Computational Thinking in Regard to Thinking and Problem-Solving

Aleksi Tiensuu

University of Tampere
School of Information Sciences
Computer Science
M.Sc. thesis
Supervisor: Eleni Berki
June 2012

University of Tampere
School of Information Sciences
Computer Science
Aleksi Tiensuu: Computational Thinking in Regard to Thinking and Problem-Solving
M.Sc. thesis, 47 pages, 2 index pages, and 2 appendixes
June 2012

Computational thinking is a relatively new concept that is still looking for its exact definition. It can be seen as the by-product of exposure to computational activities, as well as collection of information processing tools one is familiarized with through computing. However, it can be a lot more than that. Similarly as mathematical thinking can be seen to be the constructive force behind mathematics, computational thinking can be seen as the driving force behind computing. Computational thinking is approached via what is known of thinking and problem-solving. The aims of the study are to further clarify the concept and to figure out how thinking and problem-solving can be affected by the acquisition of computational thinking, and to look into if there is anything that can be done to produce more efficient computational thinkers (e.g., programmers and software developers).

The research reveals that computational thinking has notable potential to improve the "general" thinking and problem-solving ability, but there are barriers that have to be dealt with in order to meaningfully benefit from this potential. These barriers rise from the nature of thinking and they cannot be completely bypassed. However, the thesis provides methods of how to handle them more effectively.

Key words and terms: computational thinking, thinking, problem-solving, learning

Contents

1. Introduction

A few years ago, I was contemplating what would be left if “computer” was taken away from computer scientist. My initial thoughts were, that not much as the artificial languages we use and the algorithms we play with are something that require the existence of a computer to have reason to exist; there is not much point of having programming languages if there is nothing to program. And similarly, algorithms which can require more computational power than human can handle, are useless knowledge without the source of that power. However, I had noticed earlier that imperative, especially object-oriented, programming had influenced the way I observe things. It made me re-think that maybe the answer is not as simple after all. Before long I just forgot the whole question, until I accidentally came across Christos Papadimitriou’s concept of algorithmic lens. This was the point when I started to seriously think about the answer to the initial question and eventually encountered relatively new, ill-structured and formally undefined concept of computational thinking, which could be the answer to my question. Yet, there was one little problem: What precisely is computational thinking?

1.1. Research rationale

Lately, computational thinking has been attracting the attention of academics and industry (e.g., Microsoft Research and Google). It is no wonder people find it interesting, as thinking has a central role in almost everything that we do, and computing has somewhat impressive track record of changing the world. What is this thinking that draws from the concepts fundamental to computer science? What it has to offer and what are its weaknesses and limitations? These are questions well worth of research because the process can lead to findings that can, among other things, help to improve problem-solving performance and to advance thinking and computational thinking.

1.2. Research framework and approach

Lester [2005] emphasises the importance of the research framework by stating that the “notion of a research framework is central to every field of inquiry, but at the same time the development and use of frameworks may be the least understood aspect of the research process." He also states that no data without a framework makes sense, and that a good framework allows us to transcend common sense. Researcher should have a deep understanding of the phenomena he studies.

Eisenhart [1991] identified three types of research frameworks: theoretical, practical, and conceptual. "A **theoretical framework** is a structure that guides research by relying on a formal theory, that is, the framework is constructed by using an established, coherent explanation of certain phenomena and relationships" [Eisenhart, 1991]. One of the examples she lists is the Newell and Simon's theory of human problem-solving, which is quite interesting regarding my thesis topic. **Practical framework** “is not informed by formal theory, but by accumulated practical knowledge (ideas) of practitioners and administrators, the findings of previous research, and often the viewpoints of politicians or public opinions. Research hypotheses or questions are derived from this knowledge base, and research results are used to support, extend, or revise the practice” [Eisenhart, 1991]. “A **conceptual framework** is a skeletal structure of justification, rather than a skeletal structure of the explanation based of formal logic (i.e. formal theory) or accumulated experience (i.e. practitioner knowledge).” [Eisenhart, 1991] Theoretical and practical frameworks are limiting in terms of perception when it comes to wide-ranging and ill-structured topics. Lester [2005] also states that these frameworks have several serious shortcomings (look [Lester, 2005] for more details), and emphasizes the importance of the conceptual framework in mathematics education research, which can share similar objectives with my research.

In problem-solving it is important to define what belongs into the problem space and to define the concepts that are in the problem space. This makes it easier to figure out how everything works, and if something does not make sense, to figure out why; what might be wrong, what might be missing. With a similar idea, the conceptual framework was constructed by structuring the concepts of thinking, problem and problem-solving, and the research objectives were addressed via the conceptual framework. The process of figuring out what belongs to the problem space was not an easy task, because the concepts involved are wide-ranging and ill-structured. The approach was to introduce thinking, problem and problem-solving limiting them in such way that what is presented in the conceptual framework is either: (i) observable in practice or (ii) in one’s cognition, or (iii) widely (to a degree) researched and recognised. However, I had to exclude physical studies directly related to brain activity and the nervous system, if the results are not directly observable in practise, because the subjects included are approached from so many directions (from so many fields of science) that one simply can’t master them all in a master’s thesis.

This kind of approach is very similar to the conceptual-analytical approach by classification of Järvinen [2004]. An ill-structured concept, computational thinking, is structured through the integrated knowledge (of thinking and problem-solving) and further analysed in relation to it. The approach to form a conceptual framework was selected because problem-solving and thinking related to it, are topics that cannot be approached via a single theoretical framework without a huge risk of disregarding something relevant. To back this up: If I had selected, say, the Newell and Simon's theory of human problem-solving (which was given as an example of an established theory by Eisenhart [1991]) for the theoretical base to reflect computational thinking with, I would have ended up with a misconception on the foundations of my work (the misconception of the problem in the Newell and Simon's theory is implicitly addressed in Chapter 3) and I would have missed a lot of fundamental things because of it.

I have made conscious effort to keep myself aware that as a person from the field of computing I am subject to bias, and I have tried to keep the approach as objective as possible. While it is not a challenge on the conscious level (I perceive that I am open to all outcomes), that might not be the case on the unconscious level. Everyone shares the same problem caused by the nature of unconscious information processes, we are not in control and we are not aware of all of the influences.

1.3. Research objectives

The objectives of the study are to:

i. Further clarify the concept of computational thinking via thinking and problem-solving.
ii. Figure out how thinking and problem-solving can be affected by exposure to computational activities and approaches; by the acquisition of computational thinking.
iii. Find out how to increase the efficiency of computational thinkers.
iv. Figure out whether computational thinking is something that could be beneficial to be taught to people.

Research objectives presented in a question form:

i. What is computational thinking via what is known of thinking and problem-solving?
ii. How thinking and problem-solving can be affected by exposure to computational activities and approaches; by the acquisition of computational thinking?
iii. How the efficiency of computational thinkers could be increased?
iv. Is computational thinking something that could be beneficial to be taught to people?

To address the last question (iv.) a bit further: Among the main aims of activities in the fields of computing is to produce (computational) artefacts to do whatever they do or are required to do. I do not take a stand, whether or not people ought to be familiarized with the construction of these artefacts from a point of view of the importance of these artefacts, but I try to figure out what kind of "side effects" (positive and negative) computational thinking adds to that package, as it is acquired through these activities.

1.4. The journey to the discovery

The base information comes from scientific literature. When I found something relevant to read, I usually went through some of the work of the authors referenced to (not only the work, or part of the work, that was referenced), that had addressed some key point(s), or something else relevant or interesting. By doing this, new information and new researchers were introduced to me. Google Scholar helped me quite a lot to find information about the numerous subtopics involved. Probably the single most used database to find sources of information was the American Psychological Association's PsycINFO. I also used the Nelli-portal that our university provides, to search social science and information science databases, but I used it a lot less than the other methods (most common use for this portal was to access papers I had found with other means). My thesis supervisor Eleni Berki and Juri Valtanen helped me greatly by suggesting and providing relevant books and papers to read. Their research of subjects related also guided me. However, none of their work is referenced. This is because the general direction of my research has been shifting during the process due to new information presenting itself. It led to sidelining (in this context) the topics they have been working with.

There are views and research of several researchers that have clearly influenced the *direction* of the discovery. To name some: Jonassen, de Bono, Tall and Dreyfus, and Perkins and his co-authors. However, I have not blindly followed any single researcher. There is a lot of material that was selected to be discarded. The material excluded also includes publications of some of the main influencers (e.g., some of the de Bono's publications and views are not included, because I could not confirm them by empirical evidence or by explicit or implicit support of research). Detailed explanation of what was not included and why, are not provided, but the general principles were discussed earlier. There is no detailed record to present of the research process that includes, e.g., the databases used and the exact search terms. This is simply because I did not keep such a record. It slipped out of my mind as I was swamped with information soon after I started the thesis process.

2. Thinking

It is quite ironical, that for a creature that is sentient, self-aware and almost constantly thinking, it is far from obvious of what exactly constitutes the activity that we are constantly performing; what is thinking. “We may know our own thoughts, but we do not know our own thinking so well” [Swartz and Perkins, 1989]. We have concepts of cognition (refers to mental processes [W1]) and metacognition (refers to knowledge and awareness of one’s cognitive processes and anything related to them, and to knowledge of cognition in general [Flavell, 1976; Krathwohl, 2002]), several frameworks and taxonomies for thinking and we have made a huge amount of observations on how the mind works or how it possibly works. We know that the environments we are exposed to mould our ways of thinking and affect our performance in thinking, while physical structures set limitations on how much we, as biomechanical machines, can advance. However, we still know very little, judged by the degree of influence we have over thinking through the knowledge we possess. Furthermore, some of the knowledge of thinking is quite speculative as absolute proofs or falsifications are not easy to conduct with a concept as ill-structured as thinking.

In this context thinking is approached in relation to problem-solving. Thus many aspects of thinking that have no explicit or clearly relevant implicit relation to problem-solving in general context, are not included (e.g., why we like certain colours more than others, why prefer round shapes and soft materials, and so on). While human information processing (arguably) bases (mostly) on the computation of the brain, we do not possess enough knowledge of this process to effectively approach problem-solving through it to form a general overview. Thus, the approach is on the level of thinking and not on the level of brain mechanics.

Thinking can be roughly divided into the process and the outcomes. While this is quite an obvious division, it is a very important one. Thinking consists of certain dispositions/attitudes, knowledge, and mental operations. Any act of thinking engages elements of all of these three components. Mental operations constitute the mental activities, and they can be divided into cognitive and metacognitive operations. Cognitive operation can also be seen as cognitive skills when they involve several cognitive operations (e.g., analogy is an operation, where analysing could also be called a cognitive skill). Knowledge component of thinking includes three main concepts: how to execute various thinking operations, knowing of the nature of knowledge itself (e.g., what is knowledge), and the domain-specific knowledge (e.g., how internal organs are positioned inside a snake). [Swartz and Perkins, 1989] Dispositions refer to, say, attitudes towards the subject of thinking, the general feeling of the day (mood), personality features (e.g., open-mindedness, scepticism, hedonism), and so on. Thinking is an activity or a skill that can be influenced and improved, and is often approached on many levels of abstraction. It is a less context-free activity than one might think.

Thinking involves automatic and controlled processing. From a philosophical standpoint, controlled processing is a tricky concept, because if one is to control or direct his thinking (processes), he should give the impetus for the thought and for the thought about the thought to be and so on, and this would result an infinite chain. It could just be the case that controlled thinking processes are the section of thinking process where we feel in charge, or where we have a higher level of metacognitive awareness of the direction, patterns and reason of what we result (i.e., if we are directed to think about why we though like we did, we can explain it, as it felt like we reasoned it in a controlled fashion). Whatever it actually is, there is a difference between it and automated processing. And these two levels of processing clearly exist.

Automatic processing is attained into a long-term store (memory), from where the automatic processes can be triggered by the appropriate inputs. They execute independently of the subject's control. They do not require attention, thus they do not use the short-term memory capacity that is used by the controlled processes. The automatic processes can be learned through or without the controlled processing. They are greatly resistant to change, difficult to alter, ignore and suppress, but they can be unlearned with considerable effort. The automatic processes can contain components that control information flow, attract attention, cause govern overt (immediately apparent) responses, and they can overwhelm the controlled processing and cause the attention to be allocated to irrelevant positions. [Schneider and Shiffrin, 1977; Shiffrin and Schneider, 1977]

Controlled processes are something that execute under control and through the attention of a subject. They utilize the short-term store (memory), and are limited by the capacity limitations of the short-term store. These capacity limitations are balanced by the benefits of the controlled processes, deriving from the ease of setting up, alter and apply in novel situations for which automatic sequences have never been learned. [Schneider and Shiffrin, 1977; Shiffrin and Schneider, 1977]

Intrapersonal (existing or occurring within the individual's mind) and interpersonal (existing or occurring between persons) information processing are meaningful concepts in thinking and problem-solving, as is internalization of concepts and skills (translation of the interpersonal form into intrapersonal processing). This is because thinking process involves intrapersonal (mental) representations of interpersonal presentations and communication involves interpersonal presentations of the mental presentations.

If one is to solve a Burr puzzle and manages to do so, it would be quite simple to describe algorithmically how the puzzle can be solved. That, however, is not the description of the process how the puzzle was solved. The algorithmic description is an outcome. Similarly, mathematics is not being created in the final and polished form, but through an intrapersonal process in which formal logic might not be directly involved. Mathematical ideas are not derived or deducted, but generated through trials and errors, partially correct and partially wrong statements, visualizations, intuition (involves automated processing), and so on. [Dreyfus, 1991, p.27; Hanna, 1991, p.59]. (This is not to state that the process is a complete chaos, it is just less well-structured than one might think.)

The outcomes (of process) are what is commonly taught to people, while the process of creation is not as commonly addressed. Dreyfus [1991, p.28] sees this as a big problem in mathematics, because people end up lacking the know-how that allows them to use their knowledge in a flexible manner to solve problems of a type unknown to them. This problem exists across domains. Ervynck [1991, p.42] and Tall [1991, p.18] emphasize that the creation of new mathematics is meaningfully different process than the common process of learning the results of mathematics and applying them for solutions.

