



# On the cognitive process of human problem solving

Action editor: Ning Zhong

Yingxu Wang\*, Vincent Chiew

*International Center for Cognitive Informatics (ICfCI) and Theoretical and Empirical Software Engineering Research Centre,  
 Department of Electrical and Computer Engineering, Schulich School of Engineering, University of Calgary,  
 2500 University Drive, NW, Calgary, Alberta, Canada T2N 1N4*

Received 25 February 2008; accepted 4 August 2008

## 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, Wang, Patel, & Patel, 2006; 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).

The history of studies on problem solving has a humble beginning (Mayer, 1992; Newell & Simon, 1972; Payne & Wenger, 1998; Polya, 1954; Robertson, 2001; Tuma & Reif, 1980; Wallas, 1926; Wang, 2008d; Wilson & Clark, 1988; Zadeh, 2008a). Traditional *Gestalt* psychology related human problem solving ability to learning and perception (Matlin, 1998; Ormrod, 1999; Wang, 2007e). Since early 1970s, psychologists acknowledged the benefits of using computer technologies to simulate the mechanisms of problem solving (Goldstein, 1978; Smith, 1991). Matlin (1998) expressed that psychology may not prove that a

\* Corresponding author. Present address: Department of Computer Science, Stanford University, Stanford, CA 94305-9010, USA.

E-mail addresses: yingxu@ucalgary.ca, yingxuw@stanford.edu (Y. Wang), vincent.chiew@ieeee.org (V. Chiew).

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, 2007b; 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.

- 160 • *Heuristic* – **adopting** rule of thumb or the most possible  
161 solutions.
- 162 • *Analogy* – **reducing** a new problem to an existing or  
163 similar one for which solutions have already been  
164 known.
- 165 • *Hill climbing* – **making** any move that approaches closer  
166 to the problem goal step by step.
- 167 • *Algorithmic deduction* – **applying** a known and well  
168 defined solution for a problem.
- 169 • *Exhaustive search* – **using** a systematic search for all possible  
170 solutions.
- 171 • *Divide-and-conquer* – **solving** a whole problem via  
172 decomposing it into a set of subproblems.
- 173 • *Analysis and synthesis* – **reducing** a given problem to a  
174 known category and then finding particular solutions.  
175

176 Adoptions of the above approaches to problem solving  
177 may not guarantee a problem goal to be reached, espe-  
178 cially when a solution is not within the problem solver's  
179 solution space. There are a number of factors that may  
180 hinder the process of problem solving (Matlin, 1998;  
181 Smith, 1991) such as: (a) *mental set* in which a fixed or  
182 improper method is adopted for a new problem while eas-  
183 ier solutions could have been utilized; (b) *meta-cognition*  
184 in which a problem solving process may require the sup-  
185 port of other metacognitive processes to achieve the solu-  
186 tion goal; and (c) *lack of knowledge* in which either the  
187 problem or the goal could not be well represented or  
188 modeled, and no method or solution could be applied  
189 to the problem.

## 190 2.2. Cognitive characteristics of problem solvers

191 The typical psychological traits that may be of use and  
192 of benefit to a successful problem solver are as follows:  
193 To correctly identify problem goals, to be persistent, to  
194 adopt efficient strategies in search, and to be able to trace  
195 back to a certain previous point in the solution process.  
196 In cognitive psychology, what differentiates an expert  
197 from a novice problem solver is studied. It is observed  
198 that not everyone possesses the same ability for problem  
199 solving. These differences may provide insight to explain  
200 the nature of problem solving. The most significant traits  
201 between experts and novices in problem solving are iden-  
202 tified as follows (Payne & Wenger, 1998; Polya, 1954;  
203 Smith, 1991): **scope** of knowledge on accumulated infor-  
204 mation, problem solving schemas, skills, expertise, mem-  
205 ory capacity, problem representation ability, **abstraction**,  
206 and categorization abilities, **analysis**, and synthesis skills,  
207 long-term concentration ability, motivation, efficiency,  
208 and accuracy.

209 It is interesting to contrast and analyze the differences  
210 between professionals and amateurs in software engineer-  
211 ing problem solving. Professional software engineers are  
212 persons with professional cognitive models and knowledge  
213 on software engineering. They are trained with: (a) **funda-**  
214 **mental** knowledge that governs software and software engi-

215 neering practices; (b) **basic** principles and laws of software  
216 engineering; (c) **proven** algorithms; (d) **problem** domain  
217 knowledge; (e) **problem** solving experience; (f) **knowledge**  
218 about program developing tools/ environments; (g) **solid**  
219 programming skills in multiple programming languages;  
220 and (h) **a** global and insightful view on system develop-  
221 ment, including its required functionalities as well as excep-  
222 tion handling and fault-tolerance strategies. However,  
223 amateurish programmers are persons who know only one  
224 or a couple of programming languages but lack formal  
225 training. They may be characterized as follows: (a) **ad**  
226 hoc structure of programming knowledge; (b) **limited** pro-  
227 gramming experience and skills; (c) **eager** to try what is  
228 directly required before a system architecture is designed;  
229 and (d) **tend** to focus on details without a global and sys-  
230 tematic view.

