On the System Algebra Foundations for Granular Computing

Yingxu Wang, University of Calgary, Canada

Lotfi A. Zadeh, University of California, Berkeley, USA

Yiyu Yao, University of Regina, Canada

ABSTRACT

Granular computing studies a novel approach to computing system modeling and information processing. Although a rich set of work has advanced the understanding of granular computing in dealing with the "to be" and "to have" problems of systems, the "to do" aspect of system modeling and behavioral implementation has been relatively overlooked. On the basis of a recent development in denotational mathematics known as system algebra, this paper presents a system metaphor of granules and explores the theoretical and mathematical foundations of granular computing. An abstract system model of granules is proposed in this paper. Rigorous manipulations of granular systems in computing are modeled by system algebra. The properties of granular systems are analyzed, which helps to explain the magnitudes and complexities of granular systems. Formal representation of granular systems for computing is demonstrated by real-world case studies, where concrete granules and their algebraic operations are explained.

Keywords: *abstract systems; data granule; denotational mathematics; cognitive granule; computing granule; engineering applications; granules; granular computing; granular operations; granular systems; information granule; system algebra; system granule*

INTRODUCTION

The term *granule* is originated from Latin *granum*, i.e., grain, to denote a small compact particle in physics and in the natural world. The *taxonomy of granules* in computing can be classified into the data granule, information granule, concept granule, computing granule, cognitive granule, and system granule (Zedeh, 1979, 2003; Lin, 1998; Skowron and Stepaniuk, 2001; Yao, 2001, 2004a; Wang, 2007a, 2008c). The study of granular computing as an emerging filed appeared in 1997 (Zadeh, 1997, 1998; Lin, 1998). Granular computing may be viewed as an umbrella term covering theories, strategies, methodologies, techniques, tools, and systems that explore multilevel granularity in information processing, knowledge manipulation, and problem solving (Yao, 2001, 2004a, 2004b, 2005).

The concept of granules in data and information modeling and its fuzzy set treatment can be traced back to the work of L.A. Zedeh in 1979 as given below (Zadeh, 1979, 2003).

Definition 1. *The data granule g is a set with the elements x as a member of a fuzzy set $\tilde{G}$ to the degree of λ, 0 ≤ λ ≤ 1, i.e.:*

$$g \triangleq \{x \mid x \in_{\lambda} \tilde{G} \subseteq U\} \quad (1)$$

where U is the universal discourse.

Many studies investigated into granular computing based on rough sets (Lin, Yao, and Zadeh, 2002). Pawlak (1998) studied *knowledge granularity* using rough sets. Skowron and Stepaniuk (2001) proposed a rough set treatment of *information granules*. Polkowski and Skowron (1998) introduced the *granular calculus*. Lin (1998) studied *relational granules*. Pedrycz (2001) as well as Bargiela and Pedrycz (2002) suggested that granular computing may adopt a pyramid model toward various information granulations. Yao developed a trarachic perspective on granular computing with the facets of philosophy, methodology, and computational implementation (Yao, 2001, 2004a, 2005), which explains the structures of granular computing by multiple levels and views. These studies have advanced the theories of granular computing in dealing with the aspects of system "*to be*" and "*to have*" problems, particularly system architectures and high-level system conceptual designs in computing, software engineering, system engineering, and cognitive informatics. Wang initiated a set of *denotational mathematics* (Wang, 2002b, 2007a, 2007c, 2007d, 2008a) known as *concept algebra* (Wang, 2008b), *system algebra* (Wang, 2008c), and *Real-Time Process Algebra* (RTPA) (Wang, 2002a, 2003b, 2007a, 2008d), which were recognized as an expressive mathematical means for modeling and manipulating all types of granules in granular computing such as the *computing, cognitive, concept, information, data granules*, and *knowledge* granules.

This article presents a new perspective on the system metaphor of granules and granular computing, which extends the conventional set metaphors (Zadeh, 1979; Klir, 1992; Wang, 2007a). The following discusses the relationships between granules/systems and granular computing/system algebra. It will demonstrate that systems may be treated rigorously as a new mathematical structure beyond conventional mathematical entities. Based on this view, the concept of granules and granular computing are discussed below.

Definition 2. *A computing granule, shortly a granule, is a basic mathematical structure that possesses a stable topology and at least a unit of computational capability or behavior.*

It is noteworthy that, comparing Definitions 1 and 2, the computing granule is not a set, but an abstract system (Wang, 2008c) with both a given structure and a set of certain behaviors. The structural and functional models of a granule will be derived in the next section.

Definition 3. *Granularity in system design is the level of abstraction or the extent of details presented in a granule and its computational behaviors in a given level of system hierarchy.*

Definition 4. *Granulation in system design is a process to partite or decompose a computing system into its smallest components step-by-step in a given system hierarchy.*

Definition 5. *Granulometric is a measurement of granularity of a computing system with a certain granulation.*

Based on the taxonomy of granules and granulation, as well as their *system metaphor*, it is naturally perceived that computational behaviors and computing systems can be design and implemented by a set of granules and a process of granulation. In this context, the concept of computing can be described as follows (Zedeh, 2003; Wang, 2007a).

Definition 6. *Computing, in a narrow sense, is an application of computers to solve a given computational problem by imperative instructions; while in a broad sense, it is a process to implement the instructive intelligence by a system that transfers a set of given information or instructions into expected behaviors.*

On the basis of Definitions 1 - 6, the concept of granular computing can be derived as follows.

Definition 7. *Granular computing is a new computational methodology that models and implements computational structures and functions by a granular system, where each granule in the system carries out a predefined function or behavior by interacting to other granules in the system.*

For instance, according to Definition 7, the Internet may be perceived as a granular system at a certain level of granularity, where each distributed computer in the web is a granule. In this Internet granular system, with an individual computer as a reference, a higher-level granularity in the system can be a Local Area Network (LAN), and a lower level granularity can be one of the threads or processes executing on a certain granular machine. This is perfectly in-line with the system philosophy (Ellis and Fred, 1962; Klir, 1992; Wang, 2007a) and models defined in system algebra (Wang, 2007a, 2008c).

This article indicates that, although a rich set of literature on granular computing exists, the implications and rigorous models of granules and granular computing are yet to be systematically studied, particularly the design and implementation of fundamental computing behaviors via granular systems. A new approach is presented in this article with a system metaphor toward granules and the theoretical and mathematical foundations of granular computing on the basis of the recent development in denotational mathematics known as system algebra. The abstract system model of granules is investigated. Rigorous manipulations of granular systems in computing are modeled by system algebra, and the properties of granular systems are qualitatively and quantitatively analyzed. Formal representation and treatment of concrete granules are explained with case studies on real-world granular systems. This article may also be perceived as a paradigm that demonstrates the generality and expressive power of system algebra and the mathematical model of abstract systems.

THE ABSTRACT SYSTEM MODEL OF GRANULES

In order to formally describe the system metaphor of granules and granular computing, the mathematical model of abstract systems and system algebra are introduced in this section. These preparations lead to the establishment of the mathematical models of granules in system algebra, which reveal the nature and fundamental mechanisms of granular systems and granular computing.

The Mathematical Model of Abstract Systems

Systems are the most complicated entities and phenomena in abstract, physical, information, and social worlds across all science and engineering disciplines. The system concept can be traced back to the 17th Century when R. Descartes (1596-1650) noticed the interrelationships among scientific disciplines as a system. Then, the general system notion was proposed by Ludwig von Bertalanffy in the 1920s (von Bertalanffy, 1952; Ellis and Fred, 1962). The theories of system science have evolved from classic theories (Ashby, 1958, 1962; Ellis and Fred, 1962; Rapoport, 1962; Heylighen, 1989; Klir, 1992;) to contemporary theories in the mid of the 20th century such as I. Prigogine's dissipative structure theory (Prigogine et al., 1972), H. Haken's synergetics (Haken, 1977), and M. Eigen's hypercycle theory (Eigen and Schuster, 1979). Then, during late of the last century, there are proposals of complex systems theories (Zadeh, 1973; Klir, 1992), fuzzy theories (Zadeh, 1965, 1973), and chaos theories (Ford, 1986; Skarda and Freeman, 1987). Yingxu Wang found that, because of their extremely wide and frequent usability, systems may be treated rigorously as a new mathematical structure beyond conventional mathematical entities known as the *abstract systems* (Wang, 2008c). Based on this view, the concept of abstract systems and their mathematical models are introduced below.

Definition 8. *An abstract system is a collection of coherent and interactive entities that has stable functions and a clear boundary with the external environment.*

An abstract system forms the generic model of various real-world systems and represents the most common characteristics and properties of them. The granularity of granular computing can be explained by the following lemma in the abstract system theory.

Lemma 1. *The generality principle of system abstraction states that a system can be represented as a whole in a given level k of reasoning, $1 \leq k \leq n$, without knowing the details at levels below k-1.*

Definition 9. *Let E be a finite nonempty set of entities, F a finite nonempty set of functions, V_E a finite nonempty set of domains of E, and V_F a finite nonempty set of domains of F, then the universal system $\mathfrak{U}$, which forms the discourse of abstract systems, is denoted as a 4-tuple, i.e.:*

$$\mathfrak{U} \triangleq (E, F, V_E, V_F) \tag{2}$$

Abstract systems can be classified into two categories known as the *closed* and *open* systems. Most practical and useful systems in nature are open systems in which there are interactions between the system and its environment. That is, they need to interact with external world known as the *environment* in order to exchange energy, matter, and/or information. Such systems are called open systems. Typical interactions between an open system and the environment are inputs and outputs.

