A Cognitive Informatics Reference Model of Autonomous Agent Systems (AAS)

Yingxu Wang, University of Calgary, Canada

ABSTRACT

Despite the fact that the origin of software agent systems has been rooted in autonomous artificial intelligence and cognitive psychology, their implementations are still based on conventional imperative computing techniques rather than autonomous computational intelligence. This paper presents a cognitive informatics perspective on autonomous agent systems (AAS's). A hierarchical reference model of AAS's is developed, which reveals that an autonomous agent possesses intelligent behaviors at three layers known as those of imperative, autonomic, and autonomous from the bottom up. The theoretical framework of AAS's is described from the facets of cognitive informatics, computational intelligence, and denotational mathematics. According to Wang's abstract intelligence theory, an autonomous software agent is supposed to be called as an intelligent-ware, shortly, an intelware, parallel to hardware and software in computing, information science, and artificial intelligence.

Keywords: *AI; abstract intelligence; agent systems; autonomic computing; behavioral models; cognitive model; computational intelligence; enotational mathematics; intelligence models; intelware; LRMB; mathematical models; multi-agent; RMAAS; software agent; system behaviors; tonomous agent*

INTRODUCTION

A *software agent* is an intelligent software system that autonomously carries out robotic and interactive applications based on goal-driven cognitive mechanisms. The studies on software agent are rooted in the essences of computing science and cognitive science such as automata theory [von Neumann, 1946, 1958, 1963, 1966; Shannon, 1956], Turing machines [Turing, 1950], cognitive psychology [Newell, 1990; Sternberg, 1997; Anderson and Rosenfeld, 1998; Matlin, 1998], artificial intelligence [*McCarthy,* 1955, 1963; McCulloch, 1943, 1965; Barr and Feigenbaum, 1981], computational intelligence [Poole et al., 1997; Wang, 2008a], and decision theories [Wald, 1950; Newell and Simon, 1972; Berger et al., 1990; Bronson and Naadimuthu, 1997; Wang and Ruhe, 2007; Wang, 2008b].

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

The history towards software agents may be traced back to the work as early as in the 1940s. J. *McCarthy,* W. McCulloch, *M.L. Minsky, N. Rochester, and C.E. Shannon* proposed the term *Artificial Intelligence* (AI) [*McCarthy,* 1955, 1963; McCulloch, 1943, 1965]. S.C. Kleene analyzed the relations of *automata* and nerve nets [Kleene, 1956]. Then, Bernard Widrow developed the technology of *artificial neural networks* in the 1950s [Widrow and Lehr, 1990]. The concepts of *robotics* [Brooks, 1970] and *expert systems* [Giarrantans and Riley, 1989] were developed in the 1970s and 1980s, respectively. In 1992, the notion of *genetic algorithms* was proposed by J.H. Holland [Holland, 1992]. Then, *distributed artificial intelligence* and *intelligent system* technologies emerged since late 1980s [Bond and Gasser, 1988; Kurzweil, 1990; Chaib-Draa et al., 1992; Meystel and Albus, 2002, Meystel and Albus, 2002].

The origin of the term *autonomous agent* is based on Carl Hewitt and his colleagues' *artificial intelligence actor models* proposed in 1973 [Hewitt et al., 1973, 1991]. Then, as a novel approach of artificial intelligence, agent technologies have been proliferated since the early 1990s [Foner, 1993; Genesereth and Ketchpel, 1994; Hayes-Roth, 1995; Axelrod, 1997; Huhns and Singh, 1997; Wooldridge and Jenings, 1995; Wooldridge, 2002, Wang, 2003b]. Pattie Maes perceived that a software agent is a process that lives in the world of computers and networks and that can operate autonomously to fulfill a set of tasks [Maes, 1991]. Dimitris N. Chorafas described a software agent as a new software paradigm of things that think [Chorafas, 1998]. Software agents are characterized by knowledge, learning, reasoning, and adaptation, which are rational to the extent that their behaviors are predictable by given goals and the solution environment [Russell and Norvig 1995; Poole, Mackworth, and Goebel 1997; Nilsson 1998].

Multi-agent systems are proposed in [Wittig, 1992; Wellman, 1999] as distributed intelligent systems [Bond and Gasser, 1988] in which each node is an autonomous software agent. The key technology of autonomous agent systems is how a variety of heterogeneous agents allocate their roles, coordinate their behaviors, share their resources, and communicate their information, beliefs, and needs [Maes, 1991]. The interaction mechanisms of multi-agent systems, such as cooperation, negotiation, belief reconciliation, information sharing, and distributed decision making, are identified as important issues in the design and implementation of multi-agent systems.

Autonomic computing is one of the fundamental technologies of software agents, which is a mimicry and simulation of the natural intelligence possessed by the brain using general computers. Autonomic computing was first proposed by IBM in 2001, where it is perceived that "Autonomic computing is an approach to self-managed computing systems with a minimum of human interference. The term derives from the body's autonomous nervous system, which controls key functions without conscious awareness or involvement [IBM, 2006]." Various studies on autonomic computing have been reported following the IBM initiative [Kephart and Chess, 2003; Murch, 2004; Wang, 2004].

According to Wang's *abstract intelligence* theory [Wang, 2008a, 2009], software agents are a paradigm of abstract and computational intelligence, which is a subset of or an application-specific virtual brain. Behaviors of a software agent are mirrored human behaviors. Therefore, a software agent may be more accurately named as an *intelligent-ware*, shortly, an *intelware*, parallel to hardware and software in computing, information science, and artificial intelligence. In this notion, intelware will be treated as a synonym of an autonomous agent system.

This paper presents a coherent theoretical framework of autonomous agent systems (AAS's) or *intelware* from the facets of cognitive informatics, computational intelligence, and denotational mathematics. The nature of software agents and intelware is elaborated. A reference model of AAS with intelligent behaviors at three layers known as those of imperative, autonomic, and autonomous is developed from the bottom up. The theoretical

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

framework of AAS's/intelware is presented on the basis of cognitive informatics and computational intelligence theories. A set of denotational mathematics is introduced in order to provide a fundamental mathematical means for formally and rigorously dealing with the highly complicated architectures and intricate behaviors of AAS's and intelware.

