On the cognitive process of human problem solving

Yingxu Wang *, Vincent Chiew

Abstract

One of the fundamental human cognitive processes is problem solving. As a higher-layer cognitive process, problem solving interacts with many other cognitive processes such as abstraction, searching, learning, decision making, inference, analysis, and synthesis on the basis of internal knowledge representation by the object–attribute-relation (OAR) model. Problem solving is a cognitive process of the brain that searches a solution for a given problem or finds a path to reach a given goal. When a problem object is identified, problem solving can be perceived as a search process in the memory space for finding a relationship between a set of solution goals and a set of alternative paths. This paper presents both a cognitive model and a mathematical model of the problem solving process. The cognitive structures of the brain and the mechanisms of internal knowledge representation behind the cognitive process of problem solving are explained. The cognitive process is formally described using real-time process algebra (RTPA) and concept algebra. This work is a part of the cognitive computing project that designed to reveal and simulate the fundamental mechanisms and processes of the brain according to Wang’s layered reference model of the brain (LRMB), which is expected to lead to the development of future generation methodologies for cognitive computing and novel cognitive computers that are capable of think, learn, and perceive.

© 2008 Published by Elsevier B.V.

Keywords: Cognitive informatics; Cognitive computing; Brain informatics; Computational intelligence; Reference model of the brain; Cognitive processes; Problem solving; Mathematical model; Concept algebra; RTPA

1. Introduction

The attempt to understanding the advance intelligent ability of human beings for problem solving has intrigued researchers from multiple disciplines, which can be traced back to the Aristotle’s era (384–322BC). Problem solving is identified as one of the basic life functions of the natural intelligence of the brain (Polya, 1954; Wallas, 1926; Wang, 2008d; Wilson & Clark, 1988). Most decisions that an individual makes everyday are related to certain problems needed to be solved no matter how trivial or critical they are. Problem solving as a process may be embodied in many forms, where research itself is essentially a typical problem solving paradigm (Beveridge, 1957).

theory is correct, but may demonstrate so by using computer simulations. In 1972, Newell and Simon developed a computer program called the general problem solver (GPS), which was quite successful in solving limited types of carefully defined problems (Newell & Simon, 1972). In his paper entitled “Developing a Computational Representation for Problem-Solving Skills,” Ira Goldstein mentioned that computer technology could be of use as personal assistants in problem solving, and to provide a cognitive programming and simulating environments, to achieve a deeper understanding of the subject matter and to explore learning strategies (Goldstein, 1978).

Cognitive informatics is the transdisciplinary study of cognitive and information sciences that investigate into the internal information processing mechanisms and processes of the natural intelligence – human brains and minds – and their engineering applications (Wang, 2002b, 2003a, 2003b; Zadeh, 2008a; Zadeh, 2008b; Zhong, 2006). Cognitive informatics provides a coherent framework of contemporary theories for explaining human cognitive processes, such as problem solving, learning, decision making, and consciousness. A layered reference model of the brain (LRMB) is developed in Wang et al. (2006), which reveals that the brain and human intelligent behaviors may be explained by 39 cognitive processes at six layers known as the sensation, memory, perception, action, meta-cognition, and higher cognition layers. It is recognized that, in order to enable rigorous modeling and description of the brain and its cognitive mechanisms, the formal structures of internal knowledge representation and new forms of denotational mathematical means are essential and necessary. The former is implemented by the object-attribute-relation (OAR) model (Wang, 2007c) on the basis of neural informatics (Wang, 2007b). The latter is expressed by a collection of denotational mathematics (Wang, 2007a, 2007d, 2008a), such as concept algebra (Wang, 2008b) and real-time process algebra (RTPA) (Wang, 2002a, 2003b, 2008c), which is a category of expressive mathematical structures for dealing complex mathematical entities shared by many modern scientific and engineering disciplines.

In cognitive informatics, problem solving is identified as a cognitive process of the brain at the higher cognitive layer that searches a solution for a given problem or finds a path to reach a given goal (Wang, 2007b). Problem solving is one of the 39 fundamental cognitive processes modeled in the LRMB model (Wang et al., 2006). It is recognized that there is a need to seek an axiomatic and rigorous model of the cognitive process of human problem solving in order to develop a solid and coherent theoretical foundation for integrating various theories, models, and practices of problem solving (Wang, 2007b).

