

# Chapter 7.26

## On System Algebra: A Denotational Mathematical Structure for Abstract System Modeling

**Yingxu Wang**  
*University of Calgary, Canada*

### ABSTRACT

Systems are the most complicated entities and phenomena in abstract, physical, information, and social worlds across all science and engineering disciplines. System algebra is an abstract mathematical structure for the formal treatment of abstract and general systems as well as their algebraic relations, operations, and associative rules for composing and manipulating complex systems. This article presents a mathematical theory of system algebra and its applications in cognitive informatics, system engineering, software engineering, and cognitive informatics. A rigorous treatment of abstract systems is described, and the algebraic relations and compositional operations of abstract systems are analyzed. System algebra provides a denotational mathematical means that can be used to model, specify, and manipulate generic “to be” and “to have” type problems, particularly system architectures and high-level system designs, in computing, software

engineering, system engineering, and cognitive informatics.

### INTRODUCTION

Systems are the most complicated entities and phenomena in abstract, physical, information, and social worlds across all science and engineering disciplines. Systems are needed because the physical and/or cognitive power of an individual component or a person is not enough to carry out a work or solve a problem. System philosophy intends to treat everything as a system, and it perceives that a system always belongs to other super system(s) and contains more subsystems.

The system concept can be traced back to the 17th century, when René Descartes (1596-1650) noticed the interrelationships among scientific disciplines as a system. The general system notion was then proposed by Ludwig von Bertalanffy in the 1920s (von Bertalanffy, 1952; Ellis & Fred,

1962). The theories of system science have evolved from classic theories (Ashby, 1958, 1962; Ellis & Fred, 1962; Heylighen, 1989; G. J. Klir, 1992; R. G. Klir, 1988; Rapoport, 1962) to contemporary theories in the mid-20th century, such as I. Prigogine's dissipative structure theory (Prigogine et al., 1972), H. Haken's synergetics (Haken, 1977), and Eigen's hypercycle theory (Eigen & Schuster, 1979). Then, during the late part of the last century, there are proposals of complex systems theories (G. J. Klir, 1992; Zadeh, 1973), fuzzy theories (Zadeh, 1965, 1973), and chaos theories (Ford, 1986; Skarda & Freeman, 1987).

System algebra is an abstract mathematical structure for the formal treatment of abstract and general systems as well as their algebraic relations, operations, and associative rules for composing and manipulating complex systems. System algebra (Wang, 2005, 2006a, 2006b, 2007a, 2007c) presented in this article is the latest attempt to provide a formal and rigorous treatment of abstract systems and their properties. This article treats systems as a mathematic entity and it studies the generic rules and theories of abstract systems. A new mathematical structure of abstract systems as the most complicated mathematical entities beyond sets, functions, and processes is presented. Properties of abstract systems are modeled and analyzed. System algebra is introduced as a set of relational and compositional operations for manipulating abstract systems and their composing rules. The relational operations of system algebra are described encompassing *independent*, *related*, *overlapped*, *equivalent*, *subsystem*, and *super-system*. The compositional operations of system algebra are explored encompassing *inheritance*, *tailoring*, *extension*, *substitute*, *difference*, *composition*, *decomposition*, *aggregation*, and *specification*. A wide range of applications of system algebra are identified in cognitive informatics, system science, system engineering, computing, software engineering, and intelligent systems.

## THE ABSTRACT SYSTEM THEORY

This section demonstrates that systems may be treated rigorously as a new mathematical structure beyond conventional mathematical entities. Based on this view, the concept of abstract systems and their mathematical models are introduced.

**Definition 1.** *An abstract system is a collection of coherent and interactive entities that has stable functions and a clear boundary with the external environment.*

An abstract system forms the generic model of various real-world systems and represents the most common characteristics and properties of them.

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

**Definition 2.** *Let  $\mathcal{C}$  be a finite or infinite nonempty set of components, and  $\mathcal{B}$  a finite or infinite nonempty set of behaviors, then the universal system environment  $U$  is denoted as a triple, i.e.:*

$$\mathfrak{U} \triangleq (\mathcal{C}, \mathcal{B}, \mathcal{R}) \\ = \mathcal{R} : \mathcal{C} \rightarrow \mathcal{C} \mid \mathcal{C} \rightarrow \mathcal{B} \mid \mathcal{B} \rightarrow \mathcal{C} \mid \mathcal{B} \rightarrow \mathcal{B} \quad (1)$$

where  $\mathcal{R}$  is a set of relations between  $\mathcal{C}$  and  $\mathcal{B}$ , and  $\mid$  demotes alternative relations.

Abstract systems can be classified into two categories known as the *closed* and *open* systems. Most practical and useful systems in nature are open systems in which there are interactions between the system and its environment. However, in order to develop the theoretical framework of abstract systems, the closed systems in which there is no interaction with the external environment will be modeled first in the following subsection.

24 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/system-algebra-denotational-mathematical-structure/29551](http://www.igi-global.com/chapter/system-algebra-denotational-mathematical-structure/29551)

## Related Content

---

### HOD2MLC: Hybrid Ontology Design and Development Model With Lifecycle

Rishi Kanth Saripalle, Steven A. Demurjian, Michael Blechner and Thomas Agresta (2018). *Application Development and Design: Concepts, Methodologies, Tools, and Applications* (pp. 659-685).

[www.irma-international.org/chapter/hod2mlc/188228](http://www.irma-international.org/chapter/hod2mlc/188228)

### Learning to Innovate: Methodologies, Tools, and Skills for Software Process Improvement in Spain

Félix A. Barrio and Raquel Poy (2014). *Agile Estimation Techniques and Innovative Approaches to Software Process Improvement* (pp. 272-297).

[www.irma-international.org/chapter/learning-to-innovate/100283](http://www.irma-international.org/chapter/learning-to-innovate/100283)

### Gamification: Model-Driven Engineering Approaches

Pedro Aguiar and Isabel Azevedo (2020). *Software Engineering for Agile Application Development* (pp. 141-163).

[www.irma-international.org/chapter/gamification/250440](http://www.irma-international.org/chapter/gamification/250440)

### Power-Aware Mechanism for Scheduling Scientific Workflows in Cloud Environment

Kirankumar V. Kataraki and Sumana Maradithaya (2021). *International Journal of Information System Modeling and Design* (pp. 22-38).

[www.irma-international.org/article/power-aware-mechanism-for-scheduling-scientific-workflows-in-cloud-environment/273225](http://www.irma-international.org/article/power-aware-mechanism-for-scheduling-scientific-workflows-in-cloud-environment/273225)

### Design and Integration of JMF-Based Multimedia-Supported Environment in a Product Data Management System

Pengfei Zeng (2020). *International Journal of Information System Modeling and Design* (pp. 28-51).

[www.irma-international.org/article/design-and-integration-of-jmf-based-multimedia-supported-environment-in-a-product-data-management-system/255111](http://www.irma-international.org/article/design-and-integration-of-jmf-based-multimedia-supported-environment-in-a-product-data-management-system/255111)