Definition 10. *An open system S on $\mathfrak{U}$, $S \sqsubseteq \mathfrak{U}$, is a 7-tuple, i.e.:*

$$S \triangleq (C, R^c, R^i, R^o, B, \Omega\ \Theta) \tag{3}$$

where

- $\sqsubseteq$ denotes that a set or structure (tuple) is a substructure or derivation of another structure known.
- C is a finite nonempty set of *components* of system S, $C \subseteq ÞE \sqsubseteq \mathfrak{U}$ and $C \sqsubseteq S$, where Þ denotes a power set.
- $R^c \subseteq C \times C$ is a finite nonempty set of *internal relations* between pairs of the components $C \sqsubseteq S$.
- $R^i \subseteq C_\Theta \times C$ is a finite nonempty set of *input relations*, where C_Θ is a finite nonempty set of external component, $C_\Theta \subseteq ÞE \sqsubseteq \mathfrak{U}$ and $C_\Theta \not\sqsubseteq S$.
- $R^o \subseteq C \times C_\Theta$ is a finite nonempty set of *output relations*.
- B is a finite nonempty set of *behaviors* of $C \sqsubseteq S$, $B \subseteq ÞF \sqsubseteq \mathfrak{U}$ and $B \sqsubseteq S$.
- Ω is a finite nonempty set of *constraints* of C and B, $\Omega \subseteq ÞV_E \cup ÞV_F \sqsubseteq \mathfrak{U}$ and $\Omega \sqsubseteq S$.
- Θ is the *environment* of S with a finite nonempty set of external components outside S, i.e., $\Theta = C_\Theta \subseteq ÞE \sqsubseteq \mathfrak{U}$. and $\Theta = C_\Theta \not\sqsubseteq S$.

It will be demonstrated throughout this article that the abstract system is an ideal model for rigorously describing both the structures and behaviors of granules in granular computing.

System Algebra

System algebra is an abstract mathematical structure for the formal treatment of abstract and general systems as well as their algebraic relations, operations, and associative rules for composing and manipulating complex systems (Wang, 2007a, 2008c).

Definition 11. *A system algebra SA on a given universal system environment* $\mathfrak{U}$ *is a triple, i.e.:*

$$SA \triangleq (S, OP, \Theta) = (\{C, R^c, R^i, R^o, B, \Omega\}, \{\bullet_r, \bullet_c\}, \Theta) \quad (4)$$

where OP = $\{\bullet_r, \bullet_c\}$ *are the sets of relational and compositional operations, respectively, on abstract systems as defined below (Wang, 2008c).*

Definition 12. *The relational operations* $\bullet_r$ *in system algebra encompass 6 comparative operators for manipulating the algebraic relations between abstract systems, i.e.:*

$$\bullet_r \triangleq \{\nleftrightarrow, \leftrightarrow, \prod, =, \sqsubseteq, \sqsupseteq\} \quad (5)$$

where the relational operators stand for independent, related, overlapped, equivalent, subsystem, and supersystem, respectively.

Definition 13. *The compositional operations* $\bullet_c$ *in system algebra encompass 9 associative operators for manipulating the algebraic compositions among abstract systems, i.e.:*

$$\bullet_c \triangleq \{\Rightarrow, \overset{-}{\Rightarrow}, \overset{+}{\Rightarrow}, \overset{\sim}{\Rightarrow}, \boxminus, \uplus, \pitchfork, \Lleftarrow, \vdash\} \quad (6)$$

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

System algebra provides a denotational mathematical means for algebraic manipulations of all forms of abstract systems. System algebra can be used to model, specify, and manipulate generic "*to be*" and "*to have*" type problems, particularly system architectures and high-level system designs, in computing, software engineering, system engineering, and cognitive informatics.

The Mathematical Model of Granules in System Algebra

It is recognized that any abstract or concrete granule can be formally modeled by abstract systems in system algebra. On the basis of Definition 10, an abstract granule can be formally described as follows.

Definition 14. *A computing granule G on* $\mathfrak{U}$*, G* $\sqsubseteq \mathfrak{U}$*, is a 7-tuple, i.e.:*

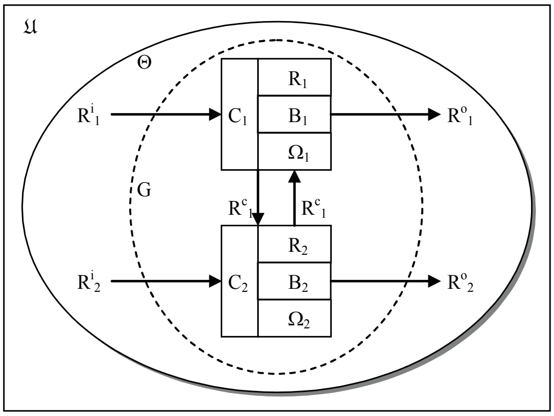
$$G \triangleq S = (C, R^c, R^i, R^o, B, \Omega, \Theta) \quad (7)$$

where

- C is a finite nonempty set of *cells* of system S, $C \subseteq \text{Þ}E \sqsubseteq \mathfrak{U}$ and $C \sqsubseteq S$, where Þ denotes a power set.
- $R^c \subseteq C \times C$ is a finite nonempty set of *internal relations* between pairs of the components $C \sqsubseteq S$.
- $R^i \subseteq C_\Theta \times C$ is a finite nonempty set of *input relations*, where C_Θ is a finite nonempty set of external component, $C_\Theta \subseteq \text{Þ}E \sqsubseteq \mathfrak{U}$ and $C_\Theta \not\sqsubseteq S$.
- $R^o \subseteq C \times C_\Theta$ is a finite nonempty set of *output relations*.
- B is a finite nonempty set of *behaviors* of $C \sqsubseteq S$, $B \subseteq \text{Þ}F \sqsubseteq \mathfrak{U}$ and $B \sqsubseteq S$.
- Ω is a finite nonempty set of *constraints* of C and B, $\Omega \subseteq \text{Þ}V_E \cup \text{Þ}V_F \sqsubseteq \mathfrak{U}$ and $\Omega \sqsubseteq S$.
- Θ is the *environment* of S with a finite nonempty set of external components outside S, i.e., $\Theta = C_\Theta \subseteq \text{Þ}E \sqsubseteq \mathfrak{U}$. and $\Theta = C_\Theta \not\sqsubseteq S$.

A granule as an abstract open system $G(C, R^c, R^i, R^o, B, \Omega, \Theta)$ can be illustrated in Figure 1. Based on the above generic structure of the abstract system model of granules, a set of relational and compositional operations on granular systems, $OP = \{\bullet_r, \bullet_c\}$, will be rigorously defined in the next section.

Figure 1. The abstract model of a granule in granular computing



It is noteworthy that system behaviors *B* is the most broad set of system actions implemented or embodied on the given layout of the system, including any kind of system functions, interactions, and communications. This is the major difference that distinguishes an abstract system from other static mathematical structures such as a set, lattice, group, or abstract concept (Wang, 2008b). With the structural layout of a granule as given in Definition 14, its functional model or its set of implemented behaviors can be described in RTPA (Wang, 2002a, 2003b, 2006, 2007a, 2008d) as given below.

Definition 15. *The functional model of a granule in computing is equivalent to one of the 17 meta-processes, 𝔓, as defined in RTPA, i.e.:*

$$\mathfrak{P} = \{:=, �, \Rightarrow, \Leftarrow, \nLeftarrow, \succ, \prec, |\succ, |\prec, \underline{\underline{@}}, \triangleq, \uparrow, \downarrow, !, \otimes, \boxtimes, \S\} \quad (8)$$

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

Definition 16. *The software composing rules state that the RTPA process relation system, ℜ, encompasses 17 fundamental algebraic and relational operations elicited from basic computing needs, i.e.:*

$$\mathfrak{R} = \{\rightarrow, \curvearrowright, |, |\ldots|\ldots, R^{*}, R^{+}, R^{i}, \circlearrowleft, \rightarrowtail, ||, ∯, |||, », ↯, \hookrightarrow_{t}, \hookrightarrow_{e}, \hookrightarrow_{i}\} \quad (9)$$

where, the RTPA relational process operators stand for those of sequence, jump, branch, switch, while-loop, repeat-loop, for-loop, recursion, function call, parallel, concurrence, interleave, pipeline, interrupt, time-driven dispatch, event-driven dispatch, and interrupt-driven dispatch (Wang, 2002a, 2003b, 2006, 2007a, 2008d).

Definition 17. *In a larger scope, the functional model of a granule may be denoted as a cognitive granule, which possesses one of the 39 basic cognitive processes as identified in the Layered Reference Model of the Brain (LRMB).*

According to the LRMB model (Wang et al., 2006; Wang and Wang, 2006), the 39 cognitive processes can be categorized at the layers of *sensation, memory, perception, action, meta cognitive function,* and *higher cognitive function*, such as the memorization, comprehension, and learning processes.

Definition 18. *A granular system S_G is a composition of multiple granules in a system where all granules interact with each other for a common goal of system functionality.*

Properties of granular systems obey the generic properties of abstract systems, which will be described in the fourth section of the article.

MANIPULATION OF GRANULAR SYSTEMS BY SYSTEM ALGEBRA

With the formal model of abstract granules as given in Definition 14, granular systems can be modeled and implemented using a set of relational and compositional operations in system algebra as summarized in Definitions 12 and 13, respectively.

