 and its evaluation	21-24
7.	Practice task continued	1-2
8.	Written brief for Task #2	1
9.	Task #2 and its evaluation	21-24
10	Concluding questionnaire	2-3
	Approximate total	1h 5 min

Phases

During the introduction subjects were informed about the general aims of the study and notified what sort of data would be recorded. They were also told how the experiment would proceed and how they would be rewarded. Participants filled in two questionnaires. The initial questionnaire documented subjects' background information (age, sex, design experience etc.) and alertness before the test. The concluding questionnaire assessed the subjective experience and domain knowledge related to the tasks. Both questionnaires are included in Appendix A.

The application of the think-aloud method for inexperienced subjects may be difficult. Hence, it is usually suggested to give the subjects an unrelated training task for practicing the talk-aloud (van Someren et al., 1994). 15-puzzle, a variation of 8-puzzle, was constructed for this purpose. The puzzle has been used previously on several problem-solving studies see (e.g. O'Hara & Payne, 1998). The puzzle is shown in Figure 8.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Figure 8. 15-puzzle contains fifteen numbered tiles an empty block. The figure displays the goal state of the puzzle. The initial state of the puzzle is a random arrangement of tiles. The puzzle is solved by moving one tile a time to the empty block to reach the goal state.

The practice was arranged so that the experimenter first turned on the video camera and then began to think-aloud while solving the puzzle. After a few moves, the task was transferred to the participant, who continued the task. While solving the puzzle, the experimenter constantly encouraged the participant to voice out her ongoing thoughts. After a couple of minutes, as the participant seemed to comprehend the talk-aloud method, the work on the puzzle was postponed and the next phase, general briefing, began.

The task administration began with a spoken, general task description that covered both tasks (see appendix B). It enclosed the general aim of the conceptual design and stated the objectives of this experiment; the need for quantity and variety of idea production together with the deferred criticism of solutions. These guidelines are a modification of the Brainstorming principles (Osborn, 1957), although the reuse of previous ideas was not mentioned here. The intended presentation of a concept was defined as a sketch; a free hand drawing, supported by explanatory words. The subjects were earlier trained to think-aloud and were now encouraged to do so during the tasks. They were also told to refrain from elaborating their concepts and also to limit their textual descriptions to minimum. They were informed that the experimenter would not answer their questions during the task, and were therefore suggested to present their inquiries prior to tasks. After that, the subjects received answer sheets for presenting concepts and the first task brief. When the first task was finished, the subjects continued to work on the puzzle for one or two minutes in order to wipe out the previous task from their short-term memory.

Tasks

Two novel and realistic design tasks were invented for the study, called Forest and Plant. The Forest task required a design of a tree trunk removal apparatus and the Plant task was about an automatic plant watering device. The tasks were designed to be equally challenging, but the possible difference of difficulty between two tasks should not be critical for a study of decomposition, as the previous design studies have even employed different tasks for each subjects participating in the same study (Ball et al., 1994; Goel & Pirolli, 1992).

The task brief was presented to the subjects as a paper printed on both sides. The task description was on the first side and the manipulation on the reverse. The description (Appendix C) consisted of a short paragraph describing the current situation in the problem domain and the design assignment at a general level. The manipulation on the reverse side contained four example concepts in the examples condition and a schematic figure of the trunk or the plant in the no-examples condition. The schematic figure was an extract from the example concepts (see Appendix C), but it is considered to be irrelevant for this study.

Both task briefs were given in textual form to subjects and their comprehension was confirmed before starting the task. The time for reading the brief was unlimited. After reading, both tasks were allowed twenty minutes timed from the flipping of the brief pap and participants were informed approximately two minutes before the end about the remaining time. During the twenty minutes, subjects drew concepts on the given sheets, two concepts on a sheet. If necessary, the subject was requested to pile the finished concepts on her other side, so that they were not directly accessible during the task and that their possible later inspection and manipulation could be captured on the video.

The experimenter avoided interaction with the subjects. If subjects, despite the instruction, asked questions during the tasks, the experimenter either answered by quoting the task brief or the general task description. In other cases, he stated that he cannot answer the question. If the subject stopped verbalizing at some point the experimenter encouraged her to continue by saying, for instance, “please continue thinking aloud”, “what are you thinking”, or “what’s on your mind.” This is a standard procedure within the paradigm (Ericsson & Simon, 1984).

After completing a task, the subjects had to briefly evaluate their sketches on a scale from five to zero according to their “general quality and creativity” and to present some supporting arguments. No time limit was given for this procedure, but the experimenter encouraged the subjects to be prompt. During this phase, the subjects were also asked to clarify their sketches with text descriptions, if the experimenter could not otherwise identify the solution principles.

Analysis of behavior

This study applied the think-aloud method together with video observation for gathering data. Think-aloud is a method that has been used for over a century in psychological research (van Someren et al., 1994). Think-aloud produces verbal protocols that are analyzed to investigate processes behind the verbal behavior with a method called protocol analysis. Protocol analysis can be applied to the study of cognitive processes to gain information inaccessible with other behavioral methods. A more recent variant of the talk-aloud method is verbal analysis, which stresses more the knowledge structures than cognitive processes (Chi, 1997). Features of both methods were incorporated in the current exploratory experiment. Think-aloud can be applied in two ways, as concurrent or retrospective verbalizing. In general, concurrent verbalization is a more reliable data source than retrospective protocols (Ericsson & Simon, 1984). Previous studies have not revealed significant differences between concurrent and retrospective protocols of design (Gero & Tang, 2001) and thus this study also used concurrent verbalizations. In addition to verbal protocols audio-video recordings and sketches were used in this study in order to provide more correct interpretations of the designer's cognitive activity. Verbal protocols alone provide a deficient data as designers habitually use indexical (or deictic) references (Dix *et al.*, 2004). This makes it impossible to understand the design process by using only the audio recording as the targets of the reference are absent.

Data preparation

The experiments were captured on a MiniDV recorder. The recordings were then transferred to VideoCD format with high picture and sound quality (sound sampling rate 48 kHz). The transcription of verbal protocols from the videos was done using the Subcreator software and the resulting text files were imported into Excel spreadsheet with the aid of a VBS macro for coding the protocols. After this procedure, the data was exported back into text format. The codes were later summarized with scripts written in PHP scripting language. Eventually, all numerical data from the experiments were imported into SPSS statistics software for analysis.

In addition to the transcribed protocols and video recordings, the product concept sketches were prepared for the analysis. Initially, concepts were simply calculated and numbered. This numbering is referred to concept identification number and it was later on joined with the protocols. While this should have been a simple operation, as the answer sheets were numbered beforehand, some exceptions occurred. All concepts drawn on the sheet were calculated without discarding any concepts. However, if the subject had drawn several concepts on a sheet that was deserved for a single concept, the ideas were calculated as separate concepts.

Coding verbal protocols

Coding schemes

The segmentation and the coding of protocols are the two most significant parts of the analysis. Initially, several schemes were considered that were all partially compatible with the cognitive model (Akin, 1986; Gero & McNeill, 1998; Kavakli & Gero, 2002; Suwa *et al.*, 1998; Goel & Pirolli, 1992). Akin has explicitly described how control structures can be traced by using problem description graphs (a PDG, see Akin, 1986). A PDG contains a description of design abstraction levels that are also found in the works of Gero and associates. Eventually, a modified version of the coding scheme used by Gero and McNeill (1998) was selected in order to obtain a representation comparable to a problem description graph. It will be described next.

Most of the aforementioned schemes have included codes for several cognitive processes, e.g. for proposing, analyzing, evaluating, and synthesizing solutions (Gero & Tang, 2001). As this study is about describing how designers decompose a problem it was seen unnecessary to denote the segments with a process code. Instead, a general decomposition code (DEC code) was included to mark segments that contain explicit decomposition.

The coding was done in two phases. First general schemes were abstracted and then the protocols were coded using these schemes. A general *problem decomposition scheme* (PDS code) was abstracted for both tasks, corresponding to a functional decomposition of the main problem (see section 2.5). A PDS contains four abstraction levels: the main function, solution types, subfunctions and subfunction solution principles. For solution types, there are two options in relation to the task brief. The rationale is that most designers stick to the problem statement, but in a few designs, they disobey this constraint and propose qualitatively different solutions (see Table 3 below). Each device has several subfunctions, also called subgoals or subproblems where each subfunction can have several solution principles, some of which maybe shared between two subfunctions.

Table 3

Problem decomposition schemes. This table includes only the main function, solution types, and subfunctions. Solution principles are omitted. The prefixed number is the PDS code

	Plant
Main function	0. Plant watering device
Solution types	1. Devices attached to pot
	2. Independent devices
Subfunctions	3. Water supply
	4. Regulation
	5. Transfer
	6. Mediator
	7. Energy source
Principles	[a]

	Forest
Main function	0. Trunk disposal
Solution types	1. Integrated devices
	2. External devices
Subfunctions	3. Destruction
	4. Removal
	5. Dispersal
	6. Recovery
	7. Attachment to machine
	8. Mediating mechanisms
Principles	[a]

Note

[a] Principles are listed in the appendix

The problem decomposition schemes were created using the protocols bottom-up, so that all protocols and sketches were first inspected and generalized into one scheme that would encompass all solution variants (see van Someren et al., 1994). All solution principles that were considered functionally relevant were included regardless of their commonality, i.e. observed frequency. The relation of some particle principles and subfunctions can be questioned, but it is indifferent for this study, as the main interest is in the abstraction level of the solution statements. Although it is likely that this generalized scheme will clash with some individual protocols, it was still considered to be a better option than devising customized schemes for each designer (cf. Gero & McNeill, 1998). The general problem decomposition schemes are presented in Table 3 and the detailed versions of both schemes are available in Appendix D.

