

Towards the abstract system theory of system science for cognitive and intelligent systems

Yingxu Wang¹

Received: 19 February 2015 / Accepted: 4 March 2015 / Published online: 23 June 2015
© The Author(s) 2015

Abstract Basic studies in system science explore the theories, principles, and properties of abstract and concrete systems as well as their applications in system engineering. Systems are the most complicated entities and phenomena in abstract, physical, information, cognitive, brain, and social worlds across a wide range of science and engineering disciplines. The mathematical model of a general system is embodied as a hyperstructure of the abstract system. The theoretical framework of system science is formally described by a set of algebraic operations on abstract systems known as system algebra. A set of abstract structures, properties, behaviors, and principles is rigorously formalized in contemporary system theories. Applications of the formal theories of system science in system engineering, intelligent engineering, cognitive informatics, cognitive robotics, software engineering, cognitive linguistics, and cognitive computing are demonstrated, which reveals how system structural and behavioral complexities may be efficiently reduced in system representation, modeling, analysis, synthesis, inference, and implementation.

Keywords System science · System theories · Abstract systems · Formal systems · Mathematical models · System algebra · Formal system theories · Denotational mathematics · Cognitive informatics · Cognitive robotics · System philosophy · Engineering applications · System engineering

✉ Yingxu Wang
yingxu@ucalgary.ca

¹ Laboratory for Computational Intelligence and Software Science, International Institute of Cognitive Informatics and Cognitive Computing (ICIC), Department of Electrical and Computer Engineering, Schulich School of Engineering and Hotchkiss Brain Institute, University of Calgary, 2500 University Drive NW, Calgary, AB T2N 1N4, Canada

Introduction

The primitive aims of science are to explore and denote knowledge about the structures and functions of the nature [1, 10, 25, 30]. *System science* is a discipline that studies the structures, mechanisms, behaviors, principles, properties, theories, and formal models of abstract systems and their applications in concrete systems in engineering and societies [2, 5, 6, 12, 14, 16, 19, 20, 29, 35, 42, 45, 63, 86]. Systems are the most complicated entities and phenomena in abstract, physical, information, cognitive, brain, and social worlds across a wide range of science and engineering disciplines. Systems are widely needed because the physical and/or cognitive power of an individual component or a person is always insufficient to carry out a work or solving a problem. The system philosophy is an important and the most general scientific philosophy that intends to treat everything as a system where it perceives that a system always belongs to other supersystem(s) and contains more subsystems.

The notion of systems can be traced back to the seventeenth century when R. Descartes (1596–1650) noticed the interrelationships among scientific disciplines as a system. The general system concept was proposed by Ludwig von Bertalanffy in the 1920s [12, 36]. The theories of system science have evolved from classic theories [2, 3, 5, 22, 29] to contemporary theories in the middle of the twentieth century such as Prigogine's *dissipative structure* theory [28], Haken's *synergetics* [15], and Eigen's *hypercycle* theory [11]. Since late of the last century, there are proposals of *complex systems* theories [17, 19, 20, 32, 42, 45, 92, 94], *fuzzy system* theories [19, 24, 26, 71, 72, 74, 75, 91–94], *chaos system* theories [13, 33], and *intelligent system* theories [4, 8, 38–40, 46, 47, 51, 52, 56, 59, 61, 65, 73, 83, 84, 90]. With regard to the metrics of system complexity and complex systems, a novel type of *long life-span systems* is identified [63], which reveals

system complexities in the time domain in supplement to the conventional focus on the size-oriented complexities and magnitudes of systems.

Fundamental studies on system science explores if there is a theory of the general system and/or a general theory of systems [5, 14, 29, 36, 42, 45, 63, 77]. This paper reveals that both these wonders are true. There exist not only the mathematical model of a general system, but also a framework of general theories for the abstract structures, properties, behaviors, and operations across all forms of systems. The former are known as the mathematical model of *abstract systems* [42, 45, 86]; while the latter are denoted by *system algebra* [45] and a set of general system properties and principles [42]. The general system and the system of systems can be formally modeled by an abstract system theory, which leads to a rigorous theoretical framework of system science for dealing with the general properties shared by all concrete systems.

This paper presents the abstract system theory for contemporary complex systems such as cognitive systems and intelligent systems. In the remainder of this paper, complex systems are formally treated as a generic mathematic structure of an abstract system in “The abstract system theory of formal systems”. A denotational mathematics of system algebra is adopted in “System algebra for formal system manipulations” to rigorously manipulate the structures and behaviors of abstract systems. Fundamental properties of abstract systems are modeled and analyzed in “Formal principles and properties of complex systems”, which rigorously explain the nature of real-world concrete systems. Applications of the abstract system theory and system algebra are demonstrated in “Paradigms of complex cognitive and intelligent systems” on complex brain and cognitive systems.

The abstract system theory of formal systems

Because of the extremely frequent and diverse usability as well as intricate complexity, systems and their properties attract a wide range of interests and intensive studies. This section presents the abstract system theory for the formal structures and properties of abstract systems. It reveals that real-world systems may be rigorously treated as a generic mathematical structure known as the hyperstructure beyond conventional mathematical entities. On the basis of this view, the concept of abstract systems, the denotational mathematical model of abstract systems, and the classification of concrete systems according to the formal system model are introduced.

The essences of abstract systems

The conceptual model of a system is a collection of a set of coherent and interactive entities that has a stable structure and

functions, as well as a clear boundary with the external environment. The natural and abstract worlds as typical systems can be perceived as an enclosure of entities and relations, respectively [20, 29, 42, 44, 45, 91]. Therefore, the discourse of universal systems can be defined as follows.

Definition 1 Let \mathcal{C} be a finite or infinite nonempty set of *components*, \mathcal{B} a finite nonempty set of *behaviors*, and \mathfrak{R} a finite nonempty set of *relations* where $\mathfrak{R} = \mathcal{C} \times \mathcal{C} | \mathcal{B} \times \mathcal{B} | \mathcal{B} \times \mathcal{C}$. The *discourse of the universal system*, \mathcal{U} , is denoted as a triple, i.e.:

$$\begin{aligned} \mathcal{U} &\hat{=} (\mathcal{C}, \mathcal{B}, \mathfrak{R}) \\ \text{where } \mathfrak{R} &= \{\mathfrak{R}_c, \mathfrak{R}_b, \mathfrak{R}_f\} \\ &= \mathfrak{R}_c : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C} \\ &\quad | \mathfrak{R}_b : \mathcal{B} \times \mathcal{B} \rightarrow \mathcal{B} \\ &\quad | \mathfrak{R}_f : \mathcal{B} \times \mathcal{C} \rightarrow \mathcal{B}, \end{aligned} \quad (1)$$

where \mathfrak{R}_c , \mathfrak{R}_b , and \mathfrak{R}_f is called the *structural*, *behavioral*, and *functional relations* in \mathcal{U} , and $|$ demotes alternative relations.

Any system may be formally modeled in the discourse of universal systems \mathcal{U} . A category of simple systems in \mathcal{U} are formally described in this subsection known as the closed systems. However, the general system model in \mathcal{U} for open systems will be derived in “The mathematical model of abstract systems”.

Definition 2 An *abstract closed system* $\widehat{\mathcal{S}}$ is a 5-tuple in the system discourse \mathcal{U} , i.e.:

$$\begin{aligned} \widehat{\mathcal{S}} &\hat{=} (C, B, R^c, R^b, R^f), C \subset \mathcal{P}\mathcal{C}, B \subset \mathcal{P}\mathcal{B}, \\ R &\subset \mathcal{P}\mathfrak{R}, \widehat{\mathcal{S}} \sqsubset \mathcal{U}, \end{aligned} \quad (2)$$

where C is a finite set of *components*, $C \subset \mathcal{P}\mathcal{C} \sqsubset \mathcal{U}$ where \mathcal{P} denotes a power set and \sqsubset denotes that a set is a substructure of an entire hyperstructure of a system; B a finite set of *behaviors* (or functions), $B \subset \mathcal{P}\mathcal{B} \sqsubset \mathcal{U}$; R^c a finite set of component relations, $R^c \subset \mathcal{P}\mathfrak{R}_c \sqsubset \mathcal{U}$; R^b a finite set of behavioral relations, $R^b \subset \mathcal{P}\mathfrak{R}_b \sqsubset \mathcal{U}$; and R^f a finite set of functional relations, $R^f \subset \mathcal{P}\mathfrak{R}_f \sqsubset \mathcal{U}$. All relations $R = \{R^c, R^b, R^f\} \subset \mathcal{P}\mathfrak{R} \sqsubset \mathcal{U}$ are functions as follows:

$$\begin{cases} R^c \hat{=} f_c : C \times C \rightarrow C \\ R^b \hat{=} f_b : B \times B \rightarrow B \\ R^f \hat{=} f_f : B \times C \rightarrow B \end{cases} \quad (3)$$

Example 1 According to Definition 2, the *primitive closed system* $\widehat{\mathcal{S}}^0$ as the most elemental system with only a single component and a single behavior can be denoted as follows:

$$\begin{aligned} \widehat{S}^0 &\triangleq (C_0, B_0, R_0^c, R_0^b, R_0^f), |C_0| = |B_0| = |R_0^c| = |R_0^b| \\ &= |R_0^f| = 1 \end{aligned} \tag{4}$$

Similarly, the *empty* system Φ and the *universal* system Ω in \mathcal{U} can be denoted as $\Phi = \widehat{S}^\Phi = (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$ and $\Omega = \widehat{S}^\Omega = (\infty, \infty, \infty, \infty, \infty)$, respectively.

The mathematical model of abstract systems

Although there are a wide variety of concrete systems in both the natural and symbolic worlds, there is a unified model of abstract systems that constitutes the most common properties of real-world systems. In the discourse of universal systems \mathcal{U} , the mathematical model of the closed abstract system as given in Definition 2 can be extended to that of open systems by considering additional attributes such as the input and output relations to the system environment.

Definition 3 The *abstract open system* S in the discourse of systems \mathcal{U} is a 7-tuple, i.e.:

$$S \triangleq (C, B, R^c, R^b, R^f, R^i, R^o), \tag{5}$$

where

- C is a finite set of components of system S , $C \subset \mathcal{P} \mathcal{C} \subset \mathcal{U}$ where \mathcal{P} denotes a power set.
- B is a finite set of behaviors (or functions), $B \subset \mathcal{P} \mathcal{B} \subset \mathcal{U}$.
- $R^c = C \times C$ is a finite set of component relations, $R^c \subset \mathcal{P} \mathcal{R}_c \subset \mathcal{U}$.
- $R^b = B \times B$ is a finite set of behavioral relations, $R^b \subset \mathcal{P} \mathcal{R}_b \subset \mathcal{U}$.
- $R^f = B \times C$ is a finite set of functional relations, $R^f \subset \mathcal{P} \mathcal{R}_f \subset \mathcal{U}$.
- $R^i = \Theta \times S$ is a finite set of input relations, $R^i \subset \mathcal{P} \mathcal{R} \subset \mathcal{U}$, where Θ is a set of external systems, $\Theta = \sqcup_{i=1}^n S_i' \mathcal{U} \wedge S_i' \not\subseteq S$.
- $R^o = S \times \Theta$ is a finite set of output relations, $R^o \subset \mathcal{P} \mathcal{R} \subset \mathcal{U}$.

The formal model of abstract systems as described in Definition 3 does not only elicit the generic model of widely various real-world systems, but also represent the most common attributes and properties of arbitrary systems. The structure of the formal abstract system model $S = (C, B, R^c, R^b, R^f, R^i, R^o)$ can be illustrated in Fig. 1 where C, B , and \mathcal{R} , $\mathcal{R} = \{R^c, R^b, R^f, R^i, R^o\}$, denote the components, behaviors, as well as its structural/behavioral/functional/input/output relations, respectively.

Definition 4 The *environment* Θ_{S^k} of a system S^k at the k th layer of a system hierarchy encompasses its parent system at a higher layer S^{k+1} and n peer systems at the same layer $(S_1^k, S_2^k, \dots, S_n^k)$, i.e.:

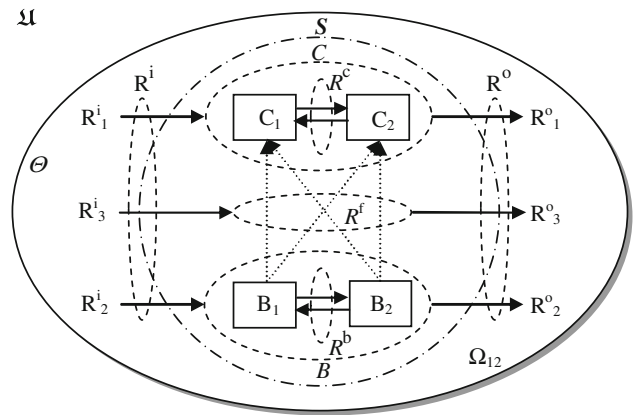


Fig. 1 The hyperstructural model of abstract systems

$$\Theta_{S^k} = S^{k+1} \sqcup_{i=1}^n S_i^{k'}, S \subset \mathcal{U} \wedge S_i^{k'} \not\subseteq S, \tag{6}$$

where the interactions between the system and its environment is via the sets of input relations R_k^i and output relations R_k^o .