Algebraic Relations of Granules and Granular Systems

As summarized in Definition 12, relationships between two granular systems can be independent, overlapped, related, equivalent, being subsystem, and being supersystem. The relational operations of granular systems are static and comparative operations that do not change the systems involved. These six relational operations on granular systems are described below.

Definition 19. *Two granules G_1 and G_2 are independent, denoted by $\nleftrightarrow$, if both their cell sets and external relation sets are disjoint, i.e.:*

$$G_1(C_1, R_1^c, R_1^i, R_1^o, B_1, \Omega_1, \Theta_1) \nleftrightarrow G_2(C_2, R_2^c, R_2^i, R_2^o, B_2, \Omega_2, \Theta_2) \triangleq C_1 \cap C_2 = \varnothing \wedge R_1^i \cap R_2^i = \varnothing \wedge R_1^o \cap R_2^o = \varnothing \quad (10)$$

Definition 20. *Two granules G_1 and G_2 are overlapped, denoted by Π, if their cell sets are joint, i.e.:*

$$G_1(C_1, R_1^c, R_1^i, R_1^o, B_1, \Omega_1, \Theta_1) \prod G_2(C_2, R_2^c, R_2^i, R_2^o, B_2, \Omega_2, \Theta_2) \triangleq C_1 \cap C_2 \neq \varnothing \quad (11)$$

Definition 21. *Two granules G_1 and G_2 are related, denoted by $\leftrightarrow$, if either the sets of their input or output relations are overlapped, i.e.:*

$$G_1(C_1, R_1^c, R_1^i, R_1^o, B_1, \Omega_1, \Theta_1) \leftrightarrow G_2(C_2, R_2^c, R_2^i, R_2^o, B_2, \Omega_2, \Theta_2) \triangleq R_1^i \cap (R_2^0)^{-1} \neq \varnothing \vee R_2^i \cap (R_1^0)^{-1} \neq \varnothing \quad (12)$$

where $(R^0_1)^{-1}$ or $(R^0_2)^{-1}$ denotes an inverse relation, i.e., $\forall a \in C_1 \wedge b \in C_2$, $r(a, b) \in R^0_1 \Rightarrow r(b, a) \in (R^i_2) = (R^0_1)^{-1}$.

Definition 22. *Two granules G_1 and G_2 are equivalent, denoted by =, if all sets of cells, relations, behaviors, constraints, and environments are identical, i.e.:*

$$G_1(C_1, R_1^c, R_1^i, R_1^o, B_1, \Omega_1, \Theta_1) = G_2(C_2, R_2^c, R_2^i, R_2^o, B_2, \Omega_2, \Theta_2) \triangleq C_1 = C_2 \wedge R_1^c = R_2^c \wedge R_1^i = R_2^i \wedge R_1^o = R_2^o \wedge B_1 = B_2 \wedge \Omega_1 = \Omega_2 \wedge \Theta_1 = \Theta_2 \quad (13)$$

Definition 23. *A subgranule G' is a subsystem that is implicated in another granule G, denoted by $\sqsubseteq$, i.e.:*

$$G'(C', R^{c'}, R^{i'}, R^{o'}, B', \Omega', \Theta') \sqsubseteq G(C, R^c, R^i, R^o, B, \Omega, \Theta) \triangleq C' \subseteq C \wedge R^{c'} \subseteq R^c \wedge R^{i'} \subseteq R^i \wedge R^{o'} \subseteq R^o \wedge B' \subseteq B \wedge \Omega' \subseteq \Omega \wedge \Theta' = \Theta \quad (14)$$

The above definition indicates that the subgranule of a given granule is a coherent part of the parent granule with all integrated cells, internal/input/output relations, behaviors, and constraints. However, they share the same environment.

Definition 24. *A supergranule G is a supersystem that consists of one or more subgranules G', denoted by* $\sqsupseteq$*, i.e.:*

$$G(C, R^c, R^i, R^o, B, \Omega\ \Theta) \sqsupseteq G'(C', R^{c'}, R^{i'}, R^{o'}, B\ , \Omega', \Theta') \triangleq C' \subseteq C \wedge R^{c'} \subseteq R^c \wedge R^{i'} \subseteq R^i \wedge R^{o'} \subseteq R^o \wedge B' \subseteq B \wedge \Omega' \subseteq \Omega \wedge \Theta' = \Theta \quad (15)$$

Compositional Operations of Granular Systems

The compositional operations on granules in system algebra are dynamic and integrative operations that manipulate all granules involved in parallel. Compositional operations on granules provide a set of fundamental mathematical means to construct complex granular systems on the basis of simple ones or to derive new granular systems on the basis of exiting ones. The compositional operations on granules as summarized in Definition 13 can be classified into the operations of reproduction, composition, and integration.

(1) *Reproduction of Granular Systems*

Reproduction operations on granules are a category of composition operations in system algebra. The reproduction operations of granules encompass the algebraic manipulations of inheritance, tailoring, extension, substitution, and instantiation.

Definition 25. *The inheritance of a granule G_1 from the parent granule G, denoted by ⇒, is the creation of the new granule G_1 by reproducing G, and the establishment of new associations between them,(see Box 1), where || denotes that a granular inheritance may create new associations between G_1 and G in parallel via* $(R^o{}_1, R^i{}_2)$ and $(R^o{}_2, R^i{}_1)$.

Definition 26. *The multiple inheritance of a granule G from n parent granules G_1, G_2, …, G_n, denoted by ⇒, is an inheritance that creates the new granule G via a set of n conjoint granules and establishes new associations among them, see Box 2, where*

$$\mathop{R}_{i=1}^{n} R_i$$

is a calculus known as the big-R notation (Wang, 2002a, 2008e), which denotes a repetitive behavior or recurrent structure as defined in RTPA (Wang, 2007a, 2008d).

Box 1.

$$\begin{aligned}
&G(C, R^c, R^i, R^o, B, \Omega, \Theta) \Rightarrow G_1(C_1, R_1{}^c, R_1{}^i, R_1{}^o, B_1, \Omega_1, \Theta_1) \\
&\triangleq\ G_1(C_1, R_1{}^c, R_1{}^i, R_1{}^o, B_1, \Omega_1, \Theta_1 \mid C_1 = C, R_1{}^c = R^c, \\
&\qquad R^i{}_1 = R^i \cup (C \times C_1), R^o{}_1 = R^o \cup (C_1 \times C), \\
&\qquad B_1 = B, \Omega_1 = \Omega, \Theta_1 = \Theta) \\
&\parallel G(C, R^c, R^{i'}, R^{o'}, B, \Omega, \Theta' \mid R^{i'} = R^i \cup (C_1 \times C), \\
&\qquad R^{o'} = R^o \cup (C \times C_1), \Theta' = \Theta \cup C_1)
\end{aligned} \quad (16)$$

Box 2.

$$\overset{n}{\underset{i=1}{R}} G_i \Rightarrow G(C, R^c, R^i, R^o, B, \Omega, \Theta)$$
$$\triangleq G(C, R^c, R^i, R^o, B, \Omega, \Theta \mid C = \bigcup_{i=1}^{n} C_i, R^c = \bigcup_{i=1}^{n} R^c{}_i, R^i = \bigcup_{i=1}^{n} R^i{}_i \cup \{\overset{n}{\underset{i=1}{R}} (C_i \times C)\},$$
$$R^o = \bigcup_{i=1}^{n} R^o{}_i \cup \{\overset{n}{\underset{i=1}{R}} (C \times C_i)\}, B = \bigcup_{i=1}^{n} B_i, \Omega = \bigcup_{i=1}^{n} \Omega_i, \Theta = \bigcup_{i=1}^{n} \Theta_i)$$
$$\| \overset{n}{\underset{i=1}{R}} G_i(C_i, R_i^c, R_i^i, R_i^o, B_i, \Omega_i, \Theta'_i \mid R^{i'}{}_i = R^i{}_i \cup (C \times C_i),$$
$$R^{o'}{}_i = R^o{}_i \cup (C_i \times C), \Theta'_i = \Theta_i \cup C) \tag{17}$$

Definition 27. *The tailoring of a granule G_1 from the parent granule G, denoted by $\overset{-}{\Rightarrow}$, is a special granular inheritance that creates the new granule G_1 based on G with the removal of specific subsets of cells C ', behaviors B′, and constraints Ω′; and at the same time, it establishes new associations between them, see Box 3.*

Definition 28. *The extension of a granule G_1 from the parent granule G, denoted by $\overset{+}{\Rightarrow}$, is a special granular inheritance that creates the new granule G_1 based G with additional subsets of components C′, behaviors B′, and constraints Ω′; and at the same time, it establishes new associations between the two systems, see Box 4.*

Definition 29. *The substitute of a granule G_1 from the parent granule G, denoted by $\overset{\sim}{\Rightarrow}$, is a flexible granular inheritance that creates the new granule G_1 based on G with the new subsets of cells C'_{G1}, behaviors B'_{G1}, and constraint attributes Ω'_{G1} to replace the corresponding inherited ones C'_{G}, B'_{G}, and Ω'_{G} that share the same identifiers; and at the same time, it establishes new associations between the two concepts, see Box 5, where $C'_{G1} \subset C_1 \wedge C'_{G2} \subset C_2 \wedge \#C'_{G1} = \#C'_{G2}$; $B'_{G1} \subset B_1 \wedge B'_{G2} \subset B_2 \wedge \#B'_{G1} = \#B'_{G2}$; and $\Omega'_{G1} \subset \Omega_1 \wedge \Omega'_{G2} \subset \Omega_2 \wedge \#\Omega'_{G1} = \#\Omega'_{G2}$.*

The binary tailoring, extension, and substitution can also be extended to corresponding *n*-nary operations, similar to that of inheritance as given in Definitions 25.