The most important detail in the schemes is the difference between subfunctions and solutions. They present two levels of abstraction: immaterial subfunctions and concrete principles. Subfunctions are the functionalities that together compose the main function. It is assumed that they a correspond to the subgoals of the goal stack, which makes their existence an indirect indicator of decomposition. The solution principle codes, numbered from thirty to eighty, are listed in Appendix D. Their intention is to identify a particular solution adapted for a subproblem, e.g. an electronic timer for regulating water flow or a corkscrew for removing a trunk. These codes are not very specific; instead of nominating one very distinctive principle, they represent a *solution category* similar to radial categories discussed in 2.2. For instance, a water regulation solution called container degradation (number 48 for Plant) could refer to a degradation caused by water, sun, or oxygen. In coding the Forest task, the coding rules were relaxed so that if a solution principle code from another subfunction category could be used if it served the same function. For instance, a mechanism that destroys the trunk using a chemical was classified as number 35, even though the designer had included a drill which would normally be classified as #41 or #56. As a consequence, this flexibility makes it more difficult to trace the control strategy, but keeps the PDS simpler.

The schemes exclude some solution principles that were evaluated less significant, but were used by some subjects, usually related to the alternative solution type (PDS code # 2, Table 3). Also, the propositions that redefined the target (the plant or the trunk) or its immediate surroundings were ignored, as they were considered to be functionally indifferent. In the Forest task, designers sometimes decomposed a solution principle, which had been dedicated a single code, e.g. by stating that exploding a trunk was a three-staged process. In these cases, an appropriate code for each solution principle was selected, even though this contradicts the basic rationale of the scheme it allowed to identify decomposition of a subproblem in this case.

During the observations at the time of experiments and in the bottom-up analysis of the protocols, it appeared that the designers were very solution-oriented. It was noted that the designer may never state a problem, but just introduces a solution. While this makes no problem for coding solution principles, it has an effect on subfunctions. It was hence acknowledged that a subfunction can be referred to in an enquiring voice or as a general solution statement. For instance, the designer of a trunk shredder could say “what could be the shredder like” or “and then we have some sort of a shredder” when referring to the same subfunction, the dispersal of the trunk. Therefore, the subfunctions presented as problems were coded with a question mark post-fixed to the PDS code.

Coding rules

A PDS was given to all segments that explicitly demonstrated idea development. If a segment contained several solution statements, the segment was split in two and PDS code was dedicated to each separately. If some solution detail seemed to serve several purposes, only the main function of the solution statement was considered and possible secondary functions were discarded e.g. a conveyer belt for plants was considered to be primarily a mean to move plants, rather than to regulate the transferred water. Segments that contained literal or semantic repetition were left out if they were temporally close, i.e. uttered during the development of a single concept (see Ericsson & Simon, 1984).This rule excludes also the cases where the subject had first vocalized an idea and then repeated it while drawing. However, if *alternative* principles falling into the same solution principle category were presented in a row, they were given separate PDS codes. In a situation, where the same principle was merely elaborated, e.g. the designer considered the water basin details very carefully, only a single code was used.

There were also segments that introduced new solution principles, but these principles were not implemented in any of the documented ideas. These segments were also included in the analysis normally, but their concept identification number was set to zero. Similarly, the identification numbers for the phrases that referred to earlier solutions, usually phrases about what had been done before, were also set zero. In practice, the zero was post fixed to the PDS code separated by a slash, e.g. 42/0. This procedure intends to identify the statements that advanced the design, and make it possible to trace the functional structure of each concept by selecting only those PDS coded segments that share the same identification number.

There were situations where the protocol was clearly incomplete or incomprehensible. On some occasions, a designer drew a whole concept or some solution principles without any verbal, spoken or written, reference. These design suggestions were called implicit segments and they were added to the transcribed protocol with the implicit PDS code after the last explicit segment of the concept-in-preparation and before the first segment of the following concept. To minimize false interpretations, implicit segments were identified by their main principle and all possible forms of elaboration were discarded. Also situations where a designer had jotted down a principle instead of a sketch occurred. They were inserted to the verbal protocol as implicit segments and given an appropriated PDS code. To sum up, all written and spoken design ideas were coded once with an explicit PDS code and ideas in figural form only were coded with an implicit PDS code.

Coding

The coding of the sixteen protocols was done after the transcriptions had been converted to a table format. In the table, each transcribed segment had a time code and four additional columns for the concept number, decomposition code, and explicit and implicit PDS codes. The details about each code will be provided next.

At the start, concept numbers were initialized by marking all segments with zero. Then the first segment clearly associated to a drawn concept was identified and the corresponding number was assigned to it. This segment was called 'concept initiating segment'. This was done for all concepts. After identifying all concept initiating segments, all segments between the two initiating segments where numbered with the number of the former segment.

Next, the explicit decomposition code (DEC), was determined. It had a Boolean value of 1 or 0 (true or false). As stated in section 2.6, for a positive DEC, the subject was required to state that she is partitioning the problem. The actual phrase could differ, but it was required to semantically match the idea of decomposition. These segments and the neighboring segments that were clearly associated with explicit decomposition were coded as true, others false.

In the following phase, segments that contained references to a solution type, to a solution principle, or a statement of the problem, were coded with a PDS code. In these segments, the subject introduced a solution, reconsidered it, or presented a problem formulation or an objective. Coding was done with the aid of sketches and video recordings. Based on the differences between verbal protocols and documented ideas, implicit solutions were also PDS coded. The implicit PDS code(s) was added to the dedicated column of the last segment of the concept in development. If there were no segments, or an insufficient number of segments, a new segment was added with the time code of the following segment. After inserting all PDS codes, all remaining segments without any PDS or DEC code were discarded from analysis. The PDS coding was carried out twice order by the author to minimize interpretation errors. After the coding, all coded protocols were checked thrice for overall coherence. The full, segmented, coded protocols are available in Appendix E in the original language.

During the last phase of the coding process, the PDS codes were used to infer the control strategy used during the development of each design idea. First, some ideas were identified as recognition-based and were thus skipped, as a control strategy was not needed in those cases. If concept development began with a recognition-based idea, but the idea was refined, it was still coded as recognition-based. Then the breadth-first and depth-first strategies were distinguished by inspecting what sort of segments made up the solution (Ball et al., 1994). This was relatively easy, as the PDS codes already contained information about the abstraction level, bigger number enumerating a more specific and a smaller number a more abstract level (see Appendix D). A breadth-first strategy was detected when a subject presented solutions to all relevant subproblems at the same level of abstraction, e.g. a solution consisted of PDS codes 30, 40, and 50. If a documented idea contained solution(s) to only one subproblem, then it was considered to be depth-first, e.g. PDS code 30, or 30 and 31. Some might argue that if several depth-first solutions were presented in a row as separate ideas, then this would indicate a breadth-first schema. For instance, PDS codes 30, 40, and 50. However, it was considered that designer's decision to sketch solutions separately is more determining than the temporal sequence in this case. Some concepts could not be classified to neither of the categories mentioned above. Although these concepts might be labeled as "opportunistic" (Ball & Ormerod, 1995), the term 'mixed strategy' is used here. A concept produced by a mixed strategy could be for example: 30,40,5,60,7. A mixture could result from leaving out subsolutions, presenting only a single solution principle, but referring to other subproblems. Identified control strategies, CS coding, were gathered to a separate table (Appendix F).

The final step was to inspect the PDS coded segments for temporal patterns, because it was assumed that explicit decomposition could occur without any particular verbal report. The criterion for explicit decomposition was that the essential subfunctions must be mentioned in a row, e.g. 3, 4, and 5. If explicit decomposition was found, these segments were marked with the positive DEC code. An example of the application of the presented coding scheme is displayed in Table 4 below.

Table 4
A real, translated extract from the complete coded protocol of the subject number thirteen. The extract displays twenty seconds and seven last segments of her protocol that contained total 227 segments. Four segments have been coded, two non-implemented, one implemented and one implicit.

Time	Concept #	DEC	Explicit PDS	Implicit 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	31/0		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

4. Results

This chapter presents results from the sixteen experiments. In each experiment, one participant completed two tasks, one with example concepts and one without. Only the data from the no-examples condition has been analyzed. Subjects who completed the Plant task make up the Plant group and those who did the corresponding Forest task are referred to as the Forest group. First, a general overview about the results is given and then the measures related to the hypotheses will be evaluated.

