Toward Theoretical Foundations of Autonomic Computing[1]

Yingxu Wang, University of Calgary, Canada

ABSTRACT

Autonomic computing (AC) is an intelligent computing approach that autonomously carries out robotic and interactive applications based on goal- and inference-driven mechanisms. This article attempts to explore the theoretical foundations and technical paradigms of AC. It reviews the historical development that leads to the transition from imperative computing to AC. It surveys transdisciplinary theoretical foundations for AC such as those of behaviorism, cognitive informatics, denotational mathematics, and intelligent science. On the basis of this work, a coherent framework toward AC may be established for both interdisciplinary theories and application paradigms, which will result in the development of new generation computing architectures and novel information processing systems.

Keywords: *autonomic computing; AI; behavioral models; cognitive informatics; cognitive models; imperative computing; intelligence models; knowledge science; mathematical models; natural intelligence; software engineering; software science*

INTRODUCTION

Autonomic computing (AC) is a mimicry and simulation of the natural intelligence possessed by the brain using generic computers. This indicates that the nature of software in AC is the simulation and embodiment of human behaviors, and the extension of human capability, reachability, persistency, memory, and information processing speed.

The history toward AC may be traced back to the work on automata by Norbert Wiener, John von Neumann, Alan Turing, and Claude E. Shannon as early as in the 1940s (Rabin & Scott, 1959; Shannon, 1956; Turing, 1950; von Neumann, 1946, 1958, 1963, 1966; Wiener, 1948). In the same period, Warren McCulloch proposed the term of artificial intelligence (AI) (McCulloch, 1965, 1993; McCulloch & Pitts, 1943), and 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 & Lehr, 1990). The concepts of robotics (Brooks, 1970) and expert systems (Giarrantans & Riley, 1989) were developed in the 1970s and 1980s, respectively. Then, intelligent systems (Meystel & Albus, 2002) and software agents (Negreponte, 1995; Jennings, 2000) emerged in the 1990s. These events and developments lead to the formation of the concept of AC.

AC was first proposed by IBM in 2001 where it is perceived that, "AC is an approach to self-managed computing systems with a minimum of human interference. The term derives from the body's autonomic nervous system, which controls key functions without conscious awareness or involvement" (IBM, 2001). Various studies on AC have been reported following the IBM initiative (Kephart & Chess, 2003; Murch, 2004; Pescovitz, 2002). The cognitive informatics foundations of AC have been revealed by Wang (2002a, 2003a, 2003b, 2004, 2006b, 2006f, 2007a, 2007c) and Wang and Kinsner (2006). A paradigm of AC in term of cognitive machine has been surveyed by Kinsner (2007) and investigated by Wang (2006a, 2007b).

Based on cognitive informatics theories (Wang, 2002a, 2003a, 2007b), AC is proposed as a new and advanced technology for computing built upon the routine, algorithmic, and adaptive systems as shown in Table 1.

The first three categories of computing techniques, such as routine, algorithmic, and adaptive computing, as shown in Table 1, are imperative. In contrary, the AC systems do not rely on imperative and procedural instructions, but are dependent on goal-, perception-, and inference-driven mechanisms.

- **Definition 1:** An *imperative computing* (IC) *system* is a passive system that implements deterministic, context-free, and stored-program controlled behaviors.
- **Definition 2:** An *autonomic computing system* is an intelligent system that implements non-deterministic, context-dependent, and adaptive behaviors based on goal- and inference-driven mechanisms.

This article attempts to explore the theoretical foundations and engineering paradigms of AC. It is extended from two invited keynote speeches of the author in the Third IEEE International Conference on Cognitive Informatics (ICCI'04) (Wang, 2004) and the First International Conference on Agent-Based Technologies and Systems (ATS'03) (Wang, 2003b). In the remainder of this article, the historical development that transfers IC to AC is reviewed. Then, a comprehensive set of theoretical foundations and paradigms for AC is explored, encompassing those of behaviorism, cognitive informatics, denotational mathematics, and intelligent science. On the basis of this work, a coherent framework toward AC may be established for the development of interdisciplinary theories and application paradigms.

FROM IMPERATIVE TO AUTONOMIC COMPUTING

The general-purpose computers may do anything unless a specific program is loaded, in which the stored program transfers a computer as a general behavioral implementing machine to specific intelligent applications. The approaches to computing, or ways for embodiment of intelligent behaviors, can be classified into two categories known as IC and AC as given in Definitions 1 and 2.

Table 1. Classification of computing methodologies and systems

		Behavior (O)	
		Constant	Variable
Event (I)	Constant	**Routine**	**Adaptive**
	Variable	**Algorithmic**	**Autonomic**
Type of behavior		*Deterministic*	*Nondeterministic*

The IC system is a traditional and passive system that implements deterministic, context-free, and stored-program controlled behaviors, where a behavior is defined as a set of observable actions of a given computing system. However, the AC system is an active system that implements non-deterministic, context-dependent, and adaptive behaviors. 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.

In its AC manifesto, IBM proposed eight conditions setting forth an AC system known as *self-awareness, self-configuration, self-optimization, self-maintenance, self-protection* (security and integrity), *self-adaptation, self-resource-allocation,* and *open-standard-based* (IBM, 2001). Kinsner pointed out that these characteristics indicate that IBM perceives AC is a mimicry of human nervous systems (Kinsner, 2007). In other words, self-awareness (consciousness) and non-imperative (goal-driven) behaviors are the main characteristics of AC systems (Wang, 2007c).

According to cognitive informatics, these eight characteristics of AC identified by IBM may be sufficient to identify an adaptive system rather than an autonomic system because *adaptive* behaviors can be implemented by IC techniques, but *autonomic* behaviors may only be implemented by non-imperative and intelligent means. This leads to the formal description of the conditions and basic characteristics of AC, and what distinguish AC systems from conventional IC systems.