(2) *Composition of Granular Systems*

Composition operations on granules in system algebra encompass a pair of composition and decomposition operations, where decomposition is defined based on the operation of granule difference.

Definition 30. *The composition of two granules G_1 and G_2, denoted by $\uplus$, results in a supergranule G that is formed by the conjunction of both sets of cells and environment $C_1 \cup C_2$ and $\Theta_1 \cup \Theta_2$, as well as incremental unions of all sets of relation $R^c = R_1^c \boxplus R_2^c$, $R^i = R_1^i \boxplus R_2^i$, $R^o = R_1^o \boxplus R_2^o$, behaviors $B = B_1 \boxplus B_2$, and constraints $\Omega = \Omega_1 \boxplus \Omega_2$, respectively, see Box 6, where the mathematical calculus of incremental union $\boxplus$ between sets of relations may be referred to (Wang, 2007a).*

It is noteworthy that granule compositions result in the *incremental relations* ΔR_{12},

Box 3.

$$
\begin{aligned}
&G(C, R^c, R^i, R^o, B, \Omega, \Theta) \stackrel{-}{\Rightarrow} G_1(C_1, R_1^{\ c}, R_1^{\ i}, R_1^{\ o}, B_1, \Omega_1, \Theta_1) \\
&\triangleq G_1(C_1, R_1^{\ c}, R_1^{\ i}, R_1^{\ o}, B_1, \Omega_1, \Theta_1 \mid C_1 = C \setminus C', R_1^{\ c} = R^c \setminus \{(C \times C')\}, \\
&\qquad R^i{}_1 = R^i \cup (C \times C_1), R^o{}_1 = R^o \cup (C_1 \times C), \\
&\qquad B_1 = B \setminus B', \Omega_1 = \Omega \setminus \Omega', \Theta_1 = \Theta) \\
&\ || \, G(C, R^c, R^{i'}, R^{o'}, B, \Omega, \Theta' \mid R^{i'} = R^i \cup (C_1 \times C), \\
&\qquad R^{o'} = R^o \cup (C \times C_1), \Theta' = \Theta \cup C_1)
\end{aligned} \tag{18}
$$

Box 4.

$$
\begin{aligned}
&G(C, R^c, R^i, R^o, B, \Omega, \Theta) \stackrel{+}{\Rightarrow} G_1(C_1, R_1^{\ c}, R_1^{\ i}, R_1^{\ o}, B_1, \Omega_1, \Theta_1) \\
&\triangleq G_1(C_1, R_1^{\ c}, R_1^{\ i}, R_1^{\ o}, B_1, \Omega_1, \Theta_1 \mid C_1 = C \cup C', R_1^{\ c} = R^c \cup (C \times C') \\
&\qquad R^i{}_1 = R^i \cup (C \times C_1), R^o{}_1 = R^o \cup (C_1 \times C), \\
&\qquad B_1 = B \cup B', \Omega_1 = \Omega \cup \Omega', \Theta_1 = \Theta) \\
&\ || \, G(C, R^c, R^{i'}, R^{o'}, B, \Omega, \Theta' \mid R^{i'} = R^i \cup (C_1 \times C), \\
&\qquad R^{o'} = R^o \cup (C \times C_1), \Theta' = \Theta \cup C_1)
\end{aligned} \tag{19}
$$

Box 5.

$$
\begin{aligned}
&G(C, R^c, R^i, R^o, B, \Omega, \Theta) \stackrel{\sim}{\Rightarrow} G_1(C_1, R_1^{\ c}, R_1^{\ i}, R_1^{\ o}, B_1, \Omega_1, \Theta_1) \\
&\triangleq G_1(C_1, R_1^{\ c}, R_1^{\ i}, R_1^{\ o}, B_1, \Omega_1, \Theta_1 \mid C_1 = (C \setminus C'_G) \cup C'_{G_1}, \\
&\qquad R_1^{\ c} = (R^c \setminus (C \times C'_G)) \cup (C \times C'_{G_1}), R^i{}_1 = R^i \cup (C \times C_1), \\
&\qquad R^o{}_1 = R^o \cup (C_1 \times C), B_1 = (B \setminus B'_G) \cup B'_{G_1}, \Omega_1 = (\Omega \setminus \Omega'_G) \cup \Omega'_{G_1}, \Theta_1 = \Theta) \\
&\ || \, G(C, R^c, R^{i'}, R^{o'}, B, \Omega, \Theta' \mid R^{i'} = R^i \cup (C_1 \times C), R^{o'} = R^o \cup (C \times C_1), \\
&\qquad \Theta' = \Theta \cup C_1)
\end{aligned} \tag{20}
$$

behaviors (functions) ΔB_{12}, and *constraints* $\Delta\Omega_{12}$, which are solely a property of the new supergranule G, but not belong to any of the independent subgranules, i.e.:

$$\Delta R_{12} \sqsubseteq S \wedge (\Delta R_{12} \not\sqsubseteq S_1 \wedge \Delta R_{12} \not\sqsubseteq S_2) \tag{22a}$$

$$\Delta B_{12} \sqsubseteq S \wedge (\Delta B_{12} \not\sqsubseteq S_1 \wedge \Delta B_{12} \not\sqsubseteq S_2) \tag{22b}$$

$$\Delta \Omega_{12} \sqsubseteq S \wedge (\Delta \Omega_{12} \not\sqsubseteq S_1 \wedge \Delta \Omega_{12} \not\sqsubseteq S_2) \tag{22c}$$

where $\sqsubseteq$ denote a membership relation of a given set in a system.

Box 6.

$$
\begin{aligned}
& G_1(C_1, R_1^{\,c}, R_1^{\,i}, R_1^{\,o}, B_1, \Omega_1, \Theta_1) \uplus G_2(C_2, R_2^{\,c}, R_2^{\,i}, R_2^{\,o}, B_2, \Omega_2, \Theta_2) \\
& \triangleq\ G(C, R^c, R^i, R^o, B, \Omega, \Theta) \mid C = C_1 \cup C_2, R^c = R_1^{\,c} \boxplus R_2^{\,c}, R^i = R_1^{\,i} \boxplus R_2^{\,i}, \\
& \quad R^o = R_1^{\,o} \boxplus R_2^{\,o}, B = B_1 \boxplus B_2, \Omega = \Omega_1 \boxplus \Omega_2, \Theta = \Theta_1 \cup \Theta_2) \\
& \ \| \mathop{R}_{i=1}^{2} G_i(C_i, R_i^c, R_i^{i'}, R_i^{o'}, B_i, \Omega_i, \Theta'_i \mid R^{i'}{}_i = R^i{}_i \cup (C \times C_i), \\
& \qquad R^{o'}{}_i = R^o{}_i \cup (C_i \times C), \Theta'_i = \Theta_i \cup C) \\
& =\ G(C, R^c, R^i, R^o, B, \Omega, \Theta) \mid C = C_1 \cup C_2, R^c = R_1^{\,c} \cup R_2^{\,c} \cup \Delta R_{12}^{\,c}, \\
& \quad R^i = R_1^{\,i} \cup R_2^{\,i} \cup \Delta R_{12}^{\,i}, R^o = R_1^{\,o} \cup R_2^{\,o} \cup \Delta R_{12}^{\,o}, \\
& \quad B = B_1 \cup B_2 \cup \Delta B_{12}, \Omega = \Omega_1 \cup \Omega_2 \cup \Delta\Omega_{12}, \Theta = \Theta_1 \cup \Theta_2) \\
& \ \| \mathop{R}_{i=1}^{2} G_i(C_i, R_i^c, R_i^{i'}, R_i^{o'}, B_i, \Omega_i, \Theta'_i \mid R^{i'}{}_i = R^i{}_i \cup (C \times C_i), \\
& \qquad R^{o'}{}_i = R^o{}_i \cup (C_i \times C), \Theta'_i = \Theta_i \cup C)
\end{aligned} \tag{21}
$$

The above discovery is known as the *system fusion principle* (Wang, 2008c), which reveal that the nature of granular system utilities can be rigorously explained as the newly generated relations ΔR_{12}, as well as behaviors ΔB_{12} and constraints $\Delta\Omega_{12}$, during the composition of two or more granules. The empirical awareness of this key system property has been intuitively or qualitatively described in the literature of system science (Ellis and Fred, 1962; Klir, 1992). However, the above system fusion principle provides the first mathematical explanation for the mechanism of system gains during granular system compositions.

Granule compositions as modeled in Definition 30 can be extended to *n*-nary compositions as follows.

Definition 31. *The composition of multiple granules, denoted by*

$$\biguplus_{i=1}^{n} G_i,$$

is an iterative integration of a pair of them, which cumulatively creates the new supergranules G, see Box 7.

According to Definitions 30 and 31, a granular system can be integrated from the bottom up by a series of compositions level-by-level in a system hierarchy.