It is noteworthy that, given an arbitrary system, the user is a typical parent system or a default peer system to the given system in the environment.

Example 2 A concrete clock system, S_1 (clock), can be formally modeled according to Definition 3 as follows:

$$\begin{aligned} S(\text{clock}) &\triangleq S(C, B, R^c, R^b, R^f, R^i, R^o) \\ &= \begin{cases} C = \{c_1, c_2, c_3, c_4\} = \{\text{Processor, Keypad, LED, Pulse}\} \\ B = \{b_1, b_2, b_3\} = \{\text{SetTime, Tick, ShownTime}\} \\ R^c \subseteq C \times C \\ \quad = \{(\text{Keypad}|\text{SM} \parallel \text{Pulse}|\text{SM}) \mid > \text{Processor}|\text{SM} \mid < \text{LED}|\text{SM}\} \\ R^b \subseteq B \times B \\ \quad = \{\text{SetTime}|\text{PM} \rightarrow \overset{T}{R}_{\text{Shutdown}|\text{B}=\text{F}}(\text{Tick}|\text{PM} \parallel \text{ShownTime}|\text{PM})\} \\ R^f \subseteq B \times C = \{\text{SetTime}|\text{SM}(\text{Keypad, Processor}), \\ \quad \text{Tick}|\text{SM}(\text{Pulse, Processor}), \text{ShownTime}|\text{SM}(\text{Processor, LEDs})\} \\ R^i \subseteq \Theta \times C = \{\text{SetTime}(\text{User, Keypad})\} \\ R^o \subseteq C \times \Theta = \{\text{ShowTime}(\text{LEDs, User})\} \end{cases} \end{aligned} \tag{7}$$

In Eq. 7, the static components of the clock system specified in C are further refined by the component relations R^c , which are represented by a set of structure models (SMs) and their interactions in Real-Time Process Algebra (RTPA) [37, 47, 48]. The dynamic behaviors of the clock system specified in B are further refined by the behavioral relations R^b and the functional relations R^f . R^b is embodied by a set of interactive relations among the process models (PMs) in RTPA. R^f is embodied by a set of cross-relations between PMs and SMs. The RTPA models for system behaviors can be implemented in any programming language in order to realize the expected functions and behaviors of the system on the basis of its structural models.

The abstract system model as given in Definition 3 represents a generic *open* system where interactions between the system and its *environment* Θ are modeled by the sets of input and output relations. Compare Definitions 3 and 2, it is obvious that the formal model of a closed systems is a special case of that of the open systems.

Lemma 1 *The relationship between an open and a closed system in \mathfrak{A} states that the open system is an extension of the closed system, and the closed system is a special case of the open system, i.e.:*

$$\begin{cases} S = \widehat{S} \sqcup (R^i, R^o) \\ \widehat{S} = S \setminus (R^i, R^o) \end{cases} \quad (8)$$

Proof Lemma 1 can be directly proved according to Definitions 2 and 3 as follows:

$$\begin{aligned} S &= \widehat{S} \sqcup (R^i, R^o) = (C, B, R^c, R^b, R^f) \sqcup (R^i, R^o) \\ &= (C, B, R^c, R^b, R^f, R^i, R^o) \\ \widehat{S} &= S \setminus (R^i, R^o) = (C, B, R^c, R^b, R^f, R^i, R^o) \setminus (R^i, R^o) \\ &= (C, B, R^c, R^b, R^f, \emptyset, \emptyset) \\ &= (C, B, R^c, R^b, R^f) \end{aligned} \quad (9)$$

□

Therefore, all abstract systems will be denoted by the unified mathematical model as given in Definition 3 unless there is a specific need to distinguish a close system.

Theorem 1 *The generality of systems states that a system S is a recursively embedded topological structure in \mathfrak{A} where each k th layer of it, S^k , in the system hierarchy can be represented or refined by its next layer, S^{k-1} , with the same structure in the generic form of Definition 3, i.e.:*

$$\begin{aligned} S &\triangleq \bigwedge_{k=1}^n S^k(S^{k-1}), S^0 = (C_0, B_0, R_0^c, R_0^b, R_0^f, R_0^i, R_0^o) \\ &= S^n(S^{n-1}(\dots(S^1(S^0)))) \end{aligned} \quad (10)$$

where S^0 is a known primitive system, and the big- R notation, $\bigwedge_{i=1}^n S_i$, denotes a repetitive behavior or a recurrent structure [43,48].

Proof The deductive structure of a general system S in Theorem 1 can be proved inductively.

Given $S^0 = (C^0, B^0, \mathcal{R}^0) = (C_0, B_0, R_0^c, R_0^b, R_0^f, R_0^i, R_0^o)$ as a primitive system where all its attributes are known and concrete, i.e., $|C_0| = |B_0| = |R_0^c| = |R_0^b| = |R_0^f| = |R_0^i| = |R_0^o| = 1$, then each of the higher layer systems can be realized based on its immediate subsystems as follows:

$$\begin{aligned} S^1 &= \bigwedge_{i_1=1}^{k_1} S_{i_1}^0 = \bigwedge_{i_1=1}^{k_1} (C_{i_1}^0, B_{i_1}^0, \mathcal{R}_{i_1}^0) \\ S^2 &= \bigwedge_{i_2=1}^{k_2} S_{i_2}^1 = \bigwedge_{i_2=1}^{k_2} \bigwedge_{i_1=1}^{k_1} S_{i_2 i_1}^0 = \bigwedge_{i_2=1}^{k_2} \bigwedge_{i_1=1}^{k_1} (C_{i_2 i_1}^0, B_{i_2 i_1}^0, \mathcal{R}_{i_2 i_1}^0) \\ &\dots \\ S^n &= \bigwedge_{i_n=1}^{k_n} \dots \bigwedge_{i_2=1}^{k_2} \bigwedge_{i_1=1}^{k_1} S_{i_n \dots i_2 i_1}^0 \\ &= \bigwedge_{i_n=1}^{k_n} \dots \bigwedge_{i_2=1}^{k_2} \bigwedge_{i_1=1}^{k_1} (C_{i_n \dots i_2 i_1}^0, B_{i_n \dots i_2 i_1}^0, \mathcal{R}_{i_n \dots i_2 i_1}^0) \\ &= S^n(S^{n-1}(\dots(S^1(S^0)))) = S \end{aligned} \quad (11)$$

□

According to Theorem 1, the following properties of abstract systems can be derived.

Corollary 1 *The general topological structure of abstract systems in \mathfrak{A} is an embedded hierarchical structure with recursive and embedded relations between any two adjacent layers of systems.*

Corollary 2 *The abstraction principle of systems states that any system S in \mathfrak{A} can be inductively integrated and composed with decreasing details at different layers, $0 \leq k \leq n$, from the bottom up.*

Corollary 3 *The refinement principle of systems states that any system S in \mathfrak{A} can be deductively specified and analyzed with increasing details at different layers, $0 \leq k \leq n$, from the top down.*

Corollary 4 *An abstract system $S = (C, B, R^c, R^b, R^f, R^i, R^o)$ in \mathfrak{A} is asymmetric and reflective, because its relations $\mathcal{R} = \{R^c, R^b, R^f, R^i, R^o\}$ are constrained by the following properties:*

(a) Structural asymmetry:

$$\forall c_1, c_2 \in C \wedge c_1 \neq c_2 \wedge r \in R^c, r(c_1, c_2) \neq r(c_2, c_1) \quad (12)$$

(b) Behavioral asymmetry:

$$\forall b_1, b_2 \in B \wedge b_1 \neq b_2 \wedge r \in R^b, r(b_1, b_2) \neq r(b_2, b_1) \quad (13)$$

(c) Functional asymmetry:

$$\forall b_1, b_2 \in B \times C \wedge b_1 \neq b_2 \wedge r \in R^f, r(b_1, b_2) \neq r(b_2, b_1) \quad (14)$$

(d) Externally asymmetry:

$$\forall c \in C, \forall x \in \Theta, r \in (R^i, R^o), r(x, c) \neq r(c, x) \quad (15)$$

(e) Reflective relations:

$$\forall c \in C, r(c, c) \in R^c \wedge \forall b \in B r(b, b) \in R^b \quad (16)$$

Taxonomy of systems according to the formal system model

On the basis of the abstract system model as given in Definition 3, systems as the most complex entities in the physical and abstract worlds may be rigorously classified into various categories according to one or multiple sets of the essential attributes of systems such as the components (C), behaviors (B), relations (R), and/or environments (Θ). A summary of the system taxonomy is shown in Table 1 according to the structural and behavioral characteristics of systems.

Table 1 indicates that all types of systems fit the unified framework of system taxonomy. Many system attributes are pairwise in structures and/or behaviors such as the static/dynamic system, closed/open system, and black-/white-box system. Complex systems in the facet of properties are usually with hybrid multi-categories of characteristics such as a dynamic nonlinear system and a discrete fuzzy social system.

Corollary 5 *The empty system Φ and the universal system Ω as a pair of unique singularities in the system hierarchy are both closed systems in \mathfrak{U} , i.e.:*

$$\left\{ \begin{aligned} \Phi &= (C_\Phi, B_\Phi, R_\Phi^c, R_\Phi^b, R_\Phi^f, R_\Phi^i, R_\Phi^o) \\ &= (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset) = \widehat{S}^\Phi \\ \Omega &= (C_\Omega, B_\Omega, R_\Omega^c, R_\Omega^b, R_\Omega^f, R_\Omega^i, R_\Omega^o) \\ &= (\infty, \infty, \infty, \infty, \infty, \infty, \infty) = \widehat{S}^\Omega \end{aligned} \right. \quad (17)$$

Corollary 6 *Any subsystem S^{k-1} of a closed system \widehat{S}^k is an open system, i.e.:*

$$\begin{aligned} \forall S^{k-1} \sqsubset \widehat{S}^k &\Rightarrow R_{k-1}^i \neq \emptyset \wedge R_{k-1}^o \neq \emptyset \\ &\Rightarrow S^{k-1} = \widehat{S}^{k-1} \sqcup (R_{k-1}^i, R_{k-1}^o) \end{aligned} \quad (18)$$

Most practical and useful systems in nature are open systems interacting to its environment Θ , $\Theta \sqsubset \mathfrak{U}$, in order to exchange information, energy, and/or matter. Typical entities in system environment are peers systems, users, and parent systems as described in Definition 4.

System algebra for formal system manipulations

On the basis of the mathematical models of abstract systems as developed in preceding section, formal manipu-

Table 1 Taxonomy of systems

No.	System (S)	Key characteristics of S			
		Components (C)	Behaviors (B)	Relations (R)	Environment (Θ)
1	Concrete	Real entities			
2	Abstract	Mathematical entities			
3	Physical	Natural entities			
4	Social	Humans and organizations			
5	Finite/infinite	$ C \neq \infty$ / $ C = \infty$			
6	Empty/universal	$ C = 0$ / $ C = \infty$			
7	Static/dynamic	Invariable/variable			
8	Linear/nonlinear	Linear/nonlinear functions			
9	Continuous/discrete	Continuous/discrete functions			
10	Precise/fuzzy	Precise/fuzzy functions			
11	Determinate /indeterminate	Responses predictable /unpredictable to the same stimulates			
12	Closed/open			$R^i = R^o = \emptyset$ / $R^i \neq \emptyset \wedge R^o \neq \emptyset$	$\Theta = \emptyset$ / $\Theta \neq \emptyset$
13	White/black-box	Observable/ unobservable	Fully/partially observable	Transparent/nontransparent	
14	Intelligent/ nonintelligent			Autonomic/imperative	Adaptive/nonadaptive
15	Maintainable/ nonmaintainable	Fixable/nonfixable	Recoverable/ nonrecoverable		

lations of abstract and concrete systems can be described by a denotational mathematics known as 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 efficiently analyzing and composing complex systems.

The architecture of system algebra

System algebra is a denotational mathematics for rigorous system modeling, computing, and manipulation [45,77]. System algebra deals any concrete system according to the general abstract system model. System algebra manipulates complex system structures and behaviors as algebraic operations on abstract systems.

Definition 5 The *system algebra*, SA , in the discourse of universal systems \mathcal{U} is a triple, i.e.:

$$SA \hat{=} (S, \bullet, \mathcal{U}) = ((C, B, R^c, R^b, R^f, R^i, R^o), (\bullet_r, \bullet_p, \bullet_c), \mathcal{U}), \quad (19)$$

where $\bullet = (\bullet_r, \bullet_p, \bullet_c)$ denotes the sets of *relational*, *reproductive* and *compositional* operators, respectively, on abstract systems.

The architecture of system algebra can be illustrated as shown in Fig. 2 where the type suffixes H and BL represent the hyperstructure of abstract systems and the Boolean type, respectively. Three categories of the relational, reproductive, and compositional operators are summarized in Fig. 2. Detailed descriptions of the algebraic operators in system algebra will be formally described in the following subsections [77].

System algebra provides a denotational mathematical means for algebraic manipulations of abstract systems.

System algebra can be used to model, specify, and manipulate system designs, analyses, syntheses, refinements, and validations in a wide range of applications in system science, system engineering, cognitive informatics, cognitive computing, software engineering, and intelligent systems.