This paper presents a formal model of human problem solving and its cognitive process. It will proceed in Section 2 with literature surveys on problem solving and related work developed in psychology, cognitive science, and computational intelligence. In Section 3, a set of cognitive, mathematical, and process models of problem solving will be developed and elaborated. Section 4 introduces useful paradigms of denotational mathematics in the forms of concept algebra and RTPA. Based on them, the process model of problem solving is formalized for machine simulations in computational intelligence. Applications of this work in brain simulation and cognitive informatics are discussed toward the development of future generation cognitive computers.

2. Cognitive psychology of problem solving

This section surveys related studies on problem solving and its cognitive foundations in psychology and computational intelligence. The approaches to and factors affecting problem solving are explored. The cognitive characteristics of problem solvers and the impact of their knowledge in problem solving are discussed. A set of empirical problem solving procedures proposed in the field of psychology is comparatively analyzed, after the presentation of the context framework of problem solving in form of the LRMB model.

2.1. Approaches and factors in problem solving

It is proposed that a problem consists of three components known as the givens, goals, and operations (Ormrod, 1999; Polya, 1954). The givens are information available as part of the problem. The goals are defined as the desired termination state of a solution to the problem. The operations are potential actions that can be executed to achieve the goals of a solution. For any given problem there is an associated problem space (Bender, 1996; Wilson & Clark, 1988), which is all the possible goals and paths potentially related to the problem known by a problem solver. A solution to a certain problem might not exist within the solver’s current solution space. This could be caused by many factors, such as that the problem could be ill-defined, expected goals are ambiguous, and/or no method (path) is available that links the give problem object to the goal(s).

A similar perception on the problem space is presented by Tuma and Reif (1980), as well as Payne and Wenger (1998), which identified two elements in problem solving: (a) a description of all possible states of the task and problem solver (representation); and (b) a list of the ways of moving among those states (search). The first element supports a problem solver to understand the problem by abstraction and identification. The second element enables the problem solver to search for a possible solution in memory.

A various of approaches to problem solving have been studied and proposed in psychology, cognitive informatics, and computational intelligence (Matlin, 1998; Ormrod, 1999; Rubinstein & Firstenberg, 1995; Wang et al., 2006; Wang & Ruhe, 2007) as follows, inter alia:

- Direct facts – finding a direct solution path based on known solutions.
• **Heuristic** – adopting rule of thumb or the most possible solutions.
• **Analogy** – reducing a new problem to an existing or similar one for which solutions have already been known.
• **Hill climbing** – making any move that approaches closer to the problem goal step by step.
• **Algorithmic deduction** – applying a known and well-defined solution for a problem.
• **Exhaustive search** – using a systematic search for all possible solutions.
• **Divide-and-conquer** – solving a whole problem via decomposing it into a set of subproblems.
• **Analysis and synthesis** – reducing a given problem to a known category and then finding particular solutions.

Adoptions of the above approaches to problem solving may not guarantee a problem goal to be reached, especially when a solution is not within the problem solver’s solution space. There are a number of factors that may hinder the process of problem solving (Matlin, 1998; Smith, 1991) such as: (a) mental set in which a fixed or improper method is adopted for a new problem while easier solutions could have been utilized; (b) meta-cognition in which a problem solving process may require the support of other metacognitive processes to achieve the solution goal; and (c) lack of knowledge in which either the problem or the goal could not be well represented or modeled, and no method or solution could be applied to the problem.

### 2.2. Cognitive characteristics of problem solvers

The typical psychological traits that may be of use and of benefit to a successful problem solver are as follows:

1. To correctly identify problem goals, to be persistent, to adopt efficient strategies in search, and to be able to trace back to a certain previous point in the solution process.
2. In cognitive psychology, what differentiates an expert from a novice problem solver is studied. It is observed that not everyone possesses the same ability for problem solving. These differences may provide insight to explain the nature of problem solving. The most significant traits between experts and novices in problem solving are identified as follows (Payne & Wenger, 1998; Polya, 1954; Smith, 1991): scope of knowledge on accumulated information, problem solving schemas, skills, expertise, memory capacity, problem representation ability, abstraction, and categorization abilities, analysis, and synthesis skills, long-term concentration ability, motivation, efficiency, and accuracy.