231 In programming, Richard Mayer identified four aspects  
232 of knowledge required for solving programming problems  
233 known as the syntactic, semantic, schematic, and strategic  
234 knowledge (Mayer, 1992). He reported that there exist sig-  
235 nificant differences between the knowledge structures of  
236 experts and novices in all the four categories.

237 It is noteworthy that unlike a machine, human abilities  
238 for problem solving do change depending on ages and  
239 external influences (Payne & Wenger, 1998). A senior per-  
240 son may have a broader knowledge base leading to better  
241 performance in problem solving. However, elderliness  
242 may reduce the efficiency in problem solving physiologi-  
243 cally and psychologically. In addition, personal motiva-  
244 tion, attitude, and external influences such as social  
245 pressure and environment inconvenience may hamper  
246 human problem solving efficiency and effectiveness (Wilson  
247 & Clark, 1988).

## 248 2.3. LRMB: the context framework for problem solving

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

255 A variety of life functions and their cognitive processes  
256 have been identified in cognitive informatics, cognitive neu-  
257 ropyschology, cognitive science, and neurophilosophy. In  
258 order to formally and rigorously describe a comprehensive  
259 and coherent set of mental processes and their relation-  
260 ships, the hierarchical LRMB model is established, which  
261 encompasses 39 cognitive processes at seven layers known  
262 as the sensation, memory, perception, action, meta-cogni-  
263 tive, meta-inference, and higher cognitive layers from the  
264 bottom up.

265 LRMB reveals that the hierarchical life functions of the  
266 brain can be divided into two categories: the subconscious  
267 and conscious life functions. The former encompasses the  
268 layers of sensation, memory, perception, and action (Lay-  
269 ers 1–4). The latter includes the layers of meta-cognitive,

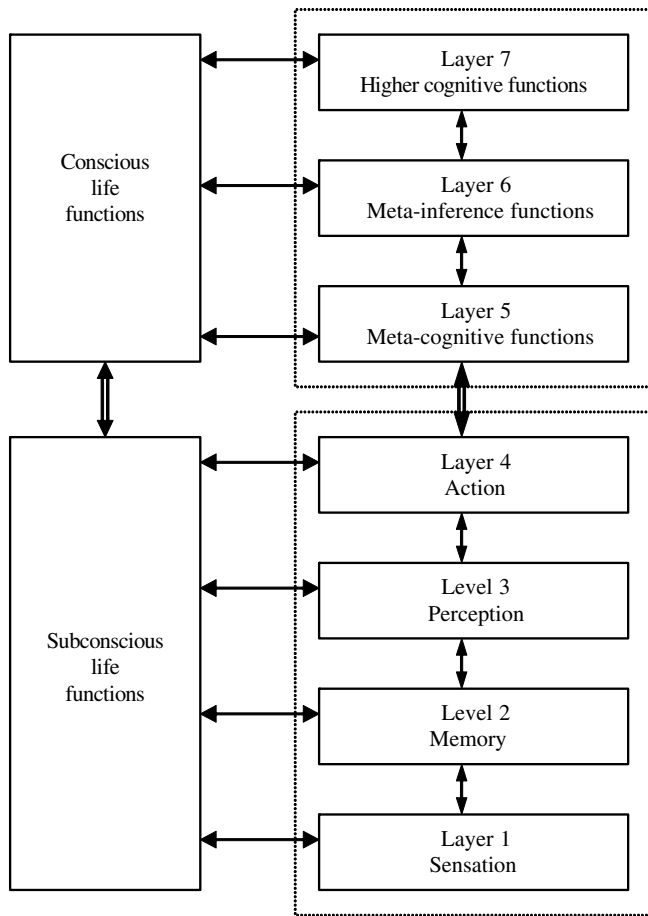


Fig. 1. The layered reference model of the brain (LRMB).

#### 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. In this section, a rigorous study on problem solving as one of the fundamental cognitive mechanisms of the brain is described. As a result, the mathematical model and process model of problem solving will be formally created on the basis

meta-inference, and higher cognitive functions (Layers 5–7).

The subconscious layers of the brain are inherited, fixed, and relatively mature when a person was born. Therefore, the subconscious function layers are not directly controlled by the conscious life function layers. This is why it is called the unconscious life functions in some literature (Matlin, 1998; Payne & Wenger, 1998). However, the conscious layers of the brain are acquired, highly plastic, and can be controlled intentionally based on willingness, goals, and motivations.