Relational operations of formal systems

Definition 6 The *relational operators* \bullet_r of system algebra encompass six associative and comparative operators for manipulating the algebraic relations between abstract systems, i.e.:

$$\bullet_r \hat{=} \{\leftrightarrow, \leftrightarrow, =, \neq, \sqsubseteq, \supseteq\}, \quad (20)$$

where each of the relational operators represent *related*, *independent*, *equivalent*, *inequivalent*, *subsystem*, and *supersystem*, respectively.

The mathematical models of relational operators of system algebra are summarized in Table 2. Detailed illustrations for the set of relational operators may be referred to [77].

Reproductive operations of formal systems

Definition 7 The *reproductive operators* \bullet_p of system algebra encompass four clone operators for deriving similar systems based on existing ones, i.e.:

$$\bullet_p \hat{=} \{\Rightarrow, \Rightarrow, \overset{+}{\Rightarrow}, \overset{\sim}{\Rightarrow}\}, \quad (21)$$

where each of the reproductive operators represent system *inheritance*, *tailoring*, *extension*, and *substitute*, respectively.

Fig. 2 The architecture of system algebra

System Algebra (SA)			
	Operation	Operator	
Relational Operations (\bullet_r)	Related	\leftrightarrow	\rightarrow \textcircled{S} Related BL
	Independent	\leftrightarrow	\rightarrow \textcircled{S} Independent BL
	Equivalent	$=$	\rightarrow \textcircled{S} Equivalent BL
	Inequivalent	\neq	\rightarrow \textcircled{S} Inequivalent BL
	Subsystem	\sqsubseteq	\rightarrow \textcircled{S} Subsystem BL
	Supersystem	\supseteq	\rightarrow \textcircled{S} Supersystem BL
Reproductive Operations (\bullet_p)	Inheritance	\Rightarrow	\rightarrow $S_2 H$
	Tailoring	$\overset{\sim}{\Rightarrow}$	\rightarrow $S_2 H$
	Extension	$\overset{+}{\Rightarrow}$	\rightarrow $S_2 H$
	Substitute	$\overset{\sim}{\Rightarrow}$	\rightarrow $S_2 H$
Compositional Operations (\bullet_c)	Composition	\uplus	\rightarrow $S H$
	Decomposition	\upharpoonright	\rightarrow $S_1 H, S_2 H, \dots, S_n H$

Table 2 Mathematical models of relational operations in system algebra

No.	Operator	Symbol	Definition
1.1	Related	\leftrightarrow	$S_1(C_1, B_1, R_1^c, R_1^b, R_1^f, R_1^i, R_1^o) \leftrightarrow S_2(C_2, B_2, R_2^c, R_2^b, R_2^f, R_2^i, R_2^o)$ $\hat{=} C_1 \cap C_2 \neq \emptyset \vee R_1^i \cap (R_2^o)^{-1} \neq \emptyset \vee R_1^o \cap (R_2^i)^{-1} \neq \emptyset$
1.2	Independent	\nleftrightarrow	$S_1(C_1, B_1, R_1^c, R_1^b, R_1^f, R_1^i, R_1^o) \nleftrightarrow S_2(C_2, B_2, R_2^c, R_2^b, R_2^f, R_2^i, R_2^o)$ $\hat{=} C_1 \cap C_2 = \emptyset \wedge R_1^i \cap (R_2^o)^{-1} = \emptyset \wedge R_1^o \cap (R_2^i)^{-1} = \emptyset$
1.3	Equivalent	$=$	$S_1(C_1, B_1, R_1^c, R_1^b, R_1^f, R_1^i, R_1^o) = S_2(C_2, B_2, R_2^c, R_2^b, R_2^f, R_2^i, R_2^o)$ $\hat{=} C_1 = C_2 \wedge B_1 = B_2 \wedge R_1^c = R_2^c \wedge R_1^b = R_2^b \wedge R_1^f = R_2^f$ $\wedge R_1^i = R_2^i \wedge R_1^o = R_2^o$
1.4	Inequivalent	\neq	$S_1(C_1, B_1, R_1^c, R_1^b, R_1^f, R_1^i, R_1^o) \neq S_2(C_2, B_2, R_2^c, R_2^b, R_2^f, R_2^i, R_2^o)$ $\hat{=} C_1 \neq C_2 \vee B_1 \neq B_2 \vee R_1^c \neq R_2^c \vee R_1^b \neq R_2^b \vee R_1^f \neq R_2^f$ $\vee R_1^i \neq R_2^i \vee R_1^o \neq R_2^o$
1.5	Subsystem	\sqsubseteq	$S_1(C_1, B_1, R_1^c, R_1^b, R_1^f, R_1^i, R_1^o) \sqsubseteq S_2(C, B, R^c, R^b, R^f, R^i, R^o)$ $\hat{=} C_1 \subseteq C \wedge B_1 \subseteq B \wedge R_1^c \subseteq R^c \wedge R_1^b \subseteq R^b \wedge R_1^f \subseteq R^f$ $\wedge R_1^i \subseteq R^i \wedge R_1^o \subseteq R^o$
1.6	Supersystem	\supseteq	$S_2(C, B, R^c, R^b, R^f, R^i, R^o) \supseteq S_1(C_1, B_1, R_1^c, R_1^b, R_1^f, R_1^i, R_1^o)$ $\hat{=} C \supseteq C_1 \wedge B \supseteq B_1 \wedge R^c \supseteq R_1^c \wedge R^b \supseteq R_1^b \wedge R^f \supseteq R_1^f$ $\wedge R^i \supseteq R_1^i \wedge R^o \supseteq R_1^o$

Table 3 Mathematical models of reproductive operations in system algebra

No.	Operator	Symbol	Definition	N-ary operation
2.1	Inheritance	\Rightarrow	$S(C, B, R^c, R^b, R^f, R^i, R^o) \Rightarrow S_1(C_1, B_1, R_1^c, R_1^b, R_1^f, R_1^i, R_1^o)$ $\hat{=} S_1(C_1 = C, B_1 = B, R_1^c = R^c, R_1^b = R^b, R_1^f = R^f,$ $R_1^i = R^i \cup (S, S_1), R_1^o = R^o \cup (S_1, S))$	$\prod_{i=1}^n S_i \xrightarrow{n} S_0$
2.2	Tailoring	\Rightarrow	$S(C, B, R^c, R^b, R^f, R^i, R^o) \Rightarrow S_1(C_1, B_1, R_1^c, R_1^b, R_1^f, R_1^i, R_1^o), C_1 \subset C \wedge B_1 \subset B$ $\hat{=} S_1(C_1 = C \setminus C', B_1 = B \setminus B', R_1^c = R^c \setminus \{(C \times C') \cup (C' \times C)\},$ $R_1^b = R^b \setminus \{(B \times B') \cup (B' \times B)\}, R_1^f = R^f \setminus (B' \times C'),$ $R_1^i = R^i \cup (S, S_1) \setminus (\Theta, C'), R_1^o = R^o \cup (S_1, S) \setminus (C', \Theta))$	$\prod_{i=1}^n S_i \xrightarrow{n} S_0$
2.3	Extension	\Leftarrow	$S(C, B, R^c, R^b, R^f, R^i, R^o) \Leftarrow S_1(C_1, B_1, R_1^c, R_1^b, R_1^f, R_1^i, R_1^o), C_1 \not\subset C \wedge B_1 \not\subset B$ $\hat{=} S_1(C_1 = C \cup C', B_1 = B \cup B', R_1^c = R^c \cup \{(C \times C') \cup (C' \times C)\},$ $R_1^b = R^b \cup \{(B \times B') \cup (B' \times B)\}, R_1^f = R^f \cup (B' \times C'),$ $R_1^i = R^i \cup (S, S_1) \cup (\Theta, C'), R_1^o = R^o \cup (S_1, S) \cup (C', \Theta))$	$\prod_{i=1}^n S_i \xleftarrow{n} S_0$
2.4	Substitute	\Rightarrow	$S(C, B, R^c, R^b, R^f, R^i, R^o) \Rightarrow S_1(C_1, B_1, R_1^c, R_1^b, R_1^f, R_1^i, R_1^o),$ $C_1 \not\subset C \wedge B_1 \not\subset B, C' \subset C \wedge B' \subset B$ $\hat{=} S_1(C_1 = C \cup C' \setminus C', B_1 = B \cup B' \setminus B',$ $R_1^c = R^c \cup \{(C \times C') \cup (C' \times C)\} \setminus \{(C \times C') \cup (C' \times C)\},$ $R_1^b = R^b \cup \{(B \times B') \cup (B' \times B)\} \setminus \{(B \times B') \cup (B' \times B)\},$ $R_1^f = R^f \cup (B' \times C') \setminus (B' \times C'),$ $R_1^i = R^i \cup (S, S_1) \cup (\Theta, C') \setminus (\Theta, C'),$ $R_1^o = R^o \cup (S_1, S) \cup (C', \Theta) \setminus (C', \Theta))$	$\prod_{i=1}^n S_i \xRightarrow{n} S_0$

The mathematical models of reproductive operators of system algebra are summarized in Table 3. Detailed illustrations for the set of reproductive operators may be referred to [77].

creating complex systems based on existing ones, and vice versa, i.e.:

$$\bullet_c \hat{=} \{\uplus, \uplus\}. \tag{22}$$

Compositional operations of formal systems

Definition 8 The *compositional operators* \bullet_c of system algebra encompass a pair of synthetic and analytic operations for

The mathematical models of compositional operators of system algebra are summarized in Table 4. Detailed illustrations for the set of compositional operators may be referred to [77].

Table 4 Mathematical models of compositional operations in system algebra

No.	Operator	Symbol	Definition	N-ary operation
3.1	Composition	\uplus	$S_1(C_1, B_1, R_1^c, R_1^b, R_1^f, R_1^i, R_1^o) \uplus S_2(C_2, B_2, R_2^c, R_2^b, R_2^f, R_2^i, R_2^o)$ $\triangleq S(C = C_1 \cup C_2, B = B_1 \cup B_2, R^c = R_1^c \cup R_2^c \cup \Delta R_{12}^c,$ $R^b = R_1^b \cup R_2^b \cup \Delta R_{12}^b, R^f = R_1^f \cup R_2^f \cup \Delta R_{12}^f,$ $R^i = R_1^i \cup R_2^i \cup \{(S_1, S), (S_2, S)\},$ $R^o = R_1^o \cup R_2^o \cup \{(S, S_1), (S, S_2)\})$	$S \triangleq \biguplus_{i=1}^n S_i$
3.2	Decomposition	\pitchfork	$S(C, B, R^c, R^b, R^f, R^i, R^o) \pitchfork_{i=1}^2 S_i(C_i, B_i, R_i^c, R_i^b, R_i^f, R_i^i, R_i^o),$ $C = \bigcup_{i=1}^2 C'_i \wedge B = \bigcup_{i=1}^2 B'_i$ $\triangleq S_1(C_1 = C'_1, B_1 = B'_1, R_1^c = R^c \setminus \{(C'_1 \times C) \cup (C \times C'_1)\},$ $R_1^b = R^b \setminus \{(B'_1 \times B) \cup (B \times B'_1)\}, R_1^f = \{B_1 \times C_1 B_1 \times C_1 \subset R^f\},$ $R_1^i = R^i \setminus \{(\Theta, C'_2)\}, R_1^o = R^o \setminus \{(C_2, \Theta)\})$ $\pitchfork S_2(C_2 = C'_2, B_2 = B'_2, R_2^c = R^c \setminus \{(C'_2 \times C) \cup (C \times C'_2)\},$ $R_2^b = R^b \setminus \{(B'_2 \times B) \cup (B \times B'_2)\}, R_2^f = \{B_2 \times C_2 B_2 \times C_2 \subset R^f\},$ $R_2^i = R^i \setminus \{(\Theta, C'_1)\}, R_2^o = R^o \setminus \{(C'_1, \Theta)\})$	$S \triangleq \bigpitchfork_{i=1}^n S_i$

As defined in Eq. 3.1 in Table 4, the *composition* of two given systems S_1 and S_2 , denoted by $S = S_1 \uplus S_2$, results in a supersystem S with the newly created relations ΔR_1^c and ΔR_2^c . However, the system gain will be diminished when S is decomposed into subsystems in an inverse operation, i.e., $S = \pitchfork_{i=1}^2 S_i$ as formally explained in Eq. 3.2 in Table 4.

Formal principles and properties of complex systems

The *abstract system theory* centered by the mathematical model of abstract systems and system algebra is the latest attempt to provide a rigorous treatment for the formal properties and principles of general systems. This section describes fundamental principles of system science on the basis of abstract system theories and system algebra. A comprehensive set of system phenomena, properties, and empirical principles are formally explained in this section.

The structural complexity of formal systems

According to Theorem 1 and Corollary 5, abstract and real-world systems may be very small or extremely large between (Φ, Ω) where $|\Phi| = 0$ and $|\Omega| = \infty$. A formal model of system magnitudes can be quantitatively introduced to classify the structural sizes of systems and their relationship with other basic attributes of systems. In order to derive such a model, a set of measures on system sizes, magnitudes, and complexities is defined below.