Intra- and interpersonal information processing will be addressed more with the mathematical thinking (that is closely related to computational thinking [CTWorkshop, 2010, p.33]) because these concepts fit nicely into the same package, and I have to address both (to make the conceptual framework more complete). However, before proceeding there, generative and evaluative thinking are approached abstractly.

2.1. Critical and creative thinking

Critical and creative thinking are concepts that are hard to miss when reading about thinking in an educational context, and they are among the ones typically included in thinking skill programme(s) [Moseley *et al.*, 2005, p.24]. There are several different models for critical and creative thinking. Some approach them as dispositions (general thinking skills applied with the aim to evaluate or to generate (look, e.g., Presseisen [2001] for that kind of approach)), and others approach them as more specific sets of cognitive operations (e.g., in Integrated Thinking Model critical thinking involves three cognitive skills (analyzing, evaluating, and connecting) and so does creative thinking (synthesizing, imagining, and elaborating) [Jonassen, 1996]).

While the general concept of **critical** thinking is quite straightforward, there are still disputes concerning the exact definition [Swartz and Perkins, 1989]. Swartz and Perkins [1989] are in line with Ennis [1962], and perceive that it involves "precise, persistent, and objective analysis of any claim, source or belief to judge its accuracy, validity, or worth". They view critical thinking as a collection of specific (cognitive) operations with a clear objective. The Integrated Thinking Model approaches critical thinking similarly, but sets the aim differently: "Critical thinking involves the dynamic reorganization of knowledge in meaningful and usable ways" [Jonassen, 1996]. There are also disagreements about which (cognitive) operations are important for critical thinking [Swartz and Perkins, 1989].

I have a shared view with de Bono [1971] that critical thinking is tightly connected to creative thinking (an obvious case of this interaction is presented in Figure 2). If one is not able to generate competing alternatives (views, approaches, explanations, etc.), the evaluation is limited. Views of Swartz and Perkins [1989] support this, and also the Integrated Thinking Model supports the interaction of critical and creative thinking in the level of "complex thinking" (Figure 1).

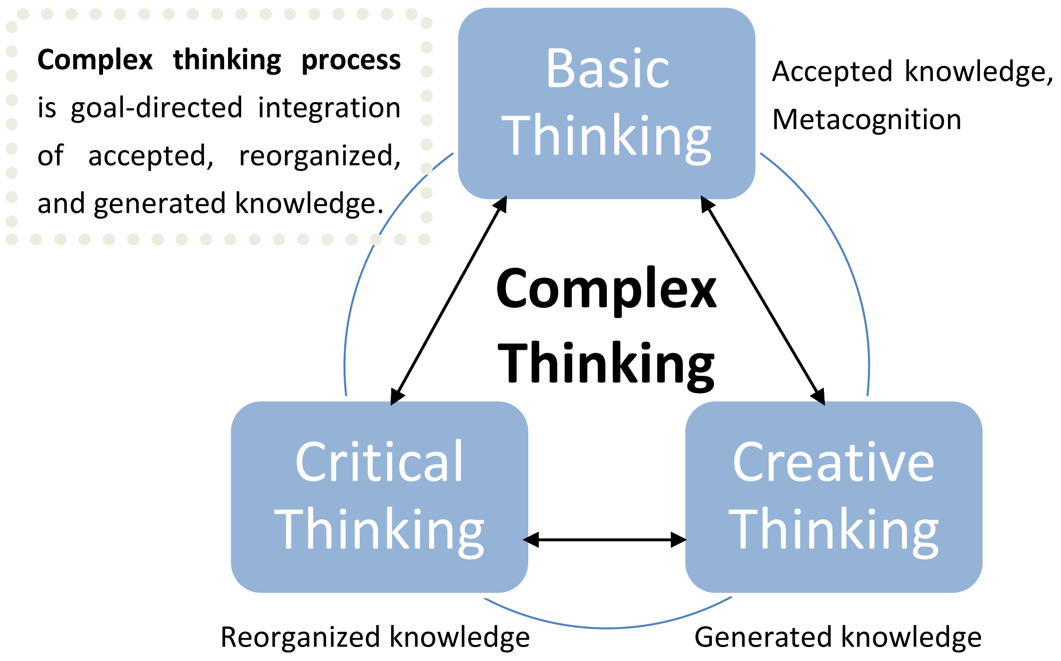
Complex thinking process is goal-directed integration of accepted, reorganized, and generated knowledge.
Basic Thinking
Accepted knowledge, Metacognition
Complex Thinking
Critical Thinking
Creative Thinking
Reorganized knowledge
Generated knowledge

Figure 1: Integrated Thinking Model (Iowa Department of Education) [Jonassen, 1996]

Richard Paul has listed some of the fundamental quality components of critical thinking to be clarity, precision, specificity, accuracy, relevance, consistency, logic, depth, completeness, significance, adequacy (for purpose), and fairness [Moseley *et al.*, 2005, p.165]. Robert Ennis sees that critical thinkers should care that their beliefs are true, their decisions are justified, and they should represent their positions honestly and clearly [Moseley *et al.*, 2005, p.153]. Presenting extensive listing of critical thinking dispositions would cause an information flood and derail the chapter. So, if interested about the topic, more detail can be easily found elsewhere.

Not so surprisingly, disagreements are also evident when it comes to the definitions of creativity [Kampylis, 2010, p.19]. "**Creative** thinking requires going beyond accepted knowledge to generate new knowledge" [Jonassen, 1996]. According to Presseisen [2001, p.50], the task of creative thinking is to create novel or aesthetic ideas or products. Creative thinking differs from the concept of creativity sometimes associated with arts, where the artistic process is automatically assumed to be creative. Creative thinking can appear anywhere e.g., in fields as well-structured as mathematics, while it might not be present in fields as ill-structured as arts. It clearly exists in relation to something, e.g., mathematical creativity does not exist in vacuum [Ervynck, 1991, p.42], but is bound to mathematical objectives. A handful of researchers consider that creative thinking is closely related to problem-solving [Swartz and Perkins, 1989]. To avoid misconceptions, I prefer to use the term **generative** thinking as a synonym for the creative thinking. It emphasizes the generative process, and it does not associate with globally unique ideas in the same way creativity does. This term is also used by some psychologists [McPeck, 1981].

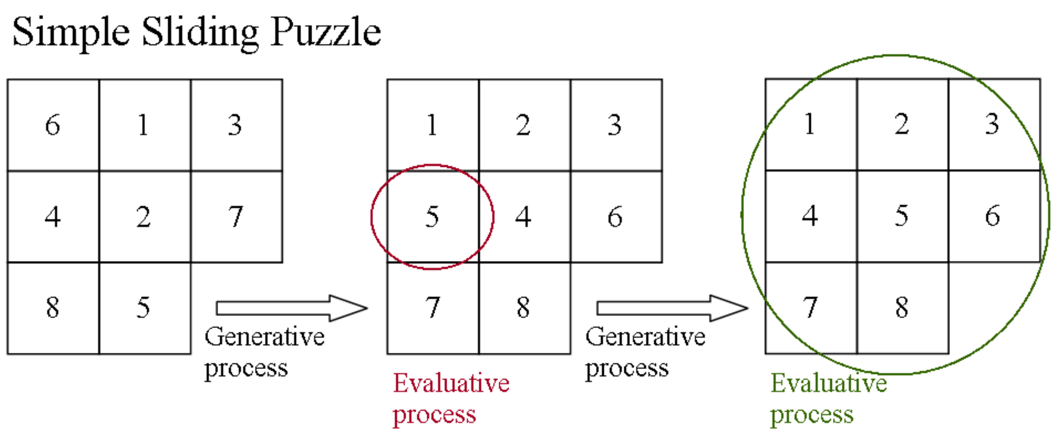

Figure 2: Concept of creative and critical thinking and their interaction, on highly abstract level

While the most novel creative acts can be quite effortlessly recognised as creative acts, the problem with creativity is, that it is hard, or even impossible, to measure [Kaufman, 2009], and the more subtle acts of creativity are not as obvious to figure out. One problematic thing is to separate the creative something (idea, solution, approach, act, etc.) from the unusual something, as the unusual something can easily be perceived as creative something without actually being creative, e.g., the outcome or the process can be copied from somewhere. It does not ease things at all that a creative thinker can commonly produce outcomes that are not globally unique, novel or creative (results that are already generated by someone else (the process is creative)).

Current creativity research mainly assumes that creativity (generativity) can be learned [Kampylis, 2010, p.55]. However, it is still open how. Some assume that creativity skills and abilities must be learned through specific introduction and training, while others assume that the creative potential is inherent in everyone; there is a need to just increase the individual's awareness of their potential [Kampylis, 2010, p.44].

de Bono has originated few creative thinking frameworks. The ones addressed here are called lateral [de Bono, 1970] and parallel [de Bono, 1994] thinking. The mind often has a habit of following the familiar paths. The key idea of lateral thinking is to restructure, escape and to provoke these fixed patterns the mind creates. Parallel thinking has similar aims, but it is more about design and dispositions/attitudes. Feelings have a bigger role in thinking that one might think. It is likely, that in a different state of mind, the mind produces different kind of reasoning (e.g., consider how thinking is affected by the current mood, say, when one is in a playful mood, he might not think as critically as usual); different dispositions results different outcomes. The importance of dispositional perspective in thinking as a source for different outcomes has been emphasized by Perkins and his colleagues for years, and it has been employed by several philosophers and psychologists [Perkins, 2001]. Some of these strategies (that I suggest to take a look at) of lateral and parallel thinking are introduced in Appendix 1 to provide concrete examples of the generative approaches to thinking. They also represent a subset of something called “the weak problem-solving methods” (concept will be introduced later).

2.2. Intra- and interpersonal information processing with the introduction of mathematical thinking

Mathematics and mathematical thinking are to be grown into. In general terms, progressive mathematizing can be seen as a sequence of horizontal and vertical mathematizing activities [Treffers, 1987]. Horizontal mathematizing means transferring a problem situation to a form that is amenable to further mathematical analysis [Treffers, 1987; Rasmussen *et al.*, 2005], and might include (but is not limited to) activities such as experimenting, pattern snooping, classifying, conjecturing, organizing [Rasmussen *et al.*, 2005, p.54] and identifying. Vertical mathematizing consists of those activities that are grounded in and built on horizontal activities and might include activities such as reasoning about abstract structures, generalizing, and formalizing [Rasmussen *et al.*, 2005, pp.54-55].

According to Tall [1992, p.1] and Rasmussen *et al.* [2005] *advanced* mathematical thinking is characterized by two important components: **precise** mathematical **definitions** (including the statement of axioms in axiomatic theories) and **logical deductions** of theorems based upon them. To translate this into terminology that will be used later: mathematical artefacts are well-defined and well-structured.

Interpersonal representations have an important function in mathematics (as they have in communication in general). The object of these representations (such as symbols and words) is to structure information to the interpersonal domain in such way that there is a relation between the representation and meaning. Mental (intrapersonal) representation refers to the internal schemata that a person has about the concept. [Dreyfus, 1991, p.30] For example, "+" is an interpersonal representation of the sum(mation) operation. We have mental schemas (intrapersonal representation) of it: what it is about, what can be done with it, what it is associated with, etc. The meaning bound to that symbol is global (to some extent), but the mental representations we have for it are local and "may differ from person to person [Dreyfus, 1991, p.30]". Thus, the well-defined and -structured mathematical concepts have more ill-structured intrapersonal counterparts (same applies in general context). What makes these mental representations so important is that they are used in thinking, not the interpersonal ones. Tall and Vinner [1981] made similar division between the individual's way of thinking of a concept and its formal definition, and Tall [1991, p.6] emphasizes the distinction between mathematics as a mental activity and mathematics as a formal system.

This distinction is one of the most important things in this conceptual framework. To emphasize it, here is another, more extreme example: Consider a normal multiplication of two integers. There is only one correct answer for the multiplication of a pair of integers and it can be given as an integer. However, there are several different ways to form the mental representations required for this process. Daniel Tammet (who is a somewhat well-known high-functioning autistic savant) is (likely) using a very different approach from you (or me): "In his mind, he says, each positive integer up to 10,000 has its own unique shape, colour, texture and feel. He can intuitively "see" results of calculations as synaesthetic landscapes without using conscious mental effort and can "sense" whether a number is prime or composite." [W2]

According to Dreyfus [1991, pp.31-33], to be successful in mathematics, it is desirable to possess rich mental representations of concepts; representations that contain many linked aspects of that concept. However, only their existence itself is not enough to allow flexible use of the concept in problem-solving. They also need to be “correctly” and strongly linked, and one has to be able to switch from one representation to other, if the other one is more competent for the next step to take. These kinds of views are supported by some considerable research which has shown that a successful problem-solver builds an internal representation of the problem in terms the solver understands [Smith, 1991]. (Because the concept of problem is not yet presented, it should be pointed out that problems as Smith refer to them, can also be wide concepts, not just puzzles, e.g., how to cure some disease.)

There are few more key operations to present. Translating is one of them. It can have many meanings, but among them is one’s ability to reformulate problems. [Dreyfus, 1991] Internalization is also related to translating, as it is involved in building new links between the interpersonal presentations and the mental representation; it involves internal structuring of the external information (transfer from a interpersonal domain to an intrapersonal domain (abstract representation of the process presented in Figure 3)). There are several more cognitive processes involved in mathematical thinking and thinking in general. Listing all of them would serve no purpose, as a list of over twenty items is hard to recall even without any additional detail. Abstraction, however, is something that will still be addressed: it is an important concept when it comes to problem-solving, it is related to computational thinking, and there is one “example” to come that involves the concept of abstraction.