It is interesting to contrast and analyze the differences between professionals and amateurs in software engineering problem solving. Professional software engineers are persons with professional cognitive models and knowledge on software engineering. They are trained with: (a) fundamental knowledge that governs software and software engineering practices; (b) basic principles and laws of software engineering; (c) proven algorithms; (d) problem domain knowledge; (e) problem solving experience; (f) knowledge about program development tools/environments; (g) solid programming skills in multiple programming languages; and (h) a global and insightful view on system development, including its required functionalities as well as exception handling and fault-tolerance strategies. However, amateurish programmers are persons who know only one or a couple of programming languages but lack formal training. They may be characterized as follows: (a) ad hoc structure of programming knowledge; (b) limited programming experience and skills; (c) eager to try what is directly required before a system architecture is designed; and (d) tend to focus on details without a global and systematic view.

In programming, Richard Mayer identified four aspects of knowledge required for solving programming problems known as the syntactic, semantic, schematic, and strategic knowledge (Mayer, 1992). He reported that there exist significant differences between the knowledge structures of experts and novices in all the four categories.

It is noteworthy that unlike a machine, human abilities for problem solving do change depending on ages and external influences (Payne & Wenger, 1998). A senior person may have a broader knowledge base leading to better performance in problem solving. However, elderliness may reduce the efficiency in problem solving physically and psychologically. In addition, personal motivation, attitude, and external influences such as social pressure and environment inconvenience may hamper human problem solving efficiency and effectiveness (Wilson & Clark, 1988).

### 2.3. LRMB: the context framework for problem solving

It is noteworthy that the cognitive process of problem solving is not a trivial and isolated mental process rather than a complex and dynamic one interacting with all other cognitive processes at the meta-cognition, higher cognition, and lower layers according to the LRMB model as shown in Fig. 1 (Wang et al., 2006).

A variety of life functions and their cognitive processes have been identified in cognitive informatics, cognitive neuropsychology, cognitive science, and neurophilosophy. In order to formally and rigorously describe a comprehensive and coherent set of mental processes and their relationships, the hierarchical LRMB model is established, which encompasses 39 cognitive processes at seven layers known as the sensation, memory, perception, action, meta-cognition, meta-inference, and higher cognitive layers from the bottom up.

LRMB reveals that the hierarchical life functions of the brain can be divided into two categories: the subconscious and conscious life functions. The former encompasses the layers of sensation, memory, perception, and action (Layers 1–4). The latter includes the layers of meta-cognitive,
2.4. Problem solving procedures

Wallas (1926) and Polya (1954) proposed two sets of classical problem solving procedures. The former studied problem solving in a creativity context; while the latter proposed a generic empirical process of problem solving.

Wallas’ creative problem solving procedure is known as follows (Wallas, 1926):

(a) **Preparation**: defining the problem and gathering information relevant to its solution.
(b) **Incubation**: thinking about the problem at a subconscious level while engaging in other activities.
(c) **Inspiration**: having a sudden insight into the solution of the problem.
(d) **Verification**: checking to be certain that the solution is correct.

Following Wallas’ work, an influential problem solving procedure was proposed by Polya as described below (Polya, 1954):

(a) **Understanding the problem**: identifying the problem’s knowns (givens) and unknowns and, if appropriate, using suitable notation, such as mathematical symbols, to represent the problem.
(b) **Devising a plan**: determining appropriate actions to take to solve the problem.
(c) **Carrying out the plan**: executing the actions that have been determined to solve the problem and checking their effectiveness.
(d) **Looking backward**: evaluating the overall effectiveness of the approach to the problem, with the intention of learning something about how similar problems may be solved on future occasions.

It is apparent that both Wallas and Polya derive their conceptual procedures for problem solving based on introspections and informal observations. However, mathematical abstraction, symbolic reasoning, and formalism have not been introduced to rigorously model the cognitive process of problem solving in order to enable machine simulation of the human cognitive process. These important topics will be addressed in the following sections throughout this article.

3. Cognitive informatics models of problem solving

As reviewed in previous sections, the lack of suitable mathematical models and formal inference treatments has kept studies on problem solving at the empirical level based on observations and subjective interpretations. However, mathematical abstraction, symbolic reasoning, and formalism have not been introduced to rigorously model the cognitive process of problem solving in order to enable machine simulation of the human cognitive process. These important topics will be addressed in the following sections throughout this article.
of empirical studies in cognitive psychology and computational intelligence.