THE NATURE OF SOFTWARE AGENTS AND INTELWARE

Definition 1. *A software agent, or more actually an intelware, is an intelligent software system that autonomously carries out robotistic and interactive applications based on goal-driven cognitive mechanisms.*

On the basis of Definition 1, an autonomous agent is a software agent that possesses high-level autonomous ability and behaviors beyond conventional imperative computing technologies.

Definition 2. *An Autonomous Agent System (AAS) is a composition of distributed agents that possesses autonomous computing and decision making abilities as well as interactive communication capability to peers and the environment.*

The classification of agent/intelware technologies can be described in Table 1, where *I* and *O* denote the inputs/outputs of a given AAS. When both input event (I) and output behavior (O) are constant, it denotes a *routine* intelware; while when both *I/O* are variable, it represents the most complicated *autonomous* intelware. Otherwise, the combinations of variable event/constant behavior and constant event/variable behaviors indicate an *algorithmic* or *autonomic* intelware, respectively.

In Table 1, the routine and algorithmic AAS's may be implemented by computational imperative behaviors. However, the autonomic AAS's should be implemented by autonomic computing, as that of the autonomous AAS's by autonomous mechanisms and behaviors.

THE REFERENCE MODEL OF INTELWARE/AUTONOMOUS AGENT SYSTEMS

The reference model of intelware/AAS's is a hierarchical model with three layers known as those of imperative, autonomic, and autonomous behaviors. This section elaborates the mathematical models of the imperative and intelligent behaviors of intelware/AAS's in the layered reference model of agent intelligence.

The Hierarchical Behavioral Model of Intelware/AAS's

Behaviorism is a doctrine of psychology and cognitive informatics that describes the association between a given stimulus and an observed response of human brains and AAS's. Cognitive informatics reveals that human and AAS behaviors may be classified into four categories known as the *perceptive, cognitive, instructive,* and *reflective* behaviors [Wang, 2007b].

Table 1. Classification of Intelware / AAS's

		Behavior (O)	
		Constant	Variable
Event (I)	Constant	*Routine*	*Autonomic*
	Variable	*Algorithmic*	*Autonomous*

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

The reference model of AAS's (RMAAS) is a hierarchical behavioral model of agent intelligence as illustrated in Fig. 1. In the RMAAS model, the hierarchy of agent behaviors can be divided into the imperative, autonomic, and autonomous layers. Conventional computing machines are implemented only by imperative behaviors. However, the autonomic computing systems and AAS's are implemented by advanced cognitive behaviors. *Imperative computing* is an enclosure of instructive and passive behaviors. The *autonomic computing* is an enclosure of internally motivated behaviors beyond those of the imperative space. The *autonomous computing* is an enclosure of perceptive- and inference-driven behaviors beyond those of both imperative and autonomic computing. More formal descriptions of the three types of behaviors of AAS's will be presented in the following subsections.

The Imperative Behavioral Layer of Intelware/AAS's

According to the RMAAS model as illustrated in Fig. 1, the imperative behavioral intelligence of intelware and AAS's can be formally modeled and elaborated in this subsection.

Definition 3. *The imperative behavioral layer of AAS's, B_I, is a set of instruction-based behaviors such as the event-driven behaviors (B_e), time-driven behaviors (B_t), and interrupt-driven behaviors (B_{int}), i.e.:*

$$B_I \triangleq \{B_e, B_t, B_{int}\} \quad (1)$$

An imperative system implemented with B_I may do nothing unless a specific program is loaded, in which the stored program transfers a general-purpose computer to a specific intelligent application. The imperative system is a passive system that implements deterministic, context-free, and stored-program controlled behaviors.

Definition 4. *An event is an abstract variable that represents an external stimulus to a system or the occurring of an internal change of status, such as an action of users, an updating of the environment, and a change of the value of a control variable.*

The types of events that may trigger a behavior can be classified into operational ($@e\mathbf{S}$), time ($@t\mathbf{TM}$), and interrupt ($@int\odot$) events, where @ is the *event prefix*, and $\mathbf{S}$, $\mathbf{TM}$, and $\odot$ the type suffixes, respectively. The *interrupt event* is a kind of special event that models the interruption of an executing process, the temporal handover of controls to an Interrupt Service Routine (ISR), and the return of control after its completion.

Definition 5. *An interrupt, denoted by $\lightning$, is a parallel process relation in which a running process P is temporarily held by another higher priority process Q via an interrupt event $@int\odot$ at the interrupt point $\odot$, and the interrupted process will be resumed when the high priority process has been completed, i.e.:*

$$P \lightning Q \triangleq P \,||\, (@int\odot \nearrow Q \searrow \odot) \quad (2)$$

where $\nearrow$ and $\searrow$ denote an interrupt service and an interrupt return, respectively.

In general, all types of events, including the operational events, timing events, and interrupt events, are captured by the system in order to dispatch a designated behavior.

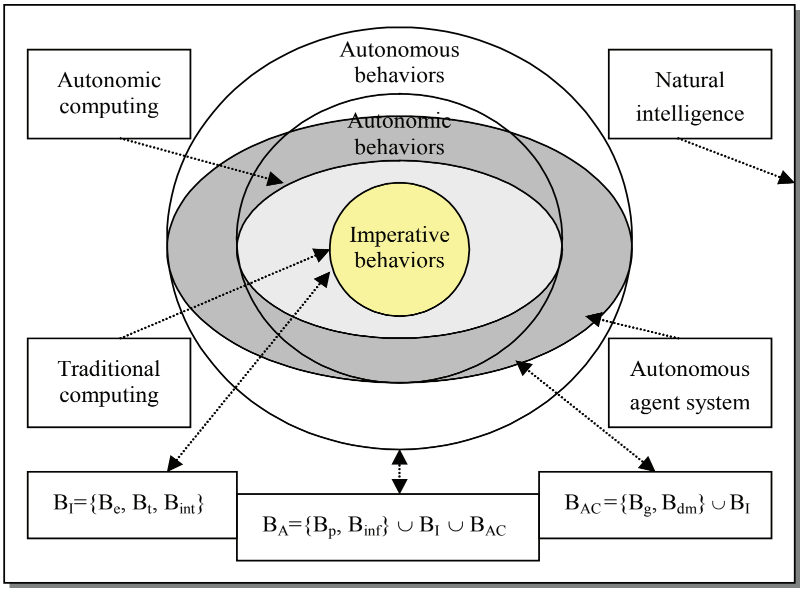
Definition 6. *An event-driven behavior B_e, denoted by $\hookrightarrow_e$, is an imperative process in which the ith behavior in term of a designated process P_i is triggered by a predefined event $@e_i\mathbf{S}$, i.e.:*

$$B_e \triangleq \mathop{\mathrm{R}}_{i=1}^{n} (@e_i\mathbf{S} \hookrightarrow_e P_i) \quad (3)$$

where the big-R notation is a mathematical calculus that denotes a sequence of repetitive/iterative behaviors or a set of recurring structures [Wang, 2007a].