- **Theorem 1:** The *necessary and sufficient conditions of IC,* C_{IC}, are the possession of *event* B_e, *time* B_t, and *interrupt* B_{int} driven computational behaviors, that is:

$$C_{IC} = (B_e, B_t, B_{int}) \quad (1)$$

- **Theorem 2.** The *necessary and sufficient conditions of AC,* C_{AC}, are the possession of *goal* B_g and *inference* B_{inf} driven computational behaviors, in addition to the *event* B_e, *time* B_t, and *interrupt* B_i driven behaviors, that is:

$$C_{AC} = (B_g, B_{inf}, B_e, B_t, B_{int}) \quad (2)$$

- **Corollary 1.** The behavioral space of *IC,* C_{IC}, is a subset of AC, C_{AC}; or in other words, C_{AC} is a natural extension of C_{IC}, that is:

$$C_{IC} \subseteq C_{AC} \quad (3)$$

The theory and philosophy behind AC are cognitive informatics (Wang, 2002a, 2003a, 2007b). Cognitive processes of the brain, particularly the perceptive and inference cognitive processes, are the fundamental means for describing AC paradigms, such as robots, software agent systems, and distributed intelligent networks. In recent research in cognitive informatics, *perceptivity* is recognized as *the sixth sense* that serves the brain as the thinking engine and the kernel of the natural intelligence. Perceptivity implements self-consciousness inside the abstract memories of the brain. Almost all cognitive life functions rely on perceptivity such as *consciousness, memory searching, motivation, willingness, goal setting, emotion, sense of spatiality,* and *sense of motion*.

In cognitive informatics research, it is recognized that AI is a subset of natural intelligence (NI) (Wang, 2007d). Therefore, AC may be referred to as the natural intelligence and behaviors of human beings. A Layered Reference Model of the Brain (LRMB) was developed (Wang, 2006g) that provides a reference model for the design and implementation of AC systems. LRMB expresses a systematical view toward the formal description and modeling of architectures and behaviors of AC systems, 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 six layers known as the *sensation, memory, perception, action, meta,* and *higher cognitive* *layers* from the bottom up. The cognitive model of the brain can be used as a reference model for goal-driven technologies in AC.

BEHAVIORISM FOUNDATIONS OF AUTONOMIC COMPUTING

Behaviorism is a doctrine of psychology and intelligence science that reveals the association between a given stimulus and an observed response of NI or AI systems, which is developed on the basis of associationism (Sternberg, 1998).

Cognitive informatics reveals that human and machine behaviors may be classified into four categories known as the *perceptive behaviors, cognitive behaviors, instructive behaviors,* and *reflective behaviors* (Wang, 2007d). This section investigates the behavioral spaces and their basic properties of IC and AC.

The Behavioral Model of Imperative Computing

Before the development of the behavioral model of IC, a discussion of the types of events that trigger a conventional computational behavior is needed.

- **Definition 3:** An event is an advanced type in computing that captures the occurring of a predefined external or internal change of status, such as an action of users, an external change of environment, and an internal change of the value of a control variable.

The event types that may trigger a behavior can be classified into operational (@*e***S**), time (@*t***TM**), and interrupt (@*int*⦿) events as shown in Table 2, where @ is the *event prefix*, and **S**, **TM**, and ⦿ are the type suffixes, respectively.

In Table 2, the time type suffix **TM** as defined in RTPA (Wang, 2002b) can be extended to three subtypes:

$$\mathbf{TM} = \mathbf{hh:mm:ss:ms} \mid \mathbf{yy:MM:dd} \mid \mathbf{yyyy:MM:dd: hh:mm:ss:ms} \quad (4)$$

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. In a real-time environment, an ISR should just conduct the most necessary functions and must be short enough compared with the time slice scheduled for a normal process. A formal model of interrupt can be described next.

- **Definition 4:** An *interrupt*, denoted by ↯, is a process relation in which a running process *P* is temporarily held before termination by another higher priority process *Q* via an interrupt event @*int*⦿ at the interrupt point ⦿, and the interrupted process will be resumed when the high priority process has been completed, that is:

$$P \text{↯} Q \triangleq P \,\|\, (@int\odot \nearrow Q \searrow \odot) \quad (5)$$

Table 2. Types of events in imperative computing

No.	Type	Syntax	Usage in system behavioral description	Category
1	Operational event	@*e***S**	$@e_k\mathbf{S} \hookrightarrow_t P_k$	External or internal
2	Time event	@*t***TM**	$@t_k\mathbf{TM} \hookrightarrow_e P_k$	Internal
3	Interrupt event	@*int*⦿	$@int_k\odot \hookrightarrow_{int} P_k$	External or internal

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

According to Theorem 1, the three generic behaviors of IC known as the event-, time-, and interrupt-driven behaviors, can be defined as follows.

- **Definition 5:** An *event-driven behavior*, denoted by $\hookrightarrow_e$, is a machine cognitive process in which the *k*th behavior in term of process P_k is triggered by a predefined system event $@e_k\textbf{S}$, that is:

$$\mathop{R}\limits_{k=1}^{n} @e_k\textbf{S} \hookrightarrow_{\text{e}} P_k \qquad (6)$$

- **Definition 6:** A *time-driven behavior*, denoted by $\hookrightarrow_t$, is a machine cognitive process in which the *k*th behavior in terms of process P_k is triggered by a predefined time point $@t_k\textbf{TM}$, that is:

$$\mathop{R}\limits_{k=1}^{n} @t_k\textbf{TM} \hookrightarrow_{\text{t}} P_k \qquad (7)$$

where the time point $@t_i$ may be a system timing or an external timing event.

- **Definition 7:** An *interrupt-driven behavior*, denoted by $\hookrightarrow_i$, is a machine cognitive process in which the *k*th behavior in terms of process P_k is triggered by a predefined system interrupt $@int_k\odot$, that is:

$$\mathop{R}\limits_{k=1}^{n} @int_k \odot \hookrightarrow_{\text{int}} P_k \qquad (8)$$

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.