Definition 32. *The difference between a granule G and a subgranule of it G_1, denoted by $\boxminus$, results in another subgranule G_2 that is formed by the difference of sets of cells and I/O relations (C_1, $R^i{}_1$, $R^o{}_1$), and the differences of the internal relations, behaviors, and constraints ($R^c{}_1$, B_1, Ω_1) with their incremental counterparts ($\Delta R^c{}_{12}$, ΔB_{12}, $\Delta\Omega_{12}$), see Box 8.*

A granular system decomposition is an inverse operation of granule composition that breaks up a given granule into two or more subgranules. Granule decomposition can be described based on the concept of granule difference. The latter is an inversed operation of the incremental union of sets as given in Eq. 21.

Definition 33. *The decomposition of a granule G, denoted by $\pitchfork$, is to break up G into two or more subgranules at the same level of the sys-*

Box 7.

$$G(C,R^c,R^i,R^o,B,\Omega,\Theta) \triangleq \biguplus_{i=1}^{n} G_i(C_i,R_i^c,R_i^i,R_i^o,B_i,\Omega_i,\Theta_i)$$
$$= G(C,R^c,R^i,R^o,B,\Omega,\Theta \mid C = \bigcup_{i=1}^{n} C_i, R^c = \boxplus_{i=1}^{n} R^c{}_i, R^i = \boxplus_{i=1}^{n} R^i{}_i,$$
$$R^o = \boxplus_{i=1}^{n} R^o{}_i, B = \boxplus_{i=1}^{n} B_i,\ \Omega = \boxplus_{i=1}^{n} \Omega_i, \Theta = \bigcup_{i=1}^{n} \Theta_i)$$
$$\parallel \mathop{R}_{i=1}^{n} G_i(C_i,R_i^c,R_i^{i'},R_i^{o'},B_i,\Omega_i,\Theta'_i \mid R^{i'}{}_i = R^i{}_i \cup (C \times C_i),$$
$$R^{o'}{}_i = R^o{}_i \cup (C_i \times C), \Theta'_i = \Theta_i \cup C) \quad (23)$$

Box 8.

$$G(C,R^c,R^i,R^o,B,\Omega,\Theta) \boxminus G_1(C_1,R_1^c,R_1^i,R_1^o,B_1,\Omega_1,\Theta_1)$$
$$\triangleq G_2(C_2,R_2^c,R_2^i,R_2^o,B_2,\Omega_2,\Theta_2 \mid C_2 = C \setminus C_1,$$
$$R_2^c = R^c \setminus (R_1^c \cup \Delta R_{12}^c), R_2^i = R^i \setminus R_1^i,$$
$$R_2^o = R^o \setminus R_1^o, B_2 = B \setminus (B_1 \cup \Delta B_{12}),$$
$$\Omega_2 = \Omega \setminus (\Omega_1 \cup \Delta\Omega_{12}), \Theta_2 = \Theta_1) \quad (24)$$

tem hierarchy lower than the current granule by a series of iterative difference operations, see Box 9.

As specified in Definition 33, the decomposition operation results in the removal of all the incremental internal relations $\Delta R_{ij}^c = C_i \times C_j$, $1 \le i, j \le n$ that are no longer belong to any of its subgranules.

(3) *Integration of Granular Systems*

Integration operations on granules are a category of composition operations in system algebra. The integration operations of granules encompass a pair of algebraic manipulations of granules known as aggregation and specification.

Definition 34. *The aggregation of a granule G from a set of n peer granules S_i, $1 \le i \le n$, denoted by $\Lleftarrow$, is an aggregation of G with the elicitation of interested subsets of cells C'_i, behaviors B'_i, and constraints Ω'_i; and at the same time, it establishes new associations among all aggregated granules, see Box 10.*

Granule aggregation is also known as granule *elicitation*. According to Definitions 30 and 34, the difference between granule composition and aggregation is that the former constructs a new granule by integrating a set of subgranules as a whole; while the latter constructs a new granule by eliciting interested subsets of cells, behaviors, and/or constraints from a set of individual and independent subgranules.

A granule specification is an inverse operation of granule aggregation. A granular system specification is usually operated by a series of refinements.

Box 9.

$$G(C, R^c, R^i, R^o, B, \Omega, \Theta) \pitchfork_{i=1}^{n} G_i(C_i, R_i^c, R_i^i, R_i^o, B_i, \Omega_i, \Theta_i)$$
$$\triangleq \mathop{R}_{i=1}^{n} \{\ G_i(C_i, R^c{}_i, R^{i'}{}_i, R^{o'}{}_i, B_i, \Omega_i, \Theta'_i \mid R^{i'}{}_i = R^i{}_i \cup (C \times C_i),$$
$$R^{o'}{}_i = R^o{}_i \cup (C_i \times C), \Theta'_i = \Theta_i \cup C)$$
$$\parallel G(C, R^c, R^i, R^o, B, \Omega, \Theta) \boxminus G_i(C_i, R_i^c, R_i^i, R_i^o, B_i, \Omega_i, \Theta_i)$$
$$\}$$
$$(25)$$

Box 10.

$$G(C, R^c, R^i, R^o, B, \Omega, \Theta) \Lleftarrow \mathop{R}_{i=1}^{n} G_i(C_i, R_i^c, R_i^i, R_i^o, B_i, \Omega_i, \Theta_i)$$
$$\triangleq G(C, R^c, R^i, R^o, B, \Omega, \Theta \mid C = \bigcup_{i=1}^{n} C'_i \subseteq C_i, R^c = C \times C,$$
$$R^i = \bigcup_{i=1}^{n} (R^i{}_i \cup (C_i \times C)), R^o = \bigcup_{i=1}^{n} (R^o{}_i \cup (C \times C_i)),$$
$$B = \bigcup_{i=1}^{n} B'_i \subseteq B_i,\ \Omega = \bigcup_{i=1}^{n} C'_i \subseteq C_i,\ \Theta \subseteq C_\Theta \cup \bigcup_{i=1}^{n} \Theta_i)$$
$$\parallel \mathop{R}_{i=1}^{n} G_i(C_i, R_i^c, R_i^{i'}, R_i^{o'}, B_i, \Omega_i, \Theta'_i \mid R^{i'}{}_i = R^i{}_i \cup (C \times C_i),$$
$$R^{o'}{}_i = R^o{}_i \cup (C_i \times C), \Theta'_i = \Theta_i \cup C)$$
$$(26)$$

Definition 35. *The specification of a granule G by a set of n refined granules G_i, $1 \le i \le n$, denoted by $\vdash$, is a specification of G with a total order of a series of refinements by increasingly specific and detailed cells C_i, behaviors B_i, and constraints Ω_{ij}; and at the same time, it establishes new associations among all refining systems, see Box 11.*

Granule specification is a refinement process where more specific and detailed cells, behaviors, and constraints are developed in a consistent and coherent top-down hierarchy. The major tasks of granule specifications are system architecture (component) and behavior specifications, which can be further refined by the *Component Logical Models* (CLMs) and *processes* as provided in RTPA (Wang, 2002a, 2007a, 2008d).

PROPERTIES OF GRANULAR SYSTEMS

Based on the abstract system models of granules and granular computing, the topological properties of granules and the granularity of granular systems can be rigorously analyzed in this section.

Box 11.

$$
\begin{aligned}
&(G_n \vdash ... \vdash G_2 \vdash G_1) \vdash G_0(C_0, R_0^{\ c}, R_0^{\ i}, R_0^{\ o}, B_0, \Omega_0, \Theta_0) \\
&\triangleq G_0(C_0, R_0^{\ c}, R_0^{\ i}, R_0^{\ o}, B_0, \Omega_0, \Theta_0 \mid C_0 = \mathop{R}_{i=1}^{n}(C_{i-1} \subset C_i), R_0^{\ c} = (C_0 \times C_0), \\
&\quad R_0^{\ i} = \bigcup_{i=1}^{n}(R^i{}_i \cup (C_i \times C_0)), R_0^{\ o} = \bigcup_{i=1}^{n}(R^o{}_i \cup (C_0 \times C_i)), \\
&\quad B_0 = \mathop{R}_{i=1}^{n}(B_{i-1} \subset B_i),\ \Omega_0 = \mathop{R}_{i=1}^{n}(B_{i-1} \subset B_i),\ \Theta_0 = \Theta_1 = \Theta_2 = ... = \Theta_n) \\
&\parallel \mathop{R}_{i=1}^{n} G_i(C_i, R_i^{\ c}, R_i^{\ i'}, R_i^{\ o'}, B_i, \Omega_i, \Theta'_i \mid R_i^{\ i'} = R_i^{\ i} \cup (C_0 \times C_i), \\
&\quad\quad R_i^{\ o'} = R_i^{\ o} \cup (C_i \times C_o), \Theta'_i = \Theta_i \cup C_0))
\end{aligned}
\tag{27}
$$

Topological Properties of Granular Systems

Theorem 1. *A granule G(C, R^c, R^i, R^o, B, Ω, Θ) on $\mathfrak{U}$ is an asymmetric and reflective system because its relations R^c, R^i, and R^o, are constrained by the following rules:*

(a) *Internally asymmetric*:
$\forall a, b \in C \wedge a \neq b \wedge r \in R^c, r(a,b) \not\Rightarrow r(b,a)$ (28)

(b) *Externally asymmetric*:
$\forall a \in C \wedge \forall x \in C_\Theta \wedge r \in R^i \vee r \in R^o, r(x,a) \not\Rightarrow r(a,x)$ (29)

(c) *Reflective*:
$\forall c \in C, r(c, c) \in R^c$ (30)

Corollary 1. *The maximum number of binary relations n_r in a granule G(C, R^c, R^i, R^o, B, Ω, Θ) is determined by the numbers of internal relations R^c as well as external relations R^i and R^o, i.e.:*

$$
\begin{aligned}
n_r(S) &= \#R^c + \#R^i + \#R^o = n_c^{\ 2} + 2(n_c \bullet n_{c_\Theta}) \\
&= n_c(n_c + 2n_{c_\Theta})
\end{aligned}
\tag{31}
$$

Corollary 2. *If all reflective self-relations are not considered among each cells in a granule, then the maximum number of binary relations n'_r is:*

$$
\begin{aligned}
n'_r(S) &= n_r(S) - n_c = n_c \bullet (n_c + 2n_{c_\Theta}) - n_c \\
&= n_c \bullet (n_c + 2n_{c_\Theta} - 1)
\end{aligned}
\tag{32}
$$

According to Corollaries 1 and 2, it is apparent that a simple granule system may result in a huge number of relations n_r and the exponential increases of complexity, when the number of cells possessed in it is considerably large. Therefore, system algebra is introduced to formally and efficiently manipulate the complex abstract and concrete granular systems.

