Chapter 18
The Formal Design Model of a Telephone Switching System (TSS)

Yingxu Wang
University of Calgary, Canada

ABSTRACT

A typical real-time system, the Telephone Switching System (TSS), is a highly complicated system in design and implementation. This paper presents the formal design, specification, and modeling of the TSS system using a denotational mathematics known as Real-Time Process Algebra (RTPA). The conceptual model of the TSS system is introduced as the initial requirements for the system. Then, the architectural model of the TSS system is created using the RTPA architectural modeling methodologies and refined by a set of Unified Data Models (UDMs). The static behaviors of the TSS system are specified and refined by a set of Unified Process Models (UPMs) such as call processing and support processes. The dynamic behaviors of the TSS system are specified and refined by process priority allocation, process deployment, and process dispatching models. Based on the formal design models of the TSS system, code can be automatically generated using the RTPA Code Generator (RTPA-CG), or be seamlessly transformed into programs by programmers. The formal model of TSS may not only serve as a formal design paradigm of real-time software systems, but also a test bench of the expressive power and modeling capability of exiting formal methods in software engineering.

INTRODUCTION

Telephone Switching Systems (TSS's) are one of the typical real-time and mission-critical systems, as those of air-traffic control and banking systems, characterized by their high degree of complexity, intricate interactions with hardware devices and users, and necessary requirement for domain knowledge (Labrosse, 1999; Liu, 2000; McDermid, 1991; Ngolah et al., 2004; Wang, 2007a). All these factors warrant a TSS system as a complex but ideal design paradigm in real-world large-scale software system design in general and in real-time system modeling in particular.

DOI: 10.4018/978-1-4666-0261-8.ch018

There is no systematical and detailed repository and formal documentation of design knowledge and modeling prototypes of a TSS system nor a formal model of it in denotational mathematics and formal notation systems. This article presents the formal design, specification, and modeling of a TSS system using a denotational mathematics known as Real-Time Process Algebra (RTPA) (Wang, 2002, 2008a, 2008b). RTPA introduces only 17 meta-processes and 17 process relations to describe software system architectures and behaviors with a stepwise refinement methodology (Wang, 2007a, 2008a, 2008c, 2008d). According to the RTPA methodology for system modeling and refinement, a software system can be specified as a set of *architectural components* and *operational components*. The former are modeled by the Unified Data Models (UDMs, also known as *Component Logical Models* CLMs), which is an abstract model of the system hardware interface, an internal logic model of hardware, and/or a control structure of a system. The latter are modeled by *static* and *dynamic processes* in term of the Unified Process Models (UPMs) (Hoare, 1978; Milner, 1980; Hoare et al., 1987; Baeten and Bergstra, 1991; Corsetti and Ratto, 1991; Vereijken, 1995; Dierks, 2000; Wang, 2007a, 2008a; Wang and King, 2000).

This article develops a formal design model of the TSS system in a top-down approach on the basis of the RTPA methodology. In the remainder of this article, the conceptual model of the TSS system is described as the initial requirements of the system. The architectural model of the TSS system is created based on the conceptual model using the RTPA architectural modeling methodologies and refined by a set of CLMs. Then, the static behaviors of the TSS system are specified and refined by a set of processes. The dynamic behaviors of the TSS system are specified and refined by process priority allocation, process deployment, and process dispatching models. With the formal and rigorous models of the TSS system, code can be automatically generated by the RTPA Code Generator (RTPA-CG) (Wang, 2007a, Wang et al., 2009), or be seamlessly transferred into programs manually. The formal model of TSS may not only serve as a formal design paradigm of real-time software systems, but also a test bench of the expressive power and modeling capability of exiting formal methods in software engineering.

THE CONCEPTUAL MODEL OF THE TSS SYSTEM

A Telephone Switching System (TSS) is a complex real-time system (Thompson, 2000; Wang, 2007a). The functional structure of the TSS system can be described by a conceptual model as illustrated in Figure 1, which consists of four subsystems known as the call processing, subscribers, routes, and signaling subsystems.

The configuration of the TSS system encompasses 1 call processor and 16 subscribers. There are 5 internal switching routes and a set of 5 signaling trunks providing the dial, busy, ringing, ring-back, and special tones. The call processor modeled by a set of functional processes operates on the line scanners, call records, digits receivers, signaling trunks, system clock, and routes in order to implement a coherent program-controlled switching functions.

The framework of the TSS system, encompassing its architecture, static behaviors, and dynamic behaviors, can be specified using RTPA as follows (Wang, 2002, 2008a):

$$\begin{aligned} \textbf{§(TSS)} \triangleq\ & \text{TSS}\textbf{§}.\text{Architecture}\textbf{ST} \\ & \parallel \text{TSS}\textbf{§}.\text{StaticBehaviors}\textbf{PC} \\ & \parallel \text{TSS}\textbf{§}.\text{DynamicBehaviors}\textbf{PC} \end{aligned} \tag{1}$$

where || indicates that these three subsystems related in parallel, and §, **ST**, and **PC** are type suffixes of *system, system structure,* and *process*, respectively.