3.1. The mathematical model of problem solving

**Definition 1.** A problem space, or solution space, $\Theta$ is a Cartesian product of a nonempty set of problem objects $X$, a nonempty set of paths $P$, and a nonempty set of goals $G$, i.e.:

$$\Theta \equiv X \times P \times G$$  \hspace{1cm} (1)

where $\times$ represents a Cartesian product.

**Definition 2.** Assuming the layout of a problem solving process is a function $f: X \rightarrow \cdot \cdot \cdot \rightarrow Y$ on $\Theta$, the problem $\rho$ is the domain of $f$, $X$, in general, and a specific instance $x, x \in X$, in particular, i.e.:

$$\rho \equiv (X[f: X \rightarrow \cdot \cdot \cdot \rightarrow Y]), \quad \rho \in X$$  \hspace{1cm} (2)

Eq. (2) denotes that, in problem solving, a problem $\rho$ is the fix point of a denotational function in general, and the input of the function in particular. The former represents the broad sense of the problem, and the latter is the narrow sense of the problem.

Problem solving is a process that seeks the generic function for a layout of problem, and determines its domain and codomain. Then, a solution in problem solving can be perceived as a concrete instance of the given function for the layout of the problem.

**Definition 3.** Problem solving is a cognitive process of the brain that searches or infers a solution for a given problem in the form of a set of paths to reach a set of expected goals.

**Definition 4.** A goal $G$ in problem solving is the terminal result $Y$ of satisfactoriness in the solution space of the problem $\rho$, which deduces $X$ to $Y$ by a sequence of inference in finite steps, i.e.:

$$G \equiv (Y[X \rightarrow \cdot \cdot \cdot \rightarrow Y]), \quad G \in G$$  \hspace{1cm} (3)

**Definition 5.** A path $P$ in problem solving on $\Theta$ is a 3-tuple with a nonempty finite set of problem inputs $X$, a nonempty finite set of traces $T$, and a nonempty finite set of goals $G$, i.e.:

$$P \equiv (X,T,G) = X \times T \times G$$  \hspace{1cm} (4)

where the a trace $t \in T$ is an internal node or subpath, $t:X_{i} \rightarrow Y_{j}$, that maps an intermediate subproblem $X_{i}$ to a subgoal $Y_{j}$.

According to Definitions 1–5, there are two categories of problems in problem solving: (a) **the convergent** problem where the goal of problem solving is given, but the paths of problem solving is unknown; and (b) **the divergent** problem where the goal of problem solving is unknown and the path of problem solving are either known or unknown.

The combination of the above cases in problem solving can be summarized in Table 1, which identifies four types of problem solving, i.e., proof, instance, case study, and explorative/creative problem solving. A special case in Table 1 is that when both the goal and path are known, the case is only a solved instance for a given problem. In a related work (Wang, 2008d), the cognitive process of creation is recognized as a novel or unexpected but useful solution to a given problem. Therefore, a creation may be perceived as a special novel solution where the problem, goal, and/or path are usually unknown. With this view, the study of the generic theory of creativity can be reduced to the theory of problem solving.

**Definition 6.** A solution $s$ to a given problem $\rho$ on $\Theta$ is an instance of a set of selected relation or function, $S$, which is a subset of the solution paths in $P$, i.e.:

$$\{ S =^\wedge (X,T,G) \subseteq P, X,T,G \neq \emptyset \}$$

$$\{ s \in S \}$$

According to Definition 6, in case $\#X = 0$, $\#G = 0$, or $\#T = 0$, there is no solution for the given problem. For a convergent problem, i.e. $\#G = 1$, the number of possible solutions is $\#X \cdot \#T$. This leads to the following theorem.

**Theorem 1.** The polymorphic solution principle for problem solving states that the size of the solution space (SS, SS $\subseteq \Theta$), $N_{SS}$, of a given problem $\rho$ is a product of the numbers of problem inputs $N_{x}$, traces $N_{t}$, and goals $N_{g}$, i.e.:

$$N_{SS} = N_{x} \cdot N_{t} \cdot N_{g} = \#X \cdot \#T \cdot \#G$$  \hspace{1cm} (5)

Theorem 1 provides a fundamental mathematical model for problem solving, which reveals that the factors determining a solution to a given problem are the Cartesian space of all possible goals $G$, problem inputs $X$, and solution paths $P$ of the problem.