LRMB provides a coherent framework to explain the functional mechanisms and cognitive processes of the natural intelligence and human mind. According to LRMB, the functioning of any **higher-layer** process is supported by the lower layer processes from bottom up. For instance, the problem solving process at the top layer invokes lower layer processes' support such as object identification (Layer 1), abstraction (Layer 2), search (Layer 3), memorization (Layer 4/2), and a number of inference processes (Layer 5). It also interacts with processes at the same layer such as comprehension, learning, and decision making. Detailed discussions on the relationships and interactions of the problem solving process with other processes in the context of LRMB will be extended in Section 4.3.

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  $\mathcal{X}$ , a nonempty set of paths  $\mathcal{P}$ , and a nonempty set of goals  $\mathcal{G}$ , i.e.:

$$\Theta \triangleq \mathcal{X} \times \mathcal{P} \times \mathcal{G} \quad (1)$$

where  $\times$  represents a Cartesian product.

**Definition 2.** Assuming the layout of a problem solving process is a function  $f: X \rightarrow \dots \rightarrow Y$  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.:

$$\rho \triangleq (X | f : X \rightarrow \dots \rightarrow Y), \quad \rho \in \mathcal{X} \quad (2)$$

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 satisfactory in the solution space of the problem  $\rho$ , which deduces  $X$  to  $Y$  by a sequence of inference in finite steps, i.e.:

$$G \triangleq (Y | X \rightarrow \dots \rightarrow Y), \quad G \in \mathcal{G} \quad (3)$$

**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.:

$$P \triangleq (X, T, G) = X \times T \times G \quad (4)$$

where the a *trace*  $t \in T$  is an internal node or subpath,  $t: X_t \rightarrow Y_t$ , that maps an intermediate subproblem  $X_t$  to a subgoal  $Y_t$ .

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 path 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.:

$$\begin{cases} S = \wedge(X, T, G) \subseteq P, X, T, G \neq \emptyset \\ s \in S \end{cases} \quad (5)$$

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.  $\#G \equiv 1$ , 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.:*

$$N_{SS} \triangleq N_x \cdot N_t \cdot N_g = \#X \cdot \#T \cdot \#G \quad (6)$$

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 1  
Classification of problems and problem solving methods

Type of problem	Goal	Path	Type of solution
Convergent	Known	Unknown	Proof (specific)
	Known	Known	Instance (specific)
Divergent	Unknown	Known	Case study (open-ended)
	Unknown	Unknown	Explorative/creative (open-ended)

453 for a given problem  $\rho$  is proportional to how many  
 454 subproblems,  $n$ , that  $\rho$  is partitioned, and/or how many path  
 455 segments,  $m$ , that the solution path  $P$  is partitioned, i.e.:

$$\begin{cases} \eta_\rho = 1 - \frac{n}{n(n-1)} = 1 - \frac{1}{n-1} \\ \eta_P = 1 - \frac{\sum_{i=1}^m N_{P_i}}{\prod_{i=1}^m N_{P_i}} \approx 1 - \frac{m \cdot \bar{N}_P}{(\bar{N}_P)^m} = 1 - \frac{m}{(\bar{N}_P)^{m-1}} \end{cases} \quad (7)$$

458 where  $\bar{N}_P$  is the average size or alternatives of all subpaths in  
 459 SS.

460 **Corollary 1** provides a mathematical model that formally  
 461 explains the long-existing empirical heuristic principle of  
 462 divide-and-conquer in problem solving for both human and  
 463 computing systems. According to **Corollary 1**, it can be  
 464 expected that the greater the partitions, the higher the  
 465 efficiency gains in both ways. For instance, given  $n = 3, m = 3$   
 466 and  $\bar{N}_P = 5$ , the efficiency gains of  $\rho$  and  $P$  partitions are  
 467  $\eta_\rho = 50\%$  and  $\eta_P = 88\%$ , respectively. However, when  
 468  $n = 5, m = 5$  and  $\bar{N}_P = 5$ , the efficiency gains of  $\rho$  and  
 469  $P$  partitions are  $\eta_\rho = 75\%$  and  $\eta_P = 99.2\%$ ,  
 470 respectively.

471 3.2. The cognitive process of problem solving

472 On the basis of the mathematical models of problem  
 473 solving as established in Section 3.1, the cognitive process  
 474 of problem solving can be derived logically in this subsection.  
 475 In the problem solving process, the representation of the  
 476 problem is crucial. Problem representation includes a  
 477 description of the given situation, predefined operators for  
 478 changing the situation, and assessment criteria to determine  
 479 whether the goal has been achieved. In most problems of  
 480 interest, the solution space can be too large to be searched  
 481 exhaustively, because it was noted in (Wang, Liu, & Wang,  
 482 2003) that the maximum human memory capacity is up to  
 483  $10^{8.432}$  bits in the brain. Typical approaches to reduce  
 484 search complexity in problem solving, according to **Corollary 1**,  
 485 are to adopt divide-and-conquer and heuristic principles in  
 486 order to simplify the problem into subproblems and/or to  
 487 reduce the paths into a set of short ones.