Granularity of Granular Systems

The study on system magnitudes (Wang, 2007a) can be adopted to classify the size properties of granular systems and their relationship with other basic system attributes. This results in a set of measures on the sizes, magnitudes, and complexities of granular systems as described below.

Definition 36. *The size of a granule G, S_G, is the number of cells encompassed in the granule, i.e.:*

$$S_G = \#C = n_c \tag{33}$$

Definition 37. *The magnitude of a granule G, M_G, is the number of asymmetric binary relations among the n_c components of the granule including the reflexive relations, i.e.:*

$$M_G = \#R = n_r = \#(C \times C) = n_c^2 \tag{34}$$

If all self-reflective relations are ruled out in n_r, the pure number of binary relations M'_G in the given granular system is determined as follows:

$$M'_G = M_s - n_c = n_c^2 - n_c = n_c(n_c - 1) \tag{35}$$

Corollary 3. *The pure number of binary relations M'_G in a granule equals to exactly two times of the number of pairwise combinations among n_c, i.e.:*

$$M'_G = n_c(n_c - 1) = 2 \bullet \frac{n_c(n_c - 1)}{2} = 2 \bullet C_{n_c}^2 \tag{36}$$

where the factor 2 represents the asymmetric binary relation r among all cells in the granule, i.e., arb ≠ bra.

The magnitude of a granule determines its complexity. The complicities of a granular system can be classified based on if they are fully or partially connected. The former is the theoretical upper-bound complexity of a granule in which all cells are potentially interconnected with each other in all *n*-nary ways, $1 \le n \le n_c = \#C$. The latter is the more typical complexity of a granule where cells are only pairwisely connected.

Definition 38. *The complexity of a fully connected granular system C_{max} is a closure of all possible n-nary relations R*, $1 \le n \le n_c$, among all components of the given granule $n_c = \#C$, i.e.:*

$$C_{max} = R^* = 2\sum_{k=0}^{n_r} C_{n_r}^k \approx 2 \bullet 2^{n_r} = 2^{n_r+1} = 2^{n_c^2+1} = 2^{M_G+1} \tag{37}$$

where C_{max} is also called the maximum complexity of a granular system.

According to Definition 38, the closure of all possible *n*-nary relations *R** may easily result in an extremely huge degree of complexity for a granule system with a few cells. For example, when $n_c = 10$, $C_{max} = 2^{101}$. This indicates that a granular system would be too hard to be modeled and handled by conventional modeling techniques.

It is noteworthy that almost all functioning systems are partially connected, because a fully connected system may not represent or provide anything meaningful. Therefore, the complexity of partially connected systems can be simplified as follows on the basis of Definition 38.

Definition 39. The *complexity of a partially connected granular system C_r* is determined by the number of asymmetric binary relations M'_G of the given granule, i.e.:

$$C_r = M'_G = 2 \bullet C_{n_c}^2 = n_c(n_c - 1) \tag{38}$$

where C_r can be referred to be the relational complexity of a granular system.

It is recognized that the extent of granule magnitudes can be classified at seven levels known as the *empty, small, medium, large, giant, immense,* and *infinite* granule systems from the bottom up. A summary of the relationships between granule magnitudes, sizes, internal relations, and complexities can be described in the *granule magnitude model* as shown in Table 1.

Table 1 indicates that the complexity of a small granule system may easily be out of control of human cognitive manageability. This leads to the following principle.

Theorem 2. *The holism complexity of granular systems states that within the 7-level scale of the granule magnitude model, almost all granular systems are too complicated to be cognitively understood or mentally handled as a whole, except small granule systems or those that can be decomposed into small granule systems.*

According to Theorem 2, the basic principle for dealing with complicated granular systems is system decomposition or modularity, in which the complexity of a lower-level granule must be small enough to be cognitively manageable. Details of granular system decomposition methods have been provided in Definition 33.

REPRESENTATION OF GRANULAR SYSTEMS

The abstract system model of granules and granular computing, developed in preceding sections enable the formal representation of both abstract and concrete granular systems. A case study on a digital clock is presented in this section to demonstrate the representation of computing granules and granular systems, as well as the expressive power of the abstract system models of granules and system algebra.

Concrete Models of Granules in Granular Computing

Two concrete real-world granular systems in granular computing, the *Clock* and *Alarm*, are described below for illustrating how the generic abstract granule models and their algebraic operations may be applied in granule-based system design and modeling based on system algebra.

Example 1. *A digital clock, Clock, can be described as a granule, or a granular system, G_1, as follows:*

$$Clock \triangleq G_1(C_1, R^c{}_1, R^i{}_1, R^o{}_1, B_1, \Omega_1, \Theta_1) \tag{39}$$

where the configuration of the Clock granule is given in Figure 2.

Table 1. The magnitude model of abstract granular systems

Level	Category	Size of granule ($S_s = n_c$)	Magnitude of granule ($M_s = n_r = n_c^2$)	Relational complexity of granule ($C_r = n_c(n_c - 1)$)	Maximum complexity of granule ($C_{max} = 2^{n_c^2}$)
1	The empty granule (𝔒)	0	0	0	-
2	Small granule	[1, 10]	$[1, 10^2]$	[0, 90]	$[2, 2^{100}]$
3	Medium granule	$(10, 10^2]$	$(10^2, 10^4]$	$(90, 0.99 \bullet 10^4]$	$(2^{100}, 2^{10,000}]$
4	Large granule	$(10^2, 10^3]$	$(10^4, 10^6]$	$(0.99 \bullet 10^4, 0.999 \bullet 10^6]$	∞
5	Giant granule	$(10^3, 10^4]$	$(10^6, 10^8]$	$(0.999 \bullet 10^6, 0.9999 \bullet 10^8]$	∞
6	Immense granule	$(10^4, 10^5]$	$(10^8, 10^{10}]$	$(0.9999 \bullet 10^8, 0.99999 \bullet 10^{10}]$	∞
7	The infinite granule (𝔘)	∞	∞	∞	∞

Figure 2. The Clock granule

The Clock Granule $G_1(C_1, R^c{}_1, R^i{}_1, R^o{}_1, B_1, \Omega_1, \Theta_1)$

- The set of cells:
  C_1 = {*Processor, Keypad, LEDs, ClockPulse*}
- The set of internal relations:
  $R^c{}_1 \subseteq C_1 \times C_1$ = {*Input* (*Keypad, Processor*),
  Tick (*ClockPulse, Processor*),
  Output (*Processor, LEDs*)}
- The set of input relations:
  $R^i{}_1 \subseteq C_{\Theta 1} \times C_1$ = {*SetTime* (*User, Keypad*)}
- The set of output relations:
  $R^o{}_1 \subseteq C_1 \times C_{\Theta 1}$ = {*ShowTime* (*LEDs, User*)}
- The set of behaviors:
  B_1 = {*SetTime*, *ShownTime*, *Tick*}
- The set of constraints:
  Ω_1 = {*Time* = hh × mm × ss}
- The environment:
  Θ_1 = {*User*}

In Figure 2, the behaviors of the clock granule defined in B_1 can be further refined by a set of processes in RTPA.

Example 2. *The alarm subsystem for the digital clock, Alarm, can be described as another granule G_2 as follows:*

$$Alarm \triangleq G_2(C_2, R^c{}_2, R^i{}_2, R^o{}_2, B_2, \Omega_2, \Theta_2) \qquad (40)$$

where the configuration of the Alarm granule is given in Figure 3.

In Figure 3, the behaviors of the alarm granule defined in B_2 can be further refined by a set of processes in RTPA.

Granular System Composition in Granular Computing

The composition of two concrete granules in real-world system design for granular computing are described below, which illustrates how the generic abstract granule composition operation may be implemented in real-world granule-based system design and modeling.

Example 3. *According to Definition 30, the composition of the two granules G_1(Clock) and G_2(Alarm) as given in Examples 1 and 2 results in a new supergranule G(Alarm_Clock) as shown in Box 12, where the configuration of the Alarm_Clock granule is given in Figure 4.*

Note that the newly generated relations *Select*(*Clock, Alarm*), as well as the new behaviors *SelectClock* and *SelectAlarm* in the granule system *G*(*Alarm_Clock*) are the results of the incremental unions of systems as defined in Eq. 21. Therefore, they do not belong to either subgranule G_1(*Clock*) or G_2(*Alarm*) rather than purely the properties of the supergranule *G*(*Alarm_Clock*).