The polymorphic characteristic of the solution space contributes greatly to the complexity of problem solving. It is noteworthy that the path $p(x,t,g) \in P$ in Definition 5 can be a simple or a complex function. A complex function that mapping a given problem into a solution goal may be very complicated depending on the nature of the problem.

**Corollary 1.** The divide-and-conquer principle for problem solving states that the efficiency gain or complexity reduction

<table>
<thead>
<tr>
<th>Type of problem</th>
<th>Goal</th>
<th>Path</th>
<th>Type of solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convergent</td>
<td>Known</td>
<td>Unknown</td>
<td>Proof (specific)</td>
</tr>
<tr>
<td>Divergent</td>
<td>Unknown</td>
<td>Known</td>
<td>Case study (opened)</td>
</tr>
<tr>
<td></td>
<td>Unknown</td>
<td>Unknown</td>
<td>Explorative/creative (opened)</td>
</tr>
</tbody>
</table>
for a given problem $\rho$ is proportional to how many subproblems, $n$, that $\rho$ is partitioned, and/or how many path segments, $m$, that the solution path $P$ is partitioned, i.e.:

\[
\begin{align*}
\eta_p &= 1 - \frac{n}{m} = 1 - \frac{1}{n} \\
\eta_p &= 1 - \frac{\sum_{i=1}^{m} N_i}{m} \approx 1 - \frac{n \cdot \eta_p}{(m / C_0)} = 1 - \frac{m}{(m / C_0)}
\end{align*}
\]  

(7)

where $N_p$ is the average size or alternatives of all subpaths in $SS$.

Corollary 1 provides a mathematical model that formally explains the long-existing empirical heuristic principle of divide-and-conquer in problem solving for both human and computing systems. According to Corollary 1, it can be expected that the greater the partitions, the higher the efficiency gains in both ways. For instance, given $n = 3, m = 3$ and $N_p = 5$, the efficiency gains of $\rho$ and $P$ partitions are $\eta_p = 50\%$ and $\eta_p = 88\%$, respectively. However, when $n = 5, m = 5$ and $N_p = 5$, the efficiency gains of $\rho$ and $P$ partitions are $\eta_p = 75\%$ and $\eta_p = 99.2\%$, respectively.

3.2. The cognitive process of problem solving

On the basis of the mathematical models of problem solving as established in Section 3.1, the cognitive process of problem solving can be derived logically in this subsection. In the problem solving process, the representation of the problem is crucial. Problem representation includes a description of the given situation, predefined operators for changing the situation, and assessment criteria to determine whether the goal has been achieved. In most problems of interest, the solution space can be too large to be searched exhaustively, because it was noted in (Wang, Liu, & Wang, 2003) that the maximum human memory capacity is up to $10^{8.432}$ bits in the brain. Typical approaches to reduce search complexity in problem solving, according to Corollary 1, are to adopt divide-and-conquer and heuristic principles in order to simplify the problem into subproblems and/or to reduce the paths into a set of short ones.

On the basis of Definitions 1–6, a generic problem solving process is modeled as shown in Fig. 2, where the cognitive process of problem solving can be divided into the following five steps:

(a) To define the problem: This step describes the problem and its input layout $X$ by identifying its object $O_X$ and attributes $A_X$ in a sub-OAR model (Wang, 2007c).

(b) To search the solution goals and paths: In this step, the brain performs a parallel search for possible goals $G$ and paths $T$ for the solution. External memory and resources may be searched if there is no available or sufficient $G$ or $T$ in the internal knowledge of the problem solver.

(c) To generate solutions according to Eq. (6): This step forms a set of possible solutions by a Cartesian product $S = X \times T \times G$.

(d) To select suitable solutions: This step evaluates each possible solution $s \in S = X \times T \times G$ as obtained in Step (c). Recursive searching actions may be executed if the obtained solution(s) could not satisfy the problem solver’s expectation.

(e) To represent the problem solving result: This step incorporates and memorizes the solution(s), $s \in S$, as the result of problem solving into the entire OAR model in the long-term memory of the problem solver. Before memorization, the solution $S$ is represented as a part of the relations, $R$, in the sub-OAR model.