Overview

There are several measures that can be used to assess the produced concepts (Shah *et al.*, 2003), but in this experiment only the quantity of ideas was considered necessary. The sixteen subjects produced eighty five concepts in the Forest task ($N = 85$) and seventy one concepts in the Plant task ($N = 71$), a total of one hundred and fifty six concepts ($N = 156$). There was a considerable variation between subjects ($M = 9.8$, $SD = 3.6$ concepts). As all subjects produced at least some concepts, all data were included in the analysis. The produced concepts varied also qualitatively. The amount of textual descriptions attached to the concepts greatly differed, as did the detail level of sketches. Some subjects produced just visual ideas and a few produced a concept in plain text. Most subjects visualized the target, the trunk or the plant, but some skipped it. These examples give an idea of how diverse the design idea generation process is, for instance, in comparison to the Electronic Brainstorming studies, in which the produced ideas are strictly formatted (e.g. Nijstad et al., 2002)

The verbal protocols were first examined quantitatively. The number of words per protocol varied considerably between all subjects ($M = 869.9$, $SD = 378.6$ words), which was anticipated as the participants were not chosen by their verbal fluency. From the protocols, over two hundred segments were detected on average, but this also varied a great deal ($M = 212.4$, $SD = 72.3$ segments). As all subjects verbalized thoughts about the majority of their concepts and video recordings facilitated the analysis, segments could be reliably associated with the sketched ideas and numbered, as described in section 3.4. It was also noted that designers discussed their work chiefly through solutions. Problems, the main problem or some of the subproblems, were typically mentioned only once by each designer ($M = 1.4$, $SD = 1.4$ segments). This emphasizes the fact that subproblems or subfunctions were more often discussed as generic solutions than specific subproblems ($M = 3.6$, $SD = 2.6$ segments), e.g. stating that “we have water supply” instead of saying “where could we get water”. Thus, the designers could be described as solution-oriented.

It was assumed that the tasks should be equally challenging, and consequently, produce quantitatively similar results. This seems to be correct. Considering the perceived difficulty based on the answers to the concluding questionnaire (Table 5), tasks were quite balanced and neither task environment was completely foreign to the designers. The median class for familiarity with the problem was “little” in both tasks. The majority of subjects felt that the Forest task was more demanding than Plant ($N = 9$), some thought the opposite ($N = 6$), and a single person estimated the tasks equally challenging[9]. Comparing the number of documented ideas, more concepts were produced for Forest than Plant ($M_{Forest} = 10.6$, $SD_{Forest} = 4.5$ and $M_{Plant} = 8.9$, $SD_{Plant} = 2.2$). To statistically test this finding, normality was first assumed and the variances were tested for inequality ($F(7,7) = 3.66$, $p > .05$). However, the difference of means was not statistically significant, $T(7) = 1.10$, $p > .10$ and therefore, the difficulty of the tasks is equal.

---

[9] This result is only possible because each subject completed both tasks. Unfortunately it is impossible to evaluate how examples affected the other task because of the small number of subjects and the experimental design.

Table 5

Frequencies of the selected results from the concluding questionnaire. Two tasks are grouped separately in the first item, but aggregated in the second

Familiarity with the task environment	Not at all	Little	Some what	Very much
Plant (*n* = 8)	2	3	0	3
Forest (*n* = 8)	1	6	1	0

Perceived difficult	Plant	Forest	Equal	Can not say
Which task was more difficult (*n*=16)	6	9	1	0

The first step in compiling the results from the protocol analyses was to summarize the problem decomposition scheme (PDS) codes. A summary of PDS codes is presented in Table 6. The classification of code types is based on two PDS:s and the specification given in section 3.4.

Table 6

Summaries of PDS the coded segments per subject and over all subjects. The upper table contains the total number of PDS codes and their relative amounts. The table below specifies the frequency distribution of PDS coded segments in the PDS category Other

Subject	N of PDS segments	Proportions: Explicit principles	Implicit principles	Other
1	31	65 %	3 %	32 %
2	30	57 %	13 %	30 %
3	30	43 %	0 %	57 %
4	36	61 %	8 %	31 %
5	16	38 %	6 %	56 %
6	39	72 %	18 %	10 %
7	32	22 %	59 %	19 %
8	35	69 %	3 %	29 %
9	29	83 %	7 %	10 %
10	47	81 %	0 %	19 %
11	41	44 %	20 %	37 %
12	30	47 %	13 %	40 %
13	47	47 %	17 %	36 %
14	27	85 %	7 %	7 %
15	16	44 %	19 %	38 %
16	31	26 %	10 %	65 %
M	32,3	55 %	13 %	32 %
SD	8,8	20 %	9 %	10 %

Subject	N of Other PDS segments (all explicit): Total	Main functions	Solution types	Subfunction problems	Subfunction solutions	Principles
1	8	1	0	3	2	2
2	9	0	2	1	2	4
3	17	0	0	0	9	8
4	11	0	2	2	2	5
5	9	0	0	0	9	0
6	4	0	0	1	2	1
7	6	0	0	2	2	2
8	10	0	3	0	2	5
9	3	0	0	0	2	1
10	9	0	1	1	3	4
11	15	0	0	2	6	7
12	12	0	1	0	2	9
13	17	0	0	4	6	7
14	2	0	1	0	0	1
15	6	0	1	1	4	0
16	20	0	3	2	4	11

The left column of Table 6 presents the number of *explicit* segments (solution principles in PDS) that were mentioned in the verbal protocol and implemented in the sketches. These make up a little more than a half of all PDS coded segments. Some solution principles, or parts of them, were identified from the sketches only. These parts are listed as *implicit* segments. Both explicit and implicit segments made additions to the design. Segments coded with an explicit PDS, but not considering any solution principle and therefore not advancing the idea, belong to the ‘Other’ category..

There was a considerable variance in the PDS code proportions categories between subjects (see Table 6). One reason for this is that some subjects (e.g. subject. #7 and #11) have produced notably more implicit segments than the others. Differences can be also understood by examining the lower portion of the table, where Other-category segments have been presented in more detail. *Other principles* are segments that evaluate, reconsider, or consider previously implemented or abandoned solution principles. It seems that some designers, for instance subjects #3, #12, and #16, analyzed their work more than the others based on the high proportion *Other principles*. Subfunctions make up two groups. Segments that stated a subfunction as a problem, e.g. “how can we dig up the trunk”, are called *Other subfunction problems*, but if these subfunctions were stated as general subfunctions “the trunk is dig up”, then segments belong to the category *Other subfunction solutions*. The few references to solution types are contained in *Other solution types* category. This includes very general references, for example, “it is an independent machine”, “it’s a …device in the pot”, or “there comes an airplane.” It makes no difference whether the solution type was implemented or just mentioned. Finally, in the single when the main goal of the task was mentioned, it was placed to the *Other main function* category.

Specific findings

Results specific to hypotheses of this study have been gathered to Tables 7 and 8. The latter presents findings about the main experimental question and the former summarizes results about the control strategies, which will be considered first.

Table 7
Summaries of the control strategies used by the subjects. The columns display the total number of concepts and which strategies that were used to produce those them. Below the individual results are the mean proportions for the two groups and at the bottom of the table summarizes the strategic orientation defined as the mode class of strategies per group

Subject	Total *N* of concepts	Recognition	Depth-first	Breadth-first	Mixed
1	16	0 %	75 %	19 %	6 %
2	9	0 %	11 %	67 %	22 %
3	10	0 %	90 %	0 %	10 %
4	9	0 %	0 %	78 %	22 %
5	3	0 %	0 %	100 %	0 %
6	13	0 %	8 %	92 %	0 %
7	13	0 %	69 %	0 %	31 %
8	7	14 %	0 %	86 %	0 %
9	10	0 %	10 %	30 %	60 %
10	11	0 %	0 %	91 %	9 %
11	13	0 %	46 %	23 %	31 %
12	6	17 %	17 %	67 %	0 %
13	14	0 %	0 %	14 %	86 %
14	8	13 %	13 %	75 %	0 %
15	5	0 %	0 %	20 %	80 %
16	8	25 %	13 %	13 %	50 %
Overall *M*	10.5	4 %	22 %	48 %	25 %
Plant group	71	8.6 %	7.6 %	70.9 %	12.9 %
Forest group	85	0.0 %	36.3 %	25.8 %	37.9 %
Strategic orientation of subjects					
Plant group (*n* = 8)		0 %	0 %	88 %	12 %
Forest group (*n* = 8)		0 %	50 %	12 %	38 %

Control strategies

All subjects seemed to have understood the task as a whole. Nobody concentrated on a single subfunction during the experiment and thus no misinterpretation-based designs were found. In general, most concepts were produced using a structured top-down, breadth-first or depth-first, strategy (70%

of 168 concepts total). The majority of these concepts were generated using the breadth-first strategy (43% of all concepts), but sometimes a depth-first strategy was also used (27%). There was still a considerable amount of non-structured approaches, which were classified as mixed strategies (26%), but might also be called opportunistic strategies. Recognition played a minor role, as only five instances were detected (4%). While the small number of these incidents makes it unpractical to statistically evaluate the relation of perceived difficulty and recognition, it should be mentioned that three of four subjects, who had used recognition in the Plant task, also found that task easier. Subfunctions were not solved in any particular order. Those subjects, who mainly used the breadth-first strategy, did not show any constant pattern in selecting subgoals, instead they started new ideas from a random subgoal. This was not analyzed in further detail.