On the basis of Theorem 1 and Definitions 5 through 7, the mathematical model of a generic IC system can be described as follows.

- **Definition 8:** The *imperative computing system*, $\S_{IC}$, is an abstract logical model of conventional computing platforms denoted by a set of parallel or concurrent computing resources and behaviors as shown in Figure 1, where || denotes the parallel relation between given components of the system.

As shown in Figure 1, an IC system $\S_{IC}$ is the executing platform or the operating system that controls all the computing resources of an abstract target machine. The IC system is logically abstracted as a set of process behaviors and underlying resources, such as the memory, ports, the system clock, and system status. An IC behavior in terms of a process P_k is controlled and dispatched by the system $\S_{IC}$, which is triggered by various external, system timing, or interrupt events (Wang, 2007d).

The Behavioral Model of Autonomic Computing

AC extends the conventional behaviors of IC as discussed in the preceding subsection to more powerful and intelligent ones such as goal-driven and inference-driven behaviors. According to Theorem 2, with the possessing of all the five forms of intelligent behaviors, AC has advanced closer to the intelligent power of human brains from IC.

- **Definition 9.** A *goal-driven behavior*, denoted by $\hookrightarrow_g$, is a machine cognitive process in which the *k*th behavior in term of process P_k is triggered by a given goal $@g_k\textbf{RT}$, that is:

$$\mathop{R}\limits_{k=1}^{n} @g_k\textbf{ST} \hookrightarrow_{\text{g}} P_k \qquad (9)$$

where the goal $@g_k\textbf{ST}$ is in the system type $\textbf{ST}$ that denotes a structured description of the goal.

In Definition 9, the goal may be formally described as follows.

- **Definition 10:** A *goal,* denoted by $@g_k\textbf{ST}$, is a triple, that is:

Figure 1. The imperative computing system model

$$\S_{IC} \triangleq \text{Imperative-SysID}\mathbf{S} ::$$

$$\{ \quad < \mathop{R}_{i\mathbf{N}=0}^{n_{proc}\mathbf{N}-1} P_i\mathbf{ST}> \qquad \text{// Processes}$$

$$\| < \mathop{R}_{addr\mathbf{P}=0}^{n_{MEM}\mathbf{H}-1} \text{MEM}[ptr\mathbf{P}]\mathbf{RT}> \qquad \text{// Memory}$$

$$\| < \mathop{R}_{ptr\mathbf{P}=0}^{n_{PORT}\mathbf{H}-1} \text{PORT}[ptr\mathbf{P}]\mathbf{RT}> \qquad \text{// Ports}$$

$$\| <\S t\mathbf{TM}> \qquad \text{// The system clock}$$

$$\| < \mathop{R}_{k\mathbf{N}=0}^{n_e\mathbf{N}-1} @e_k\mathbf{S} \hookrightarrow P_k> \qquad \text{// \underline{Event-driven behaviors}}$$

$$\| < \mathop{R}_{k\mathbf{N}=0}^{n_t\mathbf{N}-1} @t_k\mathbf{TM} \hookrightarrow P_k> \qquad \text{// \underline{Time-driven behaviors}}$$

$$\| < \mathop{R}_{k\mathbf{N}=0}^{n_{int}\mathbf{N}-1} @int_k\odot \hookrightarrow P_k > \qquad \text{// \underline{Interrupt-driven behaviors}}$$

$$\| < \mathop{R}_{i\mathbf{N}=0}^{n_V\mathbf{N}-1} V_i\mathbf{RT}> \qquad \text{// System variables}$$

$$\| < \mathop{R}_{i\mathbf{N}=0}^{n_S\mathbf{N}-1} ⓢS_i\mathbf{BL}> \qquad \text{// System statuses}$$

$$\}$$

$$@g_k\mathbf{ST} = (P, \Omega, \Theta) \tag{10}$$

where $P = \{p_1, p_2, ..., p_n\}$ is a non-empty finite set of purposes or motivations, Ω is a set of constraints for the goal, and Θ is the environment of the goal.

Therefore, to some extent, AC is a goal-driven problem solving process by machines that searches a solution for a given problem or finds a path to reach a given goal. 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 may be known or unknown; and (b) the *divergent* problem where the goal of problem solving is unknown and the path of problem solving is either known or unknown. The combination of these cases in problem solving can be summarized in Table 3. A special case in Table 3 is that when both the goal and path are known, the case is a solved instance of a given problem.

According to Theorem 2, *inference capability* is the second extension of AC on top of the capabilities of IC, which is a cognitive process that reasons 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 11.** An *inference-driven behavior*, denoted by $\hookrightarrow_{\text{inf}}$, is a machine cognitive process in which the kth behavior in terms of process P_k is triggered by a given result of inference process $@inf_k\mathbf{ST}$, that is:

$$\mathop{R}_{k=1}^{n} @inf_k\mathbf{ST} \hookrightarrow_{\text{inf}} P_k \tag{11}$$

Formal inferences can be classified into the *deductive, inductive, abductive,* and *analogical* categories (Wang, 2007d). A summary of the formal definitions of the four inference techniques will be described in Table 5 in the section of denotational mathematics.

On the basis of the definitions of the behavioral space of AC, a generic AC system may be rigorously modeled next.