It is noteworthy in Fig. 2 that a number of lower layer cognitive processes, as represented by double-ended boxes, are adopted to carry out the problem solving process. Based on the discussions in Section 2.3, the relationship among the problem solving process and other cognitive processes as modeled in LRMB will be discussed in Section 4.3.
4. Formal description of the cognitive process of problem solving in denotational mathematics

Before the formal description of problem solving is presented, the structures of denotational mathematics (Wang, 2007a, 2007d, 2008a), particularly RTPA (Wang, 2002a, 2003b, 2008c) and concept algebra (Wang, 2008b), are introduced in this section.

Definition 7. Denotational mathematics is a category of expressive mathematical structures that deals with high-level complex mathematical entities beyond numbers and sets, such as abstract objects, complex relations, behavioral information, concepts, knowledge, processes, and systems.

4.1. Formal modeling of human behaviors using RTPA

RTPA is a denotational mathematical structure for algebraically denoting and manipulating system and human behavioral processes and their attributes by a triple, i.e.:

\[ \Xi \triangleq (\mathcal{I}, \Psi, \Re) \]  

where \( \mathcal{I} \) is a set of 17 primitive types for modeling system architectures and data objects, \( \Psi \) a set of 17 meta-processes for modeling fundamental system behaviors, and \( \Re \) a set of 17 relational process operations for constructing complex system behaviors.

Definition 9. The RTPA type system \( \Xi \) encompasses 17 primitive types elicited from fundamental computing needs, i.e.:

\[ \mathcal{I} \triangleq \{N, Z, R, S, BL, B, H, P, TI, D, DT, RT, ST, \text{@eS}, \text{@TM}, \text{@intS}, \text{@S} \} \]

where all types in \( \mathcal{I} \) have been defined in Wang (2007a).

A meta-process in RTPA is a primitive computational operation that cannot be broken down to further individual actions or behaviors. A meta-process serves as a basic building block for modeling software behaviors. Complex processes can be composed from meta-processes using process relational operations.

Definition 10. The RTPA meta-process system \( \Psi \) encompasses 17 fundamental computational operations elicited from the most basic computing needs, i.e.:

\[ \Psi \triangleq \{=, \land, \implies, \leftarrow, \triangleright, \leftarrow\text{>, } \leftarrow\text{<}, \leftarrow\text{>}\text{<}, @, \triangle, \uparrow, \downarrow, !, \otimes, \otimes, \} \]  

where the 17 meta-processes stand for assignment, evaluation, addressing, memory allocation, memory release, read, write, input, output, timing, duration, increase, decrease, exception detection, skip, stop, and system, respectively.

A process relation in RTPA is an algebraic operation and a compositional rule between two or more meta-processes in order to construct a complex process, which is elicited from fundamental algebraic and relational operations in computing in order to build and compose complex processes in the context of real-time software behaviors.

Definition 11. The RTPA process relation system \( \Re \) encompasses 17 fundamental algebraic and relational operations elicited from basic computing needs, i.e.:

\[ \Re \triangleq \{\rightarrow, \cap, |, \ldots, R^*, R^+, R^, \circ, \rightarrow\} \]

where the 17 relational operators of RTPA stand for sequence, jump, branch, while-loop, repeat-loop, for-loop, recursion, function call, parallel, concurrence, interleave, pipeline, interrupt, time-driven dispatch, event-driven dispatch, and interrupt-driven dispatch, respectively.

RTPA provides a coherent notation system and a formal engineering methodology for modeling both software and intelligent systems. RTPA can be used to describe both logical and physical models of systems, where logic views of the architecture of a software system and its operational platform can be described using the same set of notations. When the system architecture is formally modeled, the static and dynamic behaviors that perform on the system architectural model, can be specified by a three-level refinement scheme at the system, class, and object levels in a top-down approach. Detailed syntaxes and formal semantics of RTPA meta-processes and process relations may be referred to Wang (2002a, 2007a, 2008c).