Definition 9 Given an abstract system $S = (C, B, R^c, R^b, R^f, R^i, R^o)$, the *structural size of the system* $\Xi(S)$ is determined by the number of components $|C| = n_c$ encompassed in the system, i.e.:

$$\Xi(S) = |S| = |C| = n_c \quad (23)$$

Definition 10 Given an abstract system $S = (C, B, R^c, R^b, R^f, R^i, R^o)$, the *magnitude of the system* $M(S)$ is the number of asymmetric binary relations including the reflexive relations among the set of components C , i.e.:

$$M(S) = |R^c| = |C \times C| = n_c^2 \quad (24)$$

If the self-reflective relations among all components are eliminated, the number of binary relations in a given abstract system represents its structural complexity.

Definition 11 Given an abstract system $S = (C, B, R^c, R^b, R^f, R^i, R^o)$, the *structural complexity of the system*, $O_c(S)$, is the number of all pairwise relations among the components in C except the self-reflective ones, i.e.:

$$\begin{aligned} O_c(S) &= M(S) - |C| \\ &= n_c^2 - n_c \\ &= n_c(n_c - 1)[E], \end{aligned} \quad (25)$$

where the unit of system structural complexity is called *compound Entity* [E] that represents the number of component relations in a given system.

Theorem 2 The structural complexity of an abstract system S , $O_c(S)$, is determined by an unordered pairwise combination among all components in C , i.e.:

$$O_c(S) = 2 \bullet C_{n_c}^2, \quad (26)$$

where the factor 2 represents the asymmetric binary relations $r(a, b) \neq r(b, a)$, $r \in R^c$.

Proof According to Definition 11 and combinatorics, Theorem 2 is proved as follows:

$$\begin{aligned}
 O_c(S) &= n_c(n_c - 1) \\
 &= 2 \cdot \frac{n_c(n_c - 1)}{2} \\
 &= 2 \cdot \frac{n_c!}{2!(n_c - 2)!} \\
 &= 2 \cdot C_{n_c}^2 \tag{27}
 \end{aligned}$$

□

According to Theorem 2 and Definition 11, the structural complexity of systems represents a fully bidirectionally connected system that is constrained by the asymmetric relations of the system. Therefore, the structural complexity of systems is the theoretical upper-bound of system complexity where all components in a system are potentially fully interconnected with each other. Certain systems may only possess partial connections where their particular structural complexity is constrained by the upper bound.

Example 3 The property of system structural complexity can be illustrated in Fig. 3, where \uplus denotes a system composition as formally defined in system algebra in “Compositional operations of formal systems”. Applying Theorem 2, the asymmetric structural complexities of the given systems, \widehat{S}^1 , \widehat{S}^2 , and \widehat{S} can be quantitatively measured, respectively, as follows:

$$O_c(\widehat{S}^1) = n_{c_1}(n_{c_1} - 1) = 3(3 - 1) = 6 [E]$$

Fig. 3 Generation of relations in abstract systems

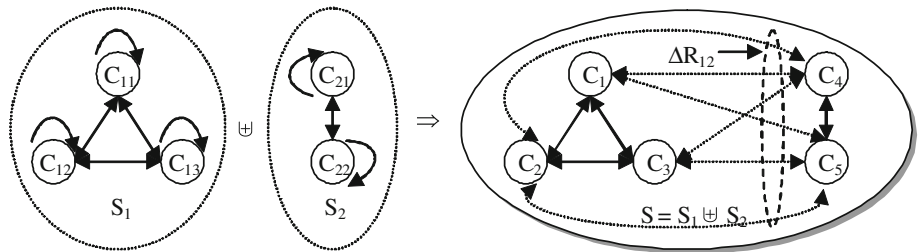


Table 5 The 7-layer model of system magnitudes

Level	Category	Size ($\Xi(S) = n_c$)	Magnitude ($M(S) = n_c^2$)	Structural complexity ($O_c(S) = n_c(n_c - 1)$)
1	The empty system (Φ)	0	0	0
2	Small system	[1, 10]	[1, 10^2]	[0, 90]
3	Medium system	(10, 10^2]	(10^2 , 10^4]	(90, 0.99×10^4]
4	Large system	(10^2 , 10^3]	(10^4 , 10^6]	(0.99×10^4 , 0.999×10^6]
5	Giant system	(10^3 , 10^4]	(10^6 , 10^8]	(0.999×10^6 , 0.9999×10^8]
6	Immense system	(10^4 , 10^5]	(10^8 , 10^{10}]	(0.9999×10^8 , 0.99999×10^{10}]
7	The infinite system (Ω)	∞	∞	∞

$$O_c(\widehat{S}^2) = n_{c_2}(n_{c_2} - 1) = 2(2 - 1) = 2 [E]$$

$$O_c(\widehat{S}) = n_c(n_c - 1) = 5(5 - 1) = 20 [E]$$

The extent of system magnitudes can be formally classified at seven levels known as the *empty, small, medium, large, giant, immense, and infinite (universal)* systems from the bottom up. Relationships between system sizes, magnitudes, and structural complexities in the system hierarchy are summarized in Table 5 known as the *7-layer Model of System Magnitudes*. The quantitative measurement scheme in Table 5 forms a reference model of system magnitudes and complexities.

Table 5 provides a new view for the taxonomy of systems based on their complexities and magnitudes in addition to the facet of their functional characteristics as summarized in Table 1. Table 5 indicates that the complexity of a small system may easily be out of control of human cognitive manageability because of the combinatorial and exponential expanding rates. It explains why most of the real-world systems are really too hard to be modeled and handled in conventional system techniques. According to Table 5, a complex system may be classified as a system that its magnitude is higher than those of large systems at Levels 5 and 6.

Corollary 7 *The holistic complexity of systems states that within the 7-level scale of system magnitudes, almost all systems are too complicated to be cognitively understood or mentally handled as a whole, except small systems or those that can be decomposed into a set of small systems according Eq. 3.2 in Table 4.*

According to Corollary 7, the basic principle for dealing with complex systems is *system decomposition* as described

in Table 4 where the complexity of any decomposed subsystem can be small enough to be cognitively manageable in system engineering.

Corollary 8 *The singularity of the minimum and maximum systems states that the uniqueness of the bottom and top levels of the system hierarchy is represented by only two abstract systems known as the empty and universal systems, respectively, in the 7-layer system hierarchy, i.e.:*

$$\begin{aligned} \widehat{S^\Phi} &\triangleq (C_\Phi, B_\Phi, R_\Phi^c, R_\Phi^b, R_\Phi^f), |C_\Phi| = |B_\Phi| = |R_\Phi^c| \\ &= |R_\Phi^b| = |R_\Phi^f| = 0 \\ &= \Phi = (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset) \end{aligned} \quad (28)$$

$$\begin{aligned} \widehat{S^\Omega} &\triangleq (C_\Omega, B_\Omega, R_\Omega^c, R_\Omega^b, R_\Omega^f), C_\Omega = \mathfrak{C}, B_\Omega = \mathfrak{B}, \\ R_\Omega^c &= \mathfrak{R}_c, R_\Omega^b = \mathfrak{R}_b, R_\Omega^f = \mathfrak{R}_f \\ &= \Omega = (\infty, \infty, \infty, \infty, \infty) \end{aligned} \quad (29)$$

It is noteworthy that according to Corollary 8, although there are infinitive concrete systems in the 7-Layer system hierarchy in the real world, there is only a unique empty system and a unique universal system. Both the empty and universal systems are an abstract closed system; however, almost all concrete systems in the real world are open systems.

The behavioral complexity of formal systems

Although system behaviors may greatly vary in applications, it is found in computer science and computational intelligence that there are only a finite set of meta-behaviors [37,41,42] shared by all applied systems in both physical and intelligent systems. Complex behaviors are algebraic compositions of these meta-behaviors. The fundamental mathematical model of system meta-behaviors is a process as revealed in RTPA [37]. A set of 17 meta-processes is identified that may be composed by a set of 17 algebraic process operators in order to build the behaviors of larger components and complex systems. The syntaxes and semantics of each meta and relational cooperators of RTPA may be referred to [47,48].

As demonstrated in Example 2, the behaviors of any concrete system, in terms of the behavioral relations R^b , the functional relations R^f , the input relations R^i , and the output relations R^o , can be rigorously specified and practically implemented in RTPA.

The behavioral complexity of system as combinatory relations

The static aspect of system behavioral complexity can be modeled in a similar approach as that of system structural complexity as developed in “The structural complexity of formal systems”.

Definition 12 Given an abstract system $S = (C, B, R^c, R^b, R^f, R^i, R^o)$, the *behavioral complexity of the system*, $O_b(S)$, is the number of all pairwise relations among the behaviors in R^b except the self-reflexive ones, i.e.:

$$\begin{aligned} O_b(S) &\triangleq 2 \bullet C_{n_b}^2, n_b = |B| \\ &= n_b(n_b - 1) \end{aligned} \quad (30)$$

The dynamic aspect of behavioral complexity is the functional interaction between the defined sets of behaviors and components of the system as a Cartesian product $B \times C$.

Definition 13 Given an abstract system $S = (C, B, R^c, R^b, R^f, R^i, R^o)$, the *functional complexity of the system*, $O_f(S)$, is determined by the number of all functional relations R^f between the sets of behaviors B and components C which forms the dynamic functions of the system, i.e.:

$$\begin{aligned} O_f(S) &\triangleq |R^f| = |B \times C| \\ &= n_b \bullet n_c \end{aligned} \quad (31)$$

Corollary 9 *The total functional complexity of an abstract system $S = (C, B, R^c, R^b, R^f, R^i, R^o)$, $O_F(S)$, is the sum of its behavioral complexity $O_b(S)$, cross-functional complexity $O_f(S)$, input complexity $|R^i|$, and output complexity $|R^o|$, i.e.:*

$$\begin{aligned} O_F(S) &\triangleq O_b(S) + O_f(S) + |R^i| + |R^o| \\ &= n_b(n_b - 1) + n_b \bullet n_c + n_i + n_o \\ &= n_b(n_b + n_c - 1) + n_i + n_o [F], \end{aligned} \quad (32)$$

where the unit of system functional complexity is abstract function (F) that represents the number of relational behaviors in the given system.

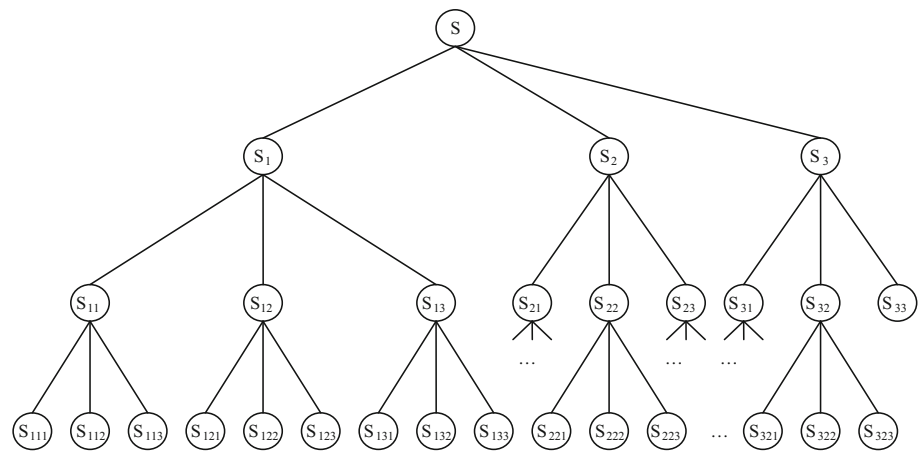
It is observed that, for any system, the higher the structural complexity and/or the functional complexity, the higher the total complexity of the system. In other words, the total complexity of a system is proportional to both its structural and functional complexities.

Corollary 10 *The total complexity of an abstract system $S = (C, B, R^c, R^b, R^f, R^i, R^o)$, $O(S)$, is a product of its total functional complexity $O_F(S)$ and structural complexity $O_c(S)$ measured in the unit of function-entity [FE], i.e.:*

$$\begin{aligned} O(S) &\triangleq O_F(S) \bullet O_c(S) \\ &= [n_b(n_b + n_c - 1) + n_i + n_o] \bullet n_c(n_c - 1) [FE] \end{aligned} \quad (33)$$

Example 4 The total complexities of the example systems \widehat{S}^1 , \widehat{S}^2 , and \widehat{S} as given in Fig. 4 can be determined according to Corollary 10, respectively, as follows:

Fig. 4 A ternary system organization tree SOT(3, 24)



$$\begin{aligned}
 O(\widehat{S}^1) &= [n_{b_1}(n_{b_1} + n_{c_1} - 1) + n_{i_1} + n_{o_1}] \bullet n_{c_1}(n_{c_1} - 1) \\
 &= [6(6 + 3 - 1) + 0 + 0] \bullet 3(3 - 1) = 288 \text{ [FE]} \\
 O(\widehat{S}^2) &= [n_{b_2}(n_{b_2} + n_{c_2} - 1) + n_{i_2} + n_{o_2}] \bullet n_{c_2}(n_{c_2} - 1) \\
 &= [4(4 + 2 - 1) + 0 + 0] \bullet 2(2 - 1) = 40 \text{ [FE]} \\
 O(\widehat{S}^2) &= [n_b(n_b + n_c - 1) + n_i + n_o] \bullet n_c(n_c - 1) \\
 &= [10(10 + 5 - 1) + 0 + 0] \bullet 5(5 - 1) = 2800 \text{ [FE]},
 \end{aligned}$$

where a composition of two simple systems has resulted in a very complicated system by $\widehat{S} = \widehat{S}^1 \uplus \widehat{S}^2$.