For control strategies, there was a considerable difference between the tasks, especially in the proportions of the breadth-first and the mixed strategy. In Plant, the bulk of concepts where created breadth-first (71%, $N = 71$), but in Forest this figure sank to a third (26%, $N = 97$) and the proportion of the depth-first strategy increased from 8% to 36% ($N = 97$). The number of mixed strategies was also greater in Forest (13% vs. 39%). The difference in control strategy distributions between the tasks is significant, χ^2 (3, $N = 155$) = 62.06, $p <= .0001$. This difference shows also in the subject orientation, which was determined as the mode class of the control strategies applied per subject (see Table 7). The most typical orientation in Plant was breadth-first (88%, $N = 8$), but in Forest, the mixed strategy was almost as popular as the depth-first strategy (38% vs. 50%, $N = 8$).

Decomposition

Several criteria for detecting the different forms of decomposition were set in section 2.6 and they were applied as described in section 3.4. The results from that inspection are presented in Table 8. Explicit decomposition was observed only in three cases ($N = 3$). Two instances were identified by the use of a verb (subjects #10 and #11) and one by the temporal sequence of subgoal presentation (subject #3). Concerning the tasks, explicit decomposition was detected twice in Forest and once in Plant. Examining the results of explicit decomposition in terms of presented subgoals (subfunctions), it appears that only a single subject verbalized a goal stack covering all essential subgoals. The normative theories state that decomposition should be carried out before generating ideas. However, of the three designers, only one behaved in this way (subject #10).

Table 8
Frequencies of the different forms of decomposition as detected from the protocols. Each subject who at least once was used a particular type of decomposition has been included

Design approach	Observed in number of subjects
Explicit decomposition	3
Of which	
-- Complete explicit decomposition	1
-- Explicit decomposition at the onset of the task	1
Implicit decomposition only	13
Total	16

As described earlier, all subjects used top-down control strategies to some extent. Inferring from the structured development of concepts, it was evident that the rest of the subjects had applied implicit decomposition (*N* = 13). Also, there were some subfunctions that were rarely considered, such as the energy source and the mediator in Plant, and the recovery and the attachment in Forest. This finding supports the view that each task has a set of essential subgoals and these goals have an effect on the problem-solving behavior. As the results have now been presented, it must be considered what they tell about the cognitive model in general and problem decomposition in particular.

5. Discussion

This study investigated the role of problem decomposition in the internal search phase of the conceptual design process. After reviewing the literature, it became evident that decomposition had not received much attention in the previous studies and was poorly understood. Therefore, a cognitive model of internal search, including decomposition, was formulated. Decomposition was incorporated as a special search method. The model also allowed making of hypotheses about how decomposition could be empirically studied. An experiment following the single comparable study (Ho, 2001) was used to investigate the subject. The results from the sixteen designers involved in two independent tasks showed that implicit decomposition was used very often, as the model predicted. Explicit decomposition, however, was applied by few subjects and did not seem to work as expected in either of the tasks. Possible explanations for these results will be discussed next, followed by the consideration of theoretical and experimental limitations of the current work.

Results

As stated, the designers were very solution-oriented. This finding was not unexpected (Cross, 2004), but provokes some thoughts. The small number of problem-focused statements found in the protocols contradicts the underlying theory of verbalization to some extent (Ericsson & Simon, 1984). If we assume that subjects use implicit decomposition, and thus possess a goal stack in their STM, it can be asked why these goals are hardly ever verbalized, and what causes this failure to report subgoals. Maybe the designers just restrained themselves from verbalizing the goals because such verbalizing was not requested in the task assignment. Or maybe the goals are not stored in STM, but in LTM or LT-WM, excluding the current subgoal. However, if they were stored in LTM, a delay would be expected between solving different subproblems with the breadth-first strategy, and data from the current research do not directly support this option. Hence, the reasons for solution orientation remain unclear.

It is interesting that some subjects verbalized solution principles which were never sketched, even though the designers had been explicitly told to take advantage of all occurring ideas. While the number of discarded ideas was small, this finding clearly points out that idea evaluation is a firmly fixed component of the process and that overriding this criticism is difficult. This fact could be explained by assuming that an evaluation process is automatic and quite independent of conscious strategic control in the model.

Recognition

The analysis found some clear cases of recognition. The number was quite small and, although no assumptions were made about this figure, there is a possibility that the current coding procedure may not have identified all recognition-based designs. On some occasions, it was clear that a subject had retrieved certain concepts from LTM, e.g. a household robot in Plant. However, these concepts were not reproduced as such, but adapted to the constraints of the particular task. Therefore, as predicted by the cognitive model, it seems that the results produced by recognition undergo the same evaluation process as all other results before the output and that they may be iterated by the solution search mechanisms to synthesize new solutions. The total lack of recognition-based concepts in the Forest task is also understandable. The trunk removal machine has not received much publicity and it is likely that none of the participants had seen one, whereas plant watering devices are much more common. In a careful examination of a two recognized concepts (subject #12, concept 1 and subject #16, concept 4), it is noticeable that a drip bottle is not a common solution for watering plants in real life, but more of an analogical solution. This implies that recognition is somewhat flexible and possibly related to the analogy-retrieval process (see Thagard et al., 1990).

It was previously claimed that the recognition which has been considered here, differs from the concept of recognition used in the memory research. This might not be the case. Others have suggested sophisticated conceptualizations of recognition (Eysenck & Keane, 2000; Raaijmakers & Shiffrin, 1992), going further than what was proposed here, and theory of recognition should definitely be considered more carefully in the future.

Control strategies

Control strategies could be determined reliably. As was expected, structured strategies were used frequently. However, the notable difference between the two tasks raises some concerns. First, it should be noted that if the depth-first and breadth-first classes are combined before the comparison, then the change from structured to mixed strategies is less radical. These two top-down strategies were applied in 79% (*N* = 71) of all designs in Plant and in 62% (*N* = 97) in Forest. The threefold increase of mixed solutions in Forest (13% vs. 38%) is more puzzling. The difference is not due to a single outlier, but caused by several subjects adopting an alternative strategy, as shown by the change in the strategic orientations (see Table 8). While it is possible that the disagreement is partially attributable to an actual difference between the tasks, it is also conceivable that the different problem decomposition schemes may have increased the proportion of mixed strategies. Especially as some exceptions (described in Experimental method chapter) were made in the PDS coding of Forest. This could partially explain the significant change in balance between the two approaches.

A qualitative analysis of the concepts produced by designers classified as mixed-strategy oriented (*N* = 4) showed that three of them applied a strategy that could be called an ‘opportunistic depth-first strategy’. This means that they neglected some of the essential subgoals and developed only one or two principles for each new idea (subject #9, #15, and #16). This neglect appeared to be random. One mixed-strategy oriented designer (subject #13) clearly used the breadth-first strategy, but constantly omitted one particular and essential subfunction from her designs and was thus considered to use the mixed strategy. Thus, her behavior should be labeled incomplete breadth-first, not opportunistic.

Finding mixed strategies and the fact that subgoal selection in structured strategies followed no clear priority order, implies that there is some opportunism or lack of discipline in the design idea-generation process. This opportunism seems to override earlier priorities given to subproblems, as there is no fixed order in which subgoals are attended to. Opportunism might be caused by the fact that a memory search related to a subgoal produces several alternative answers. For example, a subject who is applying a breadth-first strategy may find several solutions for a single subgoal and pick one of them as a starting point for the next concept because that is the cognitively easiest thing to do. This could result in apparently opportunistic designs.

Decomposition

The results about decomposition were surprising as explicit decomposition was detected in few cases. This contradicts the assumptions of educational design literature (Pahl & Beitz, 1984; Ulrich & Eppinger, 2003), which is quite fixed regarding the notion that problem decomposition should precede the solution search. There are some possible explanations for this finding. First, the subjects were not instructed to use decomposition. Also, the task assignment particularly aimed at producing a great number of alternative designs quickly. It is possible that this instruction may have guided subjects' selection of solutions search methods. Secondly, poor ecological validity, in terms of available time and commitment to the design process, may have affected designers' actions and consequently, they may not have perceived that explicit decomposition should be used in a task like this. Most previous studies have used more extensive assignments (e.g. Ball et al., 1994; Goel & Pirolli, 1992; Ho, 2001). Thirdly, the tasks were thus relatively easy as the subjects were able to generate almost ten concepts on average. It might be that in harder tasks, where implicit decomposition proved to be insufficient, the explicit decomposition would be used. Finally, it is possible that the subjects were not familiar enough with explicit decomposition to utilize it. This possibility should be considered in the future research, by ensuring subjects' familiarity with decomposition. Here it was only assumed that the senior engineering students had already become skilled at explicit decomposition in their studies or in their design occupations, as the functional decomposition is introduced on two courses of their curriculum.

The results also showed that even when decomposition was used, it succeeded poorly. Two of three subjects failed in presenting a complete functional structure. However, this discovery can be also interpreted as a failure to report the results, rather than to decompose, so another type of success indicator should be used. Also, it is possible that explicit decomposition, in the available time, simply did not produce any new discoveries about the subgoals, which had already been discovered using implicit decomposition. The observation that explicit decomposition was not performed at the onset of the task may be due to the same reasons as the general lack of decomposition - the subjects did not associate the textbook method with the current situation. There was no a relation between recognition- and decomposition-based design (first recognize, then decompose) that could have explained this behavior.