4.2. Formal treatment of internal knowledge in problem solving by concept algebra

A concept is a cognitive unit to identify and/or model a concrete entity in the real-world and an abstract object in the perceived-world. Before an abstract concept is defined, the semantic environment or context of concepts is introduced below.
Definition 12. Let \( \mathcal{O} \) be a finite nonempty set of objects, and \( \mathcal{A} \) be a finite nonempty set of attributes, then a semantic environment or context \( \Theta \) of all concepts in the discourse is denoted as a triple, i.e.:

\[
\Theta = (\mathcal{O}, \mathcal{A}, \mathcal{R})
\]

\( \mathcal{R} : \mathcal{O} \to \mathcal{O} \cup \mathcal{A} \to \mathcal{O} \to \mathcal{A} \to \mathcal{A} \to \mathcal{A} \)  

where \( \mathcal{R} \) is a set of relations between \( \mathcal{O} \) and \( \mathcal{A} \), as well as their reflective relations, and \( \setminus \) denotes alternative relations.

Concepts in denotational mathematics are an abstract structure that carries certain meaning in almost all cognitive processes such as problem solving, learning, and reasoning.

Definition 13. An abstract concept \( c \) on \( \Theta \) is a 5-tuple, i.e.:

\[
c = (O, A, R^c, R^r, R^e)
\]

where

- \( O \) is a nonempty set of objects of the concept,
- \( O = \{o_1, o_2, \ldots, o_n\} \subseteq \mathcal{O} \), where \( \mathcal{O} \) denotes a power set of \( \mathcal{O} \).
- \( A \) is a nonempty set of attributes, \( A = \{a_1, a_2, \ldots, a_m\} \subseteq \mathcal{A} \).
- \( R = O \times A \) is a set of internal relations.
- \( R^c \subseteq A^r \times A^r \) is a set of input relations, where \( C \) is a set of external concepts, \( C \subseteq \Theta \). For convenience, \( R = A^r \times A \) may be simply denoted as \( R^c = C^r \).
- \( R^r \subseteq C \times C \) is a set of output relations.

Concept algebra is an abstract mathematical structure for the formal treatment of concepts and their algebraic relations, operations, and associative rules for composing complex concepts.

Definition 14. A concept algebra \( CA \) on a given semantic environment \( \Theta \) is a triple, i.e.:

\[
CA = (O, A, R^c, R^r, R^e, \Theta)
\]

where \( OP = \{\cdot \} \) are the sets of relational and compositional operations on abstract concepts.

Definition 15. The relational operations \( \cdot \), in concept algebra encompass eight comparative operators for manipulating the algebraic relations between concepts, i.e.:

\[
\cdot = \{\sim, \leq, \leq, \geq, \geq, \cong, \sim, \}
\]

where the relational operators stand for related, independent, subconcept, superconcept, equivalent, consistent, comparison, and definition, respectively.

Definition 16. The compositional operations \( \cdot \), in concept algebra encompass nine associative operators for manipulating the algebraic compositions among concepts, i.e.:

\[
\cdot = \{\Rightarrow, \Rightarrow, \Rightarrow, \Rightarrow, \Rightarrow, \Rightarrow, \Rightarrow, \Rightarrow, \Rightarrow}\}
\]

where the compositional operators stand for inheritance, tailoring, extension, substitute, composition, decomposition, aggregation, specification, and instantiation, respectively.

Detailed descriptions of the relational and compositional operations of concept algebra may be referred to (Wang, 2008b). Concept algebra provides a powerful denotational mathematical means for algebraic manipulations of abstract concepts. Concept algebra can be used to model, specify, and manipulate generic "to be" type problems, particularly system architectures, knowledge bases, and detail-level system designs, in cognitive informatics, computing, software engineering, computational intelligence, and system engineering.

4.3. The formal model of the cognitive process of problem solving

Based on RTPA and concept algebra, the cognitive process of problem solving as elaborated in Fig. 2 can be formally described as presented in Fig. 3. According to the OAR model (Wang, 2008c) of internal knowledge representation in the brain, the result of a solution produced in problem solving in the mind of a problem solver is a new sub-OAR model, which will be used to update the entire OAR model of knowledge by concept composition (Wang, 2008b) in the long-term memory of the problem solver. The center in the formal model of the problem solving process is that knowledge about the problem and its solution(s) are represented internally in the brain as an OAR model as shown in Fig. 4, where the external world is represented by real entities (RE), and the internal world by virtual entities (VE) and objects (O). The internal world can be divided into two layers: the image layer and the abstract layer.