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Fig. 1 The hierarchical reference model of autonomous agent systems (RMAAS)



Definition 7. *A time-driven behavior B_t, denoted by $\hookrightarrow_t$, is an imperative process in which the ith behavior in term of process P_i is triggered by a predefined point of time @t_iTM, i.e.:*

$$B_t \triangleq \overset{n}{\underset{i=1}{R}} (@t_i \text{TM} \hookrightarrow_t P_i) \qquad (4)$$

where @t_iTM may be a system timing or external timing event.

Definition 8. *An interrupt-driven behavior B_{int}, denoted by $\hookrightarrow_{int}$, is an imperative process in which the ith behavior in term of process P_i is triggered by a predefined system interrupt @int_i⊙, i.e.:*

$$B_{int} \triangleq \overset{n}{\underset{i=1}{R}} (@int_i \odot \hookrightarrow_{int} P_i) \qquad (5)$$

As a summary, an imperative computing system can be described as follows.

Definition 9. *An Imperative Computing (IC) system is a passive system that implements deterministic, context-free, and stored-program controlled behaviors.*

The Autonomic Behavioral Layer of Intelware/AAS's

According to the RMAAS model as illustrated in Fig. 1, the autonomic behavioral intelligence of intelware and AAS's can be formally modeled and elaborated in this subsection.

Definition 10. *The autonomic behavioral layer of AAS's, B_C, is a set of internally motivated and self- generated behaviors such as the goal-driven behaviors (B_g) and decision-driven behaviors (B_d) on the basis of the imperative layer B_I, i.e.:*

$$\begin{aligned} B_C &\triangleq \{B_g, B_d\} \cup B_I \\ &= \{B_e, B_r, B_{int}, B_g, B_d\} \end{aligned} \qquad (6)$$

Definition 11. *A goal-driven behavior B_g, denoted by $\hookrightarrow_g$, is an autonomic process in which*

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

*the ith behavior in term of process P_i is generated by the system itself, rather than be given, corresponding to the goal @g_i**ST**, i.e.:*

$$B_g \triangleq \underset{i=1}{\overset{n}{R}} (@g_i\mathbf{ST} \hookrightarrow_g P_i) \quad (7)$$

*where the goal, denoted by g**ST**, is a triple, i.e.:*

$$g\mathbf{ST} = (P, \Omega, \Theta) \quad (8)$$

in which that $P = \{p_1, p_2, ..., p_n\}$ is a finite nonempty set of purposes or motivations, Ω is a finite set of constraints for the goal, and Θ is the environment of the goal.

Definition 12. *A decision-driven behavior B_d, denoted by $\hookrightarrow_d$, is an autonomic process in which the ith behavior in term of process P_i is generated by a given decision @d_i**ST**, i.e.:*

$$B_d \triangleq \underset{i=1}{\overset{n}{R}} (@\mathrm{d}_i\mathbf{ST} \hookrightarrow_\mathrm{d} P_i) \quad (9)$$

*where the decision, denoted by d**ST**, is a selected alternative $a \in \mathcal{A}$ from a nonempty set of alternatives $\mathcal{A}$, based on a given set of criteria C, i.e.:*

$$\begin{aligned} d &= f(\mathcal{A}, C) \\ &= f: \mathcal{A} \times C \to \mathcal{A},\ \mathcal{A} \neq \varnothing \end{aligned} \quad (10)$$

Definition 13. *An Autonomic Computing (AC) system is an intelligent system that implements nondeterministic, context-dependent, and adaptive behaviors based on goal- and decision-driven mechanisms.*

The autonomic systems do not rely on instructive and procedural information, but are dependent on internal status and willingness that formed by long-term historical events and current rational or emotional goals [Wang, 2007d].

The Autonomous Behavioral Layer of Intelware/AAS's

According to the RMAAS model as illustrated in Fig. 1, the autonomous behavioral intelligence of intelware and AAS's can be formally modeled and elaborated in this subsection.

Definition 14. *The autonomous behavioral layer of AAS's, B_A, is a set of autonomously generated behaviors by internal cognitive processes such as the perception-driven behaviors (B_p) and inference-driven behaviors (B_{inf}) on the basis of the imperative space B_I and the autonomic space B_C, i.e.:*

$$\begin{aligned} B_A &\triangleq \{B_p, B_{inf}\} \cup B_I \cup B_C \\ &= \{B_e, B_t, B_{int}, B_g, B_d, B_p, B_{inf}\} \end{aligned} \quad (11)$$

The new forms of behaviors covered in the autonomous layer can be elaborated as follows.

Definition 15. *A perception-driven behavior B_p, denoted by $\hookrightarrow_p$, is a cognitive process in which the ith behavior in term of process P_i is generated by the result of a perceptive process @p_i**PC**, i.e.:*

$$B_p \triangleq \underset{\mathrm{i}=1}{\overset{n}{R}} (@p_i\mathbf{PC} \hookrightarrow_p P_i) \quad (12)$$

*where **PC** stands for a type of process, and the perception result p**PC** is an outcome of the cognitive process of perception that an AAS may generate.*

Inferences are cognitive processes that reason about a possible causality from given premises based on known causal relations between a pair of cause and effect proven true by empirical arguments, theoretical inferences, or statistical regulations.

Definition 16. *An inference-driven behavior B_{inf}, denoted by $\hookrightarrow_{inf}$, is a cognitive process in which the ith behavior in term of process P_i is*

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

generated by the result of an inference process $@inf_i\mathbf{PC}$, *i.e.:*

$$B_{inf} \triangleq \mathop{R}_{i=1}^{n} (@inf_i\mathbf{PC} \hookrightarrow_{inf} P_i) \qquad (13)$$

where formal inferences can be classified into the deductive, inductive, abductive, and analogical categories, as well as modal, probabilistic, and belief theories [Wang, 2007e].