Figure 3. The Alarm granule

The Alarm Granule $G_2(C_2, R^c{}_2, R^i{}_2, R^o{}_2, B_2, \Omega_2, \Theta_2)$

- The set of cells:
  C_2 = {*Processor, Keypad, LEDs, Bell*}
- The set of internal relations:
  $R^c{}_2 \subseteq C_2 \times C_2$ = {*Input* (*Keypad, Processor*),
  AlarmCheck (*Time, Alarm*),
  AlarmRelease (*Keypad, Processor*),
  Output (*Processor, LEDs*),
  Ring (*Processor, Bell*)}
- The set of input relations:
  $R^i{}_2 \subseteq C_{\Theta 2} \times C_2$ = {*SetAlarm* (*User, Keypad*)}
- The set of output relations:
  $R^o{}_2 \subseteq C_2 \times C_{\Theta 2}$ = {*ShowAlarm* (*LEDs, User*)}
- The set of behaviors:
  B_2 = {*SetAlarm*, *ShownAlarm, CheckAlarm, Ring, AlarmRelease*}
- The set of constraints:
  Ω_2 = {*Alarm* = hh × mm}
- The environment:
  Θ_2 = {*User*}

Box 12.

$$
\begin{aligned}
& G_1(C_1, R_1{}^c, R_1{}^i, R_1{}^o, B_1, \Omega_1, \Theta_1) \uplus G_2(C_2, R_2{}^c, R_2{}^i, R_2{}^o, B_2, \Omega_2, \Theta_2) \\
\triangleq\ & G(C, R^c, R^i, R^o, B, \Omega, \Theta) \mid C = C_1 \cup C_2, R^c = R_1{}^c \cup R_2{}^c \cup \Delta R_{12}{}^c, \\
& \quad R^i = R_1{}^i \cup R_2{}^i \cup \Delta R_{12}{}^i, R^o = R_1{}^o \cup R_2{}^o \cup \Delta R_{12}{}^o, \\
& \quad B = B_1 \cup B_2 \cup \Delta B_{12}, \Omega = \Omega_1 \cup \Omega_2 \cup \Delta\Omega_{12}, \Theta = \Theta_1 \cup \Theta_2) \\
& \| \mathop{R}_{i=1}^{2} G_i(C_i, R_i^c, R_i^{i'}, R_i^{o'}, B_i, \Omega_i, \Theta'_i \mid R^{i'}{}_i = R^i{}_i \cup (C \times C_i), \\
& \quad\quad R^{o'}{}_i = R^o{}_i \cup (C_i \times C), \Theta'_i = \Theta_i \cup C)
\end{aligned}
\quad (41)
$$

Granular System Decomposition in Granular Computing

The decomposition of a concrete granule in real-world system design for granular computing is described below, which illustrates how the generic abstract granule decomposition operation may be implemented in real-world granule-based system design and modeling.

Example 4. *According to Definition 33, the decomposition of a supergranule G(AlarmClock) may result in two new subgranules G_1(Clock) and G_2(Alarm) as shown in Box 13, where the*

Figure 4. The Alarm_Clock granule: $G(C, R^c, R^i, R^o, B, \Omega, \Theta) \triangleq G_1 \uplus G_2$

The Composed Alarm_Clock Granule $G(C, R^c, R^i, R^o, B, \Omega, \Theta)$

- The set of cells:
  $C = C_1 \cup C_2$
  = {*Processor, Keypad, LEDs, ClockPulse*} ∪
  {*Processor, Keypad, LEDs, Bell*}
  = {*Processor, Keypad, LEDs, ClockPulse, Bell*}

- The set of internal relations:
  $R^c = R^c{}_1 \cup R^c{}_2 \cup \Delta R^c{}_{12}$
  = {*Input*(*Keypad, Processor*), *Tick*(*ClockPulse, Processor*),
  Output(*Processor, LEDs*)} ∪
  {*Input*(*Keypad, Processor*), *AlarmCheck*(*Time, Alarm*),
  AlarmRelease(*Keypad, Processor*), *Output*(*Processor, LEDs*),
  Ring(*Processor, Bell*)} ∪
  {*Select*(*Clock, Alarm*)} // $\Delta R^c{}_{12}$
  = {*Input*(*Keypad, Processor*), *Tick*(*ClockPulse, Processor*),
  AlarmCheck(*Time, Alarm*),
  AlarmRelease(*Keypad, Processor*),
  Output(*Processor, LEDs*), *Ring*(*Processor, Bell*),
  Select(*Clock, Alarm*)}

- The set of input relations:
  $R^i = R^i{}_1 \cup R^i{}_2$
  = {*SetTime*(*User, Keypad*), *SetAlarm*(*User, Keypad*)}

- The set of output relations:
  $R^o = R^o{}_1 \cup R^o{}_2$
  = {*ShowTime*(*LEDs, User*), *ShowAlarm*(*LEDs, User*)}

- The set of behaviors:
  $B = B_1 \cup B_2 \cup \Delta B_{12}$
  = {*SetTime*, *ShownTime*, *tick*} ∪
  {*SetAlarm*, *ShownAlarm, CheckAlarm, Ring, AlarmRelease*}
  {*SelectClock, SelectAlarm*} // ΔB_{12} ∪

- The set of constraints:
  $\Omega = \Omega_1 \cup \Omega_2$
  = {*Time* = hh × mm × ss, *Alarm* = hh × mm}

- The environment:
  $\Theta = \Theta_1 \cup \Theta_2$
  = {*User*}

Box 13.

$$
\begin{aligned}
& G(C, R^c, R^i, R^o, B, \Omega, \Theta) \bigcap_{i=1}^{2} G_i(C_i, R_i^c, R_i^i, R_i^o, B_i, \Omega_i, \Theta_i) \\
\triangleq\; & \mathop{R}_{i=1}^{2} \{\ G_i(C_i, R^c{}_i, R^{i'}{}_i, R^{o'}{}_i, B_i, \Omega_i, \Theta'_i \mid R^{i'}{}_i = R^i{}_i \cup (C \times C_i), \\
& \qquad R^{o'}{}_i = R^o{}_i \cup (C_i \times C), \Theta'_i = \Theta_i \cup C) \\
& \quad \| \ G(C, R^c, R^i, R^o, B, \Omega, \Theta) \boxminus G_i(C_i, R_i^c, R_i^i, R_i^o, B_i, \Omega_i, \Theta_i) \\
& \quad \}
\end{aligned}
\tag{41}
$$

configuration of the Clock granule, G_1(Clock), is given in Figure 5.

Granule G_2(*Alarm*) can be derived similarly as shown in Figure 5, which results in the second subgranule exactly as that given in Eq. 40 and Figure 3. Note that within the given supergranule *G*, because there is an overlap between the two subgranules G_1 and G_2, the difference operation for decomposition may only

Figure 5. The Clock granule as a result of decomposition

The Decomposed Clock Granule $G_1(C_1, R^c_1, R^i_1, R^o_1, B_1, \Omega_1, \Theta_1)$

- The set of cells:
  $C_1 = C \setminus C'_2$
  = {*Processor, Keypad, LEDs, ClockPulse, Bell*} \ {*Bell*}
  = {*Processor, Keypad, LEDs, ClockPulse*}

- The set of internal relations:
  $R^c_1 = R^c \setminus (R^{c'}_2 \cup \Delta R^c_{12})$
  = {*Input*(*Keypad, Processor*), *Tick*(*ClockPulse, Processor*),
  AlarmCheck(*Time, Alarm*), *AlarmRelease*(*Keypad, Processor*),
  Output(*Processor, LEDs*),*Ring*(*Processor, Bell*),
  Select(*Clock, Alarm*)
  } \
  {*AlarmCheck*(*Time, Alarm*),
  AlarmRelease(*Keypad, Processor*),
  Ring(*Processor, Bell*)
  } ∪
  {*Select*(*Clock, Alarm*)}
  = {*Input*(*Keypad, Processor*), *Tick*(*ClockPulse, Processor*),
  Output(*Processor, LEDs*)
  }

- The set of input relations:
  $R^i_1 = R^i \setminus R^i_2$
  = {*SetTime*(*User, Keypad*)}

- The set of output relations:
  $R^o_1 = R^o \setminus R^o_2$
  = {*ShowTime*(*LEDs, User*)}

- The set of behaviors:
  $B_1 = B \setminus \{B_2 \cup \Delta B_{12}\}$
  = {*SetTime*, *ShownTime*, *Tick, SetAlarm*,
  ShownAlarm, CheckAlarm, Ring, AlarmRelease,
  SelectClock, SelectAlarm
  } \
  {*SetAlarm*, *ShownAlarm, CheckAlarm, Ring, AlarmRelease*,
  SelectClock, SelectAlarm
  }
  = {*SetTime*, *ShownTime*, *tick*}

- The set of constraints:
  $\Omega_1 = \Omega \setminus \Omega_2$
  = {*Time* = hh × mm × ss}

- The environment:
  $\Theta_1 = \Theta \setminus \Theta'_2$
  = {*User*}

remove the disjoint subset C'_2, $R^{c'}{}_2$, and Θ'_2 from G. It is also noteworthy that the loss of ΔR_{12} = *Select* (*Clock, Alarm*) and ΔB_{12} = {*SelectClock, SelectAlarm*} during the operation of granular system decomposition.