The behavioral complexity of system in the time dimension

It is recognized that the complexities of systems are not only measured in the size dimension, but also in the time dimension [63]. A new facet of system functional complexity is introduced in this subsection known as the time dimensional complexity of systems as a measure of the lifespan complexity of a given system. Assume a generation in sociological term is 20 years; then, system complexity in the time dimension can be classified at three levels known as the *short*, *medium*, and *long* lifespan systems.

Definition 14 A *long-lifespan system* (LLS) is a higher order system with a lifespan longer than the creator or observer of the system, which is usually beyond three sociological generations, i.e., 60 years.

Most complex systems are an LLS, particularly those of the natural, social, knowledge-based, economical, information networks, and database systems. For instances, the Internet and many software systems are an LLS, which is complicated because not only their structural and functional magnitudes, but also their continuous lifespans that may last for a very long period well beyond those of the designers, observers, or users.

Definition 15 The *lifespan complexity* of an LLS, $O_t(S_L)$, is a product of its lifespan in time T_{S_L} [Hr] and its total functional complexity $O(S_L)$ in term of function-entity [FE], i.e.:

$$\begin{aligned}
 O_t(S_L) &\hat{=} O(S_L) \bullet T_{S_L} \\
 &= O_f(S_L) \bullet O_c(S_L) \bullet T_{S_L} \text{ [FE} \bullet \text{Hr]}, \tag{34}
 \end{aligned}$$

where the unit of the whole dimensional complexity of systems is *function-entity-hour* [FE • Hr].

The principle of system fusion

Systems are widely needed due to the advantage of system gains via system fusion in physical, abstract, and social worlds. The system fusion effect is a unique property of systems that is not possessed by any of its components before it is composed into the system.

Definition 16 The *fusion effect* of systems is a self-induction property of any system that creates the incremental usage (such as structures, relations, and functions) and complexity when the system is coherently composed as a whole.

Example 5 The *incremental relations* ΔR_{12} and *incremental complexity* $\Delta O(\widehat{S})$ created by the composition $\widehat{S} = \widehat{S}^1 \uplus \widehat{S}^2$ as given in Example 3 and Fig. 3 indicate the effects of system fusions, i.e.:

$$\begin{aligned}
 \Delta R_{12} &= R_S - (R_{S_1} + R_{S_2}) \\
 &= O_c(\widehat{S}) - (O_c(\widehat{S}^1) + O_c(\widehat{S}^2)) \\
 &= 20 - 6 - 2 \\
 &= 12 \\
 \Delta O(\widehat{S}) &= O(\widehat{S}) - (O(\widehat{S}^1) + O(\widehat{S}^2)) \\
 &= 2,800 - 288 - 40 \\
 &= 2,472
 \end{aligned}$$

An important phenomenon in system fusion is *system mutation* where the gradual increment of quantities, ΔC or ΔB , triggers the exponential generation of functionality (quality) in the system when the incremental is greater than a certain threshold.

Theorem 3 *An incremental union of two sets of relations R_1 and R_2 , denoted by $R \hat{=} R_1 \boxplus R_2$, is a union of R_1 and R_2 plus the newly generated incremental set of relations ΔR_{12} , i.e.:*

$$\begin{cases} R \hat{=} R_1 \boxplus R_2 = R_1 \cup R_2 \cup \Delta R_{12} \\ |\Delta R_{12}| \hat{=} 2|C_1| \cdot |C_2| \end{cases} \quad (35)$$

where $\Delta R_{12} \subset R_1 \boxplus R_2$ but $\Delta R_{12} \not\subset R_1 \wedge \Delta R_{12} \subset R_2$.

Proof Because ΔR_{12} is the difference between the relations of the newly generated supersystem $|R|$ and those of the individual subsystems $|R_1|$ and $|R_2|$:

$$\begin{aligned} |\Delta R_{12}| &= |R| - (|R_1| + |R_2|), \Delta R_{12} \subset S \wedge R_1 \subset S_1 \wedge R_2 \subset S_2 \\ &= (n_{c_1} + n_{c_2})^2 - (n_{c_1}^2 + n_{c_2}^2) \\ &= (n_{c_1}^2 + 2n_{c_1}n_{c_2} + n_{c_2}^2) - (n_{c_1}^2 + n_{c_2}^2) \\ &= 2n_{c_1}n_{c_2} \\ &= 2|C_1| \cdot |C_2| \end{aligned} \quad (36)$$

□

Corollary 11 *The fusion principle of systems states that the composition of two systems, $S = S_1 \uplus S_2$, results in the creation of new relations or functions ΔR_{12} , which solely belong to the supersystem S but not belong to any of the original individual subsystems S_1 or S_2 . The principle can be generally extended to n -ary dimensions of system fusions where $S = \uplus_{i=1}^n S_i$.*

A well-known expression of the system fusion effect is that “the whole of a system is always greater than the sum of all its parts [6, 12, 20].” However, the empirical statement has not been formally modeled in order to rigorously explain the key mechanisms of system gains during system fusions. Theorem 3 and Corollary 11 can be applied to formally explain the important property of system fusions when the ‘whole’ and ‘parts’ of the given system refer to its components C or behaviors B . But, it is noteworthy that the empirical statement is untrue when the ‘whole’ and ‘parts’ denote the work products of the system as analyzed in the following.

The concept of abstract work done by a system is an extension and generalization of the concept of work in concrete systems such as those of a kinetic, electrical, thermodynamic, and human social system.

Definition 17 *The abstract work done by a system S , $W(S)$, is its output of utility U in term of the implemented number of functions F , i.e.:*

$$\begin{aligned} W(S) &\hat{=} U = O_F(S) \\ &= n_b(n_b + n_c - 1) + n_i + n_o [F] \end{aligned} \quad (37)$$

The abstract function U in Eq. 37 can be perceived as energy spent in Joule in physical systems, information generated or processed in bit in intelligent systems, or tasks conducted in person-hour in human-based systems.

Theorem 4 *The functional gain of system fusions states that work done by a system is always greater than any of its components, but never be greater than the sum of them due to the system efficiency constraint $\eta(S)$, i.e.:*

$$W(S) \leq \sum_{i=1}^n W(S_i), \quad S_i \subset S \wedge \eta(S_i) \leq 1, \quad (38)$$

where $\eta(S_i)$ represents the efficiency of subsystem or a component.

Proof Because $R_{i=1}^n \eta(S_i) \leq 1$ for any of the n subsystems S_i in S , $W(S) = \sum_{i=1}^n \eta(S_i) W(S_i) \leq \sum_{i=1}^n W(S_i)$. □

Theorem 4 indicates that, although a system’s capability to carry out a work is more powerful than any of its components, the total work done by the system cannot exceed the sum of all its components because the existence of overheads or no system may reach the ideal efficiency where $\eta(S_i) = 1$. In other words, in the term of abstract work done by a system, the whole of the system is not always greater than the sum of all its parts.

Organization of complex systems

System organization is a key methodology to reduce system complexity and to improve system efficiency. A set of components may form a system by coherent organization towards a common goal of the system. It is widely and empirically observed that the tree-like architecture is a universal hierarchical prototype of systems across disciplines not only in science and engineering, but also in sociology and living systems.

Principles of structural topology and complexity reduction of systems

Although the structural topology of arbitrary unstructured system may be a network, that of structured systems is tree-like architecture according to Theorem 1. The former can be reduced to the latter by system organization methods where

the complexity of a structured system is dramatically lower than those of unstructured systems.

Corollary 12 *The general topological structure for system modeling and organization is a tree, particularly the complete n-ary tree.*

A tree said to be *complete* means that all levels of the tree are allocated with the maximum number of possible nodes except two special cases where it is at the leave level and/or on the rightmost subtrees [42, 78].

Definition 18 *A complete n-ary tree, $T_c(n, N)$, is a normalized tree where each node can have at most n children, each level k from the top-down can have at most n^k nodes, and all levels have allocated the maximum number of possible nodes except the rightmost subtrees. The number of nodes at the leave level is $N \leq n^k$.*

The advantage of complete trees is that the configuration of any complete n -ary tree $T_c(n, N)$ can be rigorously determined by only two attributes: the unified fan-out n and the number of leave nodes N at the bottom level.

Definition 19 *A normalized system is a hierarchically structured system where no direct interconnections between nodes belonging to different subtrees. Therefore, cross-functions between such nodes should be coordinated through a common parent node.*

Systems tend to be normalized into a hierarchical structure represented by a complete n -ary tree in order to maintain equilibrium, evolvability, and optimal predictability in system organization. The advantages of tree-structured system organization can be formally described in the following principle.

Corollary 13 *Advantages of the normalized tree architecture of system organization are as follows:*

- (a) *Equilibrium: Looking down from any node at a given level of the system tree, except the leave level, the structural property of fan-out or the number of coordinated components are the same and evenly distributed.*

- (b) *Evolvability: A normalized system is flexible and adaptive that does not need to change the existing structure for future growth.*
- (c) *Optimal predictability: Properties of the normalized system modeled by a complete n-ary tree, $T_c(n, N)$, are rigorously predictable as given in Table 6 once the unified fan-out n and the number of leave nodes N at the bottom level are determined.*

Based on the model of complete trees, the topology of normalized systems can be modeled by the system organization tree.

Definition 20 *A system organization tree (SOT) is an n-ary complete tree in which all leave nodes represent a component and the remainder nodes beyond the leave level represent a subsystem.*

Example 6 A ternary SOT, $SOT(n, N) = SOT(3, 24)$, is shown in Fig. 4. As a complete ternary tree, the rightmost subtrees and leaves of the SOT will be left open when the leaves (components) do not reach the possible maximum.

According to Corollary 13 and Definition 20, SOT is an ideal model for organizing a normalized structured system. A summary of the useful topological properties of SOT is as follows.

Corollary 14 *An n-ary system organization tree, $SOT(n, N)$, possesses the predictable properties as given in Table 6 when the unified fan-out n and the total number of leave nodes N are determined.*

Corollaries 13 and 14 formally explain the theories behind the universal phenomena of system science such as why systems tend to adopt tree structures in formal organizations and what advantages of hierarchically normalized systems are in system organization. SOT can be used as a formal model to rigorously analyze the architectures and efficiencies of system organizations.

Table 6 Properties of system organization trees $SOT(n, N)$

No.	Property	Mathematical model	Remark
1	The maximum number of fan-out	$\bar{n}_{fo} = n$	At any given node
2	The maximum number of nodes	$n_k = n^k$	At a given level k
3	The depth of the SOT	$d = \lceil \frac{\log N}{\log n} \rceil$	
4	The maximum number of nodes	$N_{SOT} = \sum_{k=0}^d n^k$	In the SOT
5	The maximum number of components	$N = n^d$	On all leave nodes in the SOT
6	The maximum number of subsystems	$N_m = N_{SOT} - N - 1 = \sum_{k=1}^{d-1} n^k$	Nodes except all leaves in the SOT

The principle of system equilibriums

System equilibriums are constrained by the existence of autonomous negative feedback in a given system. Feedback is a universal phenomenon that exists not only in physical systems, but also in advanced systems such as biological, neurological, physiological, economical, and social systems.

The functional structure of any system encompasses four essences known as its input (I), internal behavioral process (P), output (O), and feedback (F) denoted by IPOF. The feedback to a system is proportional to the output of the system that can be positive or negative represented as IPOF⁺ and IPOF⁻, respectively. The former is a self-stimulated system, while the latter is a self-regulated system.

Definition 21 The *equilibrium of a system* is a stable state of a given system \bar{S} where the effects of all abstract work of each component (subsystem) form a zero-sum, i.e.:

$$W(\bar{S}) = \sum_{i=1}^n W(S_i) = 0, S_i \subset \bar{S}, \tag{39}$$

where $W(S_i)$ is the abstract work of a component or subsystem in S , and $W(S)$ is the total work done by the system.

Corollary 15 *System equilibrium, as well as system self-organization, is implemented by the negative feedback mechanisms in a system, IPOF⁻, that is inversely proportional to the aggregative effect of the system’s output.*

Corollary 16 *Conservative work of equilibrium systems states that the sum of all types of works done in an equilibrium system is always zero, i.e.:*

$$\forall \bar{S} = \bigcup_{i=1}^n S_i, \sum_{i=1}^n W(S_i) \equiv 0 \tag{40}$$

Example 7 The following phenomena as shown in Table 7 are examples of the system equilibrium theory in different fields of science and engineering that fit the general principle as given in Corollary 16.

The principle of system self-organization

System organization is a process to configure and manipulate the system towards a stable and ordered state with an internal equilibrium. Self-organization is an important property of systems when there exists an equilibrium in the system’s characteristic function [3, 18, 42].

Let $f(x)$ be a continuous and deferential state function of a system defined in an arbitrary interval $[a, b]$, the following principle can be derived.

Theorem 5 *The condition of self-organization states that the necessary and sufficient condition of self-organization is the existence of at least one minimum on the state curve of a system $f(x)$, which satisfies the following requirements:*

$$\begin{cases} f'(x_{\min}|x_{\min} \in (a, b)) = 0 \\ f''(x_{\min}|x_{\min} \in (a, b)) > 0, \end{cases} \tag{41}$$

where $f'(x)$ and $f''(x)$ are the first and second derivatives of $f(x)$ in (a, b) , respectively.