The virtual entities are direct images of the external real-entities located at the image layer. The objects are abstract artifacts located at the abstract layer. The abstract layer is an advanced property of human brains. It is noteworthy that animal species have no such abstract layer in their brains. Therefore, they have no indirect or abstract thinking capability (Wang, 2007b). In other words, abstract thinking is a unique power of the human brain known as the qualitative advantage of human brains. The other advantage of the human brain is the tremendous capacity of long-term memory in the cerebral cortex known as the quantitative advantages. On the basis of these two principal advantages, mankind gains the power as human beings.

There are meta-objects (O) and derived objects (O') at the abstract layer in the OAR model. The former are concrete objects directly corresponding to the virtual entities and then to the external world. The latter are abstracted objects that are derived internally and have no direct connection with the virtual entities or images of the real-entities such as abstract concepts, notions, ideas, and states of feelings. The objects on the brain’s abstract layer can be further
extended into a network of objects, attributes, and relations according to the OAR model as shown in Fig. 4. In Fig. 4, the connections between objects/attributes (O/A) via relations are partially connected. In other words, it is not necessary to find a relation among all pairs of objects or attributes.

The OAR model can be used to describe the dynamic state changes of the problem solver during the problem solving process. The problem solving process is essentially a set of representation and search operations. According to the OAR model, a problem can be represented as an object OS. The problem object is identified once it has been comprehended as the problem of interest $X(OS,AST,RST)ST$. The problem object $XST$ is a structure as a part of the entire OARST model of the problem solver’s knowledge base or the solution space.

All meta-objects, attributes, and relations within the OARST solution space will be exhaustively searched.
Upon satisfactory of a parallel search on $G_S$ and $TST$, a set of solutions $(SAS, TST, GST)$ will be generated and represented, which is memorized by using the concept composition operations in order to update the solutions to the given problem (Wang, 2008b).

The relationships and interactions between the problem solving process and other cognitive processes can be explained according to the LRMB model as illustrated in Fig. 5. As a top layer process of the conscious life functions of the brain in LRMB, the problem solving process inter-
acts with other higher cognitive processes at layer 7, such as the learning and comprehension processes. It also involves lower layer processes such as those of abstraction, search, and memorization.

According to the cognitive model of the brain (Wang, 2007b), the thinking engine of the brain may be referred to as a real-time natural intelligence system (NI-Sys). In NI-Sys, the pre-determined operating system is defined as the natural intelligence operating system (NI-Os) and a set of acquired life applications is known as the natural intelligence applications (NI-Ap). Subsequently, the human memory has been classified into both short-term memory (STM) and long-term memory (LTM). Since problem solving is an acquired life function, it is classified as a part of NI-Ap. However, the problem solving process dynamically invokes the support of lower layer processes in NI-Os, as well as STM and LTM, where the acquired knowledge about the solution of a given problem is represented and retained in the logical form of an OAR model.

5. Conclusions

This paper has presented a generic and formal model of the fundamental cognitive process of problem solving on the basis the layered reference model of the brain (LRMB) and the object–attribute-relation (OAR) model. With the exploration of empirical studies on problem solving in cognitive psychology and computational intelligence, a set of formal and rigorous cognitive, mathematical, and process models of problem solving as a cognitive process has been developed. The cognitive structures of the brain and the mechanisms of internal knowledge representation behind the cognitive process of problem solving have been explained. In order to facilitate computer simulations in cognitive informatics and computational intelligence, the denotational mathematical structures of real-time process algebra (RTPA) and concept algebra have been introduced, which provide a powerful tool to formally model and manipulate human and system architectures and behaviors in rigorous approaches.

Future research direction related to this work will be on the modeling and explanation of the rest of the remaining cognitive processes of LRMB using denotational mathematics and formal means in hope to better understand how the brain works as a whole. The entire project will lead to the development of future generation cognitive computers and methodologies for cognitive computing that are capable of think, learn, and perceive.

6. Unicted reference

Wang (in press).

Acknowledgements

This work is partially sponsored by Natural Sciences and Engineering Research Council of Canada (NSERC). The authors would like to thank Prof. Ning Zhong, Jiming Liu, Yiyu Yao, and the anonymous reviewers for their valuable suggestions and comments that have greatly improved the quality of the final version of this article.

References