It was originally proposed that decomposition facilitates problem solving by reducing the problem space of a problem (Simon, 1996). In this paper, decomposition has been presented as a way to produce tangible problems for memory search and synthesis. Thus, I cannot agree with Simon that decomposition primarily helps by reducing the problem space. Because if a theoretical problem space of undefined complexity is associated the main problem, it is practically infinite. Decomposition is said to reduce the complexity by ruling out some portions of the problem space. But if some, even considerable, finite region of an infinite space is ruled out, the resulting portion remains infinite. Even though human LTM is finite, it is more convincing to say that problem decomposition helps by providing a set of new starting points for memory search, which is one of the most important methods of the solution search. Hence, I find the concept of a problem space problematic when talking about human ill structured problem solving.

This discussion suggests that problem decomposition is a complex process and the relation of implicit and explicit decomposition is unclear. Based on the current findings, it seems that explicit decomposition is a test for the initial subgoal structure that has been produced by implicit decomposition. This interpretation would mean that explicit decomposition would become especially important if the original goal stack was faulty or incomplete. This study showed some indirect evidence of this being possible. In the protocol of subject number five, the designer considered the possibility to replace one subgoal with another. This activity did not meet the current criteria for explicit decomposition, but it might actually be an instance of explicit decomposition, if considered in an alternative way.

It seems justified to think of explicit decomposition as a complement of implicit decomposition. If implicit decomposition happened automatically, but not very reliably, then the main purpose of explicit decomposition would be to challenge the subgoal structure and refine it. It seems likely that these two modes of decomposition are otherwise highly similar, but explicit decomposition requires more time and effort. The possible reason for this could be the retrieval of additional knowledge that is relevant to the problem under consideration, making it similar with analogical transfer. This hypothesis is partially supported by the previous studies (Ball & Ormerod, 1995), but should be investigated in the future.

Theoretical issues

The proposed cognitive model of solution search is comprised of several information processing phases. The model is serial in nature and follows the symbolic tradition in claiming that all knowledge structures and processes are carried out on a single, general-purpose processing unit in successive steps. This holds on to the unitary view of cognition (Anderson, 1983). There are several theorists of cognitive science who have questioned this type of unified architecture. Probably the best known alternative is the modular approach, which assumes that cognitive functions are implemented as specialized modules of some kind, working in a parallel (e.g. Fodor, 1983; Pinker, 1994). This view is often defended by biological evidence, and used to explain the remarkable computational capacity related to some cognitive functions.

Two views are not completely mutually exclusive, as both can include similar cognitive structures. However, there may be considerable differences between systems that operate either in serial or in parallel. For instance, if the phases of the cognitive model, that has been considered here, were executed in *parallel*, it is likely that the contents of shared STM would change more rapidly and unexpectedly than they would change in a serial architecture. A method, such as protocol analysis that is assumed to report STM contents, would thus be affected by this difference. It should be noted that in even a parallel architecture, only *some* functions would be modular, and several learned skills would still be serially processed by a general-purpose structure. In the context of design problem solving, processes such as memory search and analogical mapping might be modular, whereas other operations would be still performed in a serial manner.

It seems likely that the presented model is a serial approximation of a parallel architecture. The assumed architecture is not the only open question about the model. Referring to the aim of cognitive science to present theories as algorithms (see section 1.1), it must be asked, could this model be implemented as a program? Considering similar models, this might be possible, but challenging. Some constituent processes of the GenePlore model (see section 2.4) have been computationally modeled, but the model as a whole has not been formally explicated to this date. The same goes for the SIAM model (see section 2.4) by Nijstad (2000). Several memory models have been formalized (Raaijmakers & Shiffrin, 1981), but they present only fractions of the whole and are typically difficult to apply for realistic tasks. It is possible that a general architecture, such as ACT-R, could be used for this purpose.

There is a doubt whether the architectures and computational theories of cognition are adequate for describing human cognition (see the recent debate in Fodor, 2000 and Pinker, 2005). This is a real dilemma that potentially undermines all computational explanations. It has been argued that some ill structured problems possess properties that are incompatible with the standard symbolic information processing model, called the Computational Theory of Mind (Goel, 1995). Goel thinks that sketching, for one, is an activity that cannot be properly reduced to syntactic symbolic representations. He has also provided some evidence to support this claim, but still thinks that computational theories can, with some limitations, be used to describe cognition (*ibid*).

Finally, there are some possible difficulties with the theoretical assumptions. A simple model of declarative (semantic) knowledge representation was assumed. However, it is possible that design knowledge is non-conceptual or visual knowledge, which would make it harder to identify recognition. It has been also questioned how reliable method protocol analysis is and whether the theory of verbalization is plausible (see Ericsson & Simon, 1980).This work also assumes that the protocol analysis can be trusted, although some negative evidence was already encountered in the form of implicit design actions. Pointless to say, the reliability of current findings is connected to these underlying assumptions.

Experimental issues

Experimental work can be carried out even though the theory would be incomplete. However, an empirical study has its own limitations. First of all, it is acknowledged that this study could not control all possibly significant variables, such as attention, motivation, mood, alertness, or fatigue, which may affect a complex task such as the one used. This is unfortunately the standard situation in psychological quasi-experimental research and there was no reason to assume that any of these variables would affect problem decomposition. Another issue related to the experimental setup concerns the possible effect of examples. While the analyzed tasks did not contain any examples (no-examples condition), the half of the tasks were preceded by a task with examples (examples condition). Strategies possibly adopted in the examples condition may have affected the no-examples condition, as happens in a negative/positive transfer or set effect (Eysenck & Keane, 2000). Unfortunately, this transfer effect, caused by the examples, is indistinguishable from the effect that may have been caused by having two tasks in a row. This is because the experimental design did not consider factor interaction (see section 3.2).

Finally, it must be noted that the protocols were coded by only one person. It is usually suggested to use several independent coders to achieve greater reliability that can be measured with inter-rater agreement (kappa) (Gero & McNeill, 1998; van Someren et al., 1994). This study omitted this procedure mainly due to its explorative nature, but this measure should be taken in the following studies.

Open questions

This preliminary exploration has yielded results that raise thoughts about problem decomposition in conceptual design. As the results are still quite approximate, the future investigation should focus on at least the following issues:

- Can some task environments bring up the explicit decomposition instinctively?
- Does the use of explicit decomposition have an effect on design outcomes?
- Can the use of explicit decomposition be included in a wider range of design tasks?
- Can explicit decomposition be made more effective?
- Is implicit decomposition a domain-dependent or a domain-independent skill?
- How are implicit and explicit decomposition related to each other?
- How does expertise effect decomposition?

This study did not find any clear evidence that explicit decomposition would considerably influence design outcomes. The consequences of using implicit or explicit decomposition in concept design should be investigated. If it turned out that either mode of working would be more favorable, then the current findings should be reconsidered. If explicit decomposition proved beneficial, then it would follow that the possibilities to guide designers to take advantage of it should be investigated. On the other hand, if the use of explicit decomposition constantly produced inferior results, then it would be advisable not to waste time with it.

The possible *task- or domain specificity* of decomposition is an essential question. Explicit decomposition has been found in other studies, especially those that have used more complete design tasks, going from early planning to the detail design stage (Ball et al., 1994). A subject for a longitudinal study could be to assess decomposition as an attainable skill and its connection to expertise. It is possible that explicit decomposition is only adopted after acquiring noticeable expertise, or maybe implicit decomposition works very well right from the start. Additional investigations are also needed to clarify the relation of implicit and explicit decomposition and the underlying cognitive mechanism. In short, there are several open questions related to decomposition that should be investigated.

Conclusion

This work has applied the concepts and methods of cognitive science to investigate conceptual design. Cognitive science, being a multidisciplinary domain (Schunn *et al.*, 1998), fits well for this purpose. This work has relied on the existing cornerstones, mostly on the work done by Simon and Akin (Akin, 1986; Simon, 1973, 1996) which presents a unified picture of design as problem solving activity. In the literature review, it became evident that the design theory is still in progress, and the development of a comprehensive theory of design cognition has not been recently a major interest. However, it was noted that the recent progress in cognitive creativity research (Finke et al., 1992) has to the development of a similar, although abstract, model of creative activity. That model was used as an exemplar, together with the existing theories of design problem solving, to synthesize a cognitive model of conceptual design.

After developing the model, empirical tests were conducted. This initial exploration, in the form of sixteen psychological experiments, provided additional evidence that decomposition works in two modes, called explicit and implicit decomposition. The latter form is highly automatic and corresponds to the concept of decomposition found in the problem-solving literature. Unlike what could be predicted from the literature, explicit decomposition was observed seldom in the experiments. While this may be due to the artificial nature of the experiments, it also questions the significance of explicit decomposition in design idea generation. Current results suggest that future studies should examine the effects and constraints of decomposition in a greater detail to evaluate its practical importance. In whatever fashion problem decomposition will be interpreted in the future investigations, this study has shown that problem decomposition is an integral part of the problem-solving process, but more in the way described by the psychological literature than in the way assumed by the design literature.