Proof For an arbitrary state function of a system $f(x)$, there are two possibilities when $f'(x_0) = 0$ and $f''(x_0) > 0$, i.e.:

$$\begin{aligned} f'(x|x = x_0 \in (a, b)) = 0 \wedge f''(x|x = x_0 \in (a, b)) > 0 \\ \Rightarrow \begin{cases} f'(x|x < x_0 \in (a, b)) < 0 \\ f'(x|x > x_0 \in (a, b)) > 0 \end{cases} \\ \Rightarrow x_0 = x_{\min}, \end{aligned} \tag{42}$$

where $x_0 = x_{\min}$ guarantees that the system can autonomously reach an equilibrium state from both sides of $x_{\min} \in (a, b)$ without other external effort. □

Table 7 Properties of conservative system equilibriums

No.	Category	Phenomenon	Mathematical model	Description
1	Kinetic system	Newton’s 1st law of motion	$\vec{F} = \sum_{i=1}^n \vec{F}_i = 0 \Rightarrow \vec{a} = 0$	An object remains at rest or a state of motion at a constant velocity, if the sum of all forces exerted on it, \vec{F} , is zero
2	Energy system	Sum of work	$\sum_{i=1}^n F_i d_i = 0$	The sum of all work done by a force F in a circle of movement d is zero
3	Energy system	Energy conservation	$\sum_{i=1}^n E_i = 0$	The sum of all forms of energy E in a closed system is zero
4	Electrical system	Kirchhoff’s rule	$\sum_{i=1}^n P_i = 0$	The sum of all potentials P in a closed circuit system is zero
5	Economic equilibrium	Economic equilibrium	$\sum_{i=1}^n (P_i(D) + P_i(S)) = 0$	The effect of all demands D and supplies S on the price P in an ideal market is zero

Because negative feedback is the only means to regulate the states of a system, the following corollary can be derived based on Theorem 5.

Corollary 17 *The functional condition of a self-organization system is the existence of the negative feedback mechanism that is inversely proportional to the incremental of the aggressive effects of the system.*

Definition 22 A conservative system is an inertial system that has a tendency to remain unchanged under external influence in a certain scope.

A conservative system may autonomously adapt to an equilibrium state by proportional negative feedback against a change. An inertial or conservative system possesses the tendency to remain constant at a given equilibrium within a certain scope of stability against external changes. In the dynamic aspect, when the current equilibrium state of an inertial system cannot be maintained, it transits to another equilibrium without abrupt changes. Many physical, economical, and social systems are inertial, because of the existence of negative feedback. A typical conservative system in physics is Newton's inertial system of kinetics where if the net force imposed on a body is zero, the body either remains at rest or moves at a constant velocity [9]. The economic system equilibrium, empirically known as the invisible hand by Adam Smith [34], is proved in mathematics by Wang as an inertial and self-organization system [55].

The principle of system dissimulation

Dissimulation is a universal property of any system such as physical, economic, living, or social systems. According to the system taxonomy as summarized in Table 1, there are maintainable and nonmaintainable systems. The properties of dissimulation of both types of systems are analyzed in this subsection.

Let the *availability* of a system, $\alpha(t)$, be denoted by its designed utility, function, efficiency, or reliability in the time dimension. Then, system dissimulation is the tendency that a system undergoes an apparent or hidden destructive change against its original purposes or designed availability. System dissimulation can be analyzed on the basis of how systems maintain their functional availability and against the loss of it.

Definition 23 The *dissimulation of a nonmaintainable system*, D_n , is determined by its degradation of availability $\alpha(t)$ over time during its *lifecycle* T , i.e.:

$$D_n = k(1 - e^{-t-T}), \quad 0 \leq t \leq T \wedge k = \alpha(t|t=0), \quad (43)$$

where k is a positive constant called the *initial availability* of the system.

Definition 24 The *rate of dissimulation*, δ_n , of a nonmaintainable system can be derived as follows:

$$\begin{aligned} \delta_n &= \frac{d}{dt} D_n(t) \\ &= \frac{d}{dt} k(1 - e^{t-T}) \\ &= -ke^{t-T}, \quad 0 < t \leq T \end{aligned} \quad (44)$$

The trend of dissimulation of a nonmaintainable system D_n (the red curve) and its rate δ_n (the blue curve) are shown in Fig. 5, where they are normalized by $k = 1$. The unit of δ_n adopts a different scale from that of D_n in order to better contrast them. Figure 5 indicates that a system is exponentially dissimulating during its lifecycle, and the rate of the dissimulation reaches the maximum in the last phase of its lifecycle.

Corollary 18 *System dissimulation states that any system tends to undergo a continuous degradation that leads to the eventual loss of its designed utility and against its initial purposes to build the system.*

Corollary 18 indicates that any concrete system has a certain lifecycle, in which dissimulation is being undergone since the moment when it is put into operation. The *most critical period of system dissimulation* is its exiting phase where the rate of dissimulation of the system is uncontrollably increased.

If the development or building period is considered as a part of the lifecycle of the system, the dissimulation of a nonmaintainable system during this phase is negative, because of the continuous effort spent in system development.

Definition 25 The *dissimulation of a nonmaintainable system* during the development phase, D'_n , is determined as follows:

$$D'_n = k(1 - e^{-t-T'}), \quad T' \leq t \leq 0, \quad (45)$$

where T' may be different from T .

Therefore, in the entire lifecycle of a nonmaintainable system, $T' + T$, the trend of dissimulation is shown in Fig. 6, where $t = 0$ is the time that the system is put into operation.

However, a maintainable system provides an opportunity to introduce external effort to cope with system dissimulation.

Definition 26 The *dissimulation of a maintainable system*, D_m , is described by a periodical function with recovered availability corresponding to the maintenance cycle T as follows:

Fig. 5 System dissimulation (nonmaintainable system, $T = 20$)

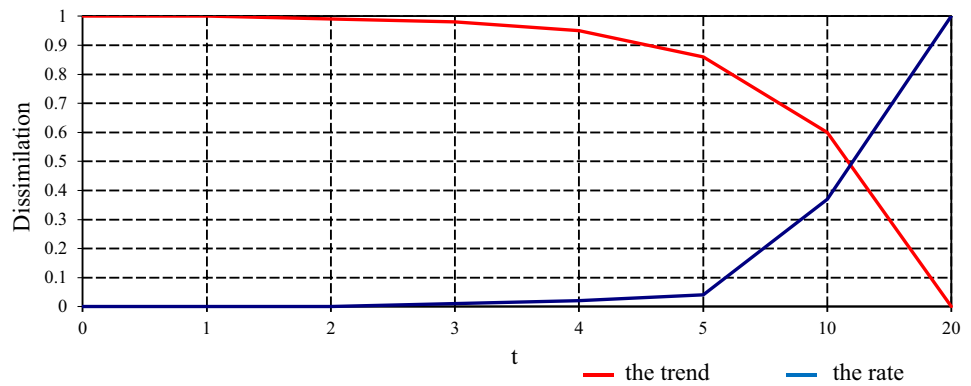


Fig. 6 The entire lifecycle of system dissimulation (nonmaintainable system, $T = T' = 20$)



$$D_m = \begin{cases} k(1 - e^{-t-T'}), & T' \leq t < 0 \\ k(1 - e^{-t-T}), & nT \leq t < (n+1)T, n \geq 0, \end{cases} \quad (46)$$

where n is the number of operation periods.

The dissimulation of a maintainable system can be derived based on those of nonmaintainable systems when the maintenance effect is treated as a recovery of the original availability of the system. Therefore, the dissimulation during the whole lifecycle of a maintainable system is determined by Eq. 46 as illustrated in Fig. 7.

System dissimulation may be considered as an inverse process of system fusion as described in “The principle of system fusion”. The property of system dissimulation can be used to explain a wide range of phenomena in system theories, such as system availability, efficiency, reliability, and the trends of systems during the entire lifecycle. Real-world cases of system dissimulation and anti-dissimulation are those such as in social welfare system maintenance, software system version upgrading, strategic business transformation, and health care system improvement.

Paradigms of complex cognitive and intelligent systems

Recent studies in cognitive informatics and brain informatics reveal many insights about the brain as the most complex natural intelligence system. *Cognitive informatics* is a transdisciplinary enquiry of computer science, information

science, cognitive science, and intelligence science, which investigates into the internal information processing mechanisms and processes of the brain and natural intelligence, as well as their engineering applications in cognitive computing [38,44,50,53,57,64,68,76,80,81,84,87–90]. *Brain informatics* is a joint field of brain and information sciences that studies the information processing mechanisms of the brain at the physiological level by computing and medical imagination technologies [51,52,56,60,62,65,67,69,70,79]. Brain informatics explains how the most complicated physiological system, the human brain, is formed based on complex nervous systems and neurological foundations as observed in brain anatomy and neurophysiology [7,23,65].

It is recognized that the exploration of the brain is a complicated recursive system problem. On the basis of the abstract system theory and formal system principles as presented in the preceding sections, contemporary cognitive and intelligent systems can be rigorously analyzed for exploring the complex brain systems and memory systems.

The complex system model of the brain

Conventional studies on the structures and functions of the brain mainly applied the *abductive* methodology, which attempts to explain the brain system by psychological and clinical evidences particularly abnormal functions [7,21,23,27]. There is a lack of a logical structure and an *inductive* theory for explaining the brain. Cognitive psychology and

Fig. 7 The entire lifecycle of system dissimulation (maintainable system, $T = 20$)

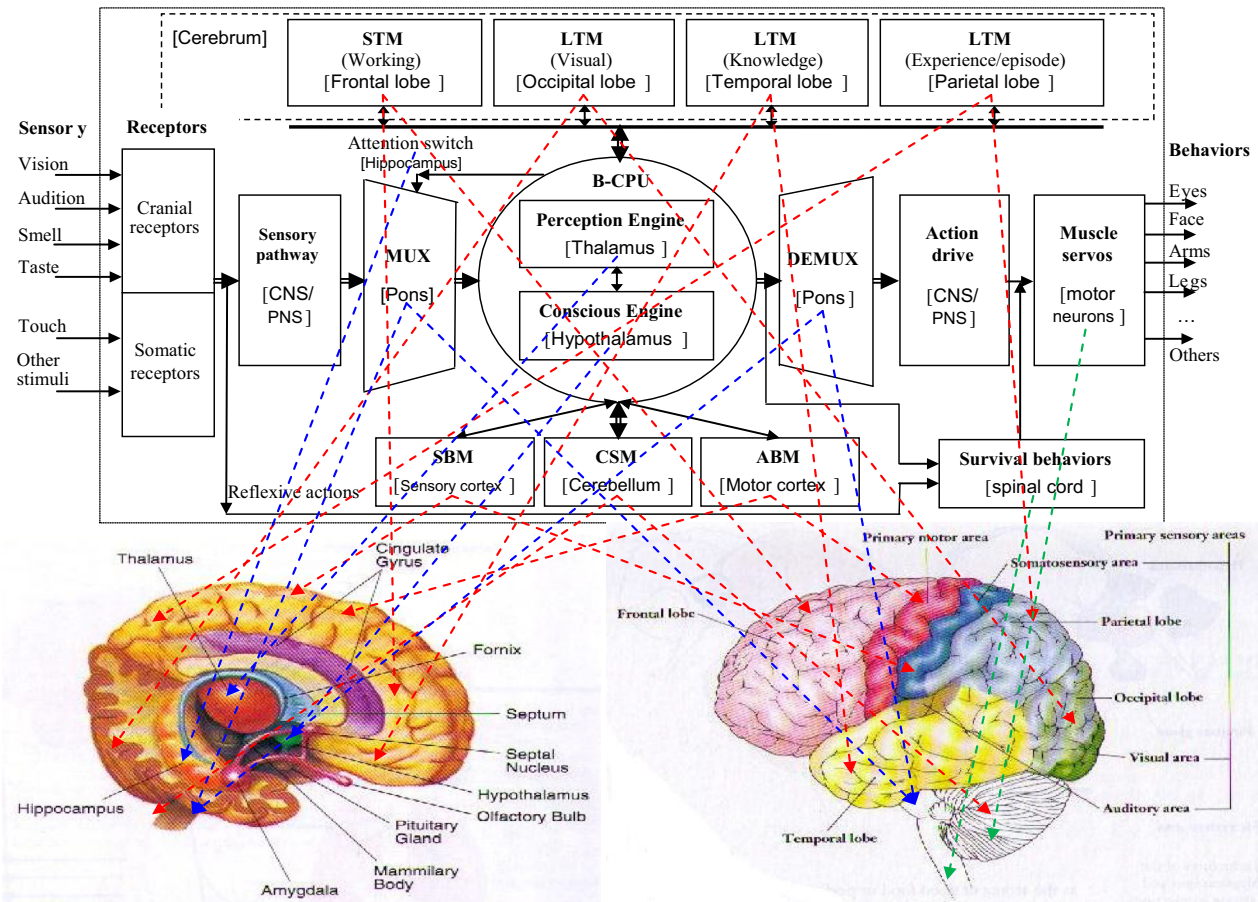
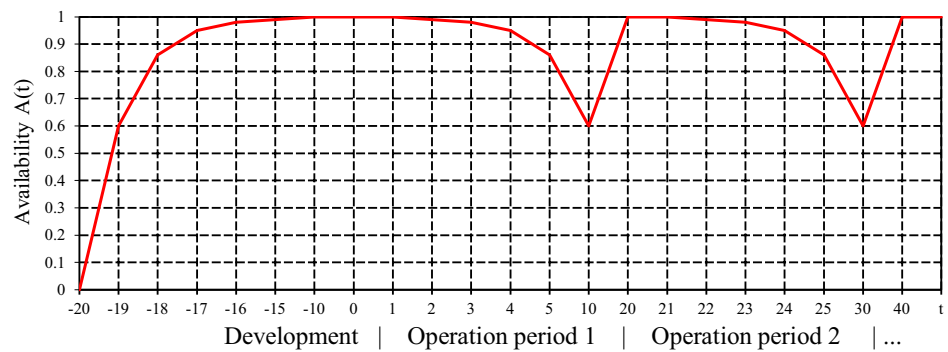


Fig. 8 The abstract intelligence model of the brain (AIMB) and the neurophysiological structures of the brain

brain science were used to explain that the brain works in a certain way based on empirical observations of corresponding activities in usually overlapped brain areas. However, the lack of precise models and rigorous causality in brain studies has dissatisfied the formal expectations of computer scientists and mathematicians, because a computer, the logical counterpart of the brain, might not be explained in such a vague and empirical approach without the support of a formal theory and a rigorous means according to the *abstract intelligence* (αI) theory [65].