CONCLUSION

A new approach to rigorously construct the theoretical foundations of granular computing has been introduced on the basis of the recent development in denotational mathematics known as system algebra. The abstract system model of granules has been developed. The rigorous treatment of granular systems in computing has been studied using system algebra. The properties of granular systems and system granularity have been formally analyzed. Case studies on concrete granules and real-world granule-based systems have been presented. A wide range of applications on the system algebra theory of granular computing have been identified in computing, software engineering, system engineering, cognitive informatics, and computational intelligence.

This paper has presented a paradigm that demonstrates the generality and expressive power of system algebra and the generic abstract systems. System algebra has been introduced as a set of relational and compositional operations for manipulating abstract systems and their composing rules. The former have been elicited as the algebraic operations of *independent, related, overlapped, equivalent, subsystem,* and *supersystem*. The latter have been identified as the algebraic operations of *inheritance, tailoring, extension, substitute, difference, composition, decomposition, aggregation,* and *specification*. The rigorous treatment of the granular computing paradigm by system algebra has established a solid foundation for granular computing and granule-based systems modeling.

ACKNOWLEDGMENT

The authors would like to acknowledge the Natural Science and Engineering Council of Canada (NSERC) for its partial support to this work. The authors would like to thank Dr. *W. Pedrycz*, Dr. *A. Skowron*, and anonymous reviewers for their valuable comments and suggestions.

REFERENCES

Ashby, W.R. (1958), Requisite Variety and Implications for Control of Complex Systems, *Cybernetica*, (1), 83-99.

Ashby, W.R. (1962), Principles of the Self-Organizing System, in von Foerster H. and Zopf G. eds, *Principles of Self-Organization*, Pergamon, Oxford, 255-278.

Bargiela, A. and W. Pedrycz (2002), Granular Computing: an Introduction, Kluwer Academic Publishers, Boston.

Eigen, M. and P. Schuster (1979), *The Hypercycle: A Principle of Natural Self-Organization*, Springer, Berlin.

Ellis, D. O. and J. L. Fred (1962), *Systems Philosophy*, Prentice-Hall.

Ford, J. (1986), Chaos: Solving the Unsolvable, Predicting the Unpredictable, in *Chaotic Dynamics and Fractals*, Academic Press.

Haken, H. (1977), *Synergetics*, Springer-Verlag, NY.

Heylighen, F. (1989), Self-Organization, Emergence and the Architecture of Complexity, in *Proc. 1st European Conf. on System Science* (AFCET), Paris, 23-32.

Klir, G.J. (1992), *Facets of Systems Science,* Plenum, New York.

Lin, T.Y. (1998), Granular Computing on Binary Relations (I): Data Mining and Neighborhood Systems, *Proc. Rough Sets in Knowledge Discovery*, Physica-Verlag, Heidelberg, 107-120.

Lin, T.Y., Yao, Y.Y. and Zadeh, L.A. eds. (2002), Data Mining, Rough Sets and Granular Computing, Physica-Verlag, Heidelberg, 2002.

Pawlak, Z. (1987), Rough Logic, *Bulletin of the Polish Academy of Science, Technical Science*, (5-6), 253-258.

Pawlak, Z. (1998), Granularity of Knowledge, Indiscernibility and Rough Sets, *Proc. 1998 IEEE International Conference on Fuzzy Systems*, 106-110.

Pedrycz, W. ed. (2001), *Granular Computing: An Emerging Paradigm*, Physica-Verlag, Heidelberg, 2001.

Polkowski, L. (2008), On Foundations and Applications of the Paradigm of Granular Rough Computing, *The International Journal of Cognitive Informatics* (IJCINI), IGI Publishing, USA, 2(2), April, 80-94.

Polkowski, L. and Skowron, A. (1998), Towards Adaptive Calculus of Granules, *Proc. 1998 IEEE International Conference on Fuzzy Systems*, 111-116.

Prigogine, I. and Nicolis, G. (1972), Thermodynamics of Evolution, *Physics Today*, (25), 23-28.

Rapoport, A., (1962), Mathematical Aspects of General Systems Theory, *General Systems Yearbook*, (11), 3-11.

Skarda, C.A. and W.J. Freeman (1987), How Brains Make Chaos into Order, *Behavioral and Brain Sciences*, (10).

Skowron, A. and Stepaniuk, J. (2001), Information Granules: Towards Foundations of Granular Computing, *International Journal of Intelligent Systems*, (16), 57-85.

von Bertalanffy, L. (1952), *Problems of Life: An Evolution of Modern Biological and Scientific Thought*, C.A. Watts, London.

Wang, Y. (2002a), The Real-Time Process Algebra (RTPA), *Annals of Software Engineering*, Springer, (14), USA, 235-274.

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

Wang, Y. (2003a), On Cognitive Informatics, *Brain and Mind: A Transdisciplinary Journal of Neuroscience and Neurophilosophy,* 4(2), 151-167.

Wang, Y. (2003b), Using Process Algebra to Describe Human and Software System Behaviors, *Brain and Mind: A Transdisciplinary Journal of Neuroscience and Neurophilosophy*, 4(2), 199–213.

Wang, Y. (2006), On the Informatics Laws and Deductive Semantics of Software, *IEEE Transactions on Systems, Man, and Cybernetics (C),* 36(2), March, 161-171.

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

Wang, Y. (2007b), The Theoretical Framework of Cognitive Informatics, *International Journal of Cognitive Informatics and Natural Intelligence*, IGI Publishing, USA, 1(1), Jan., 1-27.

Wang, Y. (2007c), Keynote: On Theoretical Foundations of Software Engineering and Denotational Mathematics, *Proc. 5th Asian Workshop on Foundations of Software*, Beihang Univ. Press, Xiamen, China, 99-102.

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

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

Wang, Y. (2008c), On System Algebra: A Denotational Mathematical Structure for Abstract Systems Modeling, *International Journal of Cognitive Informatics and Natural Intelligence*, IGI Publishing, USA, 2(2), April, 20-43.

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

Wang, Y. (2008e), On the Big-R Notation for Describing Iterative and Recursive Behaviors, *International Journal of Cognitive Informatics and Natural Intelligence*, IGI Publishing, USA, 2(1), Jan., 17-23.

Wang, Y. and Y. Wang (2006), On Cognitive Informatics Models of the Brain, *IEEE Trans. Systems, Man, and Cybernetics (C),* 36(2), March, 203-207.

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

Yao, Y.Y. (2001), Information Granulation and Rough Set Approximation, *International Journal of Intelligent Systems*, 16(1), 87-104.

Yao, Y.Y. (2004a) A Partition Model of Granular Computing, *Transactions on Rough Sets*, *1*, Springer, LNCS 3135, 232-253.

Yao, Y.Y. (2004b) Granular Computing, *Computer Science,* Beijing, China, (31), 1-5.

Yao, Y.Y. (2005) Perspectives of Granular Computing, *Proc. 2005 IEEE International Conference on Granular Computing*, (1), 85-90.

Yao, Y.Y. and Zhou, B. (2008), A Logic Approach to Granular Computing, *International Journal of Cognitive Informatics and Natural Intelligence*, IGI Publishing, USA, 2(2), April, 63-79.

Zadeh, L.A. (1965), Fuzzy Sets and Systems, in J. Fox ed., *Systems Theory*, Polytechnic Press, Brooklyn NY, 29-37.

Zadeh, L.A. (1979), Fuzzy Sets and Information Granularity, in M.M. Gupta, R. Ragade, and R. Yager eds., *Advances in Fuzzy Set Theory and Applications*, North-Holland, Amsterdam, 3-18.

Zadeh, L.A. (1997) Towards a Theory of Fuzzy Information Granulation and its Centrality in Human Reasoning and Fuzzy Logic, *Fuzzy Sets and Systems*, (19), 111-127.

Zadeh, L.A. (1998) Some Reflections on Soft Computing, Granular Computing and Their Roles in the Conception, Design and Utilization of Information/Intelligent Systems, *Soft Computing*, (2), 23-25.

Zadeh, L.A. (2003), Some Reflections on Information Granulation and its Centrality in Granular Computing, Computing with Words. The Computational Theory of Perceptions and Precisiated Natural Language. *Proc. Data Mining, Rough Sets and Granular Computing,* Heideberg, 1-19.

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 Dept. 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 International Journal of Cognitive Informatics and Natural Intelligence *(*IJCINI*), founding editor-in-chief of* International Journal of Software Science and Computational Intelligence *(*IJSSCI*), associate editor of IEEE TSMC(A), and editor-in-chief of CRC Book Series in Software Engineering. He has published over 300 peer reviewed journal and conference papers and 11 books in cognitive informatics, software engineering, and computational intelligence. He has won dozens of research achievement, best paper, and teaching awards in the last 28 years.*

Lotfi A. Zadeh is professor and director of Berkeley Initiative in Soft Computing (BISC) in Department of Electrical Engineering and Computer science at University of California, Berkeley. He is the founder of fuzzy sets, fuzzy logic, and soft computing. He has initiated a series of fundamental AI and system theories and technologies.

Yiyu Yao received his BEng from Xi'an Jiaotong University, People's Republic of China in 1983, and his MSc and PhD from the University of Regina, Canada in 1988 and 1991, respectively. He was an assistant professor and an associate professor at Lakehead University, Canada from 1992 to 1998. He joined the Department of Computer Science at the University of Regina in 1998, where he is currently a professor of computer science. His research interests include data mining, rough sets, Web intelligence, granular computing, cognitive informatics, machine learning and information retrieval.