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

- 493 (a) *To define the problem*: This step describes the problem  
 494 and its input layout  $X$  by identifying its object  $O_X$  and  
 495 attributes  $A_X$ , in a sub-OAR model (Wang, 2007c).
- 496 (b) *To search the solution goals and paths*: In this step,  
 497 the brain performs a parallel search for possible goals  $G$   
 498 and paths  $T$  for the solution. External memory and resources  
 499 may be searched if there is no available or sufficient  $G$  or  
 500  $T$  in the internal knowledge of the problem solver.

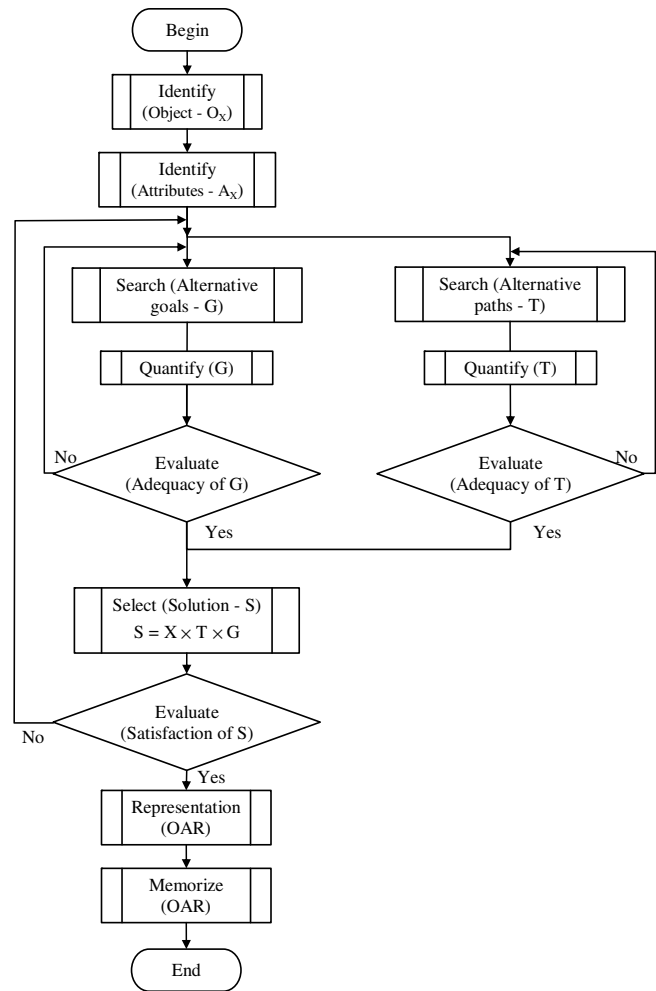


Fig. 2. The cognitive process of problem solving.

- 502 (c) *To generate solutions* according to Eq. (6): This step  
 503 forms a set of possible solutions by a Cartesian product  
 504  $S = X \times T \times G$ .
- 505 (d) *To select suitable solutions*: This step evaluates each  
 506 possible solutions  $s \in S = X \times T \times G$  as obtained in  
 507 Step (c). Recursive searching actions may be executed  
 508 if the obtained solution(s) could not satisfy the problem  
 509 solver's expectation.
- 510 (e) *To represent the problem solving result*: This step  
 511 incorporates and memorizes the solution(s),  $s \in S$ , as  
 512 the result of problem solving into the entire OAR model  
 513 in the long-term memory of the problem solver. Before  
 514 memorization, the solution  $S$  is represented as a part  
 515 of the relations,  $R$ , in the sub-OAR model.

516 It is noteworthy in Fig. 2 that a number of lower layer  
 517 cognitive processes, as represented by double-ended boxes,  
 518 are adopted to carry out the problem solving process.  
 519 Based on the discussions in Section 2.3, the relationship  
 520 among the problem solving process and other cognitive  
 521 processes as modeled in LRMB will be discussed in Section  
 522 4.3.  
 523

The impact of individual's knowledge on the outcomes and efficiencies of problem solving has been recognized in cognitive informatics and psychology. It is often true that a problem to somebody is not one to others; and a hard problem to somebody is not so tough at all to others. These phenomena may be explained empirically by the difference of expertise, knowledge, and problem solving skills. They can be more rigorously explained by the differences of individual's solution space  $SS$  as given in [Theorem 1](#) and [Corollary 1](#). That is, although a problem as given is the same to everybody, the solution spaces to the problem may be greatly different due to the differences of each problem solver's knowledge, experience, skills, and strategies in problem solving.

#### 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

**Definition 8.** RTPA is a denotational mathematical structure for algebraically denoting and manipulating system and human **behavioral** processes and their attributes by a triple, i.e.:

$$RTPA \triangleq (\mathcal{T}, \mathfrak{P}, \mathfrak{R}) \quad (8)$$

where  $\mathcal{T}$  is a set of 17 primitive types for modeling system architectures and data objects,  $\mathfrak{P}$  a set of 17 meta-processes for modeling fundamental system behaviors, and  $\mathfrak{R}$  a set of 17 relational process operations for constructing complex system behaviors.