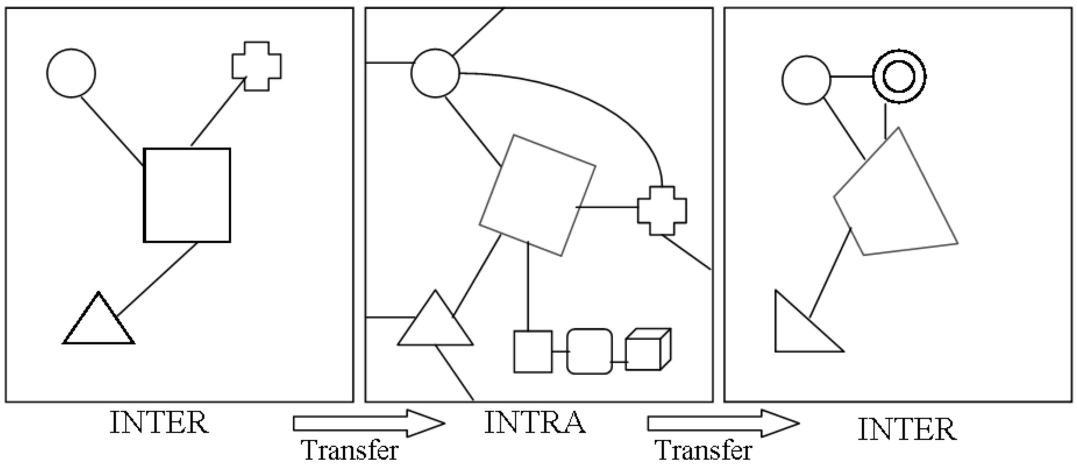

Figure 3: Abstract representation of interpersonal to intrapersonal to interpersonal transfer. Shapes represent concepts and lines represent connections.

Abstraction contains the potential for both generalization (“derive or induce from particulars, to identify commonalities, to expand domains of validity”) and synthesis (“combine or compose parts in such a way that they form a whole, an entity”), and gets its purpose mainly from this potential of generalization and synthesis [Dreyfus, 1991, pp.34-38]. de Bono [1971] addresses the use of abstraction as a level of detail, and point out that finding the suitable level of detail to approach things is important. The thing is that logical reasoning (or a solution that computes) can fail to address the reality if the abstraction level of the approach is not suitable, e.g., some crucial information is hidden because the abstraction level of the approach is too high, or there is too much detail (abstraction level is too low) to find the information required and to see the relevant connections to solve the problem.

de Bono has a concept of a black box that he calls an ignorance tool. Science is full of these black boxes (and so is programming). He gives the gravity as an example: we know its effects, how to calculate it, and how to use it with enough precision to send men around the moon, but we don't understand it. These black boxes enable us to use an effect without actually knowing the details of how it is produced. [de Bono, 1971, pp.40-46] Similarly, something he calls "meaningless words" allow us to define things on a high abstraction level: give a name to something that can, at that moment, be unknown to a great extent [de Bono, 1971, pp.24-25]. These meaningless words allow us to make definite statements and ask definite questions when we do not really know what we are talking about [de Bono, 1971, p.68].

Proof is the final concept addressed about mathematical thinking (in this chapter), because it is something that characterizes mathematics to me, it is shortly mentioned later on, and there are misconceptions about it. "A proof is a logical argument that establishes the truth of a statement beyond any doubt. A proof consists of a finite chain of steps, each one of them a logical consequence of the previous one." [Cupillari, 2005, p.3] This is the misconception. In reality, mathematicians admit that proofs can have different degrees of formal validity and still gain the same degree of acceptance [Hanna, 1991, p.55]. "A proof becomes a proof after the social act of "accepting it as a proof". This is true of mathematics as it is of physics, linguistics, and biology." [Manin, 1977, p.48] This is supported by Hanna [1991, p.58], who states that the acceptance of a theorem is a social process. Please, check [Hanna, 1991] for more details.

2.3. Feelings, perception and mistakes

de Bono [1971, p.8] states that being right is a feeling. When dealing with well-defined concepts, e.g., in mathematics or in computer science, it might be unintuitive to figure out what this actually means, as most of the time it can be figured out, whether the proposed solution is correct or not. However, when dealing with ill-structured concepts it becomes clearer what de Bono means by this statement (he does not imply that the rightness is a feeling), and what kind of roles emotions might play, especially when dealing with problems where it is very challenging, if not impossible, to know for sure if the proposed solution is the right one, or even, if the path taken is likely to lead to the solution. Emotion of being right, the gut feeling, is something not to be taken lightly; while it can be misleading (e.g., self-bias can affect the direction of thought), it can also be very useful as it can draw from past experiences (result from automated processes), e.g., in chess one can get a feeling that it would not be wise to perform the planned move after all, based on some unconsciously recognized pattern.

The role of feelings, or that feelings have a role in thinking, is not a controversial subject, e.g., changes in the strength of the feeling of knowing has been used as a factor in several studies to figure out the intuition's role in problem-solving (look Metcalfe and Wiebe [1987] and Hélie & Sun [2010]). Furthermore, it is something that should be quite easily observed from one's own cognition, e.g., it is quite unlikely that anyone who reads this has never felt unsure about the reasoning he or she have come up with (feeling of rightness is missing or it is weak, depending on how one wants to put it).

Perception is partially related to feelings (and to dispositions). Together they form a meaningful source of mistakes in the thinking process that results as errors in outcomes. According to de Bono [1976], people tend to keep the focus in logical errors, not in the errors in perception. When I earlier discussed about the abstraction, I described how a perception error in the abstraction of the approach can render (formal) logic useless. In the following, I'll present collection of sources of errors mostly based on de Bono [1971; 1976] (few are provided by me). Many of these involve emotions, perception or dispositions.

Pure perception error is conducted by observing only a part of the situation, e.g., by not including all the related components, or by abstracting incorrectly. There can be many reasons why something is not included or why some detail is missing or buried under the information. Emotional drifting can occur when a line of thought triggers emotion(s). It can, e.g., make one look deeper into something that could be misleading, or to turn thoughts away from something that might be important but unpleasant. Egocentricity and the involvement of the ego also cause emotion-based mistakes, e.g., things are approached in relation to “me” (it limits the perception), or there might be a need to be correct, which can lead to difficulty of admitting one's own mistakes and seeing the value of someone else's ideas. Initial judgement can lock the approach to some direction (limits the perception): thinking is not actually used to explore, but to support the judgement which has already been made. The initial position can be caused by the emotions toward the topic. Initial judgement is not to be confused with the initial view of what the result(s) might be. Respect, fear, presence of authority and similar influencers can cause submissive reactions, and direct thinking without intellectual reasons.

Also, there are slightly different types of perception errors, where the involvement of feelings might not be as obvious. An error in recognition can occur when something is observed in relation to some existing perception. A doctor can make a faulty diagnosis by connecting the symptoms to an illness he or she is familiar with. To put it in more general terms: if the solution to a specific problem is based on the solving mechanics of a seemingly similar problem (e.g., based on a mathematical model), which is actually somehow different, the process is likely to result in an error. “The feeling of rightness or certainty is almost inversely related to the accuracy of recognition. [de Bono, 1971, p.120]” In Chapter 4.4, an incident with the Game Show puzzle is presented. It is a concrete real life example of emotion- and perception-based sources of errors manifesting themselves in people’s views, and their willingness to hold the incorrect positions.

The uniqueness of something can also be the cause of perception-based mistakes, simply because there are no alternative options or explanations. If one is unable to generate the alternatives themselves, there is a risk that the lack of competing options causes the existing opinion to be evaluated as correct or sufficient, while it is not or might not be. It would be easy to declare that chocolate is the sweetest thing, when one has not encountered any other sweets. It is also easy to be confident in the outcomes of reasoning if the imagination does not produce alternatives. This is part of the reason why earlier on was stated that critical (evaluative) thinking is closely related to creative (generative) thinking.

Last group of perception-based mistakes presented here – does not have to involve feelings, but can – involve time-scale, magnitude and thinking with extremes. These are very straight forward. Magnitude-based mistake occur when something that works in some scale is assumed automatically to work similarly in a larger or a smaller scale. Time-scale-based mistakes are similar, but (surprisingly) related to time-scales. Other perception and feeling based mistakes exist, e.g., point of views could be addressed more closely, but I think this already gives a pretty good general perception of the subject in question.

2.4. Summary

- Thinking can be roughly separated to the process(es) and the outcomes.
- It involves cognitive operations, knowledge and dispositions/attitudes.

- Mental (intrapersonal) representations of the interpersonal concepts are used in thinking.
- It is partly controlled and partly automatic. Controlled processes are tightly capacity limited. Automatic processes can be obtained, but are not limited to be obtained, through controlled processing.
- Feelings and perception have a meaningful role in thinking. They are also a source of many errors and are involved in creative (generative) and critical (evaluative) thinking.
- Creative thinking is required for critical thinking, and vice versa.

While the chapter of thinking ends here, these concepts are revisited and further addressed in Chapter 4, as thinking and problem-solving overlap.

3. Problem

What is a problem? By a somewhat old definition, it is “a difficult question proposed for solution” [etymonline]. This might seem adequate at first, but it is not this simple to define problem. Suppose we give a solution to a problem and the solution contains no errors and it solves the given problem. Is there a problem with this? There is, as our solution could originate “problems” that are not considered problems before they are proposed for a solution. This would obviously mean that, by this kind of definition, “problems” are required to be observed before they become problems. It is safe to say that there are things that can be seen to have negative or problematic effects to some entity and thus, can be considered problematic, even if their existence has not been observed, for example, cancer.

Allen Newell and Herbert Simon stated that 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.” [Newell, 1972, p.72] Newell introduces the problem space principle, a rational activity of which people engage to solve problems, in four terms [University of Michigan]:

- a set of states of knowledge;
- operators for changing one state into another;
- constraints on applying operators;
- control knowledge for deciding which operator to apply next.

The problem space (abstract representation in Figure 4) is often represented with a tree diagram which consists of an initial state, a goal state and various intermediate positions or problem states in between. It is composed of all the possible sequences of actions that can be derived from the initial state and allowed by the operators. The problem-solver must get from the initial state to the goal state by allowed moves between problem states. [Carter, 1988] Carter calls this approach an information-processing model of the problem and considers it mostly suitable for dealing with well-structured problems, but unable to handle complex intellectual inquiry. “The fact is that in the information-processing model of problem solving, the concept of problem is unimportant and is therefore dealt with in only the most simplistic of terms. Describing a problem as existing when one is at point A but wants to be at point B says nothing about what a problem is, only that a problem exists. Thus, a problem is a problem only in terms of a specific goal.” [Carter, 1988]

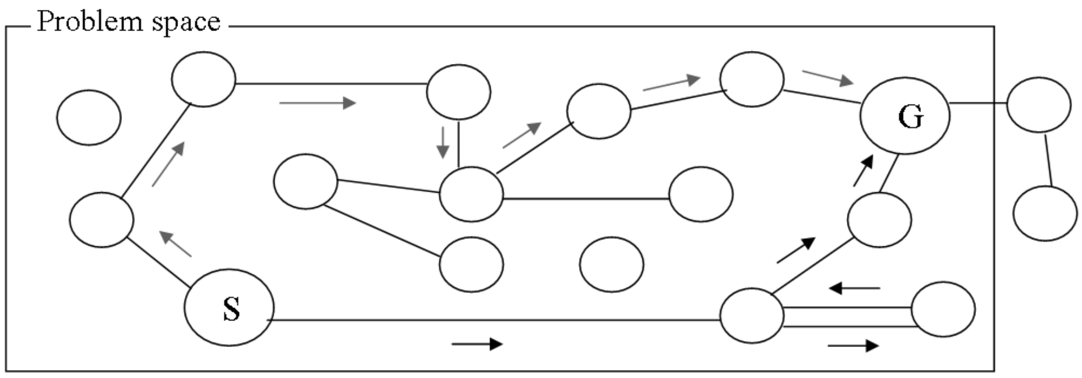

Figure 4: Abstract representation of a problem-space of a well-structured problem, and two direct solution paths (black and grey arrows) from the initial state (S) to the goal state (G). Lines represent connections between components. Notice that the black arrows make a "loop". This is to represent that the component next to the goal can only be reached by visiting the most furthest to the right (in problem space) and coming back to its left neighbour (the one accessed first (left one) leads to the next step, but only if it is expanded by the one on the right). Also, there are components outside the problem space that reach the goal state. They represent that the operations and concepts allowed to use for solving well-structured problems are limited.

Do problems exist in relation to some entity (e.g., can slippery floor turn from an observation to a problem in relation to something)? Depending on how we define relation, there are problems that are problems, just because they are declared to be problems, like some of the artificially produced puzzles. However, many problems do clearly exist in relation to something; something is or can be problematic to some entity or entities, like the slippery floor. This makes the concept of problem very ill-structured, for example, feelings can be the source of a problem, and problems and their solutions can breed totally different types of problems that might or might not be observed or might even be unobservable to a degree. As the solution for a problem can cause problems outside the problem's own problem space, or in solutions that are in same problem space, but in relation to different goals, there is a need for a different kind of description for problem than the ones presented above.

Epistemic model of the problem and problem-solving is very different from the information-processing model of the problem and problem-solving. It has four major features [Carter, 1988]:

- a problem is defined by an incongruity (confliction);
- the nature of problems as incongruity accounts for the impetus (push) for solving problems;
- problem-solving is an epistemic (knowledge-based) act, a way of learning, of coming to knowledge;
- problem and problem-solving are social concepts.

Epistemic model sees problems to be more ill-structured than the information-processing model. It takes into account that problems exist in relation to something, problems can intersect and so on. Both of the approaches are important in forming the concept of the problem, as epistemic model is not much of a use with well-structured problems and the information-processing type of approach on this level of simplicity cannot handle very ill-structured problems.

So far, there has not been an exact “short and sweet” answer to the question presented in the start of this chapter: “What is a problem?” The one presented at the beginning “a difficult question proposed for solution” can be improved quite a lot. Sockalingam *et al.* [2011] presented that “Problems are typically a set of descriptions of a phenomena or situations in need of explanations”. This is a better definition, in a way that it does not limit problems to be just this, and the “in need of explanations” does not take a stand whether the problem is observed or not. The downfall of this definition, in my opinion, is that it still states that problems are typically a set of descriptions, and while it is often the case, often it is not, as explained previously in this chapter. If one would have to state it shortly, I would say that *problems are the conflicts of interests, views, observations, and similar, of entities or entity, and some of the questions proposed for, and in the need of, answer(s) or solution(s).* However, I’d rather not try to generalize what a problem is in this manner.