6. References

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

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

Anderson, J. R. (1987). Skill acquisition: Compilation of weak-methods problem solutions. *Psychological Review, 94*(2).

Anderson, J. R. (1990). *The adaptive character of thought*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, J. R., Greeno, J. G., Kline, P. J., & Neves, D. M. (1981). Acquisition of problem-solving skill. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Baddeley, A. (2003). Working memory: Looking back and looking forward. *Nature Reviews Neuroscience, 4*(10), 829-839.

Ball, L. J., Evans, J. S. B. T., & Dennis, I. (1994). Cognitive processes in engineerign design: A longitudinal study. *Ergonomics, 37*(11), 1753-1786.

Ball, L. J., & Ormerod, T. C. (1995). Structured and opportunistic processing in design: A critical discussion. *International Journal of Human-Computer Studies, 43*(1), 131-151.

Ball, L. J., Ormerod, T. C., & Morley, N. J. (2004). Spontaneous analogising in engineering design: A comparative analysis of experts and novices. *Design Studies, 25*(5), 495-508.

Bechtel, W., Abrahamsen, A., & Graham, G. (2001). Cognitive science: History. In *International encyclopedia of the social &behavioral sciences* (pp. 2154). Oxford: Elsevier Science Ltd.

Boden, M. A. (2004). *The creative mind. Myths and mechanisms. Second edition*. London: Routledge.

Chandrasekaran, B. (1990). Design problem solving: A task analysis. *AI Magazine, 11*(4), 59-71.

Charness, N. (1991). Expertise in chess: The balance between knowledge and search. In K. A. Ericsson & J. Smith (Eds.), *Toward a general theory of expertise*. Cambridge, MA: Cambridge University Press.

Chi, M. T. H. (1997). Quantifying qualitative analyses of verbal data: A practical guide. *Journal Of The Learning Sciences, 6*(3), 271-315.

Chi, M. T. H., Glaser, R., & Farr, M. J. (1988). *The nature of expertise*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Costello, F. J., & Keane, M. T. (2000). Efficient creativity: Constraint-guided conceptual combination. *Cognitive Science, 24*(2), 299-349.

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral And Brain Sciences, 24*(1), 87-+.

Cross, N. (2004). Expertise in design: An overview. *Design Studies, 25*(5), 427.

Dennis, A., Aronson, J., Heninger, B., & Walker, E. (1996). *Task and time decomposition in electronic brainstrorming.* Paper presented at the 29th Annual Hawaii International Conference on System Sciences, Hawaii.

Diehl, M., & Stroebe, W. (1987). Productivity loss in brainstrorming groups: Toward the solution of a riddle. *Journal of Personality and Social Psychology, 53*(3), 497-509.

Dix, A., Finlay, J., Abowd, G. D., & Beale, R. (2004). *Human-computer interaction. Third edition* (3rd ed.). Harlow: Prentice-Hall.

Duncker, K. (1945). *On problem-solving* (L. S. Lees, Trans. Vol. 58). Washigton D.C.: American Psychological Association inc.

Egan, D. E., & 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*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Ericsson, K. A., & Kintsch, W. (1995). Long-term working memory. *Psychological Review, 102*(2), 211-245.

Ericsson, K. A., & Simon, H. A. (1980). Verbal reports as data. *Psychological Review, 87*(3), 215-251.

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

Ericsson, K. A., & Smith, J. (1991a). Prospects and limits of the empirical study of expertise: An introduction. In K. A. Ericsson & J. Smith (Eds.), *Toward a general theory of expertise* (pp. 1-38). Cambridge, MA: Cambridge University Press.

Ericsson, K. A., & Smith, J. (Eds.). (1991b). *Toward a general theory of expertise. Prospects and limits*. New York: Cambridge University Press.

Eysenck, M. W., & Keane, M. T. (2000). *Cognitive psychology. A student's handbook.* (4th ed.). East Sussex: Psychology Press.

Finke, R. A., Ward, T. B., & Smith, S. M. (1992). *Creative cognition. Theory, research, and applications*. Cambrige, MA: The MIT Press.

Fodor, J. A. (1983). *The modularity of mind. An essay of faculty psychology*. Cambridge, MA: MIT Press.

Fodor, J. A. (2000). *The mind doesn't work that way. Scope and limitations of computational psychology*. Cambridge, MA: MIT Press.

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

Gero, J. S., & McNeill, T. (1998). An approach to the analysis of design protocols. *Design Studies, 19*(1).

Gero, J. S., & Tang, H.-H. (2001). The differences between retrospective and concurrent protocols in revealing the process-oriented aspects of the design process. *Design Studies, 21*(3), 283-295.

Goel, V. (1995). *Sketches of thought*. Cambridge, MA: MIT Press.

Goel, V., & Pirolli, P. (1992). The structure of design problem spaces. *Cognitive Science, 16*(3), 395-429.

Hebb, D. O. (1949). The organization of behavior. In R. Cummins & D. D. Cummins (Eds.), *Minds, brains, and computers*. Malden, MA: Blackwell.

Ho, C.-H. (2001). Some phenomena of problem decomposition strategy for design thinking: Differences between novices and experts. *Design Studies, 22*(1), 27-45.

Hummel, J. E., & Holyoak, K. J. (1997). Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review, 104*(3), 427-466.

Jansson, D. G., & Smith, S. M. (1991). Design fixation. *Design Studies, 12*(1), 3-11.

Jeffries, R., Turner, A. A., Polson, P. G., & Atwood, M. E. (1981). The processes involved in designing software. In J. R. Anderson (Ed.), *Cognitive skilss and their acquisition*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Johnson-Laird, P. (1980). Mental models in cognitive science. *Cognitive Science, 4*(1), 71-115.

Kavakli, M., & Gero, J. S. (2002). The structure of concurrent cognitive actions: A case study on novice and expert designers. *Design Studies, 23*(1), 25.

Keane, M. T. (1997). What makes an analogy difficult? The effects of order and causal structure on analogical mapping. *Journal Of Experimental Psychology: Learning Memory And Cognition, 23*(4), 946-967.

Klahr, D., & Simon, H. A. (1999). Studies of scientific discovery: Complementary approaches and convergent findings. *Psychological Bulletin, 125*(5), 524-543.

Laird, J., Rosenbloom, P., & Newell, A. (1986). *Universal subgoaling and chunking: The automatic generation and learning of goal hierarchies*. Boston: Kluwer.

Lakoff, G. (1986). *Women, fire, and dangerous things: What categories reveal about the mind*. Chicago, Illinois: University of Chicago Press.

Lewis, R. L. (2001). Cognitive theory: Soar. In *International encyclopedia of the social &behavioral sciences* (pp. 2178). Oxford: Elsevier Science Ltd.

Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. New York: Freeman.

Mayer, R. E. (1992). *Thinking, problem solving, cognition. Second edition*. New York: W.H. Freeman and Company.

Mayer, R. E. (1999). Fifty years of creativity research. In R. J. Sternberg (Ed.), *Hanbook of creativity*. Cambridge: Cambridge University Press.

McDermott, D. (1976). Artificial intelligence meets natural stupidity. *SIGART Newsletter, 57*, 4-9.

Mednick, S. A. (1962). The associative basis of the creative process. *Psychological Review, 69*(3), 220-232.

Miller, G. A. (1956). The magical number 7, plus or minus 2 - some limits on our capacity for processing information. *Psychological Review, 63*(2), 81-97.

Mumford, M. D. (2003). Where have we been, where are we going? Taking stock in creativity research. *Creativity Research Journal, 15*(2-3), 107-120.

Newell, A. (1969). Heuristic programming: Ill structured problems. In J. S. Aronofsky (Ed.), *Progress in operations research, vol. 3* (pp. 360-414). New York: Wiley.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.

Nijstad, B. A. (2000). *How the group affects the mind.* Unpublished Doctoral dissertation, Utrecht University.

Nijstad, B. A., Stroebe, W., & Lodewijkx, H. F. M. (2002). Cognitive stimulation and interference in groups: Exposure effect in an idea generation task. *Journal Of Experimental Social Psychology, 38*(6), 535-544.

Nijstad, B. A., Stroebe, W., & Lodewijkx, H. F. M. (2003). Production blocking and idea generation: Does blocking interfere with cognitive processes? *Journal Of Experimental Social Psychology, 39*(6), 531-548.

O'Hara, K. P., & Payne, S. J. (1998). The effects of operator implementation cost on planfulness of problem solving and learning. *Cognitive Psychology, 35*(1), 34-70.

Osborn, A. F. (1957). *Applied imagination: Principles and procedures of creative problem-solving. Revised edition* (Revised edition ed.). New York: Scribner.

Pahl, G., & Beitz, W. (1984). *Engineering design*. London: The Design Council.

Perttula, M., & Liikkanen, L. A. (2005). Cue-based memory probing in design idea generation, *Sixth International Roundtable Conference on Computational and Cognitive Models of Creative Design*. Queensland, Australia.

Pinker, S. (1994). *The language instinct*. New York: William Morrow.

Pinker, S. (2005). So how does the mind work? *Mind & Language, 20*(1), 1-24.

Potter, R. E., & Balthazard, P. (2004). The role of individual memory and attention processes during electronic brainstorming. *Mis Quarterly, 28*(4), 621-643.