Table 3. Classification of problems and goals

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 (Open-ended)

Figure 2. The autonomic computing system model

$$\S_{AC} \triangleq \text{Autonomic-SysID}\mathbf{S} ::$$

$$\{ \; < \mathop{R}\limits_{i\mathbf{N}=0}^{n_{proc}\mathbf{N}-1} P_i\mathbf{ST}> \qquad \text{// Processes}$$

$$|| < \mathop{R}\limits_{addr\mathbf{P}=0}^{n_{MEM}\mathbf{H}-1} \text{MEM}[ptr\mathbf{P}]\mathbf{RT}> \qquad \text{// Memory}$$

$$| \quad | < \mathop{R}\limits_{ptr\mathbf{P}=0}^{n_{PORT}\mathbf{H}-1} \text{PORT}[ptr\mathbf{P}]\mathbf{RT}> \qquad \text{// Ports}$$

$$|| <\S t\mathbf{TM}> \qquad \text{// The system clock}$$

$$|| < \mathop{R}\limits_{k\mathbf{N}=0}^{n_e\mathbf{N}-1} @e_k\mathbf{S} \hookrightarrow P_k> \qquad \text{// } \textit{Event}\text{-driven behaviors}$$

$$|| < \mathop{R}\limits_{k\mathbf{N}=0}^{n_t\mathbf{N}-1} @t_k\mathbf{TM} \hookrightarrow P_k> \qquad \text{// } \textit{Time}\text{-driven behaviors}$$

$$|| < \mathop{R}\limits_{k\mathbf{N}=0}^{n_{\text{int}}\mathbf{N}-1} @int_k\odot \hookrightarrow P_k > \qquad \text{// } \textit{Interrupt}\text{-driven behaviors}$$

$$|| < \mathop{R}\limits_{k\mathbf{N}=0}^{n_t\mathbf{N}-1} @g_k\mathbf{ST} \hookrightarrow P_k> \qquad \text{// } \underline{\text{Goal-driven behaviors}}$$

$$|| < \mathop{R}\limits_{k\mathbf{N}=0}^{n_{\text{int}}\mathbf{N}-1} @inf_k\mathbf{ST} \hookrightarrow P_k > \qquad \text{// } \underline{\text{Inference-driven behaviors}}$$

$$|| < \mathop{R}\limits_{i\mathbf{N}=0}^{n_V\mathbf{N}-1} V_i\mathbf{RT}> \qquad \text{// System variables}$$

$$|| < \mathop{R}\limits_{i\mathbf{N}=0}^{n_S\mathbf{N}-1} \textcircled{S}S_i\mathbf{BL}> \qquad \text{// System statuses}$$

$$\}$$

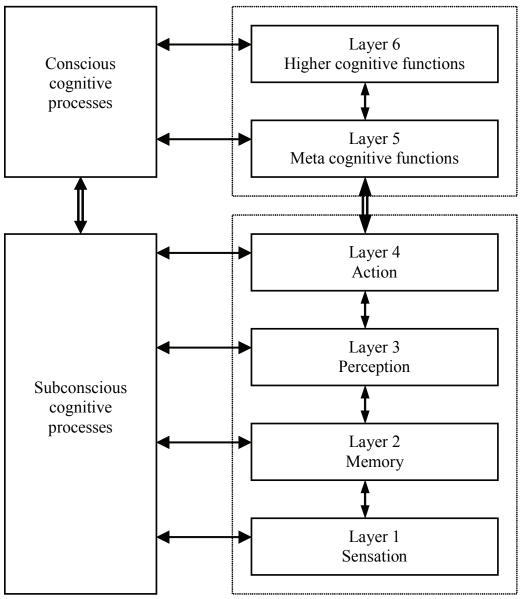
- **Definition 12:** The *AC System*, $§_{AC}$, is an abstract logical model of computing platform denoted by a set of parallel or concurrent computing resources and behaviors as shown in Figure 2, where || denotes the parallel relation between given components of the system.

COGNITIVE INFORMATICS FOUNDATIONS OF AUTONOMIC COMPUTING

Cognitive processes of the brain, particularly the perceptive cognitive processes, are the fundamental means for describing AC systems, such as robots, software agent systems, and distributed intelligent networks. In recent research in cognitive informatics, *perceptivity* is recognized as *the sixth sense* that serves the brain as the thinking engine and the kernel of the natural intelligence (Wang, Wang, Patel, & Patel, 2006). Perceptivity implements self-consciousness inside the abstract memories of the brain. Almost all cognitive life functions rely on perceptivity such as *consciousness, memory searching, motivation, willingness, goal setting, emotion, sense of spatiality,* and *sense of motion*.

The LRMB reference model (Wang et al., 2006) is developed to explain the fundamental cognitive mechanisms and processes of natural intelligence. Because a variety of life functions and cognitive processes have been identified in CI, psychology, cognitive science, brain science, and neurophilosophy, there is a need to organize all the recurrent cognitive processes in an integrated and coherent framework. The LRMB model explains the functional mechanisms and cognitive processes of natural intelligence that encompasses 39 cognitive processes at six layers known as the *sensation, memory, perception, action, meta cognitive,* and *higher cognitive layers* from the bottom-up as shown in Figure 3. LRMB elicits the core and highly repetitive recurrent cognitive processes from a 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 AC, particularly the relationships and interactions between the

Figure 3. The layered reference model of the brain



inherited and the acquired life functions as well as those of the subconscious and conscious cognitive processes.

The LRMB establishes a reference model for implementing AC systems, which provides insightful and multidisciplinary theories for the design and implementation of AC systems. According to LRMB, the brain as well as an AC system can be formally treated as a real-time information processing system at the functional level as described next.

- **Definition 13:** The *cognitive system model* of the brain or an AC system, *NI-Sys*, can be described as a real-time natural intelligent system with an inherited operating system (thinking engine) *NI-OS* and a set of acquired life applications *NI-App*, that is:

$$NI\text{-}Sys \triangleq NI\text{-}OS \; || \; NI\text{-}App \qquad (12)$$

where *NI-OS* represents the inherited life functions, *NI-App* the developed life functions, and || a parallel relation.

Corresponding to Figure 3, the Level 1 through Level 4 life functions belong to *NI-OS*, and Level 5 and Level 6 life functions are fundamental parts of *NI-App*. All everyday behaviors can be perceived as a real-time combination and invocation of these fundamental life functions at various levels of LRMB.