3.1. Well- and ill-structured dichotomy

Reitman claims that most of the problems that humans encounter in their environment are ill-defined, one or more of their elements requiring a clearer definition before solving can take place. He suggested that "problem" may not even be an appropriate term for the ill-defined counterpart of the well-defined problem, because the process to a solution is so different. [Carter, 1988] What Reitman calls the well- and ill-defined problems, are known by many names, e.g., transformation problems and formal problems. Not to cause confusion with the use of multiple terms, in this context they are called the well- and ill-**structured** problems, and what in here is called a well- and an ill-**defined** problem, has a different meaning.

Well-**structured** problems ([Jonassen, 2000; Wickelgren, 1974, pp.10-14]):

- Present all the elements of the problem (have clearly restricted problem environment, which contains all the elements required to solve the problem).
- Have a knowable (well-defined goal state) and comprehensible solutions (correctness of the solution can be confirmed).
- Require the application of a limited number of regular and well-defined rules and principles that are organized in predictive and prescriptive ways.

To put this in simpler terms (with a little loss in accuracy), the well-structured problems have: a well-defined initial state (what is known), a known goal state (what is wanted to achieve), and a constrained set of logical operators (what kind of actions are allowed). Revisit Figure 4.

Ill-**structured** problems are not constrained by the content domains (problem environment is not restricted). The ill-structured problems are ill-structured because they ([Jonassen, 2000]):

- Involve (concepts, rules and limitations, solutions, etc.) elements that are unknown or not known with any degree of confidence.
- Might have multiple solutions, solution paths, or have no solutions at all.
- There are multiple criteria for evaluating the solution.

Basically, ill-structured problems can be seen as problems that do not classify as well-structured problems. However, problems exist in a continuum, ranging from well- to ill-structured [Carter, 1988]. This means that problems classified as well-structured have a degree of structuredness. Similarly, with ill-structured problems, there are those that are closer to be well-structured and those that are extremely ill-structured.

Reason why I use the term "well-**structured** problem" is because a well-**defined** problem can, and many times do, exist in an ill-defined environment. This makes the well-defined problems (we know where we stand and what we want to achieve) to have ill-structured structure (what we can do to reach the goal). A problem can be ill-structured, but still well-defined. There are several ways to formulate a problem, e.g., vaguely, semi-precisely, precisely but implicitly, precisely and explicitly, and problem definitions usually evolve from a less specific to a more specific [Wickelgren, 1974]. Hence, a problem that is considered well-structured, can be ill-defined to some degree, for example, some of the elements of the problem can be implicitly defined/presented (like in the Game Show problem, to be presented in Chapter 4). Some of these concepts are visualized in Figure 5.

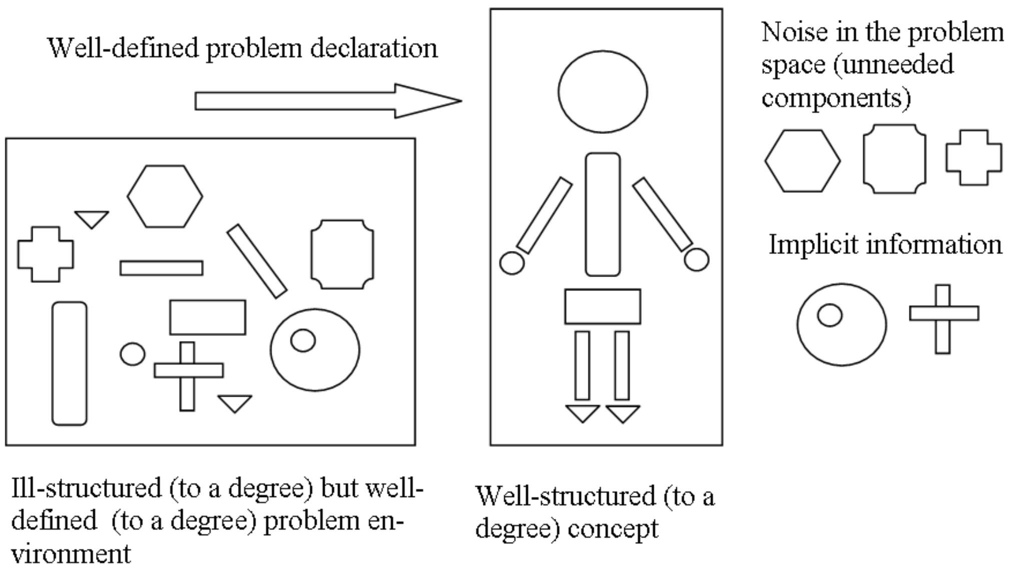
Well-defined problem declaration
Noise in the problem space (unneeded components)
Implicit information
Ill-structured (to a degree) but well-defined (to a degree) problem environment
Well-structured (to a degree) concept

Figure 5: Well-defined problem (goal state is clear) in ill-structured, but well-defined problem environment. Problem environment is well-defined because it presents all the components required for solving the problem.

To present a concrete example of the difference between a well-structured and an ill-structured problem: “Nine men and two boys want to cross a river, using an inflatable raft will carry either one man or the two boys. How many times must the boat cross the river in order to accomplish this goal? [Wickelgren, 1974, p.98]” As it is presented as a well-structured problem by Wickelgren, the answer is: 37 one-way crossings [Wickelgren, 1974, p.98]. (However, to be exact, we need to add one declaration in problem declaration: raft = boat.) If this was an ill-structured problem, 37 might not be the minimum amount of crossings, as the reality affects the rules and sets new ones. E.g., if the water was in fact water, the raft could probably support the weight of several men floating in the river while crossing it, if it would be too laborious to just swim across. Of course the safety of entering the river and the purposefulness of crossing it fast, etc., would be some of the factors to be taken into consideration when selecting the appropriate approach.

3.2. Characteristics and features

Funke [1991] characterises the complex problem-solving situation in six terms: (i) intransparency, (ii) polytely (many goals), (iii) complexity of the situation, (iv) connectivity of the variables, (v) dynamic developments, (vi) and time-delayed effects. He argues that the complex problem solving can be understood by contrasting it with "simple", non-complex problem solving, by following criteria (more in-depth description of the listing presented above):

- availability of information about the problem, that is, transparency of the problem situation;

- precision of goal definition, that is, whether a goal is defined, and whether there are multiple goals, some of which may be contradictory;
- "complexity" of the problem as defined by the number of variables, the degree of connectivity among the variables, and the type of functional relationships;
- stability of properties of the problem, that is, time dependencies in the course of the problem-solving process;
- "richness" of the problem's semantic embedding. Rich semantic embeddings often reduce the uncertainty to a large degree [Funke, 1991].

Jonassen and Hung [2008] divided problem characteristics into structuredness and complexity, and identified five components of structuredness and four major parameters of complexity. Problem difficulty can be analyzed and evaluated in terms of its complexity and structuredness dimensions. One might think that the problem difficulty can also be evaluated by how successful people are in solving the problem, and (obviously) correct they would be. However, that method is not much use with problems that have not previously been solved (these arguably are the problems most worth solving).

Components of structuredness:

- Intransparency: the higher the degree of intransparency is, the more about the problem is unknown, the more ill-structured the problem is.
- Heterogeneity of interpretations: the more open the problem is for interpretation, the more ill-structured it will be.
- Interdisciplinary.
- Dynamicity: dynamic nature of variables and operators contributes to ill-structuredness of the problem.
- Legitimacy of competing alternatives. [Jonassen and Hung, 2008]

Parameters of complexity are:

- How much domain knowledge is required.
- The degree of abstractness of the concepts.
- Intricacy of problem-solution procedures (solution path length, number of element involved).
- Relational complexity. [Jonassen and Hung, 2008]

As Jonassen and Hung's work is more recent than Funke's, and as they were familiar with Funke's and his co-authors work, it is not surprising that Funke's characteristics for the complex problems can be found in Jonassen and Hung's approach. Dynamic development and time-delayed effects can be seen to belong to the dynamicity component of structuredness and connectivity of the variables is related to relational complexity. In my view, polytely is the only one that cannot be directly associated with a single factor for structuredness or complexity defined by Jonassen and Hung, as it is on a somewhat different abstraction level. However, it can play its role in several of Funke’s characteristics (e.g., in heterogeneity of interpretations, legitimacy of competing alternatives, and degree of abstractness of the concept.).

Several more features and characteristics of the problem exist, and even more become available when the characteristics of the problem are observed in relation to the problem-solver(s), e.g., the uniqueness of the problem can be seen as a problem related feature (as global uniqueness) and it can be seen in relation to the problem-solver as well (as local uniqueness), where the perplexity of the problem is more problem-solver related (e.g., locating something by the scent can be quite a confusing challenge for a human, while it could be an easy task for a dog). However, presenting more details serves no purpose in this context.

3.3. Typology

Jonassen [2000] has gathered problem types from various sources and further structured the following typology of problems: logical problems (puzzles), algorithmic problems, rule-using problems, decision making problems, troubleshooting problems, diagnosis-solution problems, strategic performance problems, case analysis problems, design problems, and dilemmas. Some of these problem types obviously overlap with each other. He notes that this typology is not promulgated as a definitive theory, but rather as a work in progress, and welcomes experimentation, assessment and dialogue. (Look Jonassen [2000] for more details.)

Things start to get bit imprecise on the abstraction level of Jonassen’s problem typology. However, while some details might be lost, more general structure is formed and this increases our ability to communicate, observe and process similar problems. In my opinion, Jonassen’s typology is a good base for the typology of problems because it is inclusive and relatively clear. However, the direction of approach makes it not all-inclusive.

An important problem type not included in Jonassen’s typology is the “problem of no problem”. Problem of no problem is a situation where a problem exists, but is not observed, or a situation where a problem (or a better solution) exists, but it cannot be observed without restructuring the data in hand to free information which is not available from its current structure. de Bono [1970, p.53] says this kind of situation to have insight restriction. It could be argued that the problem of no problem fits under the troubleshooting problems, except troubleshooting is about making sure that something is functioning as it is supposed to. Case-analysis could find the existing problems that have not previously presented themselves. However, in case-analysis the level of restructuring might not come even close to the required level to free new information through it, meaning, while some problems might present themselves through additional information in case-analysis, the ones that are "insight restricted" might not be found.

There could be typology of generic information processing problems parallel with the problem typology presented by Jonassen. The base of all problem-solving is arguably collaborative (to be addressed in Chapter 4.3). It is one of the reasons that make the concept of intrapersonal and interpersonal information processing to be at the very core of the problem-solving; questions of how to structure, process, present, transfer and store the information, and how not to cause unnecessary noise (loss of the meaning, consistency and structuredness) when given out, taken in and while processing inside. Examples of sources of noise and mistakes have already been presented.

Some of the information processing problems could also be defined in a way they can be fitted into Jonassen's typology, for example, the describing problem (how to describe the essence of something accurately, so that other people are able to form a "picture" of what was being described), the understanding problem (how to take in the information in a way one understands its true essence) and the knowledge management problem (containing, among other, fact-finding and -verification problems). I would, however, separate these kinds of problems to their own parallel typology.

The importance of information processing in problem-solving should be something obvious, or at least somewhat obvious in accordance to what has been presented so far. To further back it up a little, Whimpey [1971] and Lochhead's checklist of errors in problem-solving: inaccuracy in reading, inaccuracy in thinking, weakness in problem analysis; inactiveness, lack of perseverance, and failure to think aloud.

3.4. Summary

- Problems exist in relation to a goal (problem declaration) or an entity or entities.
- They exist on a continuum, ranging from well- to ill- structured.
- Well-structured problems:
    - have clearly restricted problem environment, which contains all the elements required to solve the problem;

  - well-defined goal state and the correctness of the solution can be confirmed;
  - require the application of a limited number of regular and well-defined rules and principles that are organized in predictive and prescriptive ways.
- Problems can be further observed through their general characteristics, like structuredness and complexity.
- Problems can be classified into groups according to the similarities between them.

4. Problem-Solving

“Pooh looked at his two paws. He knew that one of them was the right, and he knew that when you had decided which one of them was the right, then the other was the left, but he never could remember how to begin” - Winnie the Pooh

Mayer [1998] suggests that successful problem-solving depends on three components: skill, metaskill and will. Skills basically refer to the knowledge of the nature of the concepts involved, for example, what are integers and how to add them together. Mayer states that mastering all the skill components is not enough to promote non-routine problem-solving, metaskills are required. "Metaskills (or metacognitive knowledge) involve knowledge of when to use, how to coordinate, and how to monitor various skills in problem solving." Will refers to the problem-solver’s motivation to solve the problem. Compare these to what was presented about the three components of thinking: knowledge, operations, and dispositions.

McPeck [1981] critiques de Bono’s view that thinking is a generalized skill. According to him, it is not (“good builder of sandcastle is not necassarily a good builder of everything...”). His critique is correct in relation to claim “thinking is a generalized skill”. However, de Bono clearly states that “The aim is to produce a ‘detached’ thinking skill so that the thinker can use his skill in the most effective way. [McPeck, 1981]” Thing is, thinking involves both, the generilizable components (things that can be taken out of one context and then transferred to another) and the situated components. A more detailed explanation is to follow, but stop and think for a moment about the situated and generalizable dispositions/attitudes, knowledge and mental operations.

4.1. Cognitive skill transfer

Edward Thorndike observed as early as in the beginning of 1900’s that cognitive skill transfer is quite limited, even in tasks that fall within the same function group (between similar tasks). While the absence of effective transfer was clear, Thorndike’s views of why, were questioned by several researchers, and it was observed that transfer often depends on whether the commonalities between the tasks are seen. [Singley and Anderson, 1989, pp.2-12] Unfortunately, a wide range of studies have shown that people are quite bad at noticing similarities between problems and drawing analogies [Singley and Anderson, 1989, p.20]. Routine problems (problems that are similar to those one has already learned to solve) are performed well, but failure in non-routine problems is common [Mayer, 1998, p.49]. An operator that is used automatically when embedded in a schema may not be used automatically in less familiar circumstances [Cooper and Sweller, 1987]. Perkins and Salomon [2001] suggest that there might not be a lot to transfer from skill to skill as is generally thought; knowledge and skill tend to be local rather than general (memory strategies are contextually welded to the circumstance of their acquisition [Belmont *et al.*, 1982]).