Raaijmakers, J. G. W., & Shiffrin, R. M. (1981). Search of associative memory. *Psychological Review, 88*(2), 93-134.

Raaijmakers, J. G. W., & Shiffrin, R. M. (1992). Models for recall and recognition. *Annual Review Of Psychology, 43*, 205-234.

Reitman, W. R. (1965). *Cognition and thought: An information-processing approach*. New York: Wiley.

Runco, M. A., & Sakamoto, S. O. (1999). Experimental studies of creativity. In R. J. Sternberg (Ed.), *The handbook of creativity*. Cambridge, MA: Cambridge University Press.

Russell, S., & Norvig, P. (2003). *Artificial intelligence: A modern approach. Second edition*. New Jersey: Prentice-Hall.

Schunn, C. D., Crowley, K., & Okada, T. (1998). The growth of multidisciplinarity in the cognitive science society. *Cognitive Science, 22*(1), 107.

Schön, D. A. (1995). *The reflective practitioner: How professionals think in action. Reprise edition*. Aldershot: Arena.

Shah, J. J., Kulkarni, S. V., & Vargas-Hernandez, N. (2000). Evaluation of idea generation methods for conceptual design: Effectiveness metrics and design of experiments. *Journal of Mechanical Design, 122*(4), 377-384.

Shah, J. J., Vargas-Hernandez, N., & Smith, S. M. (2003). Metrics for measuring ideation effectiveness. *Design Studies, 24*(2), 111-134.

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

Simon, H. A. (1996). *The sciences of the artificial. Third edition*. Cambridge, MA: MIT Press.

Simon, H. A., & Newell, A. (1958). Heuristic problem solving: The next advance in operations research. *Operations Research, 6*(1), 1-10.

Smith, G. F., & Browne, G. J. (1993). Conceptual foundations of design problem solving. *IEEE Transactions on Systems, Man and Cybernetics, 23*(5), 1209-1219.

Stein, B. (2004). Engineers don't search. In W. Lenski (Ed.), *Lecture notes in computer science. Logic versus approximation* (Vol. 3075, pp. 120-137). Berlin: Springer-Verlag.

Suwa, M., Purcell, T., & Gero, J. S. (1998). Macroscopic analysis of design processes based on a scheme for coding designer's cognitive actions. *Design Studies, 19*(4), 455-483.

Thagard, P., Holyoak, K. J., Nelson, G., & Gochfeld, D. (1990). Analog retrieval by constraint satisfaction. *Artificial Intelligence, 46*(3), 259-310.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind, 49*, 433-460.

Ullman, D. G., Stauffer, L. A., & Dietterich, T. G. (1986). Preliminary results of an experimental study on mechanical design process: Oregon state university, Corvallis, Oregon.

Ulrich, K. T., & Eppinger, S. D. (2003). *Product design and development. Third edition*. Boston: McGraw-Hill.

van Someren, M. W., Barnard, Y. F., & Sandberg, J. A. C. (1994). *The think aloud method. A practical guide to modelling cognitive processes.* London: Academic Press Limited.

Weisberg, R. W. (1986). *Creativity: Genius and other myths*. New York: W.H. Freeman.

Voss, J. F., & Post, T. A. (1988). On the solving of ill-structured problems. In M. T. H. Chi, R. Glaser & M. J. Farr (Eds.), *The nature of expertise*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Appendix A

Initial and concluding questionnaires

In original language

G1 Alkukysely

Tee merkintä oikean vaihtoehdon vasemmalla puolella olevaan ympyrään tai viivalle

Henkilötausta

1.1 Sukupuoli O Mies O Nainen

1.2 Syntymävuosi 19____

1.3 Äidinkieli _______________

1.4 Sähköpostiosoite ______________________________

Suunnittelutausta

1.4 Opintojen vaihe ________ opintoviikkoa (kymmenen viikon tarkkuudella)

1.5 Työkokemus suunnittelusta ________ vuotta (½ vuoden tarkkuudella)

1.6 Opintojen / työn suuntautumisala
O koneen suunnittelu
O teollinen muotoilu
O muu, mikä: ________________________

1.7 Suoritetut korkeakoulututkinnot (jos olet jo valmistunut, jätä muuten tyhjäksi)
O DI O TM O Muu, mikä _________________________

Lähtötilanne kokeeseen

Arvioi miten hyvin seuraavat väittämät kuvaavat tämän hetkistä oletilaasi, skaalalla 1-6 ympyröimällä oikea vaihtoehto

	Hyvin paljon		Jonkin verran		Ei lainkaan	
1.8 Fyysinen väsymys	1	2	3	4	5	6
1.9 Mentaalinen väsymys	1	2	3	4	5	6
1.10 Olen motivoinut osallistumaan kokeeseen						
	1	2	3	4	5	6

Kokeen kulusta

Olen tietoinen, että kokeen materiaalit ovat toistaiseksi luottamuksellisia ja en saa kertoa kokeen tehtävien varsinaisesta sisällöstä tutkimusryhmän ulkopuolisille henkilöille. Tiedän, että minulla on oikeus keskeyttää koe milloin tahansa. Kokeessa tuottamani materiaali käsitellään luottamuksellisesti hyvän tutkimusetiikan mukaan. Koe liittyy tieteelliseen työhön, joka raportoidaan niin, ettei henkilöllisyyttäni voida yhdistää tuottamaani koeaineistoon. Tiedän, että kokeenjärjestäjät voivat käyttää kokeessa tehtyjä luonnoksia, pätkiä videosta tai ääniraidasta julkaisujen ja konferenssiesitysten osana.

Espoossa,

Nimikirjoitus ___________________________

G4 Loppukysely 1/2

A) Tehtäväkohtainen tietämys

Seuraavassa on kysymyksiä äskeisiin tehtäviin liittyen. Arvioi väitteiden oikeellisuutta tekemällä merkintä vaihtoehdon edessä olevaan ympyrään.

2.1 Oliko ensimmäisessä tehtävässä suunniteltu laite ja tehtäväympäristö sinulle entuudestaan tuttu?

O Ei lainkaan
O Vähän
O Jonkin verran
O Hyvin tuttu

2.2. Oliko toisessa tehtävässä suunniteltu laite ja tehtäväympäristö sinulle entuudestaan tuttu ?

O Ei lainkaan
O Vähän
O Jonkin verran
O Hyvin tuttu

2.3 Sait toisessa tehtävistä käyttöösi esimerkkikonsepteja, olivatko ne mielestäsi hyödyllisiä?

O Ei lainkaan
O Vähän
O Jonkin verran
O Erittäin

2.4 Käytitkö esimerkkejä suunnittelun aikana?

O En lainkaan
O Vähän
O Jonkun verran
O Hyvin

2.5 Kumpi tehtävä oli mielestäsi vaikeampi?

O Ensimmäinen
O Toinen
O Yhtä vaikeita
O En osaa sanoa

2.6 Haluatko saada sähköpostilla yhteenvedon tutkimuksen tuloksista?

O Kyllä
O Ei

2.7 Voiko sinuun ottaa yhteyttä jos tarvitsemme tulevaisuudessa koehenkilöitä vastaantyyppisiin kokeisiin?

O Kyllä
O Ei

G4 Loppukysely 2/2

B) Lopputilanne

Arvioi nykyinen tilanteesi seuraavien väittämien suhteen skaalalla 1-6 ympyröimällä oikea vaihtoehto

	Hyvin paljon		Jonkin verran		Ei lainkaan	
2.8 Fyysinen väsymys	1	2	3	4	5	6
2.9 Mentaalinen väsymys	1	2	3	4	5	6

C) Esimerkkien muistaminen

Seuraavalla sivulla on kaksi aiemmin käytetyn kaltaista konseptipaperia. Nyt sinun pitäisi palauttaa mieleesi aiemmin toisen tehtävän yhteydessä sinulle esitetyt esimerkkikonseptit. Toteuta esimerkit niin hyvin kuin muistat samalla tavalla kuin aiemmatkin konseptit. Hahmottele konseptit arkille, älä kiinnitä huomiota yksityiskohtiin. Aikaa tähän tehtävään sinulla on kolme minuuttia.

D) Miten esimerkit mielestäsi vaikuttivat suunnittelutehtävään?

____

____

____

____

____

____

E) Koe loppuu – kiitos osallistumisesta

Appendix B

General brief

In original language

Annan sinulla kohta ensin tehtäväpaperin, mutta sitä ennen vielä kerron yleisesti tehtävistä. Tehtävissä pitää tehdä konseptiluonnoksia annetusta aiheesta. Konseptit piirretään erillisille vastausarkeille, jotka ovat tässä.

{vastauspaperin jakaminen}

Konsepti on siis laitteen ulkoasun luonnos, yksinkertainen piirros. Esitä konseptit niin selkeästi, että laitteen toimintaperiaate ja sen olennaisimmat komponentit ovat tunnistettavissa ja voit tarvittaessa lisätä tekstiä selventämään piirrosta. Tekstien tulee olla kuitenkin mahdollisimman lyhyitä, sillä kirjoittaminen vie aikaa suunnittelulta.

Tee siis mahdollisimman paljon, mahdollisimman erilaisia konsepteja, kaikki ideat mitkä mieleen tulee paperille, ei turhaa kritiikkiä. Ei hienostelua, ainoastaan olennaisimmat piirteet näkyviin.