**Definition 9.** The RTPA type system  $\mathcal{T}$  encompasses 17 primitive types elicited from fundamental computing needs, i.e.:

$$\mathcal{T} \triangleq \{ \mathbf{N}, \mathbf{Z}, \mathbf{R}, \mathbf{S}, \mathbf{BL}, \mathbf{B}, \mathbf{H}, \mathbf{P}, \mathbf{TI}, \mathbf{D}, \mathbf{DT}, \mathbf{RT}, \mathbf{ST}, \\ @e\mathbf{S}, @t\mathbf{TM}, @int\mathbf{O}, @s\mathbf{BL} \} \quad (9)$$

where all types in  $\mathcal{T}$  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  $\mathfrak{P}$  encompasses 17 fundamental computational operations elicited from the most basic computing needs, i.e.:

$$\mathfrak{P} \triangleq \{ :=, \blacklozenge, \Rightarrow, \Leftarrow, \neq, >, <, |>, |<, @, \triangle, \\ \uparrow, \downarrow, !, \otimes, \boxtimes, \S \} \quad (10)$$

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  $\mathfrak{R}$  encompasses 17 fundamental algebraic and relational operations elicited from basic computing needs, i.e.:

$$\mathfrak{R} \triangleq \{ \rightarrow, \curvearrowright, |, |\dots| \dots, R^*, R^+, R^i, \cup, \mapsto, ||, \\ \text{\textcircled{f}}, |||, \gg, \leftarrow, \hookrightarrow, \hookrightarrow_e, \hookrightarrow_i \} \quad (11)$$

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.:

$$\Theta = \langle \mathcal{O}, \mathcal{A}, \mathcal{R} \rangle \quad (12)$$

$$= \mathcal{R} : \mathcal{O} \rightarrow \mathcal{O} | \mathcal{O} \rightarrow \mathcal{A} | \mathcal{A} \rightarrow \mathcal{O} | \mathcal{A} \rightarrow \mathcal{A}$$

where  $\mathcal{R}$  is a set of relations between  $\mathcal{O}$  and  $\mathcal{A}$ , as well as their reflective relations, and  $|$  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.:

$$c \triangleq \langle \mathcal{O}, \mathcal{A}, R^c, R^i, R^o \rangle \quad (13)$$

where

- $\mathcal{O}$  is a nonempty set of objects of the concept,  $\mathcal{O} = \{o_1, o_2, \dots, o_m\} \subseteq \wp \mathcal{O}$ , where  $\wp \mathcal{O}$  denotes a power set of  $\mathcal{O}$ .
- $\mathcal{A}$  is a nonempty set of attributes,  $\mathcal{A} = \{a_1, a_2, \dots, a_n\} \subseteq \wp \mathcal{A}$ .
- $R^c = \mathcal{O} \times \mathcal{A}$  is a set of internal relations.
- $R^i \subseteq \mathcal{A}' \times \mathcal{A}$ ,  $\mathcal{A}' \sqsubseteq \mathcal{C}' \wedge \mathcal{A} \sqsubseteq c$ , is a set of input relations, where  $\mathcal{C}'$  is a set of external concepts,  $\mathcal{C}' \subseteq \Theta$ . For convenience,  $R^i = \mathcal{A}' \times \mathcal{A}$  may be simply denoted as  $R^i = \mathcal{C}' \times c$ .
- $R^o \subseteq c \times \mathcal{C}'$  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.:

$$CA \triangleq \langle \mathcal{C}, OP, \Theta \rangle = \langle \{\mathcal{O}, \mathcal{A}, R^c, R^i, R^o\}, \{\bullet_r, \bullet_c\}, \Theta \rangle \quad (14)$$

where  $OP = \{\bullet_r, \bullet_c\}$  are the sets of *relational* and *compositional* operations on abstract concepts.

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

$$\bullet_r \triangleq \{ \leftrightarrow, \leftrightarrow^-, \prec, \succ, =, \cong, \sim, \triangle \} \quad (15)$$

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

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

$$\bullet_c \triangleq \{ \Rightarrow, \Rightarrow^-, \Rightarrow^+, \Rightarrow^{\sim}, \uplus, \psi, \uplus, \vdash, \mapsto \} \quad (16)$$