Shiffrin and Schneider [1977] observed that “the role of categories is to improve controlled search, and possibly to speed the acquisition of automatic detection". When Cooper and Sweller [1987] investigated the relations between schema acquisition and rule automation on learning and transfer, they noticed that schemas appear to heavily influence the performance on problems sufficiently similar to those previously solved, when the connection is seen. In fact, a number of studies have shown that the interdomain cognitive skill transfer can be facilitated by manipulations designed to encourage the formation of generalized rules or schemas [Catrambone and Holyoak, 1989]. Perkins and Salomon [2001] emphasize that “skill and knowledge transfer does not take care of itself”. Jonassen [2000] sees that, if one possesses a complete schema of any problem type, he can use it to solve a problem of that type. While it might not be possible to master the complete schemas of problems classified as widely as in Jonassen’s typology, it is obviously beneficial to be familiar with how to operate with different problem types. It is advantageous to have some sort of categories to increase the change of interdomain unconscious level (low road) transfer, and to make it easier to figure out the connections for the conscious level (high road) transfer. To perceive these connections, two things are required: concept of contexts and concept of method. This claim is implicitly supported by Swartz and Perkins [1989]: they guide teachers to focus on the strategy instead of the content. Thus, de Bono is not as incorrect McPeck claims, and McPeck is not completely correct in his own claim either.

It is worth mentioning that Cooper and Sweller [1987] speculated that automation may occur more slowly than the schema acquisition. E.g., people can form schemata of chess years before they master the game; before they develop strong automated perceptions to situations (effective (automated) pattern recognition). This can also be formulated that schemas have a level of sophistication, as they obviously do (e.g., in terms of meaningful connections between mental representations).

To concretize the issue with the cognitive skill transfer with concept related to computing: There is growing evidence that object-oriented programming languages and design concepts are difficult to learn. Even persons who are very experienced in procedural design methodologies have difficulties in learning object-oriented design. [Pennington *et al.*, 1995] Not that this is new information, but it is interesting when considering the skill transfer, as object-oriented use of abstraction is so close to what we already use in natural languages (look Chapter 2.2, and think about it). And yet, the transfer is difficult even for those who are familiar with the well-structured use of language and many other basic procedures of programming (of course, the existing schema of procedural programming can also have negative impact on learning by trapping thinking into that kind of approach).

Low road and high road transfer are concepts of Salomon and Perkins [1987], and they do not contain anything that is not already addressed in this chapter, but they summarize the skill transfer nicely: Transfer occurs by the low road (unconsciously) when performance practised to near automaticity in one context becomes activated by stimulus conditions in another context (notice that these include accidental discovery). And the high road (conscious level) transfer can be mediated by deliberate mindful abstraction of ideas, procedures, skills and concept involved and deliberate applications in other domains.

4.2. Uncoconscious thinking and problem-solving

Many psychological theories about problem solving and reasoning have highlighted a role of implicit cognitive processes (known also as unconscious mental processes and automatic processing). “For instance, implicit processes are often thought to generate hypotheses that are later explicitly tested [Evans, 2006]. Also, similarity has been shown to affect reasoning through processes that are mostly implicit [Sun, 1994; Sun and Zhang, 2006].” Solutions are often found by a sudden insight, yet most theories about problem-solving focus on the explicit processes that gradually bring the problem solver closer to the solution in a deliberative way. [Hélie and Sun, 2010]

Poincaré [1913/1982, p.210] made a distinction between two kinds of mathematical minds, those that are above all preoccupied with logic; ones that make it tempting to believe that they have progressed step by step, leaving nothing to change to achieve their discovery, and those guided by **intuition**; ones that sometimes at the first stroke make quick and precarious conquest. Intuition is also variously called illumination and insight [Metcalfe and Wiebe, 1987]. "The thinker senses that a problem is soluble (and perhaps what direction the solution will take), but fails to solve it on his or her first attempt; later, after a period in which he or she has been occupied with other concerns (or, perhaps, with nothing at all), the solution to the problem emerges full-blown into conscious awareness.” [Dorfman *et al.*, 1996] This phenomenon related to intuition, called **incubation,** covers two different things, according to Wallas [1926]: during incubation (i) we do not voluntarily or consciously think of a particular problem, but (ii) series of unconscious and involuntary mental events may take place during that period. To put it in other words: solutions can form subconsciously with time.

"Introspectively, intuition is one of the most compelling and obvious cognitive processes; empirically and theoretically, it is one of the processes least understood by contemporary cognitive scientists." [Reber, 1989] Intuition, incubation and implicit processes in general make it very challenging to figure out how the exposure to different things affects thinking, e.g., we cannot simply assume that something that works somehow in the controlled processing, would work exactly the same way when it gets integrated into the unconscious processes. How things affect one’s intuition and incubation is really complicated to observe as there is so much going on, and because people do not share the same initial set up (not everyone reacts similarly to the same things).

4.3. Overview of problem-solving methods and mind extension

The most important division to be made between the problem-solving methodologies is, that there are those which are applicable across domains (called weak methods), e.g., “divide and conquer”, and those that are domain specific (called strong methods), e.g., a specific method for solving the Rubik's Cube. Not surprisingly, problem-solving methods based on a specific (domain, structural, procedural, or conceptual [Jonassen, 2000]) knowledge geared towards particular situation outperform widely applicable methods [Singley and Anderson, 1989, p.26].

Another important division to be made, is between those that are targeted to guide to understand, operate and manipulate the mind (intrapersonal things), and those that are targeted to guide to observe and operate concepts, phenomena, process, objects and similar (interpersonal things). Of course, many methods fall somewhere in between, doing a little bit of both.

So, why is this later division so meaningful? As mentioned earlier on, when we learn to solve problems by solely applying strong problem-solving methods, such as mathematical formulas, we learn to apply existing solution paths to the problem, not to generate the solution paths. Figuring out the solutions paths, the generative problem-solving, is a quite different process (abstract illustration in Figure 6) that can involve (automated and controlled) use of both, strong and weak methods. This makes the role of these problem-solving methods dealing with the intrapersonal information processing (e.g., with feelings and intrapersonal structures) quite interesting. The problem-solver is always present in the problem-solving situation; he or she has always a role in the generation of the solution(s). These methods have meaningful potential to prevent feeling-, perception- and schemata-based (presented soon) mistakes, and to make use out of these things in form of generative tools. In a way, they could be seen as subcomponents of the "strong" methods of creative problem-solving. Please notice that these methods can be either strong (e.g., how to prepare mentally for a Judo competition) or weak (e.g., how to prepare mentally for competitions in general).

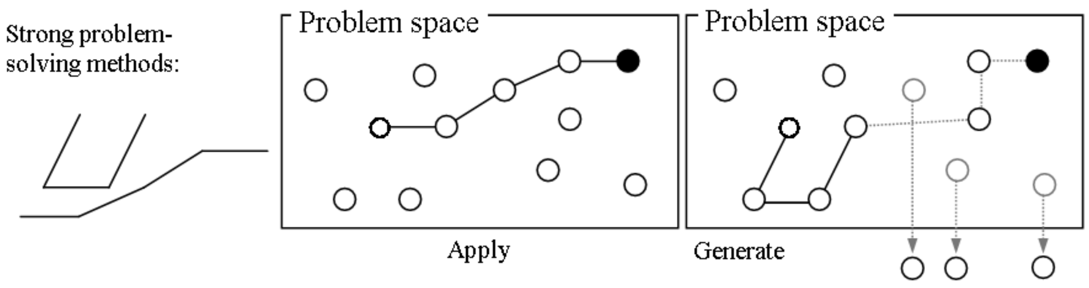

Figure 6: The solution might be the same, but the process of applying an existing solution path to solve a problem is quite different than the generative approach.

While thinking executes in the intrapersonal domain and with intrapersonal representations, all problem-solving is arguably collaborative. A large part of the mental tools, models and knowledge (e.g., scientific results) we use in problem-solving and thinking are not generated by an individual, but derived from the surrounding world (from the interpersonal domain(s)). Based on views by David Perkins [1992], David Moursund presented that the "problem-solving team" (in Figure 7) constitutes from:

- tools that extend the mental capabilities of the problem-solver,
- tools that extend the physical capabilities of the problem-solver,
- and from the education, training and experience to build one's mental and physical capabilities, to effectively use mental and physical tools individually and as a team member [CTWorkshop, 2010, p.19].

Tools that extend the mental and physical capabilities partially overlap, e.g., pen and paper (e.g., by extending the memory (mental) and by easing to formalise interpersonal communication (physical)) and calculator/computer (e.g., by changing the structure of the cognitive load (mental) and by allowing the execution of calculations that would be physically impossible (physical)).

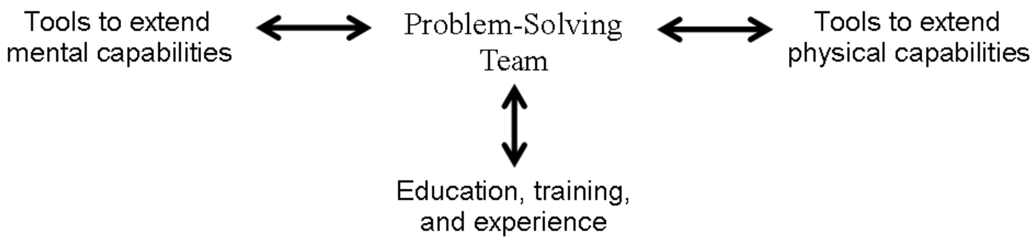
Tools to extend mental capabilities
Problem-Solving Team
Tools to extend physical capabilities
Education, training, and experience

Figure 7: Problem-Solving Team [CTWorkshop, 2010, p.19]

Obviously, with problem-solving situations that are explicitly collaborative, social things play an important role (e.g., group dynamics). For example, with design problems and decision making, it can be very important that everyone is aiming for the same goal and that the problem-solvers have shared conceptualization, joint intentions, etc. To form a shared conceptualization, transfers between intrapersonal domains and the interpersonal domain is in a meaningful role. Formalism helps, e.g., in mathematics, the several schools of thought are bound together by the concern for precision in definitions, the careful use of language and the axiomatic approach, including the intuitionists who view mathematics being essentially a languageless activity of the mind [Hanna, 1991, pp.54-55]. However, there is no point of limiting the interaction interface to a well-structured, formal language, as formal systems lack expression power and can be quite different from the intrapersonal dialogue and, thus, hard to understand. Formal representations can also take a considerable amount of time to learn. Usually it is more important that the parties involved find a common language (can be a formal language) that using a formal language to communicate with. (For example, we [Nummenmaa *et al.*, 2011] researched a (single) method on how the common "language" can be found, in interacting with a formal language (specification) in the agile software development to improve the process).

There are problem-solving methods that are not plausible to effectively perform without the assistance of a mind extension, e.g., without computer, like genetic algorithms, artificial neural nets and different kind of searches. The hi-tech problem-solving artefacts utilizing other computational problem-solving methodologies are currently evolving, e.g., IBM's Watson, that might one day be able to outperform the doctors in routine medical diagnosis-solution problems [IBM, 2011]. While some of these artefacts do not require any knowledge of computation to be used, some do, and developing these potentially very powerful problem-solving methods and artefacts certainly do. While the use of these kind of mind extension could be easily seen as something unique to computing (and to a degree it is), in fact many fields (of science) offer similar (not identical) benefits in terms of external information processing. For example, in biosciences cells are guided to form organs [Atala, 2009], and in synthetic biology life is manipulated to form itself some way differently than it naturally would: part of the problem is solved by using an external intelligence (compare these approaches to declarative programming (not a well-structured analogy, but a similarity)). Some similar approaches in different fields (of science) should be quite easy to figure out, e.g., people can solve problems in interaction with other animals.

4.4. Schemas and mistakes

This chapter is to extend Chapter 2.3 about feelings, perception and mistakes by explicitly naming an additional source of errors in thinking and problem-solving, now that this has been introduced. Schemas are important and the strong problem-solving methods are powerful, but they are also a notable source of mistakes. The problem is that while schemas can get more sophisticated and effective with time, due to skill automation they can also form a stronger grip (make the mind to wander familiar paths) and they can grow to be more emotionally trapping. While it is important to see the similarities between problems in order to use existing schemata to solve them, it is also important to perceive the differences to avoid schemata-based mistakes.

World is full of examples, but a famous incident describes the dangers of schemas nicely. It is known as the “Game Show problem” and as the “Monty Hall three doors problem”. It was presented to Marilyn vos Savant. She offered and reasoned {a correct answer}[Hogbin and Nijdam, 2010] that was hard to swallow for most, even for majority of the academics who replied [vos Savant]. (To see the failure to understand the problem in scientific publications, look Morgan *et al.* [1991] and Puza *et al.* [2005].) The problem with understanding the solution was due to a similarity with an elementary mathematical situation that one has encountered countless times before while dealing with probabilities. Basically, a familiar concept (schema about how to deal with these sorts of problems) was applied for solving the problem without taking a further look into the problem space. The perceived familiarity of the situation and the feeling of rightness toward the schema they had applied prevented people from approaching it differently.