As shown in Definition 16 and Fig. 1, an AAS implemented on B_A extends the conventional behaviors B_I and B_C to more powerful and intelligent behaviors, which are generated by internal and autonomous processes such as the perception and inference processes. With the possession of all the seven forms of intelligent behaviors in B_A, the AAS may advance closer to the intelligent power of human brains.

Relationships between the Agent Behaviors of Intelware/AAS's at the Three Layers of RMAAS

Contrasting Definitions 3, 10, and 14, the following relationships among the three-layer agent intelligent behaviors can be established on the basis of the RMAAS model as illustrated in Fig. 1.

Theorem 1. *The relationships of the imperative behaviors* B_I, *autonomic behaviors* B_C, *and cognitive behaviors* B_A *of intelware or AAS's are hierarchical and inclusive, i.e.:*

$$B_I \subseteq B_C \subseteq B_A \qquad (14)$$

Theorem 1 and Definition 14 indicate that any lower layer behavior of an intelware or AAS is a subset of those of a higher layer. In other words, any higher layer behavior is a natural extension of those of lower layers as shown in Fig. 1. Therefore, the *necessary and sufficient conditions of AAS's,* C_{AAS}, are the possession of all behaviors at the three layers.

Corollary 1. *The behavioral model of intelware or* AAS, §AAS**ST**, *can be logically modeled by a set of parallel processes that encompasses the imperative behaviors* B_I, *autonomic behaviors* B_C, *and autonomous behaviors* B_A *from the bottom-up, i.e.:*

$$\begin{aligned}
\S AAS\mathbf{ST} &\triangleq (B_I, B_C, B_A) \\
&= \{ (B_e, B_t, B_{int}) \qquad // B_I \\
&\quad || (B_e, B_t, B_{int}, B_g, B_d) \qquad // B_C \\
&\quad || (B_e, B_t, B_{int}, B_g, B_d, B_p, B_{\inf}) \quad // B_A \\
&\quad \}
\end{aligned} \qquad (15)$$

where || denotes a parallel relation in RTPA.

THEORETICAL FOUNDATIONS OF INTELWARE/AAS'S

Recent research reveals that the foundations of agent technologies root in cognitive informatics, denotational mathematics, and computational intelligence [Wang, 2002a, 2003b, 2008a]. Along with the latest advances in cognitive informatics, non-imperative autonomous agent systems known as *intelware* and *cognitive computers* are emerging. This section explores the theoretical foundations of AAS's and intelware. The latest development of fundamental theories and technologies underpinning AAS's and intelware are highlighted.

Denotational Mathematics for AAS's

Applied mathematics can be classified into two categories known as *analytic* and *denotational* mathematics [Wang, 2002b, 2007a, 2008a, 2008c]. The former are mathematical structures that deal with functions of variables as well as their operations and behaviors; while the latter are mathematical structures that formalize rigorous expressions and inferences of system architectures and behaviors with abstract concepts, complex relations, and dynamic processes. The denotational and expressive needs in cognitive informatics, com-

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

putational intelligence, software engineering, and knowledge engineering have led to new forms of mathematics collectively known as denotational mathematics.

Definition 17. *Denotational mathematics is a category of expressive mathematical structures that deals with high-level mathematical entities beyond numbers and simple sets, such as abstract objects, complex relations, behavioral information, concepts, knowledge, processes, intelligence, and systems.*

The term denotational mathematics is first introduced by Yingxu Wang in the emerging discipline of cognitive informatics [Wang, 2002a, 2007a, 2008c]. Typical paradigms of denotational mathematics are comparatively presented in Table 1, where their structures, mathematical entities, algebraic operations, and usages are contrasted. The paradigms of denotational mathematics as shown in Table 1 are *concept algebra* [Wang, 2008d], *system algebra* [Wang, 2008e], and *Real-Time Process Algebra* (RTPA) [Wang, 2002b, 2008f].

The emergence of denotational mathematics is driven by the practical needs in cognitive informatics, computational intelligence, computing science, software science, and knowledge engineering, because all these modern disciplines study complex human and machine behaviors and their rigorous treatments. Among the new forms of denotational mathematics, *concept algebra* is designed to deal with the abstract mathematical structure of concepts and their representation and manipulation in knowledge engineering. *System algebra* is created to the rigorous treatment of abstract systems and their algebraic relations and operations. RTPA is developed to deal with series of behavioral processes and architectures of human and systems.

Denotational mathematics provides a powerful mathematical means for modeling and formalizing AAS's. Not only the architectures of AAS's, but also their dynamic behaviors can be rigorously and systematically manipulated by denotational mathematics. Applications of denotational mathematics in cognitive informatics and computational intelligence have been elaborated with a wide range of real-world case studies [Wang, 2008a, 2008c], which demonstrate that denotational mathematics is an ideal mathematical means for dealing with concepts, knowledge, behavioral processes, and human/machine intelligence in ASS's and intelware.

Cognitive Informatics Theories of AAS's

Cognitive informatics is the transdisciplinary enquiry of cognitive and information sciences that investigates into the internal information processing mechanisms and processes of the brain and natural intelligence, and their engineering applications via an interdisciplinary approach [Wang, 2002a, 2003a , 2003b, 2006, 2007a, 2007b, 2007c, 2007d]. According to the abstract intelligence theory [Wang, 2008a, 2009], because cognitive informatics investigates the internal information processing mechanisms and processes of the brain and natural intelligence, its research results underlie the engineering applications of AAS's. Cognitive informatics reveals that artificial intelligence (AI) is a subset of natural intelligence (NI) [Wang, 2007a, 2007b]. Therefore, AAS's may be referred to the natural intelligence and behavioral mechanisms of human beings.