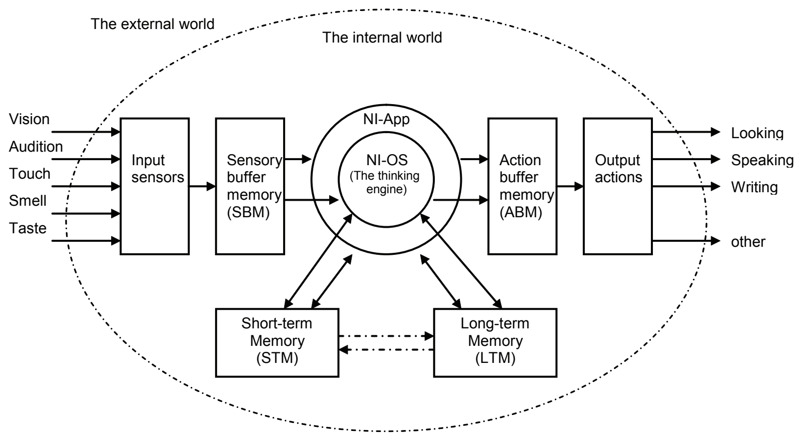
The characteristics of *NI-OS* have been observed as follows:

- Inherited
- Wired (by neural networks)
- Working subconsciously
- A real-time system
- Running subconsciously and automatically
- Person-independent, common, and similar
- Highly parallel and fault-tolerant
- With event/time/interrupt/goal/inference-driven mechanisms

In contrary to the *NI-OS*, the characteristics of *NI-App* have been identified as follows:

- Acquired
- Partially wired (frequently used functions) and partially programmed (temporary functions)
- Working consciously
- Can be trained and programmed
    - o Person-specific

Figure 4. The functional model of the brain (Wang & Wang, 2006)



The goal-driven and inference-driven mechanisms are unique features of NI and AC systems that are autonomously determined by internal events or conditions such as emotions, desires, and rational reasoning.

The LRMB model for AC can be extended to a functional model as illustrated in Figure 4 with the *NI-Sys* (*NI-OS* || *NI-App*), LTM, STM, SBM (connected with a set of sensors), and ABM (connected with a set of servos). In Figure 4, the kernel of the brain is the natural intelligence system (*NI-Sys*), which is *the thinking engine* of the brain as described in the LRMB model.

It is noteworthy that each abstract object is physically stored in a different type of the memories as given next.

- **Theorem 3.** 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), and Action-Buffer Memory (ABM), that is:

$$\begin{aligned} \text{CMM} \triangleq\ & \text{SBM} \\ & \| \ \text{STM} \\ & \| \ \text{LTM} \\ & \| \ \text{ABM} \end{aligned} \qquad (13)$$

The CMM model presents the neural informatics foundation of natural intelligence, and the physiological evidences of why natural intelligence may be classified into four forms as given in Theorem 4 in the next section.

With the CMM model, an AC system can be implemented by mimicking the following abstract brain models.

- **Definition 14:** The *functional model of the brain, BRAIN*, as a real-time system and a high-level logical model of the brain, describes the functional configuration of the brain and how the *NI-Sys* interacts with the memory system, that is:

$$\begin{aligned} BRAIN \triangleq\ & NI\_Sys \\ & \,||\, CMM \\ =\ & (\ NI\_OS \\ & \quad ||\, NI\_App \\ & \ ) \\ & ||\, (\ LTM \\ & \quad ||\, STM \\ & \quad ||\, SBM \\ & \quad ||\, ABM \\ & \ ) \end{aligned} \qquad (14)$$

Equation 14 indicates that although the thinking engine *NI-Sys* is considered the center of the natural intelligence, the memories are essential to enable the *NI-Sys* to properly function, and to keep temporary and stable results stored and retrievable.

The corresponding biological organ of *NI-OS* in neurophysiology is *thalamus*—a switching center located above the midbrain, which possesses tremendous amounts of connections to almost all parts of the brain, especially cerebral cortex, eyes, and visual cortex (Sternberg, 1998). *NI-Sys* interacts with LTM and STM in a bi-directional way, which forms the basic functionality of the brain as a thinking machine. STM provides working space for the *NI-Sys*, and LTM stores both cumulated information (knowledge) and wired and usually subconscious procedures (skills).

The *NI-Sys* communicates with the external world through inputs and outputs. The former are sensorial information, including vision, audition, touch, smell, and taste. The latter are action and behaviors of life functions, such as looking, speaking, writing, and driving. The actions and behaviors generated in the brain, either from *NI-OS* or *NI-App*, are buffered in the ABM before they are executed and outputted to implement the predetermined actions and behaviors. Therefore, ABM plays an important role in the brain to plan and implement human behaviors. However, it was overlooked in the literature of neuropsychology and cognition science (Gabrieli, 1998; Pinel, 1997; Sternberg, 1998).

It is noteworthy that unlike a computer, the brain works in two approaches: the internal *willingness-driven* processes (in *NI-OS*), and the external *event-* and *time-driven* processes (in *NI-App*). The external information and events are the major sources that drive the brain, particularly for *NI-App* functions. In this case, the brain may be perceived as a passive system, at least when it is conscious, which is controlled and driven by external information. Even the internal willingness, such as goals, desires, and emotions, may be considered as derived information based on originally external information.

A fundamental question in cognitive psychology is how consciousness can be the product of physiological processes in the brain. Similarly, the fundamental question for AC is how autonomic behaviors may be generated by non-imperative processes on generic computers. The cognitive models developed in this section reveal, as that of IC is controlled by stored-programs, AC should be controlled by predefined and learned cognitive processes as identified in LRMB.