The Game Show problem goes as follows: "Suppose you're on a game show, and you're given the choice of three doors. Behind one door is a car, behind the others, goats. You pick a door, say #1, and the host, who knows what's behind the doors, opens another door, say #3, which has a goat. He says to you, "Do you want to pick door #2?" Is it to your advantage to switch your choice of doors?" [vos Savant]

The (first) answer was (there were disputes even after a more detailed answer): "Yes; you should switch. The first door has a 1/3 chance of winning, but the second door has a 2/3 chance. Here's a good way to visualize what happened. Suppose there are a million doors, and you pick door #1. Then the host, who knows what's behind the doors and will always avoid the one with the prize, opens them all except door #777,777. You'd switch to that door pretty fast, wouldn't you?" [vos Savant]

4.5. Summary

- While thinking executes in the intrapersonal domain and with intrapersonal representations, all problem-solving is arguably collaborative. A large part of the mental tools, models and the knowledge (e.g., scientific results) we use in problem-solving and thinking are not generated by an individual, but derived from the surrounding world (from the interpersonal domains).
- The "problem-solving team" constitutes from:
    - tools that extend the mental capabilities of the problem-solver,
    - tools that extend the physical capabilities of the problem-solver,
    - from the education, training and experience to build one's mental and physical capabilities, to effectively use mental and physical tools individually and as a team member.
- Unconscious thinking has a central role in problem-solving, and it is complicated to understand and to manipulate.
- Thinking and problem-solving involve situated and generalizable components (knowledge, cognitive operations and disposition), including domain specific and general domain problem-solving methods. However, cognitive skills and knowledge by default have a tight contextual bind.
- Cognitive skill can be transferred on the conscious and on the unconscious level (high and low road), but they do not transfer easily. In general, people are bad at seeing similarities between problems. Categories and generalized schemas ease the cognitive skill transfer (helps to build the connections), but they and domain specific (strong) problem-solving methods are also sources of mistakes.
- Generative (creative) thinking is of importance when it comes to the non-routine problem-solving, especially when the problem cannot be solved with the use of strong (domain specific) problem-solving methods alone.
- *Solving well- and ill-structured problems require different cognitive skills (to be addressed in Chapter 6.1).*

5. Computing

According to Denning [2003], computing is the study of information processes, natural and artificial. He classifies, in his "The Great Principles of Computing" -framework [Denning, 2010], computing into seven categories:

- Computation - What can and cannot be computed.
- Communication - Reliably moving information between locations.
- Coordination - Effectively using many autonomous computers.
- Recollection - Representing, storing, and retrieving information from media.
- Automation - Discovering algorithms for information processes.
- Evaluation - Predicting performance of complex systems.
- Design - Structuring software systems for reliability and dependability.

What is important, in this context, is that the activities in these categories are clearly concerned only about non-intrapersonal domain information processing (this argument is further addressed later on).

In Computing Curricula 2005, computing is described differently: “In a general way, we can define computing to mean any activity of a technical nature involving computers”. However, it is notified that computing also has other meanings that are more specific and context related. People in different fields of computing have different views about what computing actually is. According to the Computing Curricula 2005, the major computing disciplines are computer engineering, computer science, information systems, information technology, and software engineering. [Computing Curricula 2005]

These views are very different, and they are not the only ones out there. Based on my experiences, I am convinced that there is no way to approach computing in a manner that would satisfy: the “academics” (e.g., those who perceive computing as Denning [2003; 2010]), the people who have more applied views about computing (e.g., those who perceive computing as it is described in the Computing Curricula [2005]), and all the other parties somehow involved (e.g., those who view computing as the computer science, and that computer science is about computers). Even if there was one commonly agreed definition for computing, one problem remains: How far apart is the definition of computing from the practices, how much they differ? This is quite important in this context, because the skill acquisition is obviously influenced by the practices.

6. Computational Thinking

Computational thinking is a relatively new concept that currently has no exact definition. There are two clearly different approaches for perceiving it, which are here called Approach 1 and Approach 2. Approach 1 is the general perception, while Approach 2 represents something more suggestive.

Approach 1:
Wing [2006] describes that "computational thinking involves solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science". She states that computational thinking is about *how humans solve problems*, not an attempt to get humans to think like computers.

Several participants of the Workshop on the Scope and Nature of Computational Thinking [CTWorkshop, 2010] perceived that computational thinking is closely related to "procedural thinking", which involves developing, representing, testing and debugging procedures. Procedures (the outcomes) are detailed step-by-step set of instructions that can be mechanically interpreted and carried out by a specific agent, such as computer or automated equipment. Cuny, Snyder and Wing have a similar view: "Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent." They, however, define that the information agent can be a machine, a human or a combination of both. [Wing, 2010] Basically, computational thinking produces outcomes that are precisely formulated; well-structured.

Google (Inc.) perceives computational thinking similarly, but has a slightly different view about what computing is. Google define computational thinking as a set of problem-solving *skills* and *techniques* that software engineers use to write programs (e.g., problem decomposition, pattern recognition, pattern generalization to define abstractions or models, algorithm design, and data analysis and visualization), and claims that they are applicable to nearly any subject [GoogleCT; Kao, 2011].

Along with a number of other workshop participants, Gerald Sussman argued that computational thinking is not equivalent to computer science [CTWorkshop, 2010]. This is a correct notion. There is a separation between the processes of thinking and the outcomes (more details are presented soon).

The report of the workshop [CTWorkshop, 2010] claims that the elements of computational thinking are reasonably well-known (in Appendix 2), given that they include the computational concepts, principles, methods, languages, models, and tools that are often found in computer science. I must point out that this list is missing a meaningful component: thinking. General view that I have formed is that the nature of thinking is not clear to everyone involved the discussions of computational thinking.

As I perceive it: Computing sets the objectives for thinking (a direction). The encountered influences (knowledge (used in a wide sense), problem-solving situations, tools, experiences and similar) together with the objectives affect thinking and shape the computational thinker. The word “computational” in “computational thinking” refers to the outcomes, not to the thinking process itself. In a way, the computer science as a study is the cradle where the thinking of the “computer scientist” develops features that could be characterized as “computational thinking” (see Figure 8). (Definition of computing was abstracted out purposely.)

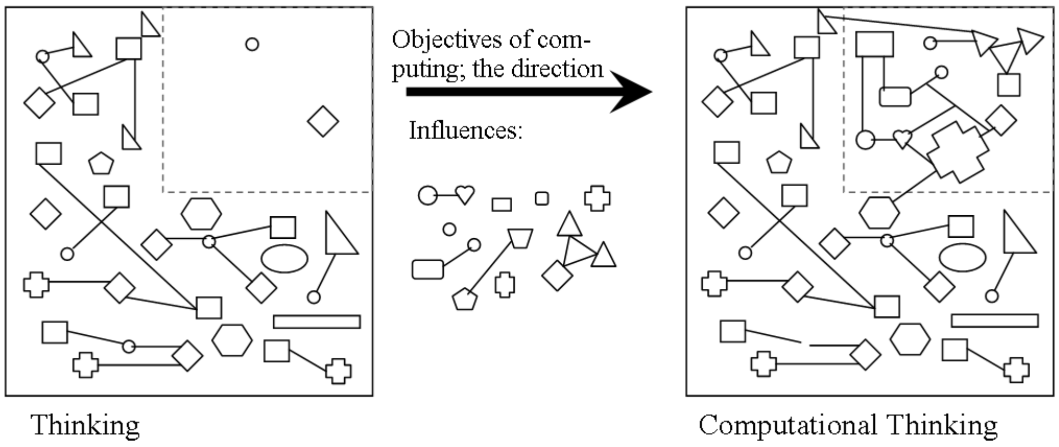

Figure 8: The effects that computing has to thinking. Notice, that while the components are quite connected in the domain area, there are only few connections out. Also, the components in the “general area” are presented in static form, not to make the image too confusing.

There is a distinction between a method and a cognitive skill. Thinking takes place in the intrapersonal domain, while computing (or computer science) as a study exists in the interpersonal domain, and so does these general purpose problem-solving methods, the collection of mental tools. To integrate these tools into thinking, inter- to intrapersonal domain transfer has to happen (Figure 3). The transfer is not a clean process; the intrapersonal representation of the interpersonal concept will not be identical (look Chapter 2.3). It is differently structured and differently connected to related (and also to unrelated) knowledge. It is likely that what was transferred in is not utilizable across domains. This means that the domain general methods of computing are not in effective across domain use by default after the interpersonal to intrapersonal domain transfer (this is the reason why in Figure 8 there is not that many connections out of the domain area).

Intrapersonal information processing includes aspects that are unconscious (e.g., intuition and incubation). Those unconscious processes have meaningful role in problem-solving (look Chapter 4.2). Thinking and problem-solving skills are not affected only by the structured information we process in a controlled fashion (e.g., a set of mental tools that we are introduced to), but also by the practical problem-solving situations that we are exposed to. As the fields of computing are vast (look Chapter 5), people do not get exposed to the same concepts, and the same concepts they do encounter are likely to be in different set ups and can have a different emphasis. This means that the mental schemas (mental representations and their connections) formed of the interpersonal knowledge (e.g., of a set of mental tools) can be meaningfully different among people in different fields of computing, even if the people shared an identical initial mental set up (which in reality is not possible).

Approach 2:
Wing and Sussman suggested that computational thinking could be seen as a meta-science about studying ways or methods of thinking that are applicable across the different disciplines. Peter Denning expressed that computer science itself is the study of information processes, and that computational thinking is a subset of computer science. [CTWorkshop, 2010]

This is an interesting notion by Denning, but does it address the reality? If computational thinking is to have the relation to thinking, it might not be a subset of computer science by his own classification (look [Denning, 2010]). While computer science studies also the natural information processes (e.g., how the brain computes, artificial intelligence and DNA translation) it, arguably, does not study thinking according to Denning’s classification. There is a difference between how the brain computes and how the mind works on the level where we experience thinking. On the level we experience thinking it is an ill-structured phenomenon. While thinking is based on the computation of the brain, we can directly interact with the brain thought the conscious level of thinking only to a degree, and not in a strictly imperative fashion. This means that if we are to study the ways and methods of thinking, we are also to study the intrapersonal information processing. To me, observed in this way, computational thinking would be in a cross section of computer science (study of the non-intrapersonal information processes) and (some branch of (yet another ill-defined concept)) cognitive science (study of the intrapersonal information processes). And it probably should not be called either, computational or thinking. For example, the study of thinking and information processes would be a lot more describing, even thought it does not sound as nice.

6.1. Cognitive impacts

The potential of the computational thinking lies in the separation of the methods (processes) and the contexts. However, while many psychologists and educators emphasize that *computer programming* can have a meaningful cognitive impact in terms of things that appear to be applicable to nearly any domain (like is the case with the “elements” of computational thinking), the track record of enhancing the cognitive skills via programming has been discouraging [Perkins and Salomon, 2001]. The problem is that the things that computing has to offer come with a contextual bind, and as we already know, cognitive skills do not generalize easily and transfer well (look Chapter 4.1).

The cognitive load of learning things that come in the same package with the information processing and problem-solving approaches that computing has to offer, can be a limiting factor when it comes to cognitive skill transfer. These “things” include (but are not limited to) practices, languages and (mathematical) notations. An overwhelming number of concepts encountered can cause a cognitive overload as the controlled processing is strictly capacity limited (look Chapter 2). It might take a considerable amount of time before one is familiar enough with the surroundings to be able to start looking behind the curtains; before there is room for separating the method and context, and the high road transfer. Also, the initial interpersonal to intrapersonal translations can be substantially distorted due to the excess information the concepts come with (e.g., they can be connected distortedly to the existing mental representations).

Earlier in his career Jonassen [1996] evaluated how computers perform as mind tools, using the Integrated Thinking Model (look Figure 1 for a general overview) as the base for the evaluation. He argued that nearly every critical, creative and complex thinking skill is engaged by building system models, and he believed that systems modelling (quite similar to software development and programming) is among the most intellectually complete activities that can be performed *in a class room*. Similarly, he saw that several critical, creative and complex thinking skills are involved in construction of databases, emphasis, however, being more on the critical and complex thinking skills.

I don’t find any reasons to disagree with him. However, when it comes to computational activities where the objectives are in computing and not in educating thinking, the activities are contextually welded and can be limited to a narrow domain. The problem with the narrow domain is that, if the challenge level does not incrementally increase, activities can become quite automated processes of applying what is already known. This would mean that the possible further positive cognitive impacts, in terms of learning to use components of thinking and problem-solving, like analyzing, synthesizing, elaborating etc., through practise, and the encounters with new problem-solving methods, would be out of the picture. Another, a more serious problem with operating in a narrow domain is the skill transfer. It can be facilitated by attempting to use the acquired skills across domains, and by forming generalized schemas, but it might not happen if acts like these are not performed.

Because the means for generative thinking and problem-solving that are directly addressed in computing are relatively limited, an interesting question arises: How likely the creative ways of thinking are to develop just by operating in an environment that has the room for creativity? That, however, is a question where the answer might not be very useful, as every one of us have the space to be creative in our lives. What about when there is room, a need and a must? Do activities in the field of computing implicitly force one to discover generative ways to think in order to function in the domain and if so, to what extent? In my view, one can develop creative ways to approach situations without being directly introduced to generative methods (they can be discovered), but direct introduction would likely result better (one’s knowledge and discoveries vs. collective knowledge and discoveries (not a fair fight)). The non-routine problem-solving can offer a platform where one must be creative in order to perform. This would require that the solution paths are at least partially generated by the problem-solver and not directly copied e.g., by applying some existing model to solve the problem. Activities in computing offer plenty of possibilities for both approaches.

From here the nature of the chapter changes, following is presented more suggestively to provide ideas and provoke thoughts.

When I was observing the similarities between computing and mathematics, I noticed that the similarities of the process are somewhat obvious, if mathematics is approached as a study that builds well-defined and well-structured knowledge artefacts of the findings that are made, that can be utilized to solve specific types of problems or to further evolve the art/science. Basically, both studies are doing somewhat the same thing, with different objectives. While this on its own might not be worth much, it gave me an idea to figure out these kinds of possible “main sources” for why the schemas, developed through activities in the fields of computing, can result mistakes when applied across domains. This would be useful information because the more aware we are of the sources of mistakes, the easier it becomes to avoid these mistakes. However, it was something I did not manage to complete (it would require a study of its own), but so-far I have identified three *possible* components:

- Well-structured problems, structures and outcomes.
- High requirement for domain knowledge (varies in different domains of computing).
- Immaterial operating environments (that set different kinds of freedoms and limitations, e.g., testing, troubleshooting and design procedures can be notably different when operating in an immaterial domain instead of a material domain).

Well-structured problems are common in computing (not to say computing is limited to them). Schraw *et al.* [1995] experimentally confirmed Kitchener's [1983] claim that the cognitive and metacognitive processes required to solve well-structured problems are insufficient to solve ill-structured problems. This was later supported in Shin *et al.* [2003] concluding that solving well-structured and ill-structured problems requires different component skills and different approaches. This can result many kinds of mistakes, e.g., one might develop a too high perception about his problem-solving ability with ill-structured problem (source of the ego/arrogance-based mistakes), one might treat definitions with ill-structured problems as they would with well-structured ones, etc.