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

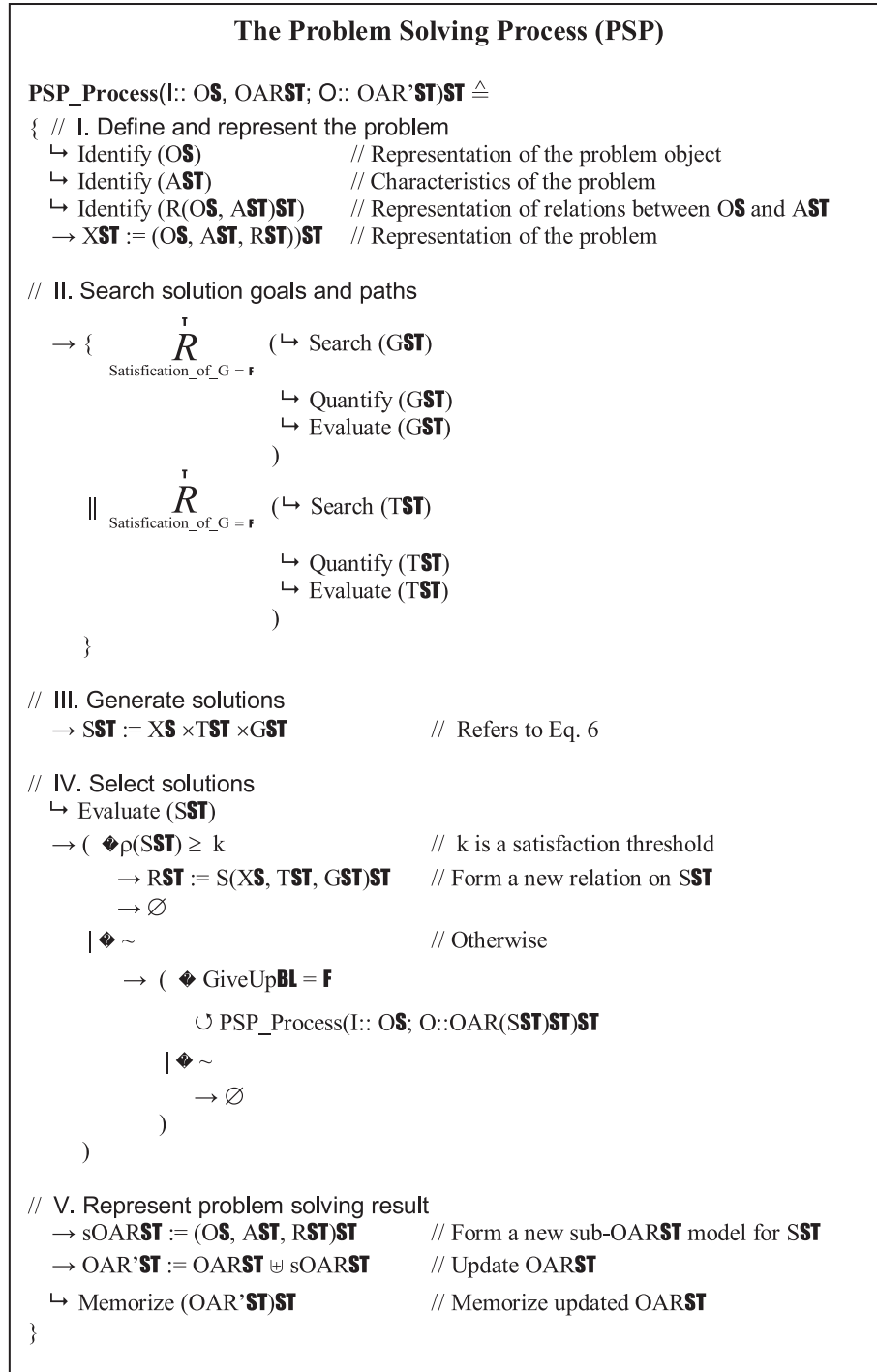


Fig. 3. Formal description of the cognitive process of problem solving in RTPA.

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.