A Layered Reference Model of the Brain (LRMB) is developed [Wang, et al., 2006] that reveals the logical model of NI and a coherent set of cognitive mechanisms. LRMB presents a systematical view toward the formal description and modeling of architectures and behaviors of AAS's, which are created to extend human capability, reachability, and/or memory capacity. The LRMB model explains the functional mechanisms and cognitive processes of the natural intelligence with 39 cognitive processes at seven layers known as the *sensation, memory, perception, action, meta-cognitive, meta-inference,* and *higher cognitive layers* from the bottom up. LRMB elicits the core and highly repetitive recurrent cognitive processes from a

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Table 2. Paradigms of Denotational Mathematics

No.	Paradigm	Structure	Mathematical entities	Algebraic operations	Usage
1	Concept algebra	$CA \triangleq (C, OP, \Theta) = (\{O, A, R^c, R^i, R^o\}, \{\bullet_r, \bullet_c\}, \Theta_C)$	$c \triangleq (O, A, R^c, R^i, R^o)$	$\bullet_r \triangleq \{\leftrightarrow, \not\leftrightarrow, \prec, \succ, =, \cong, \sim, \triangleq\}$ $\bullet_c \triangleq \{\Rightarrow, \overset{-}{\Rightarrow}, \overset{+}{\Rightarrow}, \overset{\sim}{\Rightarrow}, \uplus, \pitchfork, \Lleftarrow, \vdash, \mapsto\}$	Algebraic manipulations on abstract concepts
2	System algebra	$SA \triangleq (S, OP, \Theta) = (\{C, R^c, R^i, R^o, B, \Omega\}, \{\bullet_r, \bullet_c\}, \Theta)$	$S \triangleq (C, R^c, R^i, R^o, B, \Omega, \Theta)$	$\bullet_r \triangleq \{\not\leftrightarrow, \leftrightarrow, \prod, =, \sqsubseteq, \sqsupseteq\}$ $\bullet_c \triangleq \{\Rightarrow, \overset{-}{\Rightarrow}, \overset{+}{\Rightarrow}, \overset{\sim}{\Rightarrow}, \boxminus, \uplus, \pitchfork, \Lleftarrow, \vdash\}$	Algebraic manipulations on abstract systems
3	Real-time process algebra (RTPA)	$RTPA \triangleq (\mathfrak{T}, \mathfrak{P}, \mathfrak{N})$	$\mathfrak{P} \triangleq \{:=, ◆, \Rightarrow, \Leftarrow, \not\Leftarrow, \gtrdot, \lessdot, \mid\gtrdot, \mid\lessdot, @, , \uparrow, \downarrow, !, \otimes, \boxtimes, \S\}$ $\mathfrak{T} \triangleq \{$**N, Z, R, S, BL, B, H, P, TI, D, DT, RT, ST**$, @e$**S**$, @t$**TM**$, @int\odot, ⓢs$**BL**$\}$	$\mathfrak{R} \triangleq \{\rightarrow, \curvearrowright, \mid, \mid\ldots\mid\ldots, R^*, R^+, R^i, \circlearrowleft, \rightarrowtail, \parallel, ∯, \mid\mid\mid, », ↯, \hookrightarrow_t, \hookrightarrow_e, \hookrightarrow_i\}$	Algebraic manipulations on abstract processes

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

huge variety of life functions, which may shed light on the study of the fundamental mechanisms and interactions of complicated mental processes as well as AAS's, particularly the relationships and interactions between the inherited and the acquired life functions as well as those of the subconscious and conscious cognitive processes. The cognitive model of the brain can be used as a reference model for goal- and inference-driven technologies in AAS's.

Definition 18. *The cognitive model of the kernel of an AAS or intelware, AAS_k, can be described as a real-time intelligent system with an inherited Agent Operating System AOS and a set of Agent Intelligent Behaviors AIB in parallel, i.e.:*

$$AAS_k \triangleq AOS \,||\, AIB \tag{16}$$

Definition 19. *The Cognitive Models of Memory (CMM) states that the architecture of human memory is parallel configured by the Sensory Buffer Memory (SBM), Short-Term Memory (STM), Long-Term Memory (LTM), Conscious Status Memory (CSM), and Action-Buffer Memory (ABM), i.e.:*

$$\begin{aligned} CMM \triangleq (\ & LTM \\ & ||\, STM \\ & ||\, CSM \\ & ||\, SBM \\ & ||\, ABM \\ ) & \end{aligned} \tag{17}$$

The CMM model provides a neural informatics foundation of natural intelligence. With the CMM model, the broad sense of an AAS, AAS', can be described by mimicking the abstract architecture and mechanisms of the brain.

Definition 20. *The cognitive model of AAS's, AAS, is represented by a real-time intelligent system that encompasses the intelware and the CMM as well as their interactions, i.e.:*

$$\begin{aligned} AAS \triangleq\ & Intelware \\ & ||\, CMM \\ & = (\ AOS \\ & \quad ||\, AIB \\ & \quad) \\ & \,||\, (\ LTM \\ & \quad ||\, STM \\ & \quad ||\, CSM \\ & \quad ||\, SBM \\ & \quad ||\, ABM \\ & \quad) \end{aligned} \tag{18}$$

Eq. 18 indicates that although *intelware* is considered the center of AAS's, the memories are essential to enable it to properly function, and to keep temporary and permanent results physiologically retained and retrievable.

Computational Intelligence Theories of AAS's

According to the abstract intelligence theory [Wang, 2008a, 2009], intelligence is perceived as the driving force or the ability to acquire and use knowledge and skills, or to reason in problem solving. It was conventionally perceived that only human beings possess higher-level intelligence. However, the development of computers, robots, intelligent systems, and AAS's indicates that intelligence may also be created or implemented by machines and man-made systems.

Definition 21. *Intelligence, in the narrow sense, is a human or a system ability that transforms information into behaviors; and in a broad sense, it is any human or system ability that autonomously transfers the forms of abstract information between data, information, knowledge, and behaviors in the brain.*

Definition 22. *The Generic Abstract Intelligence Model (GAIM), as shown in Fig. 2, represents abstract intelligence in four forms known as the perceptive, cognitive, instructive, and reflective intelligence, corresponding to the specific forms of cognitive information and their memories.*

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

The GAIM indicates that different forms of intelligence are the driving force that transfers between a pair of abstract objects in the brain such as *data* (*D*), *information* (*I*), *knowledge* (*K*), and *behavior* (*B*). It is noteworthy that each abstract object is physiologically retained in a particular type of memories as given in the CMM model. This is the neural informatics foundation of natural intelligence, and the physiological evidences of why natural intelligence can be classified into four forms as shown in Fig. 2.