Operating with problems where the requirements for domain knowledge are high (includes strong problem-solving methods), one can also form a too high perception of their general problem-solving ability, due to the cumulatively increasing repertory of domain specific (strong) problem-solving methods. While this obviously applies across domains, computing might be exceptionally deceiving because many of the methods involved can easily be perceived to be in effective across domain use by default, yet they are not.

While figuring out the possible sources of mistakes, I was also trying to find out how computing could positively affect the dealing with the sources of mistakes. When dealing with the well-structured concepts, one is dealing with something that one might call the absolute rightness. Dealing with the absolute rightness could be beneficial for dealing with the feelings of rightness, and obviously, with logical correctness. Clements and Gullo [1984] observed while studying the effects of computer programming on young children's cognition, that the "ability to monitor one's own thinking and realize when one does not understand may also be positively affected by computer programming environments in which problems and solution processes are brought to an explicit level of awareness and in which consequent modification of problem solutions is emphasized." Being in situations where one feels to be right, but can be proven to be wrong, might educate them to deal with the emotions. It might make one more aware that they could be wrong even if the feelings indicate otherwise.

The role of the external execution of solutions is interesting, and it can affect the problem-solving process in many ways, for example:

- It can make the problem-solver dependent on the external verification of the proposed solution. When a mathematician or a computer scientist (from the mathematical end of computing) proposes a solution to a problem, he is required to provide a justification, a proof. For a programmer, it can be enough that the solution computes, and an easy way to figure out if it does, is to try executing the proposed solution.
- Compiler (or whatever is used to execute the algorithms) can be seen to have a role as an educator, as it tells how the current version of the proposed solution executes. This can affect learning and problem-solving process, e.g., the executions can help to form a picture of how close of the required solution is and what direction the changes are taking the process to.

6.2. How to increase the efficiency of computational thinkers

Many of the weaknesses of computational thinking (in non-special cases) regarding problem-solving and thinking are related to:

- how the intrapersonal information processing is not addressed (including structuring the domain, dealing with feelings, schemas and unconscious processes),
- how the skills transfer is not addressed,
- how the inter- to intrapersonal translation is not addressed (intra- to interpersonal is),
- and how the generative thinking might not be nurtured.

This is because computing as a study is not principally interested of the intrapersonal information processing, thinking (not to be confused with the mechanics of the brain, artificial intelligences or interpersonal information processing). Following suggestions for increasing the efficiency of computational thinking are pretty straightforward:

The intrapersonal information processing should be explicitly addressed (what is known of it and how to deal with it (problem-solving and information processing methodologies that are targeted to guide to understand, operate and manipulate the human mind)), because:

- It can help to reduce perception-, emotion- and schemata-based mistakes in thinking, by making one more aware of the dangers, and more capable of recognising when being influenced.
- It can help to improve generative (and through that evaluative) thinking and problem-solving, by providing means to break out of the existing schemas, by guiding to use the dispositions/attitudes and other feelings, and by guiding to more effectively structure the intrapersonal representations.
- Increasing metacognitive knowledge and awareness (one's knowledge and awareness of their cognitions (cognitive operations and skills)) can help disconnect the methods from the contexts, which is important for skill transfer.

The cognitive skill transfer should be emphasized, so that the computational thinking would better transfer for across domains use. This can be achieved through generalization of the approaches, taking them out of the contexts where they are familiar, and with through conscious effort to use them in other domains (look Chapter 4.1). Dealing effectively with the cognitive skill transfer does not only help to export the computational thinking, but also, one's skills outside of the fields of computing would transfer better into computing activities. Increasing the awareness of similarities between problems, e.g., by introducing problem categories, helps to form the connections for the conscious level transfer between similar types of problems. Also, increasing the awareness of the differences between the operating environments, helps to avoid the schemata-based mistakes when the approaches of one domain is applied to another. It can also help to realize when the gut feeling is incorrect. These kinds of activities increase the connectivity and richness of the mental representations (look Chapter 2.2, for importance of this).

The interpersonal to intrapersonal transfer could be eased with similar means that are used for skill transfer, connecting the concepts, and using them in various contexts and structures, with a volume that does not cause a cognitive overload. It can be quite deceiving how one can learn to apply a lot of information to solve routine problems, but that is pretty much all that can be done with the information, when there has not been time to internalize it properly.

For advancing the generative problem-solving ability with objectives in computing, it is important how the encountered problems are solved. In problem-solving situations, instead of learning solely to apply solution paths to solve problems, learning to generate the solution(s) would likely result beneficial in non-routine problem-solving. With other words: study also the process, not only the solution. This should be supported by introduction of methods presented above, but in practise it could be a pretty straight forward process, e.g., instead of learning to apply some sort algorithm to some existing data structure, one could first try to generate a few algorithms of their own and after that, learn some of the existing solutions (strong methods are important to learn, in general, and from generative perspective (creative process (can) involve both, the weak and the strong methods)). While the self-produced solutions are unlikely to be as good as the existing algorithms, the different problem-solving process (generate instead of apply) might serve a higher purpose in the long run, that is the acquisition and automatisation of the generative thinking style and the problem-solving processes. However, this is not a magic trick to make one creative in an instant as skills can take a considerable amount of time to develop.

7. Conclusions

Computational thinking is (most) effective in the environments where it has been acquired from. It has notable potential to improve the “general” thinking and problem-solving ability, but there are barriers that have to be dealt with in order to meaningfully benefit from this potential. These barriers rise from the nature of thinking and they cannot be completely bypassed.

The last research question was reserved to be concluded with. So, does computing affect the thinking and the problem-solving skills to extent where teaching it to people can also be justified by the cognitive benefits it offers?

This topic requires more research, and the answer depends greatly on how the important of computing (as a field of study) is perceived, because thinking is rather contextual bind than a general activity (it is both, but the emphasis is on the first one). There is loads of potential in terms of beneficial cognitive impacts (especially due to the separation of methods and contexts), but there are also loads of unredeemed potential. There are barriers that prevent computational thinking from being what it can be assumed to be, an integration of a collection of effective multi domain problem-solving tools, encountered and introduced in the field of computing. The lack of cognitive skill transfer (skill does not travel well into computing and it does not travel well out from computing) and other “limitations” within the nature of human information processing (dealing with intrapersonal representations, involvement of feelings and unconscious processes) can be addressed to a degree, in order to make computational thinking more of what it can be seen to be. There, however, are things that cannot be affected, such as the situatedness of the cognitive demands in problem-solving (improvements in skill transfer help to deal with this problem, but does not solve it).

From the point of view where the advances in thinking and problem-solving skills are seen as secondary benefits for the actual studies (e.g., biology, history, mathematics, etc.), there should be a relatively clear understanding of what the advances and disadvantages are in every one of these fields of study, in order for the comparisons to have any real value. I don’t see it as something that can be done currently. There are just too many factors to be taken into account, the direct cognitive benefits are not the only things to be observed (and with the assessments of the cognitive impacts, we probably should be more careful than we have been).

Figuring out and agreeing upon the criteria for the framework used to evaluate the advances in the problem-solving (not thinking) ability is something not unambiguous, because of the important role the domain knowledge (includes the strong problem-solving methods) has in problem-solving. There are some general aspects we can try to measure (at least in theory, reliable practical implementations are a challenge), such as the advances in generative problem-solving and the development of metacognitive knowledge and awareness. But left would be the other side of the story, where the advances in the problem-solving ability are in relation to one's advancing within the field of study. Not all problems are equal in the terms of meaningfulness of the potential solution. This makes it a little fuzzy to define how different studies advance the problem-solving ability, because the problem-solving domain they operate in is also a meaningful factor and links the study and advances in problem-solving skills with each other. If we rate problems in relation to the meaningfulness of the potential solution(s), the domain of the study itself is likely to define the winner. And if we measure the advances with artificial, well-structured problems, we are using problems that are likely to be the least important ones to be solved. The requirements they set for domain knowledge and cognitive operations could be totally irrelevant in relation to problems more worth of solving. Also, I argue that there exist no "abstract", context free, problems to measure with. Every problem has its context, through how it is introduced and through the environment it exists in relation to.

What has been presented and concluded here should not be approach as given, unquestionable truth (very few things should), because of the ill-structured nature of the concepts involved. However, what has been presented should neither be discarded only due to the ill-structuredness involved. The concepts involved are important, no matter how unpleasant the true nature of the human information processing might be, or how hard they are to address with high standards of exactness and detail.

7.1. Future research

The key to further understand computational thinking from here on is to understand thinking (in general) more deeply. Both vertical and horizontal activities are required for widening the view and deepening the understanding of concepts involved.

Personally, at this point I am more interested in thinking and problem-solving in general than in relation to computational thinking; I am more interested in ways of advancing the problem-solving performances than, e.g., arguing about the definition of computing. I would, however, continue with the topics suggested to be the downfalls of computational thinking, and widen the view by including the physical mechanics of the brain and the nervous system; approach the topics also in biological level. Connecting mechanics of the mind with the mechanics of the brain is likely to be ambiguous to a great degree. Unconscious learning is also a topic to include, as thinking and problem-solving ability is (also) affected on the unconscious level, and there might be something very essential to be discovered.

7.2. Experiences

While the outcomes of this process are well-structured to a degree, the process itself was quite chaotic at the beginning. It was a challenge to fit all the pieces together (synthesizing the information), because the concepts involved are wide-ranging and ill-structured to a degree that they are. And, because the concepts involved are addressed in a different light and with a different emphasis within several fields of science (cognitive science, psychology, educational sciences, philosophy, biology, information sciences, etc.).

I started the process by observing problems with an aim to figure out how they could be structured and classified into forms which could be approached algorithmically, and what kind of collection of cognitive operations and skills there are to execute in those set ups. However, then I (finally) figured out that I am working with a (partial) misconception of thinking and problem-solving, and changed the direction of the study. The work I had done was not completely wasted, as it gave me a pretty good perception of the concept of a problem.

Once I managed to figure out the general picture, more or a less, I had plenty of problems putting it into words which would be understood by the others. Not because the contents were so brilliant, but because presenting wide-ranging and vastly connected concepts in a textual form is challenging for me; it is hard to perceive whether other people can see the same connections in the text that I can, whether there is enough or too much argumentation and what is obvious and what is not. My solution was to cut down on details and concentrate presenting the general principles behind the concept to build a clearer central line of thought. I also cut out some less relevant arguments, which were mostly about the details of why something that someone had presented about a related topic is not exactly accurate.

It is probably worth of mentioning that the outcomes of this study were opposite to what I had expected and would have liked them to be; I expected computational thinking to be more. (In case someone is to critique this by stating that a researcher should not have any initial positioning or perception, or anything similar: I perceive that it is a more objective approach to identify the initial positions one has and (try to) deal with them, than make the silly assumption that one has none and end up being influenced by them.)

7.3. Acknowledgement

I want to express my gratitude to Eleni Berki and Juri Valtanen for their support, it really helped me out a lot. I also want to thank my friend Matti Karjalainen and professor Erkki Mäkinen for the much needed feedback of an earlier version of the work, and my little sister Kaisa Tiensuu and friends Tuomas Uimonen, Timo Korhonen and Anniina Peltonen for helping me with finding and fixing some of the language related errors.

To end with a “lighter note”: This has been a thought provoking journey for me. It would be interesting to know, if reading this has provoked one e.g., to listen to the tone of the voice they uses in their inner dialogue for figuring out the attitudes that might (currently) affect the path of their thought, or to introduce themselves with a problem and then leave the production of the solution to subconscious thinking, incubation. :)

References

[Atala, 2009] Anthony Atala *on growing new organs* (video). TEDMED, 2009. http://www.ted.com/talks/anthony_atala_growing_organs_engineering_tissue.html (Accessed: 25.4.2012)

[Belmont *et al.*, 1982] J.M. Belmont, E. C. Butterfield and R. P. Ferretti, To secure transfer of training instruct self-management skills. In: D. K. Detterman and R. J. Sternberg (Eds.) *How And How Much Can Intelligence Be Increased?*. Ablex, 1982, 147-154.

[Carter, 1988] Michael Carter, Problem Solving Reconsidered: A Pluralistic Theory of Problems. *College English* **50** (5), 1988, 551-565.

[Catrambone and Holyoak, 1989] Richard Catrambone and Keith J. Holyoak, Overcoming Contextual Limitations on Problem-Solving Transfer. *Journal of Experimental Psychology: Learning, Memory, and Cognition* **15** (6), 1989, 1147-1156.

[Clements and Gullo, 1984] Douglas H. Clements and Dominic F. Gullo, Effects of Computer Programming on Young Children's Cognition. *Journal of Educational Psychology* **76** (6), 1984, 1051-1058.

[Computing Curricula 2005] The Joint Task Force for Computing Curricula 2005, a cooperative project of The Association for Computing (ACM), The Association for Information Systems (AIS) and The Computer Society (IEEE-CS), *The Overview Report - The Guide to Undergraduate Degree Programs in Computing*, A volume of the Computing Curricula Series, 11 April 2005.

[Cooper and Sweller, 1987] Graham Cooper and John Sweller, Effects of Schema Acquisition and Rule Automation on Mathematical Problem-Solving Transfer. *Journal of Educational Psychology* **79** (4), 1987, 347-362.

[Cupillari, 2005] Antonella Cupillari, *The Nuts and Bolts of Proofs (Third Edition)*. Elsevier Academic Press, 2005.

[CTWorkshop, 2010] Committee for the Workshops on Computational Thinking; National Research Council, *Report of a Workshop on The Scope and Nature of Computational Thinking*. The National Academies Press, 2010.

[de Bono, 1970] Edward de Bono, *Lateral Thinking*. Penguin Books Ltd, 1990.

[de Bono, 1971] Edward de Bono, *Practical Thinking*. Penguin Books Ltd, 1991.

[de Bono, 1976] Edward de Bono, *Teaching Thinking*, Penguin Books Ltd, 1991.

[de Bono, 1994] Edward de Bono, *Parallel Thinking*. Penguin Books Ltd, 1995.

[Denning, 2003] Peter J. Denning, Great principles of computing. *Communications of the ACM* **46** (11), 2003, 15-20.

[Denning, 2010] Peter J. Denning, The great principles of computing. *American Scientist* **98** (5), 2010.