DENOTATIONAL MATHEMATICS FOUNDATIONS OF AUTONOMIC COMPUTING

As that of IC is based on mathematical foundations of Boolean algebra, the more intelligent capability of AC should be treated by more powerful mathematical structures known as denotational mathematics in the form of system algebra (Wang, 2006c), concept algebra (Wang, 2006d), and real-time process algebra (RTPA) (Wang, 2002b).

The history of sciences and engineering shows that new problems require new forms of mathematics. AC and intelligence science are new disciplines, and the problems in them require new mathematical means that are descriptive and explicit in expressing and denoting human and system behaviors in the non-imperative approach. Conventional *analytic mathematics* are unable to solve the fundamental problems inherited in AC and related disciplines. Therefore, *denotational mathematical structures and means* (Wang, 2006b) beyond set theory and mathematical logic are yet to be sought.

Although there are various ways to express facts, objects, notions, relations, actions, and behaviors in natural languages, it is found in cognitive informatics that human and AC system behaviors may be classified into three basic categories known as to *be*, to *have*, and to *do*. All mathematical means and forms, in general, are an abstract and formal description of these three categories of expressibility and their rules. Adopting this view, mathematical logic may be perceived as the abstract means for describing "to be," set theory describing "to have," and algebras, particularly process algebra, describing "to do."

Three forms of new mathematics known as concept algebra, system algebra, and RTPA are created in cognitive informatics to enable rigorous treatment of knowledge representation and manipulation in a formal and coherent framework. The three new structures of

Table 4. Conventional and contemporary mathematical means for denotational problems

Basic denotational problems in AC	Classic mathematics	Contemporary mathematics
To be	Logic	Concept algebra
To have	Set theory	System algebra
To do	Functions	RTPA

contemporary mathematics have extended the abstract objects under study in mathematics from basic mathematical entities of numbers and sets to higher levels, that is, concepts, systems, and behavioral processes as shown in Table 4. Table 4 indicates that, in general, the utility of mathematics is the means and rules to express thought rigorously and generically at a higher level of abstraction.

It is recognized that intelligent inference capability is based on the cognitive process of abstraction. *Abstraction* is not only a powerful means of philosophy and mathematics, but also a preeminent trait of the human brain identified in cognitive informatics studies (Wang, 2005). All formal logical inferences and reasoning can only be carried out on the basis of abstract properties shared by a given set of objects under study.

- **Definition 15:** *Abstraction* is a process to elicit a subset of objects that shares a common property from a given set of objects and to use the property to identify and distinguish the subset from the whole in order to facilitate reasoning, that is:

$$\forall S, p \Rightarrow \exists\, e \in E \subseteq S, p(e) \qquad (15)$$

Abstraction is a gifted capability of human beings, which is identified as a basic cognitive process of the brain at the meta cognitive layer according to LRMB (Wang et al., 2006). Only by abstraction can important theorems and laws about the objects under study be elicited and discovered from a great variety of phenomena and empirical observations in an area of knowledge inquiry.

On the basis of abstraction, formal inferences may be classified into the deductive, inductive, abductive, and analogical categories as shown in Table 5 (Wang, 2005). *Deduction* is a cognitive process by which a specific conclusion necessarily follows from a set of general premises. *Induction* is a cognitive process by which a general conclusion is drawn from a set of specific premises based on three designated samples in reasoning or experimental evidences.

Table 5. Definitions of formal inferences processes

No.	Inference technique	Formal description		Usage
		Primitive form	Composite form	
1	Deduction	$\forall x \in \mathbb{X}, p(x) \vdash \exists a \in \mathbb{X}, p(a)$	$(\forall x \in \mathbb{X}, p(x) \Rightarrow q(x))\ \forall x \in \mathbb{X}, p(x) \vdash \exists a \in \mathbb{X}, p(a)\ (\exists a \in \mathbb{X}, p(a) \Rightarrow q(a))$	To derive a conclusion based on a known and generic premises.
2	Induction	$((\exists a \in \mathbb{X}, P(a)) \wedge (\exists k, k+1 \in \mathbb{X}, (P(k) \Rightarrow P(k+1))) \vdash \forall x \in \mathbb{X}, P(x)$	$((\exists a \in \mathbb{X}, p(a) \Rightarrow q(a)) \wedge (\exists k, k+1 \in \mathbb{X}, ((p(k) \Rightarrow q(k)) \Rightarrow (p(k+1) \Rightarrow q(k+1)))) \vdash \forall x \in \mathbb{X}, p(x) \Rightarrow q(x)$	To determine the generic behavior of the given list or sequence of recurring patterns by three samples.
3	Abduction	$(\forall x \in \mathbb{X}, p(x) \Rightarrow q(x)) \vdash (\exists a \in \mathbb{X}, q(a) \Rightarrow p(a))$	$(\forall x \in \mathbb{X}, p(x) \Rightarrow q(x) \wedge r(x) \Rightarrow q(x)) \vdash (\exists a \in \mathbb{X}, q(a) \Rightarrow (p(a) \vee r(a)))$	To seek the most likely cause(s) and reason(s) of an observed phenomenon.
4	Analogy	$\exists a \in \mathbb{X}, p(a) \vdash \exists b \in \mathbb{X}, p(b)$	$(\exists a \in \mathbb{X}, p(a) \Rightarrow q(a)) \vdash (\exists b \in \mathbb{X}, p(b) \Rightarrow q(b))$	To predict a similar phenomenon or consequence based on a known observation.

Abduction is a cognitive process by which an inference to the best explanation or most likely reason of an observation or event. *Analogy* is a cognitive process by which an inference about the similarity of the same relations holds between different domains or systems, and/or examines that if two things agree in certain respects then they probably agree in others.

Detailed descriptions of the formal cognitive inference processes for AC as listed in Table 5 may be referred to Wang (2005), which can be used to simulate machine cognitions and the implementation of inference engines for AC systems on the basis of denotational mathematics.