According to Definitions 21 and 22, *computational intelligence* is a paradigm of abstract intelligence. Computational intelligence models human intelligence by computational methodologies and cognitively inspired models.

Definition 23. *The computational intelligence model of AAS's and intelware, §AAS***ST***, is a parallel structure represented by the Agent Operating System (AOS***ST***) and a set of agent intelligence represented by the Agent Intelligent Behaviors (AIB***ST***), as shown in Fig. 3.*

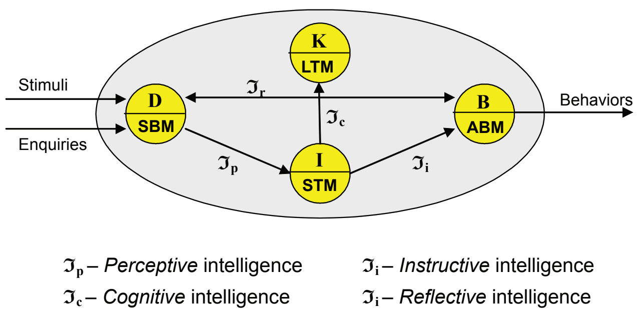
The GAIM and §AAS**ST** model reveal that NI and AI share the same cognitive informatics foundations on the basis of abstract intelligence. The *compatible intelligent capability* states that NI, AI, AAS's, and intelware are compatible by sharing the same mechanisms of intelligent capability and behaviors. In other words, at the logical level, NI of the brain shares the same mechanisms as those of AI and computational intelligence. The differences between NI and AI are only distinguishable by the means of implementation and the extent of intelligent ability. Therefore, the studies on NI and AI in general, and intelware and AAS's in particular, may be unified into a coherent framework based on cognitive informatics and computational intelligence, which are formalized by denotational mathematics.

CONCLUSION

This paper has presented a coherent theoretical framework of Autonomous Agent Systems (AAS), known as *intelware,* from the facets of cognitive informatics, computational intelligence, and denotational mathematics. A reference model of AAS has been developed with three-layer intelligent behaviors known as the imperative, autonomic, and autonomous agent intelligence from the bottom up. It has been recognized that the characteristics of an AAS is its perception-driven and inference-driven behaviors beyond the imperative and autonomic ones as provided by conventional imperative and autonomic computing.

In order to formally and rigorously deal with the highly complicated architectures and intricate behaviors of intelware and AAS's, a new mathematical means known as denotational mathematics has been developed. Typical paradigms of denotational mathematics have been

Fig. 2 The Generic Abstract Intelligence Model (GAIM)



Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Fig. 3 The computational intelligence model of AAS

The Computational Intelligent Model of AAS

§AAS**ST** ≙ AOS**ST** // Agent operating system
|| AIB**ST** // Agent intelligent behaviors
= { // AOS**ST**

<SE**ST**:

$\overset{n-1}{\underset{ptr\mathbf{P}=0}{R}}$ SENSORS[ptr**P**]**ST**> // Layer 1: Sensation engine

|| <ME**ST**: $\overset{5}{\underset{addr\mathbf{P}=0}{R}}$ MEM[addr**P**]**ST**> // Layer 2: Memory engine
= <SBM**ST** || STM**ST** || CSM**ST** || LTM**ST** || ABM**ST**>

|| <PE**ST**: $\overset{7}{\underset{i\mathbf{P}=0}{R}}$ PROC[i**N**]**ST**> // Layer 3: Perception engine
= <Attention**ST** || Motivation**ST** || Emotion**ST** || Attitude**ST**
|| SensOfSpatiality**ST** || SensOfTime**ST** || SensOfMotion**ST**>

|| <AE**ST**: $\overset{n_{SERVO}\mathbf{H}-1}{\underset{ptr\mathbf{P}=0}{R}}$ SERVOS[ptr**P**]**ST**> // Layer 4: Action engine

|| <CE**ST**: $\overset{10}{\underset{i\mathbf{P}=0}{R}}$ PROC[i**N**]**ST** > // Layer 5: Meta-cognition engine
= <ObjectIdentification**ST** || Abstraction**ST** || ConceptEstablishment**ST**
|| Search**ST** || Categorization**ST** || Comparison**ST** || Memorization**ST**
|| Qualification**ST** || Quantification**ST** || Selection**ST**>

|| <IE**ST**: $\overset{6}{\underset{i\mathbf{P}=0}{R}}$ PROC[i**N**]**ST** > // Layer 6: Meta-inference engine
= <Deduction**ST** || Induction**ST** || Abduction**ST** || Analogy**ST**
|| Analysis**ST** || Synthesis**ST**>

|| <HCE**ST**: $\overset{7}{\underset{i\mathbf{P}=0}{R}}$ PROC[i**N**]**ST** > // Layer 7: Higher cognition engine
= <Comprehension**ST** || Learning**ST** || Planning**ST** || ProblemSolving**ST**
|| DecisionMaking**ST** || Creation**ST** || PattenRecognition**ST**>

|| <§t**TM**> // Relative clock
}

|| { // AIB**ST**

|| < $\overset{n_e\mathbf{N}-1}{\underset{k\mathbf{N}=0}{R}}$ @e_k**S** ↳ P_k**ST**> // *Event*-driven behaviors (B_e)

|| < $\overset{n_t\mathbf{N}-1}{\underset{k\mathbf{N}=0}{R}}$ @t_k**TM** ↳ P_k**ST**> // *Time*-driven behaviors (B_t)

|| < $\overset{n_{int}\mathbf{N}-1}{\underset{k\mathbf{N}=0}{R}}$ @int_k⊙ ↳ P_k**ST**> // *Interrupt*-driven behaviors (B_{int})

|| < $\overset{n_t\mathbf{N}-1}{\underset{k\mathbf{N}=0}{R}}$ @g_k**ST** ↳ P_k**ST**> // *Goal*-driven behaviors (B_g)

|| < $\overset{n_t\mathbf{N}-1}{\underset{k\mathbf{N}=0}{R}}$ @d_k**ST** ↳ P_k**ST**> // *Decision*-driven behaviors (B_d)

|| < $\overset{n_t\mathbf{N}-1}{\underset{k\mathbf{N}=0}{R}}$ @p_k**ST** ↳ P_k**ST**> // *Perception*-driven behaviors (B_p)

|| < $\overset{n_{int}\mathbf{N}-1}{\underset{k\mathbf{N}=0}{R}}$ @inf_k**ST** ↳ P_k**ST**> // *Inference*-driven behaviors (B_{inf})
}

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

introduced such as concept algebra, system algebra, and RTPA. The findings of this work, particularly the necessary and sufficient conditions of imperative and autonomous computing, and the abstract intelligence model of natural and artificial intelligence, have formed a solid foundation for explaining and developing advanced autonomous computing systems and their engineering applications.

