Implementing Continuous Time SRML by Integrating IBM WebSphere with Matlab

Ning Yu and Martin Wirsing
Software and Systems Engineering, Institut für Informatik
Ludwig-Maximilians-Universität München
Oettingenstrasse 67, 80538 Munich, Germany

Service-Oriented Computing [1,2] is a paradigm for distributed computing, in which computation units are abstracted as services that are autonomous, platform-independent and can be described, published, discovered, orchestrated and programmed using standard protocols within certain networks. The SENSORIA Reference Modeling Language (SRML) [3] has been developed in the IST-FET integrated project SENSORIA. It is a prototype domain-specific language for modeling service-oriented systems at the business level abstraction. Continuous time SRML [4] extends SRML by adding syntax for differential equations that describe the continuous time processes of the systems, is interpreted over Service-Oriented Hybrid Doubly Labeled Transition Systems (SO-HL^{2}TSs) [4], which can be seen as abstractions of the behavior of hybrid systems with service-oriented features. With this continuous time extension, SRML can be used to specify models of hybrid systems in a service-oriented way.

In order to implement SRML modules (models specified by continuous time SRML), we first transform them into business state machines, which can be constructed in IBM WebSphere Integration Developer (WID) [5]. (WID is named as WebSphere Integration Designer for the latest version.) In detail, the transformation assigns a state to each transition, specifies the propositions that hold in the state, and labels the edge that leads to the state with the trigger and guard conditions of that transition.

For the implementation, we integrate WID with MATLAB [6] to work as the underlying platform. WID is a comprehensive Service-Oriented Architecture (SOA) [7] integration platform, which is based on WebSphere Application Server, and provides support for Service Component Architecture (SCA) [8] programming model. In WID, service-oriented applications are constructed as service modules, which are built by assembling service components. Service components are the atomic units in SCA that perform business logic, and they provide their capabilities as services through interfaces, and require services offered by other components through references. Service components can be implemented in various types such as Java objects, business processes, and business state machines. In our implementation, we choose business state machines because they are more close to continuous time SRML specifications. Moreover, differential equations specified in SRML modules are embedded in business state machines by giving solutions to the differential equations. When the differential equations are simple, we can give the exact solutions directly. But when the differential equations are complicated and can't be solved by getting exact solutions, we have to integrate WID with MATLAB to get the approximate solutions.

MATLAB is a multi-paradigm numerical computing environment, and it allows interfacing with programs written in other languages such as C, C++ and Java. Programming in MATLAB can solve differential equations numerically. For the differential equations that are embedded in WID business state machines, we write programs in MATLAB to solve them, and pass the result to the business state machines by calling the programs with Java codes in the state machines.

In this paper, we first introduce the Service-Oriented Hybrid Doubly Labeled Transition Systems, which serve as semantic domain for continuous time SRML. As example we take the Train-Control model, which is part of the European Train Control system [9], and show the continuous time SRML specification of this model. Then we present our systematic approach for transforming SRML specifications to business state machines, and show how to construct business state machines in the implementation platform. Finally, we show the essential coding and the results for various inputs.

Key words: continuous time SRML, SO-HL^{2}TSs, differential equations, WID, MATLAB

References:

[1] Hilia M., Chibani A., Djouani K., Amirat Y. *Formal Specification and Verification Framework for Multi-domain Ubiquitous Environment.* In: Sheng Q., Stroulia E., Tata S., Bhiri S. (eds) Service-Oriented Computing. ICSOC 2016. Lecture Notes in Computer Science, vol 9936. Springer, Cham

[2] Issarny V., Bouloukakis G., Georgantas N., Billet B. *Revisiting Service-Oriented Architecture for the IoT: A Middleware Perspective.* In: Sheng Q., Stroulia E., Tata S., Bhiri S. (eds) Service-Oriented Computing. ICSOC 2016. Lecture Notes in Computer Science, vol 9936. Springer, Cham

[3] J. A. Abreu. *Modelling Business Conversations in Service Component Architectures.* PhD thesis, University of Leicester, 2009.

[4] N. Yu. *Injecting Continuous Time Execution into Service-Oriented Computing.* PhD thesis, Munich University, 2016.

[5] IBM Knowledge Center. *Starting from WebSphere Integration Developer.* Last updated: July 2017.[online] https://www.ibm.com/support/knowledgecenter/en/SSMKHH_9.0.0/com.ibm.etools.mft.doc/bc34005_.htm

[6] Mathwork. *The Overview of MATLAB.* Latest version: R2018a. [online] https://de.mathworks.com/products/matlab.html

[7] IBM Knowledge Center. *Service-Oriented Architecture.* Part of the Rational Business Developer9.5.1.[online] https://www.ibm.com/support/knowledgecenter/en/SSMQ79_9.5.1/com.ibm.egl.pg.doc/topics/pegl_serv_overview.html

[8] IBM Knowledge Center. *Service Component Architecture*. Last updated: April 2018. [oline] https://www.ibm.com/support/knowledgecenter/en/SSGMCP_5.1.0/com.ibm.cics.ts.applicationprogramming.doc/bundleinterface/sca.html

[9] UIC, The Worldwide Railway Organisation. *The European Train Control System.* Last updated: August 2015. [online] https://uic.org/etcs#Standards-and-Specifications