[Dorfman *et al.*, 1996] Jennifer Dorfman, Victor A. Shames, & John F. Kihlstrom, Intuition, incubation, and insight: Implicit cognition in problem solving. From: http://ist-socrates.berkeley.edu/~kihlstrm/Underwood96.htm. Slightly modified version of what was published in G. Underwood (Ed.), *Implicit Cognition*. Oxford University Press, 1996, 257–296.

[Dreyfus, 1991] Tommy Dreyfus, Advanced mathematical thinking processes. In: David Tall (ed.), *Advanced Mathematical Thinking*. Kluwer Academic Publishers, 1991.

[Eisenhart, 1991] Margaret A. Eisenhart, Conceptual framework for research circa 1991: Ideas from a cultural anthropologist; implications for mathematics education researchers. In: *Proceedings of the Thirteenth Annual Meeting - North American Chapter of the International Group for the Psychology of Mathematics Education,* 1991.

[Ervynck, 1991] Gontran Ervynck, Mathematical creativity. In: David Tall (Ed.), *Advanced Mathematical Thinking*. Kluwer Academic Publishers, 1991.

[etymonline] http://www.etymonline.com, accessed 14.2.2012.

[Evans, 2006] Jonathan ST. B. T. Evans, The heuristic-analytic theory of reasoning: Extension and evaluation. *Psychonomic Bulletin & Review* **13** (3), 2006, 378–395.

[Flavell, 1976] John H. Flavell, Metacognitive aspects of problem solving. In: L. B. Resnick (Ed.), *The Nature of Intelligence*. Lawrence Erlbaum Associates, 1976.

[GoogleCT] Google - Exploring Computational Thinking
http://www.google.com/edu/computational-thinking/what-is-ct.html (28.5.2011)

[Hélie and Sun, 2010] Sébastien Hélie and Ron Sun, Incubation, insight, and creative problem solving: A unified theory and a connectionist model. *Psychological Review* **117** (3), 2010, 994-1024.

[Hogbin and Nijdam, 2010], Martin Hogbin and W. Nijdam, Letter to the Editor. *The American Statistician* **64** (2), 2010.

[IBM, 2011] Press release: *WellPoint and IBM Announce Agreement to Put Watson to Work in Health Care,*
http://www-03.ibm.com/press/us/en/pressrelease/35402.wss, accessed: 24.2.2012.

[Jonassen, 1996] David H. Jonassen, *Computers as Mindtools for Schools: Engaging Critical Thinking (Second Edition)*. Prentice Hall, 2000.

[Jonassen, 2000] David H. Jonassen, Toward a design theory of problem solving. *Educational Technology Research and Development*, **48** (4), 2000, 63-85.

[Jonassen and Hung, 2008] David H. Jonassen and Woei Hung, All problems are not equal: implications for problem-based learning. *Interdisciplinary Journal of Problem-Based Learning* **2** (2), 2008, 6-28.

[Järvinen, 2004] Pertti Järvinen, On a variety of research output types. University of Tampere Department of Computer Sciences Series of Publications D - Net

Publications, 2004. Available at: http://www.cs.uta.fi/reports/dsarja/D-2004-6.pdf.

[Kao, 2011] Elaine Kao, Exploring computational thinking at Google. *The CSTA Voice* **7** (2), 2011.

[Kampylis, 2010] Kampylis Panagiotis, *Fostering creative thinking - The role of primary teachers*, Jyväskylä Studies in Computing No 115, Jyväskylä, Finland: University of Jyväskylä.

[Kaufman, 2009] James C. Kaufman, *Creativity 101*. Springer, 2009.

[Kitchener, 1983] Karen S. Kitchener, Cognition, metacognition, and epistemic cognition. A three-level model of cognitive processing. *Human Development* **26** (4), 1983, 222-232.

[Krathwohl, 2002] David R. Krathwohl, A revision of Bloom's taxonomy: An overview. *Theory Into Practice* **41** (4), 2002.

[Lester, 2005] Frank K. Lester, Jr., On the theoretical, conceptual, and philosophical foundations for research in mathematics education. *ZDM - The International Journal on Mathematics Education* **37** (6), 2005.

[McPeck, 1981] John E. McPeck, *Critical Thinking and Education*. Martin Robertson & Company Ltd., 1981.

[Manin, 1977] Yu. I. Manin, *A Course in Mathematical Logic*. Springer-Verlag, 1977.

[Mayer, 1998] Richard E. Mayer, Cognitive. Metacognitive, and motivational aspects of problem solving. *Instructional Science* **26** (1-2) 1998, 49-63.

[Metcalfe and Wiebe, 1987] Janet Metcalfe & David Wiebe, Intuition in insight and noninsight problem solving. *Memory & Cognition* **15** (3), 1987, 238-246.

[Morgan *et al.*, 1991] J.P. Morgan, N.R. Chaganty, R.C. Dahiya, and M.J. Doviak, Let's make a deal: the player's dilemma. *The American Statistician* **45**, 1991.

[Moseley *et al.*, 2005] David Moseley, Vivienne Baumfield, Julian Elliott, Maggie Gregson, Steven Higgins, Jennifer Miller, Douglas Newton, *Framework for Thinking: A Handbook for Teaching and Learning.* Cambridge University Press, 2005.

[Newell, 1972] Allen Newell & Herbert Simon, *Human Problem Solving*. Prentice-Hall, 1972.

[Nummenmaa *et al.*, 2011] Timo Nummenmaa, Aleksi Tiensuu, Eleni Berki, Tommi Mikkonen, Jussi Kuittinen, and Annakaisa Kultima, Supporting agile development by facilitating natural user interaction with executable formal specifications. *SIGSOFT Software Engineering Notes* **36** (4), 2011, 1-10.

[Pennington *et al.*, 1995] Nancy Pennington, Adrienne Y. Lee and Bob Rehder, Cognitive Activities and Levels of Abstraction in Procedural and Object-Oriented Design. *Human-Computer Interaction* **10** (2/3), 1995.

[Perkins, 1992] David Perkins, *Smart Schools: Better Thinking and Learning for Every Child*. The Free Press, 1992.

[Perkins, 2001] David Perkins, The social side of thinking. In: *Developing Minds: A Resource Book for Teaching Thinking (Third Edition).* Association for Supervision and Curriculum Development, 2001.

[Perkins and Salomon, 2001] David Perkins and Gavriel Salomon, Teaching for transfer. In: *Developing Minds: A Resource Book for Teaching Thinking (Third Edition).* Association for Supervision and Curriculum Development, 2001.

[Poincaré, 1913/1982] Henri Poincaré, *The Foundations of Science* (translated by George B. Halstead). University Press of America, 1982.

[Presseisen, 2001] Barbara Z. Presseisen, Thinking skills: Meanings and models revisited. In: *Developing Minds: A Resource Book for Teaching Thinking (Third Edition).* Association for Supervision and Curriculum Development, 2001.

[Puza *et al.*, 2005] Borek D. Puza, David G. W. Pitt and Terence J. O'Neill, The Monty Hall three doors problem. *Teaching Statistics* **27** (1), 2005.

[Rasmussen *et al.*, 2005] Chris Rasmussen, Michelle Zandieh, Karen King, Anne Teppo, Advancing Mathematical Activity: A Practice-Oriented View of Advanced Mathematical Thinking. *Mathematical Thinking and Learning* **7** (1), 2005, 51-73.

[Reber, 1989] Arthur S. Reber, Implicit learning and tacit knowledge. *Journal of Experimental Psychology: General* **118** (3), 1989, 219–235.

[Salomon and Perkins, 1987] Gavriel Salomon and D. N. Perkins, Transfer of Cognitive Skills from Programming: When and How?, *Journal of Educational Computing Research* **3** (2), 1987.

[Schneider and Shiffrin, 1977] Walter Schneider & Richard M. Shiffrin, Controlled and Automatic Human Information Processing: I. Detection, Search, and Attention. *Psychological Review* **84** (1), 1977.

[Schraw *et al.*, 1995] Gregory Schraw, Michael E. Dunkle & Lisa D. Bendixen, Cognitive Processes in Well-Defined and Ill-Defined Problem Solving. *Applied Cognitive Psychology* **9** (6), 1995, 523-538.

[Shiffrin and Schneider, 1977] Richard M. Shiffrin & Walter Schneider, Controlled and Automatic Human Information Processing: II. Perceptual Learning, Automatic Attending, and a General Theory. *Psychological Review* **84** (2), 1977.

[Shin *et al.*, 2003] Namsoo Shin, David H. Jonassen and Steve McGee, Predictors of well-structured and ill-structured problem solving in an astronomy simulation. *Journal of Research in Science Teaching* **40** (1), 2003, 6–33.

[Singley and Anderson, 1989] Mark K. Singley and John R. Anderson, *The Transfer of Cognitive Skill*. Harvard University Press, 1989.

[Smith, 1991] Mike U. Smith, *Toward a Unified Theory of Problem Solving: Views from the Content Domains*. Lawrence Erlbaum Associates, 1991.

[Sockalingam *et al.*, 2011] Nachamma Sockalingam, Jerome Rotgans and Henk G. Schmidt, Student and tutor perceptions on attributes of effective problems in problem-based learning. *Higher Education* **62** (1), 2011, 1-16.

[Sun and Zhang, 2004] Ron Sun and Xi Zhang, Top-down versus bottom-up learning in cognitive skill acquisition. *Cognitive Systems Research* **5** (1), 2004, 63–89.

[Swartz and Perkins, 1989] Robert J. Swartz and D. N. Perkins, *Teaching Thinking: Issues and Approaches*. Midwest Publications, 1989.

[Tall, 1991] David Tall, The psychology of advanced mathematical thinking. In: David Tall (ed.), *Advanced Mathematical Thinking*. Kluwer Academic Publishers, 1991.

[Tall, 1992] David Tall, The transition to advanced mathematical thinking: functions, limits, infinity and proof. In: Grouws D.A. (ed.), *Handbook of Research on Mathematics Teaching and Learning*. Macmillan Publishing, 1992.

[Tall and Vinner, 1981] David Tall and Shlomo Vinner, Concept image and concept definition in mathematics with particular reference to limits and continuity. *Educational Studies in Mathematics* **12** (2), 1981, 151–169.

[Treffers, 1987] Adrian Treffers, *Three Dimensions. A Model of Goal and Theory Description in Mathematics Education: The Wiskohas Project.* Kluwer Academic Publishers, 1987.

[University of Michigan] *A Survey of Cognitive and Agent Architecture*, University of Michigan - Department of Electrical Engineering and Computer Science http://ai.eecs.umich.edu/cogarch0/common/theory/prob.html
(accessed: 21.1.2012).

[vos Savant] http://www.marilynvossavant.com/articles/gameshow.html, accessed 25.2.2012. Content originally published in PARADE magazine in 1990 and 1991.

[Wallas, 1926] Graham Wallas, *The Art of Thought*. Franklin Watts.

[Wickelgren, 1974] Wayne A. Wickelgren, *How to Solve Problems: Elements of a Theory of Problems and Problem Solving*. W.H. Freeman and Company, 1974.

[Wing, 2006] Jeanette M. Wing, Computational Thinking. *Communications of the ACM* **49** (3), 2006.

[Wing, 2010] Jeannette M. Wing, *Computational Thinking: What and Why?* http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf.

[W1] http://en.wikipedia.org/wiki/Cognition, accessed 20.1.2012.

[W2] http://en.wikipedia.org/wiki/Daniel_Tammet, accessed: 5.1.2012.

Lateral and parallel thinking methods/approaches

Lateral thinking methods/approaches:

- challenge the assumptions;
- try to generate alternative;
- suspend judgement to end up somewhere where the early judgement would not allow to travel;
- find dominant ideas of concepts, not to be dominated by them, but to be able to get out of the rigid patterns;
- finding the crucial factor(s) of concepts to figure out what are the reasons for holding on the current approach (crucial factor is an element of the situation which must always be included, no matter how one looks at the situation);
- fractionate for purpose of restructuring, not for explanation;
- approach things by first taking them as they are and then shaking them up to see what happens, in aim to escape standard approaches and to free information that can then come together in a new way;
- brainstorm for new ideas (creativity also exists interpersonally);
- choose an entry point and an attention area, and then start to rotate the entry points and attention areas, because there is no certainty that the solution will be found through the most obvious ones;
- use random stimulation. [de Bono, 1970, pp.58-231]

Six thinking hats is a well known framework of **parallel** thinking. There are six metaphorical hats for six approaches. Only one of the hats is used at time, and only the mode of thinking that the hat indicates is used [de Bono, 1994, pp.43-44]:

- The white hat indicates information. When the white hat is used, the focus is on laying out the information. The quality of information can range from hard checkable facts to rumours or opinions. The quality of information should be indicated, but there is no dispute, challenge or argument in this mode.
- The red hat indicates feelings, emotions, intuition and hunches. It legitimizes the expression of feelings and intuition. Even though feelings and intuition are not always correct, they can base on complex experiences.
- The black hat is for caution, for risk assessment and for criticism. It can be the most valuable of the hats, because it is essential to not to make big mistakes. However, de Bono states that the problem with the black hat is that it can be overused by those who feel it is enough to criticise. Critique only does not lead to solutions.

- Under the yellow hat there is an effort to see how things can be done when thinking positively, yet with reason and logic.
- The green hat is for creative effort. Under the green hat there is a search for alternatives and for new ideas. Above all, the green hat is concerned with possibilities.
- The blue hat is about thinking about thinking and managing of the thinking process. It is about metacognition.

Appendix 2

The elements of computational thinking, according to the Report of a Workshop on the Scope and Nature of Computational Thinking

Computational thinking might include:

- reformulation of difficult problems by reduction and transformation;
- approximate solutions;
- parallel processing;
- type checking and model checking as generalizations of dimensional analysis;
- problem abstraction and decomposition;
- problem representation;
- modularization;
- error prevention, testing, debugging, recovery, and correction;
- damage containment;
- simulation;
- heuristic reasoning;
- planning, learning, and scheduling in the presence of uncertainty;
- search strategies;
- analysis of the computational complexity of algorithms and processes;
- and balancing computational costs against other design criteria.

Following concepts from computer science also find broad applicability:

- algorithm,
- process,
- state machine,
- task specification,
- formal correctness of solutions,
- machine learning,
- recursion,
- pipelining,
- optimization.