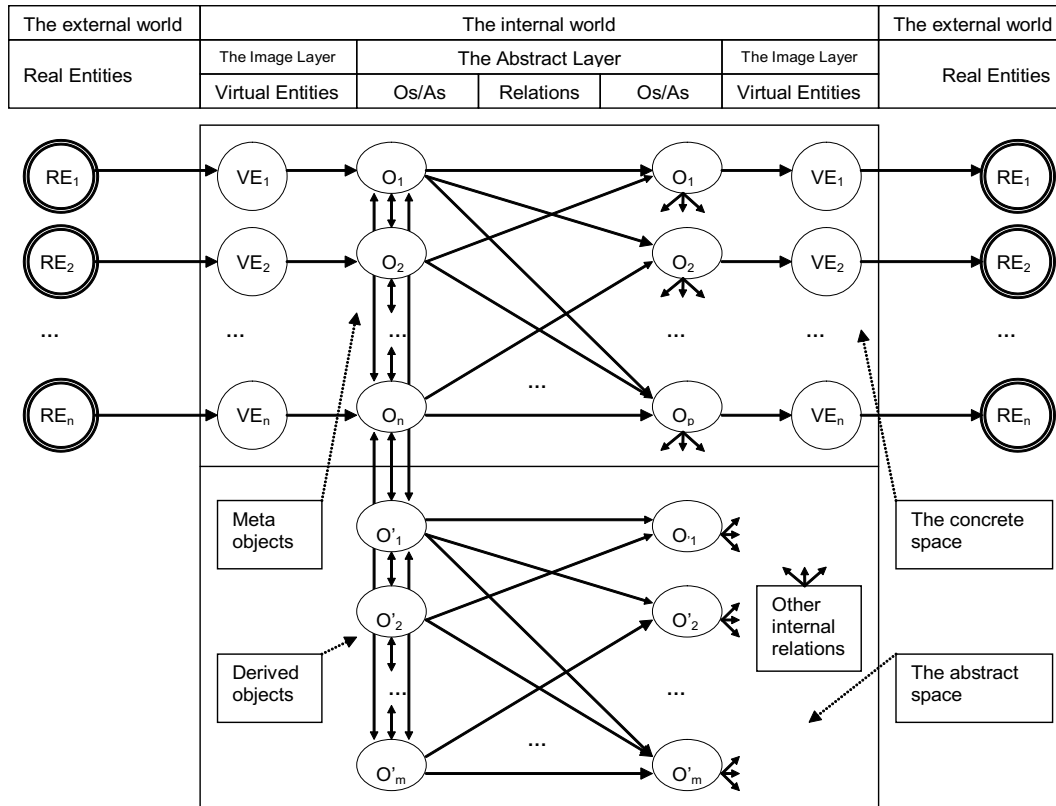


Fig. 4. The extended OAR model of the brain.

Upon satisfactory of a parallel search on *GS* and *TST*, a set of solutions (*SXS*, *TST*, *GST*)*ST* will be generated and represented, which is memorized by using the concept composition operations  $\oplus$  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-

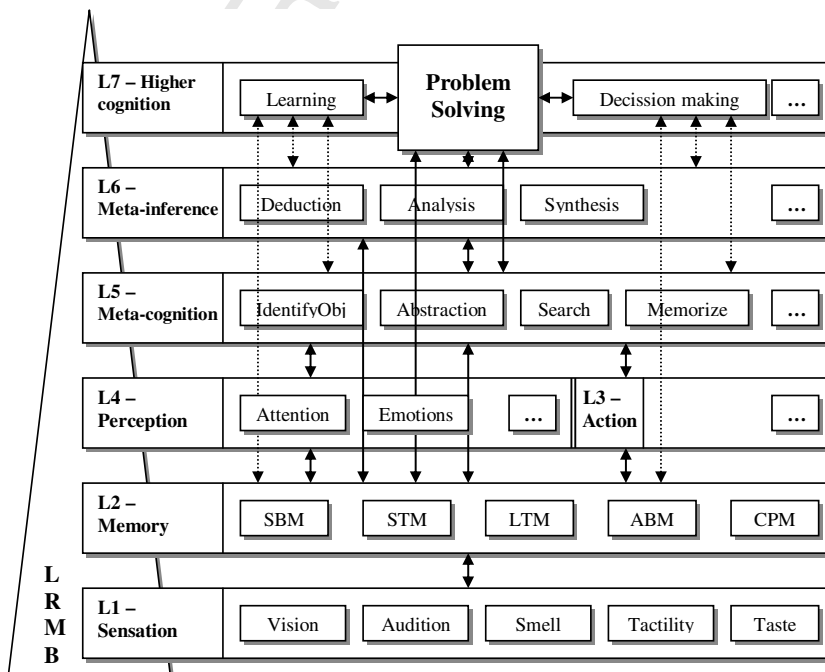


Fig. 5. Interaction between problem solving and other cognitive processes in LRMB.

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-App). 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-App. 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. Uncited reference