En tule enää kokeen aikana antamaan lisää informaatiota, ainoastaan tarvittaessa muistutan sinua jatkamaan ääneenajattelua. Voit käyttää tehtävänantopaperia tehtävän aikana. Tehtävän suorittamiseen sinulla on aikaa 20 minuuttia, ilmoitan kun aikaa on jäljellä kaksi minuuttia niin voit ottaa vielä loppukirin. Aika alkaa kun käännät tehtäväpaperin.

Aloitetaan, kerro kun olet lukenut tehtävän niin aloitan ajanoton ja voit kääntää tehtäväpaperin ja alkaa tekemään konsepteja. Tässä vaiheessa voit vielä kysyä, huomaa että kaikki mitä ei ole tehtävässä määritelty on suunnittelijan päätettävissä.

{ensimmäisen tehtäväpaperin jakaminen}

Appendix C

Task assignments

The task assignments presented here are original Finnish assignments, Kasvi and Metsä (Plant and Forest). English translations are available from the author on request. The schematic figure and concept images have been omitted from the electronic publication.

G2B Tehtävä Kasvi

K.1 Taustaa

Huonekasvien kastelu on yksinkertainen toimenpide, jonka hoitaminen omistajan poissaollessa jää kuitenkin usein muiden ihmisten tehtäväksi. Olemassaolevat automaattiset ratkaisut eivät ole teknisesti riittävän hyviä tai sosiaalisesti hyväksyttäviä.

K.2 Tehtävä

Suunnittele automaattinen huonekasvien kastelulaite, tee laitteesta mahdollisimman paljon toisistaan poikkeavia konsepteja. Laite pitää yhden tai useamman kasvin elossa vähintään kuukauden ajan.. Laitteen pitää toimittaa kasville noin yksi desilitra vettä viikossa.

Konseptisuunnittelun tueksi on paperin kääntöpuolella kuva huonekasvista, jonka on tarkoitus toimi innoittajana sunnittelulle. Saat käyttää tehtävänantoa ja esimerkkiä tehtävän aikana.

Ilmoita nyt kokeenjohtajalle, niin voit kääntää paperin ja aloittaa suunnittelun

The schematic figure used in the Plant task,
originally located on the reverse side of the paper

G3A Tehtävä Metsä

M.1 Taustatietoja

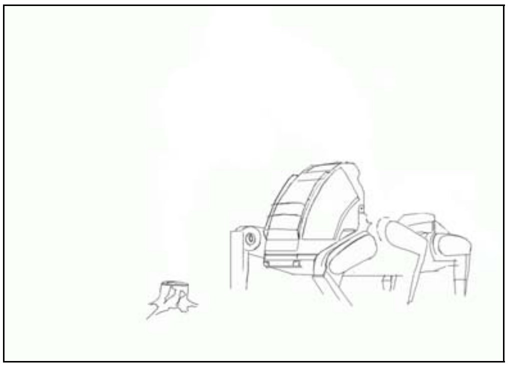
Metsähakkuun jäljiltä metsään jää kantoja. Kannoissa on kuitenkin runsaasti hitaasti hajoavaa puuta, joka pitää poistaa metsästä hyötykäytettäväksi ja nopeuttamaan metsän uusiutumista. Tällä hetkellä kannot siirretään pois metsästä yhdellä laitteella ja silputaan metsän ulkopuolella erillisessä laitteessa.

M.2 Tehtävä

Suunnittele talousmetsän kantojen hävityslaite, tee laitteesta mahdollisimman paljon erilaisia konsepteja. Laitteen tarkoitus on poistaa kannot metsästä tehokkaasti paikan päällä. Laite liikkuu jaloilla, näitä ei tarvitse konseptoida (kts. kuvaa seuraavalla sivulla).

Konseptisuunnittelun tueksi on paperin kääntöpuolella kuva kohteesta, jonka on tarkoitus toimi innoittajana sunnittelulle. Saat käyttää tehtävänantoa ja esimerkkiä myös tehtävän aikana.

Ilmoita nyt kokeenjohtajalle, niin voit kääntää paperin ja aloittaa suunnittelun



The schematic figure used in the Forest task,
originally located on the reverse side of the paper

Appendix D

Problem decomposition schemes

PDS for Plant

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	40. Mechanical timer
	31. Water pipe	41. Valve or choke
	32. Rain wall	42. Humidity sensor
	33. Condensation water	43. Pipe, hose, or string size
	34. Ice	44. Mail man
	35. Bound water	45. Weight sensor
	36. Tank in soil	46. Surface permeability
	37. Tank in pot or in plant board	47. Phone, PC, or electric timer
	38. Waterin can	48. Tank decomposition
	39. Steamer	49. Controlled vaporization
	310. Closed system	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	60. Plant mover
	51. Capillar hose, stick, or string	61. Soil changer
	52. Pouring system	62. Robot
	53. Gravity; syphon or drain	63. Rotating ground
	54. Sprinkler	64. Household air
	55. Pump, water gun, or pressure	65. Plant air
	56. Syphon	66. Humidity collector
	57. Diffusion	
	58. Absorption	
	59. Scooping or sprinkling	

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

PDS for Forest

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 agains 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

Appendix E

Complete coded verbal protocols of all subjects

Omitted from this document version.
Please, contact the author in case You wish to obtain the protocols.

Appendix F

Control strategy summaries

Specified for each subject and each concept generated by the subject including a summary of strategies over all concepts

R	Recognition
D	Depth-first
B	Breadth-first
M	Mixed

Subject #	**Concept**	**Strategy**
1	1	D
	2	D
	3	D
	4	D
	5	D
	6	D
	7	M
	8	D
	9	B
	10	D
	11	D
	12	D
	13	B
	14	B
	15	D
	16	D
	Total	
	R	0
	D	12
	B	3
	M	1

Subject #	**Concept**	**Strategy**
2	1	M
	2	M
	3	D
	4	B
	5	B
	6	B
	7	B
	8	B
	9	B
	Total	
	R	0
	D	1
	B	6
	M	2

Subject #	**Concept**	**Strategy**
9	1	M
	2	B
	3	M
	4	M
	5	D
	6	M
	7	B
	8	B
	9	M
	10	M
	11	M
	Total	
	R	0
	D	1
	B	3
	M	6

Subject #	**Concept**	**Strategy**
10	1	M
	2	B
	3	B
	4	B
	5	B
	6	B
	7	B
	8	B
	9	B
	10	B
	11	B
	Total	
	R	0
	D	0
	B	10
	M	1

Subject #	**Concept**	**Strategy**
3	1	D
	2	D
	3	D
	4	M
	5	D
	6	D
	7	D
	8	D
	9	D
	10	D
	Total	
	R	0
	D	9
	B	0
	M	1

Subject #	**Concept**	**Strategy**
4	1	B
	2	B
	3	B
	4	B
	5	B
	6	M
	7	B
	8	M
	9	B
	Total	
	R	0
	D	0
	B	7
	M	2

Subject #	**Concept**	**Strategy**
5	1	B
	2	B
	3	B
	Total	
	R	0
	D	12
	B	3
	M	1

Subject #	**Concept**	**Strategy**
11	1	B
	2	M
	3	D
	4	M
	5	B
	6	M
	7	M
	8	B
	9	D
	10	D
	11	D
	12	D
	13	D
	Total	
	R	0
	D	6
	B	3
	M	4

Subject #	**Concept**	**Strategy**
12	1	R
	2	B
	3	D
	4	B
	5	B
	6	B
	Total	
	R	1
	D	1
	B	4
	M	0

Subject #	**Concept**	**Strategy**
13	1	M
	2	M
	3	B
	4	M
	5	M
	6	M
	7	M
	8	M
	9	M
	10	M
	11	M
	12	B
	13	M
	14	M
	Total	
	R	0
	D	0
	B	2
	M	12

Subject #	**Concept**	**Strategy**
6	1	B
	2	B
	3	B
	4	B
	5	B
	6	B
	7	B
	8	B
	9	B
	10	B
	11	D
	12	B
	13	B
	Total	
	R	0
	D	1
	B	12
	M	0

Subject #	**Concept**	**Strategy**
7	1	D
	2	D
	3	D
	4	D
	5	D
	6	D
	7	D
	8	D
	9	M
	10	M
	11	D
	12	M
	13	M
	Total	
	R	0
	D	9
	B	0
	M	4

Subject #	**Concept**	**Strategy**
8	1	R
	2	B
	3	B
	4	B
	5	B
	6	B
	7	B
	Total	
	R	1
	D	0
	B	6
	M	0

Subject #	**Concept**	**Strategy**
14	1	B
	2	B
	3	B
	4	B
	5	R
	6	B
	7	D
	8	B
	Total	
	R	1
	D	1
	B	6
	M	0

Subject #	**Concept**	**Strategy**
15	1	M
	2	M
	3	M
	4	M
	5	B
	Total	
	R	0
	D	0
	B	1
	M	4

Subject #	**Concept**	**Strategy**
16	1	R
	2	D
	3	M
	4	R
	5	M
	6	M
	7	M
	8	B
	Total	
	R	2
	D	1
	B	1
	M	4