ACKNOWLEDGMENT

The author would like to acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) to this work. The author would like to thank the valuable comments and suggestions of the reviewers and colleagues.

REFERENCES

Anderson, J.A. and E. Rosenfeld, eds. (1988). *Neurocomputing: Foundations of Research*, Cambridge.

Axelrod, R. (1977), *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*, Princeton Univ. Press, Princeton, NJ.

Barr, A. and E. A. Feigenbaum, eds. (1981), *The Handbook of Artificial Intelligence*, Vol. 1. Stanford and Los Altos, CA: HeurisTech Press and Kaufmann.

Berger, J. (1990), *Statistical Decision Theory – Foundations, Concepts*, and Methods, Springer-Verlag.

Bond, A.H. and L. Gasser (1988), *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA.

Bronson, R. and G. Naadimuthu (1997), *Schaum's Outline of Theory and Problems of Operations Research*, 2nd ed., McGraw-Hill, NY.

Brooks, R.A. (1970), New Approaches to Robotics, *American Elsevier*, NY, 5, 3-23.

Chaib-Draa, B. Moulin, R. Mandiau, and P. Millot, (1992), Trends in Distributed Artificial Intelligence, *Artificial Intelligence Review* 6, 35-66.

Chorafas, D.N. (1998), *Agent Technology Handbook*, McGraw-Hill, NY.

Foner, L. (1993), What ís an Agent, Anyway? A Sociological Case Study, *Agents Memo 93-01*, MIT Media Lab, Cambridge, MA.

Genesereth, M.R. and S.P. Ketchpel (1994), Software Agents, *Communications of the ACM*, 37 (7), 48-53.

Giarrantans, J. and G. Riley (1989), *Expert Systems: Principles and Programming*, PWS-KENT Pub. Co., Boston.

Hayes-Roth, B. (1995), An Architecture for Adaptive Intelligent Systems*, Artificial Intelligence*, 72(1-2), 329-365.

Hewitt, C., R. Bishop, and R. Steiger (1973), A Universal Modular Actor Formalism for Artificial Intelligence, *Proc. 3rd Int. Joint Conf. on Artificial Intelligence*, Stanford, CA, Aug.

Hewitt, C. and J. Inman (1991), DAI Betwixt and Between: From Intelligent Agents to Open Systems Science, IEEE Trans. on System, Man, and Cybernetics, Nov/Dec.

Holland, J.H. (1992), Genetic Algorithms, *Scientific American*, 267, 66-72.

Huhns, M., and M. Singh, eds. (1997). *Readings in Agents,* Kaufmann, San Francisco.

IBM (2006), *Autonomous Computing White Paper: An Architectural Blueprint for Autonomous Computing*, 4th ed., June, 1-37.

Jennings, N.R. (2000), On Agent-Based Software Engineering, *Artificial Intelligence*, 17(2), 277-296.

Kephart, J. and D. Chess (2003), The Vision of Autonomic Computing, IEEE *Computer*, 26(1), Jan, 41-50.

Kleene, S.C. (1956), Representation of Events by Nerve Nets, in C.E. Shannon and J. McCarthy eds., *Automata Studies*, Princeton Univ. Press, 3-42.

Kurzweil, R. (1990). *The Age of Intelligent Machines.* Cambridge, MA, MIT Press.

Maes, P. ed. (1991), *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, London, The MIT press.

Matlin, M.W. (1998), *Cognition*, 4th ed., Harcourt Brace College Publishers, Orlando, FL.

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

McCarthy, J., M.L. Minsky, N. Rochester, and C.E. Shannon (1955), *Proposal for the 1956 Dartmouth Summer Research Project on Artificial Intelligence*, Dartmouth College, Hanover, NH, USA, http://www.formal.stanford.edu/jmc/history/dartmouth/*dartmouth.html.*

McCarthy, J. (1963), *Situations, Actions, and Causal Laws*, Memo 2, Stanford University Artificial Intelligence Project, Stanford, CA.

McCulloch, W. S., and W. Pitts. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics* 5, 115–137.

McCulloch, W.S. (1965), *Embodiments of Mind*, MIT Press, Cambridge, MA.

Meystel, A.M. and J.S. Albus (2002), *Intelligent Systems, Architecture, Design, and Control,* John Wiley & Sons.

Murch, R. (2004), *Autonomic Computing*, Person Education, London.

Newell, A. (1990), *Unified Theories of Cognition,* Harvard University Press, Cambridge, MA.

Newell, A., and H.A. Simon (1972), *Human Problem Solving*, Prentice-Hall Englewood Cliffs, NJ.

Nilsson, N. J. (1998), *Artificial Intelligence: A New Synthesis,* Morgan Kaufmann, San Mateo, CA.

Poole, D., A. Mackworth, and R. Goebel. (1997). *Computational Intelligence: A Logical Approach.* Oxford: Oxford University Press, Oxford, UK.

Russell, S.J., and P. Norvig. (1995), *Artificial Intelligence: A Modern Approach,* Prentice-Hall, Englewood Cliffs, NJ.

Shannon, C.E. *ed.* (1956), Automata Studies, Princeton University Press, Princeton.

Sternberg, R.J. (1997), The Concept of Intelligence and the its Role in Lifelong Learning and Success, *American Psychologist*, 52(10), 1030-1037.

Turing, A.M. (1950), Computing Machinery and Intelligence, *Mind,* 59, 433-460.

von Neumann, J. (1946), The Principles of Large-Scale Computing Machines, reprinted in *Annals of History of Computers*, 3(3), 263-273.

von Neumann, J. (1958), The Computer and the Brain, Yale Univ. Press, New Haven.

von Neumann, J. (1963), *General and Logical Theory of Automata,* A.H. Taub ed., Collected Works, Vol. 5, Pergamon, 288-328.

von Neumann, J. and A.W. Burks (1966), Theory of Self-Reproducing Automata, Univ. of Illinois Press, Urbana IL.