The analytical and formal models of the brain developed in cognitive informatics enable a systematic modeling

of the brain from the facets of cognitive informatics, brain informatics, and neuroinformatics. These models lead to the development of a coherent αI theory in order to rigorously explain the underpinning principles and mechanisms of the brain based on denotational mathematical means and brain science observations.

According to the abstract system theory and the αI theory, the *abstract intelligence model of the brain* (AIMB) is developed as a natural intelligent system as shown in Fig. 8 [65, 69]. The intelligent system of the brain as modeled in AIMB encompasses the subsystems of processor, memory, consciousness monitor, sensory, and motor, which modeled in

different color schemes. Particularly, the intelligent processor subsystem is highlighted by the blue links, memories in red, and other categories in green. AIMB explains the key organs of the brain and their cognitive functions as allocated in different regions of the brain as well as of cerebrum and cerebellum cortexes. On the basis of AIMB, a systematical model and mapping between the logical and neurophysiological models of the brain are enabled. The reductive mapping of AIMB creates cognitive connections between the logical functions and the neural structures of the entire brain system.

As that a computer may only be explained via the mapping of computer theories in logical models and the low-level implementations in integrated electronic circuits, the approach to explore the brain and the natural intelligence may be achieved by the systematical theories of α I that enables the top-down reduction and bottom-up induction on the hierarchical structures and functions of the brain.

The complex brain system as configured in the AIMB model can be formally described as given in Eq. 47. In Eq. 47, || denotes parallel subsystems and/or components, and // represents the corresponding physiological organs or cortical lobes in the brain. The system model of the brain, AIMB, described in Eq. 47 rationally explains the natural structures and cognitive functions of the brain, as well as their relationships and interactions. A set of conventionally overlapped, redundant, and even contradicted empirical observations in brain studies and cognitive psychology may be clarified based on the α I theory and the system model of AIMB as a logical intelligent system. As indicated in Fig. 8, the exploration of the brain is a complicated recursive problem where abstract system theory and contemporary denotational mathematics [53,61,66] are needed to efficiently deal with the complex system.

$$\begin{aligned}
 \text{AIMB} \triangleq & \text{ Perception Engine} && // \text{ Thalamus} \\
 & || \text{ Conscious/Attention Engine} && // \text{ Hypothalamus/pons} \\
 & || \text{ Inference Engine} && // \text{ Thalamus} \\
 & || \text{ Memory (STM (working)} && // \text{ Frontal lobe} \\
 & \quad || \text{ LTM (knowledge)} && // \text{ Temporal lobe} \\
 & \quad || \text{ LTM (vision)} && // \text{ Occipital lobe} \\
 & \quad || \text{ LTM (episode)} && // \text{ Parietal lobe} \\
 & \quad || \text{ STM (knowledge)} && // \text{ Temporal l} \\
 & \quad || \text{ CSM (knowledge)} && // \text{ Temporal l} \\
 & \quad || \text{ ABM (knowledge)} && // \text{ Temporal l} \\
 &) && \\
 & || \text{ Behavior (Basic Actions} && // \text{ Motors of eyes, face, body, limb} \\
 & \quad || \text{ Reflexive Actions} && // \text{ Un-/sub-conscious behaviors} \\
 & \quad || \text{ Skills} && // \text{ Permanently acquired behaviors} \\
 & \quad || \text{ Cognitive Behaviors} && // \text{ Temporarily generated behaviors} \\
 &) && \\
 & || \text{ Sensory (Primary Sensory} && // \text{ Vision, hearing, smell, taste, touch} \\
 & \quad || \text{ Complex Sensory} && // \text{ Spatiality, time, motion, gravity, equilibrium, posture} \\
 & \quad || \text{ Internal Sensory} && // \text{ Hungry, thirsty, breathing, excretion, pain} \\
 &) &&
 \end{aligned} \tag{47}$$

The AIMB model of the brain establishes a top-level system model that efficiently reduces the extremely intricate complexity of the structural and functional models of the brain. The system model of the brain enables the development of *cognitive computers* that perceive, think, inference, and learn mimicking the brain [56,61,64]. The functional and theoretical difference between cognitive computers and classic computers are that the latter are data processors based on Boolean algebra and its logical counterparts; while the former are knowledge processors based on contemporary denotational mathematics. A wide range of applications of cognitive computers have been developing in ICIC (<http://www.ucalgary.ca/icic/>) such as, *inter alia*, cognitive robots [56], cognitive learning engines [68,81,89], cognitive Internet [58], cognitive agents [54], cognitive search engines [58], cognitive translators [80,81], cognitive control systems, and cognitive automobiles as contemporary paradigms of highly complex intelligent systems.

The complex system model of the capacity of human memory

One of the key wonders in brain science, neurology, and cognitive informatics is what the capacity of human memory is in the brain as a complex cognitive system. The *memory model of the brain* (MMB) [69] and the *object-attribute-relation* (OAR) model [43] of long-term memory (LTM) provide insights for the system modeling of human memory.

Definition 27 The *OAR model* of LTM can be described as a triple, i.e.:

$$OAR \hat{=} (O, A, R), \tag{48}$$

where O is a set of objects representing concrete entities or abstract artefacts, A is a set of attributes for characterizing the objects, and R is a set of relations between the object and its attributes or other objects.

An illustration of the OAR model between two objects is shown in Fig. 9 where $O = \{O_1, O_2\}$, $A_1 = \{a_{11}, a_{12}, \dots, a_{1m}\}$, $A_2 = \{a_{21}, a_{22}, \dots, a_{2n}\}$, and $R = \{(O_1, O_2), (O_2, O_1), (A_1, A_2), (A_2, A_1), (O_1, A_1), (O_2, A_2)\}$. It is noteworthy according to the OAR model that the *relations* themselves represent information and knowledge in the brain. The relational metaphor is totally different from the traditional container metaphor in neuropsychology and computer science, because the latter perceive that memory and knowledge are *stored* in individual neurons and the neurons function as containers.

According to the OAR model as shown in Fig. 9, information or acquired knowledge is represented in the brain by relations implemented via a synaptic connection between

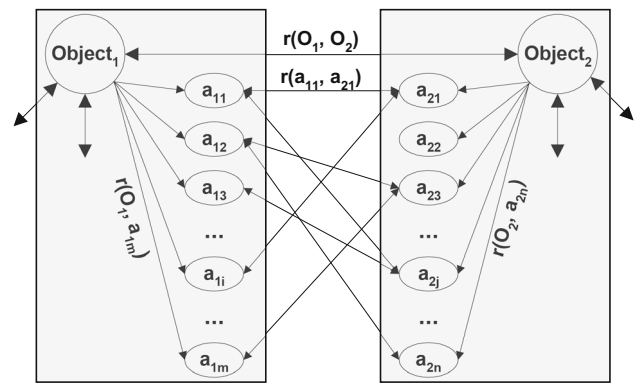


Fig. 9 The OAR model of human memory system

neurons. Therefore, the capacity of human memory is not only dependent on the number of neurons, but also the connections among them. This mechanism may result in an exponential combination to represent and store information in LTM of the brain. This also explains why the magnitude of neurons in an adult brain would seem to be stable; however, huge amount of information can be remembered throughout the entire life of a person.

Definition 28 Assume there are n neurons in the brain, and in average there are k connections between a given neuron and the rest of them. The *magnitude of the brain memory capacity*, M_b , is determined by a mathematical combination of all potential subgroups of k connections out of the base of the total number of neurons n , i.e.:

$$M_b \hat{=} C_n^k = \frac{n!}{k!(n-k)!} \tag{49}$$

Equation 49 indicates that the memory capacity problem in cognitive science and neurology can be reduced to a typical system problems solved by mathematical combinatory.

According to the empirical data about the human memory system observed in neurology and cognitive science [23,27], the total number of neurons in the brain is $n = 10^{11}$, and the average synaptic connections among the neurons are $k = 10^3$. Then, the problem is calibrated as follows for estimating the upper bound of the capacity of human memory:

$$M_b = C_n^k = \frac{n!}{k!(n-k)!} = \frac{10^{11}!}{10^3!(10^{11} - 10^3)!} \tag{50}$$

However, the problem is yet still very hard to be solved analytically because the factorials in the mathematical model are too huge to be calculated by any modern computer. Observe the nature of the problem, it is found that Eq. 50 can be

reduced to a lower order computational problem when logarithmic operations are taken at both sides of the equation, i.e.:

$$\begin{aligned} \log(M_b) &= \log(n!) - \log(k!) - \log((n-k)!) \\ &= \log\left(\prod_{i=n-k+1}^n i\right) - \log(k!) \\ &= 11,000 - 2,567.6 \\ &= 8,432.4 \end{aligned} \quad (51)$$

The solution is then obtained by a numerical program designed in MATLAB. The result reveals the following property of human memory capacity.

Corollary 19 *The capacity of human memory is bounded by 10^{8432} bits, i.e.:*

$$\begin{aligned} M_b &= C_n^k = \frac{n!}{k!(n-k)!} \\ &= \frac{10^{11}!}{10^3!(10^{11} - 10^3)!} \\ &= 10^{8432} \end{aligned} \quad (52)$$

The finding on the magnitude of the human memory capacity on the order as high as 10^{8432} bits reveals an interesting mechanism of the brain as a complex system. That is, the brain does not create new neurons to represent new information, instead it generates new synapses between existing neurons in order to represent new information. The observation in neurophysiology that the number of neurons is kept stable rather than continuously increasing in adult brains [23,27] provides empirical evidence for the relational cognitive model of information representation in the complex system of human memory.

The tremendous difference of memory magnitudes between human beings and computers demonstrates the efficiency of information representation, storage, and processing in the complex system of human brains. Computers store data in a direct and unconsumed manner; while the brain stores information by relational neural clusters. The former can be accessed directly by explicit addresses and can be sorted; while the latter may only be retrieved by content-sensitive search and matching among neuron clusters with tree-form organization where spatial connections and configurations themselves represent information in such a partially connected memory system.

Conclusions

A theoretical framework of system science and engineering has been explored. Systems have been recognized as the most complicated entities and phenomena in abstract,

physical, information, cognitive, brain, and social worlds across almost all science and engineering disciplines. The abstract system theory has been presented based on a survey on the latest advances in system algebra, complex system, and system engineering. The insights of fundamental studies on system science have been formally described based on the theories of abstract systems and the modeling of various concrete systems in system engineering. On the basis of the mathematical model of abstract systems and system algebra, system theories have been embodied by a set of formal structures, properties, behaviors, and principles. The system philosophy and methodology for efficiently reducing system structural and behavioral complexities have been formally explained in system representation, modeling, analysis, synthesis, and inference. Applications of the abstract system theory have been demonstrated and explored in complex intelligent systems in the contexts of system engineering, intelligent engineering, cognitive informatics, brain systems, cognitive robotics, software engineering, cognitive linguistics, and cognitive systems.

Acknowledgments I would like to thank the Editor-in-Chief, Prof. Janusz Kacprzyk, for his pioneer work and leadership in system science as well as his kind invitation and advice to this article. The author acknowledges the support in part of a discovery fund granted by the Natural Sciences and Engineering Research Council of Canada (NSERC). The author would like to thank the anonymous reviewers for their valuable suggestions and comments on the previous version of this paper.

