

Chapter 7.19

RTPA:

A Denotational Mathematics for Manipulating Intelligent and Computational Behaviors

Yingxu Wang

University of Calgary, Canada

ABSTRACT

Real-time process algebra (RTPA) is a denotational mathematical structure for denoting and manipulating system behavioral processes. RTPA is designed as a coherent algebraic system for intelligent and software system modeling, specification, refinement, and implementation. RTPA encompasses 17 metaprocesses and 17 relational process operations. RTPA can be used to describe both logical and physical models of software and intelligent systems. Logic views of system architectures and their physical platforms can be described using the same set of notations. When a system architecture is formally modeled, the static and dynamic behaviors performed on the architectural model can be specified by a three-level refinement scheme at the system, class, and object levels in a top-down approach. RTPA has been successfully applied in real-world system modeling and code generation for software

systems, human cognitive processes, and intelligent systems.

INTRODUCTION

The modeling and refinement of software and intelligent systems require new forms of denotational mathematics that possess enough expressive power for rigorously describing system architectures and behaviors. Real-time process algebra (RTPA) is a new denotational mathematical structure for software system modeling, specification, refinement, and implementation for both real-time and nonreal-time systems (Wang, 2002b, 2007d; Wang & King, 2000). An important finding in formal methods is that a software system can be perceived and described as the *composition* of a set of *processes*, which are constructed on the basis of algebraic operations. Theories on process algebras can be traced back

to Hoare's *Communicating Sequential Processes* (CSP) (Hoare, 1978, 1985) and Milner's *Calculus of Communicating Systems* (CCS) (Milner, 1980). The *process metaphor* of software systems has evolved from concurrent processes to real-time process systems in the area of operating system research and formal methods (Boucher & Gerth, 1987; Hoare, 1978, 1985; Milner, 1980, 1989; Reed & Roscoe, 1986; Schneider, 1991).

Definition 1. *A process is an abstract model of a unit of meaningful system behaviors that represents a transition procedure of the system from one state to another by changing values of the sets of inputs, outputs, and/or internal variables.*

It is recognized that generic computing problems are 3-D problems, known as those of the behavior, space, and time dimensions, which require a denotational mathematical means for addressing the requirements in all dimensions, particularly the time dimension (Wang, 2002b, 2003b, 2007d). However, conventional process models are hybrid (Hoare, 1985), which do not distinguish the concepts of fundamental metaprocesses and process operations between them. The CSP notation models a major part of elementary software behaviors that may be used in system specification and description. However, it lacks many useful processes that are perceived essential in system modeling, such as addressing, memory manipulation, timing, and system dispatch. CSP models all input and output (I/O) as abstract channel operations that are not expressive enough to denote complex system interactions, particularly for those of real-time systems.

A number of timed extensions and variations of process algebra have been proposed (Baeten & Bergstra, 1991; Boucher & Gerth, 1987; Cerone, 2000; Corsetti, Montanari, & Ratto, 1991; Dierks, 2000; Fecher, 2001; Gerber, Gunter, & Lee, 1992; Jeffrey, 1992; Klusener, 1992; Nicollin & Sifakis, 1991; Reed & Roscoe, 1986; Schneider, 1991; Vereijken, 1995). It is found that

the existing work on process algebra and their timed variations can be extended and refined to a new form of denotational mathematics, RTPA, based on a set of algebraic process operations and laws (Wang, 2002b, 2003b, 2006a, 2006c, 2007d, 2008a). RTPA can be used to formally and precisely describe and specify architectures and behaviors of software systems on the basis of algebraic process notations and rules.

Definition 2. *A process P in RTPA is a composed component of n metastatements s_i and s_j , $1 \leq i < n$, $j = i + 1$, according to certain composing relations r_{ij} , i.e.:*

$$P = (\dots((s_1) r_{12} s_2) r_{23} s_3) \dots r_{n-1,n} s_n) \quad (1)$$

where $r_{ij} \in \mathfrak{R}$, which is a set of relational process operators of RTPA that will be formally defined in Lemma 2.

Definition 2 indicates that the mathematical model of a process is a cumulative relational structure among computing operations. The simplest process is a single computational statement. Further discussion will be provided in the section on the unified mathematical model of programs.

Definition 3. *RTPA is a denotational mathematical structure for algebraically denoting and manipulating system behavioral processes and their attributes by a triple, i.e.:*

$$RTPA \triangleq (\mathfrak{T}, \mathfrak{P}, \mathfrak{R}) \quad (2)$$

where \mathfrak{T} is a set of 17 primitive types for modeling system architectures and data objects, \mathfrak{P} is a set of 17 metaprocesses for modeling fundamental system behaviors, and \mathfrak{R} is a set of 17 relational process operations for constructing complex system behaviors.

18 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/denotational-mathematics-manipulating-intelligent-computational/29544

Related Content

A Software Cost Model to Assess Productivity Impact of a Model-Driven Technique in Developing Domain-Specific Design Tools

Achilleas Achilleos, Nektarios Georgalas, Kun Yang and George A. Papadopoulos (2011). *Modern Software Engineering Concepts and Practices: Advanced Approaches* (pp. 333-355).

www.irma-international.org/chapter/software-cost-model-assess-productivity/51979

Formal Modeling and Verification of Security Property in Handel C Program

Yujian Fu, Jeffery Kulick, Lok K. Yan and Steven Drager (2012). *International Journal of Secure Software Engineering* (pp. 50-65).

www.irma-international.org/article/formal-modeling-verification-security-property/69393

A Semantic Web-Based Information Integration Approach for an Agent-Based Electronic Market

Maria João Viamonte and Nuno Silva (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 1458-1477).

www.irma-international.org/chapter/semantic-web-based-information-integration/29458

The Case for Privacy Awareness Requirements

Inah Omoronyia (2016). *International Journal of Secure Software Engineering* (pp. 19-36).

www.irma-international.org/article/the-case-for-privacy-awareness-requirements/152245

A Security Review of Event-Based Application Function and Service Component Architecture

Faisal Nabi, Jianming Yong and Xiaohui Tao (2020). *International Journal of Systems and Software Security and Protection* (pp. 58-70).

www.irma-international.org/article/a-security-review-of-event-based-application-function-and-service-component-architecture/259420