Wald, A. (1950), *Statistical Decision Functions*, John Wiley & Sons.

Wang, Y. (2002a), Keynote: On Cognitive Informatics, *Proc. 1st IEEE International Conference on Cognitive Informatics* (ICCI'02), Calgary, Canada, IEEE CS Press, August, 34-42.

Wang, Y. (2002b), The Real-Time Process Algebra (RTPA), *Annals of Software Engineering: An International Journal*, 14, USA, 235-274.

Wang, Y. (2003a), Cognitive Informatics: A New Transdisciplinary Research Field, *Brain and Mind: A Transdisciplinary Journal of Neuroscience and Neurophilosophy,* 4(2), 115-127.

Wang, Y. (2003b), Keynote: Cognitive Informatics Models of Software Agent Systems, Proc. 1st International Conference on Agent-Based Technologies and Systems (ATS'03), Univ. of Calgary Press, Calgary, Canada, August, 25.

Wang, Y. (2004), Keynote: On Autonomic Computing and Cognitive Processes, *Proc. 3rd IEEE International Conference on Cognitive Informatics* (ICCI'04), Victoria, Canada, IEEE CS Press, August, 3-4.

Wang, Y. (2006), Keynote: *Cognitive Informatics - Towards the Future Generation Computers that Think and Feel, Proc. 5th IEEE International Conference on Cognitive Informatics* (ICCI'06), Beijing, China, IEEE CS Press, July, 3-7.

Wang, Y. (2007a), *Software Engineering Foundations: A Software Science Perspective*, CRC Book Series in Software Engineering, Vol. II, Aurebach Publications, NY., USA.

Wang, Y. (2007b), Keynote: Cognitive Informatics Foundations of Nature and Machine Intelligence, *Proc. 6th International Conference on Cognitive Informatics* (ICCI'07), IEEE CS Press, Lake Tahoe, CA., Aug., 3-12.

Wang, Y. (2007c), The Theoretical Framework of Cognitive Informatics, *International Journal of*

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Cognitive Informatics and Natural Intelligence, IGI, USA, 1(1), Jan., 1-27.

Wang, Y. (2007d), Exploring Machine Cognition Mechanisms for Autonomic Computing, *International Journal on Cognitive Informatics and Natural Intelligence*, March, 1(2), i - v.

Wang, Y. (2007e), The Cognitive Processes of Formal Inferences, *International Journal of Cognitive Informatics and Natural Intelligence*, IGI, USA, Dec., 1(4), 75-86.

Wang, Y. (2008a), Keynote: On Abstract Intelligence and Its Denotational Mathematics Foundations, *Proc. 7th IEEE International Conference on Cognitive Informatics* (ICCI'08), Stanford University, CA., USA, IEEE CS Press, August, 5-15.

Wang, Y. (2008b), Toward a Generic Mathematical Model of Abstract Game Theories, *Transactions of Computational Science, 2,* Springer, June, 205-223.

Wang, Y. (2008c), On Contemporary Denotational Mathematics for Computational Intelligence, *Transactions of Computational Science, 2,* Springer, June, 6-29.

Wang, Y. (2008d), On Concept Algebra: A Denotational Mathematical Structure for Knowledge and Software Modeling, *International Journal of Cognitive Informatics and Natural Intelligence,* IGI, USA, April, 2(2), 1-19.

Wang, Y. (2008e), On System Algebra: A Denotational Mathematical Structure for Abstract System modeling, *International Journal of Cognitive Informatics and Natural Intelligence,* IGI, USA, April, 2(2), 20-42.

Wang, Y. (2008f), RTPA: A Denotational Mathematics for Manipulating Intelligent and Computational Behaviors, *International Journal of Cognitive Informatics and Natural Intelligence,* IGI, USA, April, 2(2), 44-62.

Y. Wang (2009), On Abstract Intelligence: Toward a Unified Theory of Natural, Artificial, Machinable, and Computational Intelligence, *International Journal of Software Science and Computational Intelligence,* IGI, USA, Jan., 1(1), 1-18.

Wang, Y. and G. Ruhe (2007), The Cognitive Process of Decision Making, *International Journal of Cognitive Informatics and Natural Intelligence*, IGI, USA, March, 1(2), 73-85.

Wang, Y., Y. Wang, S. Patel, and D. Patel (2006), A Layered Reference Model of the Brain (LRMB), *IEEE Trans. on Systems, Man, and Cybernetics (C),* March, 36(2), 124-133.

Wellman, M.P. (1999), Multiagent Systems, in R.A. Wilson and C.K. Frank eds., *The MIT Encyclopedia of the Cognitive Sciences*, MIT Press, MA.

Widrow, B. and M.A. Lehr (1990), 30 Years of Adaptive Neural Networks: Perception, Madeline, and Backpropagation, *Proc. of the IEEE*, Sept., 78(9), 1415-1442.

Wittig, T. ed. (1992), *ARCHON: An Architecture for Multi-Agent Systems*, Ellis Horwood, London.

Wooldridge, M. and N. Jennings (1995), Intelligent Agents: Theory and Practice, *The Knowledge Engineering Review* 10(2), 115-152.

Wooldridge, M. (2002), *An Introduction to Multiagent Systems*, John Wiley & Sons.

Yingxu Wang is professor of cognitive informatics and software engineering, director of International Center for Cognitive Informatics (ICfCI), and director of Theoretical and Empirical Software Engineering Research Center (TESERC) at the University of Calgary. He received a PhD in software engineering from The Nottingham Trent University, UK, in 1997, and a BSc in electrical engineering from Shanghai Tiedao University in 1983. He was a visiting professor in the Computing Laboratory at Oxford University and Department of Computer Science at Stanford University during 1995 and 2008, respectively, and has been a full professor since 1994. He is founding editor-in-chief of the International Journal of Cognitive Informatics and Natural Intelligence *(*IJCINI*), founding editor-in-chief of the* International Journal of Software

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Science and Computational Intelligence *(*IJSSCI*), associate editor of* IEEE TSMC*(A), and editor-in-chief of the CRC Book Series in software engineering. He has published over 300 journal and conference papers and 11 books in software engineering and cognitive informatics, and won dozens of research achievement, best paper, and teaching awards in the last 28 years.*

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.