INTELLIGENT SCIENCE FOUNDATIONS OF AUTONOMIC COMPUTING

Intelligence is perceived as the driving force or the ability to acquire and use knowledge and skills, or to reasoning in problem solving. It was conventionally perceived that only human beings possess higher-level intelligence. However, the development of computers, robots, and autonomic systems indicates that intelligence may also be created or implemented by machines and manufactured systems. This is the intelligent behavioral foundation for designing and implementing AC systems.

- **Definition 16:** *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.

With the clarification of the intension and extension of the generic concept of intelligence, the terms of natural and artificial intelligence can be derived next.

- **Definition 17:** *Natural intelligence* is the intelligent capability possessed or implemented by the brains of human beings and other advanced species.
- **Definition 18:** *Artificial intelligence* is the intelligent capability possessed or implemented by machines or manufactured systems.

Intelligence can be formally modeled as a set of functions that transfers a pair of abstract objects in the brain or system as given in Definitions 17 or 18.

- **Theorem 4:** The *nature of intelligence* states that *intelligence* $\Im$ can be classified into four forms called the *perceptive intelligence* $\Im_p$, *cognitive intelligence* $\Im_c$, *instructive intelligence* $\Im_i$, and *reflective intelligence* $\Im_r$ as modeled next:

$$\begin{aligned} \Im \triangleq\ & \Im_p : D \to I \ \text{(Perceptive)} \\ & \| \Im_c : I \to K \ \text{(Cognitive)} \\ & \| \Im_i : I \to B \ \text{(Instructive)} \\ & \| \Im_r : D \to B \ \text{(Reflective)} \end{aligned} \qquad (16)$$

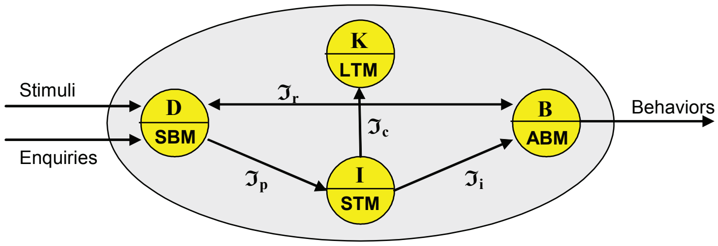
According to Definition 16 and Theorem 4, the narrow sense of intelligence corresponds to the instructive and reflective intelligence, while the broad sense of intelligence includes all four forms of intelligence, that is, the perceptive, cognitive, instructive, and reflective intelligence.

On the basis of the conceptual models developed in this subsection, the mechanisms of NI can be described by a generic intelligence model as given in Definition 19 and Figure 5.

- **Definition 19:** The *Generic Intelligence Model* (GIM) describes the mechanisms of NI, as shown in Figure 5, according to Theorem 4 on the nature of intelligence.

In Figure 5, four forms of natural intelligence are described as the driving forces that transfer between a pair of abstract objects in the brain or an AC system such as data (D), information (I), knowledge (K), and behavior (B).

Figure 5. The Generic Intelligence Model (GIM)



$\mathfrak{I}_p$ *Perceptive* intelligence
$\mathfrak{I}_i$ *Instructive* intelligence
$\mathfrak{I}_c$ *Cognitive* intelligence
$\mathfrak{I}_i$ *Reflective* intelligence

The GIM and Theorem 4 reveal that NI and AI share the same cognitive informatics foundations. In other words, they are compatible. Therefore, on the basis of Theorem 4, the studies on NI and AI in cognitive informatics, and AC in particular, may be unified into a common framework.

The intelligent behavioral foundations of AC as given in the GIM provide a new paradigm of AC, which reveals that an AC system may not only implement the reflective and instructive intelligence, but also implement the cognitive and perceptive intelligence according to the theory of the intelligent behavioral paradigm.

CONCLUSION

This article has presented a new perspective on autonomic computing as a novel computing system with the highest level of machine intelligence, which embodies the goal- and inference-driven computational behaviors on top of imperative computing techniques with event-, time-, and interrupt-driven computational behaviors. This article has explored the theoretical foundations and engineering paradigms of AC. A comprehensive set of theoretical foundations for AC, such as those of behaviorism, cognitive informatics, denotational mathematics, and intelligent science, has been identified. The findings of this work, particularly the theorems of the necessary and sufficient conditions of imperative and autonomic computing, and the generic intelligence model of natural and machine intelligence, have formed a solid foundation for understanding and developing advanced autonomic computing techniques 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

Brooks, R. A. (1970). New approaches to robotics. *American Elsevier, 5*, 3-23.

Gabrieli, J. D. E. (1998). Cognitive neuroscience of human memory. *Annual Review of Psychology, 49*, 87-115.

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

IBM. (2001). *IBM autonomic computing manifesto*. Retrieved from http://www.research. ibm.com/autonomic/

IBM. (2006). *An architectural blueprint for autonomic computing* (4th ed.) (White Paper).

Jennings, N. R. (2000). On agent-based software engineering. *Artificial Intelligence, 17*(2), 277-296.

Kephart, J., & Chess, D. (2003). The vision of autonomic computing. *IEEE Computer, 26*(1), 41-50.

Kinsner, W. (2007). Towards cognitive machines: Multiscale measures and analysis. *The International Journal on Cognitive Informatics and Natural Intelligence (IJCINI), 1*(1), 28-38.

Kleene, S. C. (1956). Representation of events by nerve nets. In C. E. Shannon & J. McCarthy (Eds.), *Automata studies* (pp. 3-42). Princeton Univ. Press.

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

McCulloch, W. S. (1993). *The complete works of Warren S. McCulloch*. Salinas, CA: Intersystems Pub.

McCulloch, W. S., & Pitts, W. H. (1943). A logic calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics, 5*.