Q1 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

- Bender, E. A. (1996). *Mathematical methods in artificial intelligence*. Los Alamitos, CA, USA: IEEE CS Press.
- Beveridge, W. I. (1957). *The art of scientific investigation*. UK: Random House Trade Paperbacks.
- Goldstein, I. (1978). *Developing a computational representation for problem solving skills*, AI memo 495. Cambridge, MA, USA: MIT.
- Matlin, M. W. (1998). *Cognition* (4th ed.). NY, USA: Harcourt Brace College Publishers.
- Mayer, R. E. (1992). *Thinking, problem solving, cognition* (2nd ed.). W.H. Freeman and Co..
- Newell, A., & Simon, H. A. (1972). *Human problem solving*. Prentice-Hall.
- Ormrod, J. E. (1999). *Human learning* (3rd ed.). Prentice-Hall Inc..
- Payne, D. G., & Wenger, M. J. (1998). *Cognitive psychology*. Boston, NY, USA: Houghton Mifflin Company.
- Polya, G. (1954). *Patterns of plausible inference*. Princeton, NJ, USA: Princeton University Press.
- Robertson, I. (2001). *Problem solving*. Psychology Press.
- Rubinstein, M. F., & Firstenberg, I. R. (1995). *Patterns of problem solving*. Englewood, NJ, USA: Prentice Hall.
- Smith, M. U. (1991). *Toward a unified theory of problem solving: Views from the content domains*. Hillsdale, NJ, USA: Lawrence Erlbaum Associates.
- Tuma, D. T., & Reif, F. (1980). *Problem solving and education*. Hillsdale, NJ, USA: Lawrence Erlbaum Associates.
- Wallas, G. (1926). *The art of thought*. NY, USA: Harcourt Brace.
- Wang, Y. (2002a). The real-time process algebra (RTPA). *Annals of Software Engineering: An International Journal*, 14, 235–274.
- Wang, Y. (2002b). *Keynote, on cognitive informatics. Proceedings of the 1st IEEE international conference on cognitive informatics (ICCI'02)*. Calgary, Canada: IEEE CS Press (pp. 34–42).
- Wang, Y. (2003a). On cognitive informatics. *Brain and Mind: A Transdisciplinary Journal of Neuroscience and Neurophilosophy*, 4(2), 151–167.
- Wang, Y. (2003b). Using process algebra to describe human and software behaviors. *Brain and Mind: A Transdisciplinary Journal of Neuroscience and Neurophilosophy*, 4(2), 199–213.
- Wang, Y. (2007a). Software engineering foundations: A software science perspective. *CRC series in software engineering* (Vol. II). NY, USA: Auerbach Publications.
- Wang, Y. (2007b). The theoretical framework of cognitive informatics. *International Journal of Cognitive Informatics and Natural Intelligence*, 1(1), 1–27.
- Wang, Y. (2007c). The OAR model of neural informatics for internal knowledge representation in the brain. *International Journal of Cognitive Informatics and Natural Intelligence*, 1(3), 64–75.
- Wang, Y. (2007d). Keynote, on theoretical foundations of software engineering and denotational mathematics. In *Proceedings of the 5th Asian workshop on foundations of software* (pp. 99–102), Xiamen, China.
- Wang, Y. (2007e). *The theoretical framework and cognitive process of learning. Proceedings of the 6th international conference on cognitive informatics (ICCI'07)*. Lake Tahoe, CA: IEEE CS Press (pp. 470–479).
- Wang, Y. (2008a). On contemporary denotational mathematics for computational intelligence. *Transactions of Computational Science*, 2, 6–29.
- Wang, Y. (2008b). On concept algebra: A denotational mathematical structure for knowledge and software modeling. *The International Journal of Cognitive Informatics and Natural Intelligence*, 2(2), 1–19.
- Wang, Y. (2008c). RTPA: A denotational mathematics for manipulating intelligent and computational behaviors. *International Journal of Cognitive Informatics and Natural Intelligence*.

- 882 Wang, Y. (2008d). *On cognitive foundations of creativity and the cognitive* 896  
 883 *process of creation. Proceedings of the 7th IEEE international confer-* 897  
 884 *ence on cognitive informatics (ICCI08)*. Stanford University, CA, 898  
 885 USA: IEEE CS Press (pp. 104–113). 899
- 886 Wang, Y. (in press). On abstract intelligence: Toward a unified theory of 900  
 887 natural, artificial, machinable, and computational intelligence. *Inter-* 901  
 888 *national Journal of Software Science and Computational Intelligence,* 902  
 889 *1(1), 1–18.* Q4 903
- 890 Wang, Y., Liu, D., & Wang, Y. (2003). Discovering the capacity of human 904  
 891 memory. *Brain and Mind: A Transdisciplinary Journal of Neuroscience* 905  
 892 *and Neurophilosophy, 4(2), 189–198.* 906
- 893 Wang, Y., & Ruhe, G. (2007). The cognitive process of decision making. 907  
 894 *International Journal of Cognitive Informatics and Natural Intelligence,* 908  
 895 *1(2), 73–85.* 909
- Wang, Y., Wang, Y., Patel, S., & Patel, D. (2006). A layered reference 896  
 model of the brain (LRMB). *IEEE Transactions on Systems, Man, and* 897  
*Cybernetics (C), 36(2), 124–133.* 898
- Wilson, L. B., & Clark, R. G. (1988). *Comparative programming language.* 899  
 Wokingham, UK: Addison-Wesley Publishing Co.. 900
- Zadeh, L. A. (2008a). Toward human level machine intelligence – Is it 901  
 achievable? The need for a paradigm shift. *IEEE Computational* 902  
*Intelligence Magazine, 11–22.* Q5 903
- Zadeh, L. A. (2008b). Is there a need for fuzzy logic? *Information Sciences:* 904  
*An International Journal, 178, 2751–2779.* 905
- Zhong, N. (2006). Impending brain informatics research from web 906  
 intelligence perspective. *International Journal of Information Technol-* 907  
*ogy and Decision Making, 5(4), 713–728.* 908  
 909

UNCORRECTED PROOF