References

1. Aristotle (384 BC–322 BC) (1989) *Prior analytics* (translated by Robin Smith, Hackett)
2. Ashby WR (1958) Requisite variety and implications for control of complex systems. *Cybernetica* 1:83–99
3. Ashby WR (1962) Principles of the self-organizing system. In: von Foerster H, Zopf G (eds) *Principles of self-organization*. Pergamon, Oxford, pp 255–278
4. Bender EA (2000) *Mathematical methods in artificial intelligence*. IEEE CS Press, Los Alamitos
5. Boulding K (1956) General systems theory—the skeleton of science. *Gen Syst Yearb* 1:11–17
6. Bunge M (1978) General systems theory challenge to classical philosophy of science. *Int J Gen Syst* 4(1):3–28
7. Carter R, Aldridge S, Page M, Parker S (2009) *The human brain*. Dorling Kindersley Ltd., New York
8. Castillo O, Melin P, Kacprzyk J (2013) *Recent advances on hybrid intelligent systems*. Springer, Berlin
9. Cutnell JD, Johnson KW (1998) *Physics*. Wiley, New York
10. Descartes R (1979) *Meditations on first philosophy*. D. Cress trans., Indianapolis: Hackett Publishing Co., Inc
11. Eigen M, Schuster P (1979) *The hypercycle: a principle of natural self-organization*. Springer, Berlin
12. Ellis DO, Fred JL (1962) *Systems philosophy*. Prentice-Hall, Englewood Cliffs
13. Ford J (1986) *Chaos: solving the unsolvable, predicting the unpredictable*. In: Barnsley MF, Demko SG (eds) *Chaotic dynamics and fractals*. Academic Press, New York

14. Gaines BR (1984) Methodology in the large: modeling all there is. *Syst Res* 1(2):91–103
15. Haken H (1977) *Synergetics*. Springer, New York
16. Hall AS, Fagan RE (1956) Definition of system. *Gen Syst Yearb* 1:18–28
17. Hassanién AE, Azar AT, Snašel V, Kacprzyk J (2015) *Big data in complex systems: challenges and opportunities*. Springer, Berlin
18. Heylighen F (1989) Self-organization, emergence and the architecture of complexity. In: *Proceedings of the 1st European conference on system science (AF CET)*, Paris, pp 23–32
19. Kacprzyk J (1997) *Multistage fuzzy control: a model-based approach to control and decision-making*. Wiley, Chichester
20. Klir GJ (1992) *Facets of systems science*. Plenum, New York
21. Kotulak R (1997) *Inside the brain*. Andrews McMeel Publishing Co., Kansas City
22. Makridakis S, Faucheux C (1973) Stability properties of general systems. *Gen Syst Yearb* 18:3–12
23. Marieb EN (1992) *Human anatomy and physiology*, 2nd edn. Benjamin Cummings Publishing Co., USA
24. Negoita CV (1989) Review: fuzzy sets, uncertainty, and information. *Cybernetes* 18(1):73–74
25. Newton I (1729) *The principia: the mathematical principles of natural philosophy*. Benjamin Motte, London
26. Pedrycz W (1981) On approach to the analysis of fuzzy systems. *Int J Control* 34:403–421
27. Pinel JPJ (1997) *Biopsychology*, 3rd edn. Allyn and Bacon, Needham Heights
28. Prigogine I, Nicolis G (1972) Thermodynamics of evolution. *Phys Today* 25:23–28
29. Rapoport A (1962) Mathematical aspects of general systems theory. *Gen Syst Yearb* 11:3–11
30. Russell B (1903) *The principles of mathematics*. W.W. Norton & Co., London
31. Schedrovitzk GP (1962) Methodological problems of systems research. *Gen Syst Yearb* 11:27–53
32. Simon H (1965) Architecture of complexity. *Gen Syst Yearb* 10:63–76
33. Skarda CA, Freeman WJ (1987) How brains make chaos into order. *Behav Brain Sci* 10(2):161–173
34. Smith A (1776) *An inquiry into the nature and causes of the wealth of nations*, vol. 1, 2. W. Strahan and T. Cadell, London
35. Takahara Y, Takai T (1985) Category theoretical framework of general systems. *Int J Gen Syst* 11(1):1–33
36. von Bertalanffy L (1952) *Problems of life: an evolution of modern biological and scientific thought*. C.A. Watts, London
37. Wang Y (2002) The real-time process algebra (RTPA). *Ann Softw Eng* 14:235–274
38. Wang Y (2003) On cognitive informatics. *Brain Mind* 4(2):151–167
39. Wang Y (2003) Using process algebra to describe human and software system behaviors. *Brain Mind* 4(2):199–213
40. Wang Y (2005) System science models of software engineering. In: *Proceedings of the 18th Canadian conference on electrical and computer engineering (CCECE'05)*, Saskatoon, SA, Canada, May 1–4, pp 1802–1805
41. Wang Y (2006) On the informatics laws and deductive semantics of software. *IEEE Trans Syst Man Cybern (C)* 36(2):161–171
42. Wang Y (2007) *Software engineering foundations: a software science perspective*. CRC series in software engineering, vol II. Auerbach Publications, New York
43. Wang Y (2007) The OAR model of neural informatics for internal knowledge representation in the brain. *Int J Cogn Inform Nat Intell* 1(3):64–75
44. Wang Y (2007) *The theoretical framework of cognitive informatics*. *Int J Cogn Inform Nat Intell (IJCINI)*, IPI Publishing, USA, 1(1):1–27
45. Wang Y (2008) On system algebra: a denotational mathematical structure for abstract system modeling. *Int J Cogn Inform Nat Intell* 2(2):20–42
46. Wang Y (2008) On contemporary denotational mathematics for computational intelligence. *Trans Comput Sci* 2:6–29
47. Wang Y (2008) RTPA: a denotational mathematics for manipulating intelligent and computing behaviors. *Int J Cogn Inform Nat Intell* 2(2):44–62
48. Wang Y (2008) Deductive semantics of RTPA. *Int J Cogn Inform Nat Intell*, IGI Publishing, USA 2(2):95–121
49. Wang Y (2008) On the big-R notation for describing iterative and recursive behaviors. *Int J Cogn Inform Nat Intell* 2(1):17–23
50. Wang Y (2008) On concept algebra: a denotational mathematical structure for knowledge and software modeling. *Int J Cogn Inform Nat Intell* 2(2):1–19
51. Wang Y (2009) On abstract intelligence: toward a unified theory of natural, artificial, machinable, and computational intelligence. *Int J Softw Sci Comput Intell* 1(1):1–17
52. Wang Y (2009) On cognitive computing. *Int J Softw Sci Comput Intell* 1(3):1–15
53. Wang Y (2009) Paradigms of denotational mathematics for cognitive informatics and cognitive computing. *Fundam Inform* 90(3):282–303
54. Wang Y (2009) A cognitive informatics reference model of autonomous agent systems (AAS). *Int J Cogn Inform Nat Intell* 3(1):1–16
55. Wang Y (2009) Toward formal models of the theoretical framework of fundamental economics. *Fundam Inform* 90(4):443–459
56. Wang Y (2010) Cognitive robots: a reference model towards intelligent authentication. *IEEE Robot Autom* 17(4):54–62
57. Wang Y (2010) On formal and cognitive semantics for semantic computing. *Int J Semant Comput* 4(2):203–237
58. Wang Y (2010) Keynote: cognitive computing and world wide wisdom (WWW+). In: *Proceedings of the 9th IEEE international conference on cognitive informatics (ICCI'10)*, Tsinghua Univ., Beijing. IEEE CS Press, London, pp 4–5
59. Wang Y (2011) Inference algebra (IA): a denotational mathematics for cognitive computing and machine reasoning (I). *Int J Cogn Inf Nat Intell* 5(4):61–82
60. Wang Y (2011) On cognitive models of causal inferences and causation networks. *Int J Softw Sci Comput Intell* 3(1):50–60
61. Wang Y (2012) On denotational mathematics foundations for the next generation of computers: cognitive computers for knowledge processing. *J Adv Math Appl* 1(1):118–129
62. Wang Y (2012) Inference algebra (IA): a denotational mathematics for cognitive computing and machine reasoning (II). *Int J Cogn Inf Nat Intell* 6(1):21–46
63. Wang Y (2012) On long lifespan systems and applications. *J Comput Theor Nanosci* 9(2):208–216
64. Wang Y (2012) Keynote: towards the next generation of cognitive computers: knowledge vs. data computers. In: *Proceedings of the 12th international conference on computational science and applications (ICCSA'12)*, Salvador, Brazil. Springer, Berlin, pp 18–21
65. Wang Y (2012) On abstract intelligence and brain informatics: mapping cognitive functions of the brain onto its neural structures. *Int J Cogn Inf Nat Intell* 6(4):54–80
66. Wang Y (2012) In search of denotational mathematics: novel mathematical means for contemporary intelligence, brain, and knowledge sciences. *J Adv Math Appl* 1(1):4–25
67. Wang Y (2012) The cognitive mechanisms and formal models of consciousness. *Int J Cogn Inf Nat Intell* 6(2):23–40
68. Wang Y (2013) On semantic algebra: a denotational mathematics for cognitive linguistics, machine learning, and cognitive computing. *J Adv Math Appl* 2(2) (in press)

69. Wang Y (2013) Neuroinformatics models of human memory: mapping the cognitive functions of memory onto neurophysiological structures of the brain. *Int J Cogn Inf Nat Intell* 7(1):98–122
70. Wang Y (2013) Formal models and cognitive mechanisms of the human sensory system. *Int J Softw Sci Comput Intell* 5(3):49–69
71. Wang Y (2014) Keynote: latest advances in neuroinformatics and fuzzy systems. In: Proceedings of 2014 international conference on neural networks and fuzzy systems (ICNF-FS'14), Venice, Italy, pp 14–15
72. Wang Y (2014) Fuzzy causal inferences based on fuzzy semantics of fuzzy concepts in cognitive computing. *WSEAS Trans Comput* 13:430–441
73. Wang Y (2014) Keynote: from information revolution to intelligence revolution. In: Proceedings of the 13th IEEE international conference on cognitive informatics and cognitive computing (ICCI*CC 2014). IEEE CS Press, London, pp 3–5
74. Wang Y (2014) Fuzzy causal patterns of humor and jokes for cognitive and affective computing. *Int J Softw Sci Comput Intell* 8(2):33–45
75. Wang Y (2014) Towards a theory of fuzzy probability for cognitive computing. In: Proceedings of the 13th IEEE international conference on cognitive informatics and cognitive computing (ICCI*CC 2014). IEEE CS Press, London, pp 19–28
76. Wang Y (2014) On granular algebra: a denotational mathematics for modeling granular systems and granular computing. *J Adv Math Appl* 3(1):60–73
77. Wang Y (2015) A mathematical theory of system science: system algebra for formal system modeling and manipulations. *J Adv Math Appl* 4(2) (in press)
78. Wang Y, Tan X (2011) The formal design models of tree architectures and behaviors. *Int J Softw Sci Comput Intell* 3(4):84–108
79. Wang Y, Fariello G (2012) On neuroinformatics: mathematical models of neuroscience and neurocomputing. *J Adv Math Appl* 1(2):206–217
80. Wang Y, Berwick RC (2012) Towards a formal framework of cognitive linguistics. *J Adv Math Appl* 1(2):250–263
81. Wang Y, Berwick RC (2013) Formal relational rules of english syntax for cognitive linguistics, machine learning, and cognitive computing. *J Adv Math Appl* 2(2):182–195
82. Wang Y, Wiebe, VJ (2014) Big data analyses for collective opinion elicitation in social networks. In: Proceedings of IEEE 2014 international conference on big data science and engineering (BDSE'14), Beijing, China, pp 630–637
83. Wang Y, Tian Y (2013) A formal knowledge retrieval system for cognitive computers and cognitive robotics. *Int J Softw Sci Comput Intell* 5(2):37–57
84. Wang Y, Wang Y (2006) Cognitive informatics models of the brain. *IEEE Trans Syst Man Cybern (Part C)*, 36(2):203–207
85. Wang Y, Zhang D, Kinsner W (eds) (2010) Advances in cognitive informatics and cognitive computing. *SCI*, vol 323. Springer, Berlin
86. Wang Y, Zadeh LA, Yao Y (2009) On the system algebra foundations for granular computing. *Int J Softw Sci Comput Intell, IGI, USA* 1(1):64–86
87. Wang Y, Kinsner W, Zhang D (2009) Contemporary cybernetics and its faces of cognitive informatics and computational intelligence. *IEEE Trans Syst Man Cybern (Part B)* 39(4):1–11
88. Wang Y, Kinsner W, Anderson JA, Zhang D, Yao Y, Sheu P, Tsai J, Pedrycz W, Latombe J-C, Zadeh LA, Patel D, Chan C (2009) A doctrine of cognitive informatics. *Fundam Inform* 90(3):203–228
89. Wang Y, Tian Y, Hu K (2011) Semantic manipulations and formal ontology for machine learning based on concept algebra. *Int J Cogn Inf Nat Intell* 5(3):1–29
90. Wang Y, Wang Y, Patel S, Patel D (2006) A layered reference model of the brain (LRMB). *IEEE Trans Syst Man Cybern (Part C)*, 36(2):124–133
91. Zadeh LA (1965) Fuzzy sets and systems. In: Fox J (ed) *Systems theory*. Polytechnic Press, Brooklyn, pp 29–37
92. Zadeh LA (1973) Outline of a new approach to analysis of complex systems. *IEEE Trans Syst Man Cybern* 1(1):28–44
93. Zadeh LA (1982) Fuzzy systems theory: framework for analysis of buerocratic systems. In: Cavallo RE (ed) *System methods in social science research*. Kluwer-Nijhoff, Boston, pp 25–41
94. Zadrozny S, Kacprzyk J (2006) Computing with words for text processing: an approach to the text categorization. *Inf Sci* 176:415–437