Meystel, A. M., & Albus, J. S. (2002). *Intelligent systems, architecture, design, and control*. John Wiley & Sons, Inc.

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

Pescovitz, D. (2002). Autonomic computing: Helping computers help themselves. *IEEE Spectrum, 39*(9), 49-53.

Pinel, J. P. J. (1997). *Biopsychology* (3rd ed.). Needham Heights, MA: Allyn and Bacon.

Rabin, M. O., & Scott, D. (1959). Finite automata and their decision problems. *IBM Journal of Research and Development, 3*, 114-125.

Shannon, C. E. (Ed.). (1956). *Automata studies*. Princeton, NJ: Princeton University Press.

Sternberg, R. J. (1998). *In search of the human mind* (2nd ed.). Orlando, FL: Harcourt Brace & Co.

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

von Neumann, J. (1946). The principles of large-scale computing machines. *Annals of History of Computers, 3*(3), 263-273.

von Neumann, J. (1958). *The computer and the brain*. New Haven, CT: Yale Univ. Press.

von Neumann, J. (1963). General and logical theory of automata. In A. H. Taub (Ed.), *Collected works* (Vol. 5) (pp. 288-328). Pergamon.

von Neumann, J., & Burks, A. W. (1966). *Theory of self-reproducing automata*. Urbana, IL: Univ. of Illinois Press.

Wang, Y. (2002a). Keynote speech: On cognitive informatics. In *Proceedings of the First IEEE International Conference on Cognitive Informatics (ICCI'02)* (pp. 34-42). Calgary, Canada: IEEE CS Press.

Wang, Y. (2002b). The Real-Time Process Algebra (RTPA). *Annals of Software Engineering: An International Journal, 14*, 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 speech: Cognitive informatics models of software agent systems and autonomic computing. In Proceedings of the First International Conference on Agent-Based Technologies and Systems (ATS'03) (p. 25). Calgary, Canada: Univ. of Calgary Press.

Wang, Y. (2004). Keynote speech: On autonomic computing and cognitive processes. In *Proceedings of the Third IEEE International Conference on Cognitive Informatics (ICCI'04)* (pp. 3-4). Victoria, Canada: IEEE CS Press.

Wang, Y. (2005). The cognitive processes of abstraction and formal inferences. In *Proceedings of the Fourth IEEE International Conference on Cognitive Informatics (ICCI'05)* (pp. 18-26). Irvin, CA: IEEE CS Press.

Wang, Y. (2006a). Keynote speech: Cognitive informatics—Towards the future generation computers that think and feel. In *Proceedings of the Fifth IEEE International Conference on Cognitive Informatics (ICCI'06)* (pp. 3-7). Beijing, China: IEEE CS Press.

Wang, Y. (2006b). Cognitive informatics and contemporary mathematics for knowledge representation and manipulation: Invited plenary talk. In *Proceedings of the First International Conference on Rough Set and Knowledge Technology (RSKT'06)* (pp. 69-78). Chongqing, China: Springer.

Wang, Y. (2006c). On abstract systems and system algebra. In *Proceedings of the Fifth IEEE International Conference on Cognitive Informatics (ICCI'06)* (pp. 332-343). Beijing, China: IEEE CS Press.

Wang, Y. (2006d). On concept algebra and knowledge representation. In *Proceedings of the Fifth IEEE International Conference on Cognitive Informatics (ICCI'06)* (pp. 320-331). Beijing, China: IEEE CS Press.

Wang, Y. (Ed.). (2007a). Special issues on autonomic computing. *The International Journal on Cognitive Informatics and Natural Intelligence (IJCINI), 1*(3).

Wang, Y. (2007b). The theoretical framework of cognitive informatics. *The International Journal of Cognitive Informatics and Natural Intelligence (IJCiNi), 1*(1), 1-27.

Wang, Y. (2007c). Exploring machine cognition mechanisms for autonomic computing. *The International Journal on Cognitive Informatics and Natural Intelligence (IJCINI), 1*(2), i-v.

Wang, Y. (2007d). *Software engineering foundations: A software science perspective, CRC Book Series in Software Engineering* (Vol. II). CRC Press.

Wang, Y., & Kinsner, W. (2006). Recent advances in cognitive informatics. *IEEE Transactions on Systems, Man, and Cybernetics (C), 36*(2), 121-123.

Wang, Y., (2006). On cognitive informatics models of the brain. *IEEE Transactions on Systems, Man, and Cybernetics (C), 36*(2), 16-20.

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

Widrow, B., & Lehr, M. A. (1990). 30 years of adaptive neural networks: Perception, madeline, and backpropagation. In *Proc. of the IEEE, 78*(9), (pp. 1415-1442).

Wiener, N. (1948). *Cybernetics or control and communication in the animal and the machine*. Cambridge, MA: MIT Press.

ENDNOTE

[1] This article is an extension based on the keynote lectures of the author in the Third IEEE International Conference on Cognitive Informatics (ICCI'04) "*On Autonomic Computing and Cognitive Processes* (Wang, 2004)," and in the First International Conference on Agent-Based Technologies and Systems (ATS'03) on "*Cognitive Informatics Models of Software Agents and Autonomic Computing* (Wang, 2003b)."

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 (1997), and a BSc in electrical engineering from Shanghai Tiedao University (1983). He was a visiting professor in the Computing Laboratory at Oxford Uni.versity (1995), and has been a full professor since 1994. He is editor-in-chief of International Journal of Cognitive Informatics and Natural Intelligence *(IJCiNi), editor-in-chief of* IGP Book Series of Advances in Cognitive Informatics and Natural Intelligence*, and editor of* CRC Book Series in Software Engineering. *He has published more than 290 papers and 10 books in software engineering and cognitive informatics, and won dozens of research achievement, best paper, and teaching awards in the last 28 years, particularly the IBC 21st Century Award for Achievement "in recognition of outstanding contribution in the field of cognitive informatics and software science."*