Journal of Systems Architecture 59 (2013) 331–340

Contents lists available at SciVerse ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

Modeling distributed real-time systems with MAST 2

Michael González Harbour *, J. Javier Gutiérrez, José M. Drake, Patricia López Martínez, J. Carlos Palencia

Computers and Real-Time Group, Universidad de Cantabria, 39005 Santander, Spain

ARTICLE INFO

Article history:
Available online 24 February 2012

Keywords:
Real-time
Distributed systems
Modeling
Network switches
Clock synchronization

ABSTRACT

Switched networks have an increasingly important role in real-time communications. The IEEE Ethernet standards have defined prioritized traffic (802.1p) and other QoS mechanisms (802.1q). The Avionics Full-Duplex Switched Ethernet (AFDX) standard defines a hard real-time network based on switched Ethernet. Clock synchronization is also an important service in some real-time distributed systems because it allows a global notion of time for event timing and timing requirements. In the process of defining the new MAST 2 model, clock synchronization modeling capabilities have been added, and the network elements have been enhanced to include switches and routers. This paper introduces the schedulability model that will enable an automatic schedulability analysis of a distributed application using switched networks and clock synchronization mechanisms.

1. Introduction

MAST (Modeling and Analysis Suite for Real-Time Applications) [7,14] defines a model to describe the timing behavior of real-time systems designed to be analyzable via schedulability analysis techniques. MAST also provides an open-source set of tools to perform schedulability analysis or other timing analysis, with the goal of assessing whether the system will be able to meet its timing requirements, and, via sensitivity analysis, how far or close is the system from meeting its timing requirements.

The model defined in MAST is very similar to the model defined in the Schedulability Analysis Modeling chapter (SAM) of the MARTE profile (The UML Profile for Modeling and Analysis of Real-Time and Embedded Systems) [16]. A new enhanced model is currently being defined as a project called MAST 2, trying to incorporate new modeling elements that can be found in real systems. It is expected that the ideas introduced in MAST 2 will contribute to the future evolution of the MARTE standard.

Some of the new elements being defined in MAST 2 are network switches and routers. Switched networks are being used increasingly to build real-time systems, as new network switches incorporate the real-time mechanisms being defined in standards such as IEEE 802.1p with prioritized traffic [9], 802.1q with various QoS mechanisms [23], or the Avionics Full-Duplex Switched Ethernet (AFDX) [1] that defines a hard real-time network based on switched Ethernet.

This paper introduces the model elements required to add network switches and routers into the MAST model. These elements will allow an automatic schedulability analysis of applications using switched networks.

Another addition to MAST 2 is the capability to model and analyze time-triggered systems, such as those developed according to the ARINC 653 standard [3]. In these systems the scheduler uses a table to drive the generation of the events that trigger the execution of operations in the processing nodes, and the transmission of messages across the networks. In many cases these time-triggered systems require the capability of having a global notion of time, and thus require clock synchronization services. In addition, another kind of scheduling that requires clock synchronization is global EDF [22], which has been shown to obtain better schedulability results than the non-synchronized local EDF [18].

Kopetz has a very interesting introduction to clock synchronization in [11] that describes basic concepts about global time. It provides definitions of the digital physical clock, the granularity, the reference clock, the clock drift, and the precision or the accuracy of the global time base. MAST 2 defines a means to model the clock synchronization service and to analyze its effects.

The paper is organized as follows. Section 2 contains a brief summary of related work. In Section 3 we present a general overview of the MAST 2 model, and we focus on the network modeling elements in Section 4. The model of clock synchronization services is described in Section 5. The new elements introduced to model network switches are presented in Section 6, and similarly in Section 7 for network routers. Section 8 introduces the new modeling elements for AFDX networks and switches together with a simple example using these elements. Finally, Section 9 gives our conclusions.

* Corresponding author.
E-mail addresses: mgh@unican.es (M. González Harbour), gutierjj@unican.es (J.J. Gutiérrez), drakej@unican.es (J.M. Drake), lopezpa@unican.es (P. López Martínez), palencij@unican.es (J.C. Palencia).

1383-7621/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.sysarc.2012.02.001

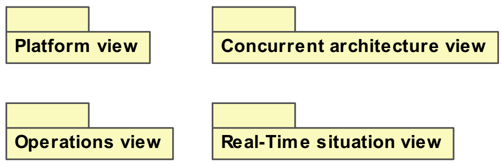

Fig. 1. Real-Time Model in MAST 2.

2. State of the art in modeling real-time properties of distributed systems

There are some frameworks that allow modeling real-time systems for different purposes of analysis and simulation. For instance, Cheddar [21] is a framework that allows defining a simple model of a real-time system with the purpose of making an analysis of its properties via simulation and feasibility tests. It does not support distributed systems.

UPPAAL [5] is a popular model-checker that can be used to model and analyze many different kinds of systems, including distributed systems, but given its nature it is only practical for small systems.

Papyrus MDT [17] is an Eclipse project that provides advanced support for editing UML2 models, including also support for the MARTE profile. As it happens with MARTE, this tool does not directly support some of the modeling elements being defined in MAST 2, such as network switches, network routers, and drivers. The MARTE profile defines a huge variety of modeling constructs that can be used to model an application from different viewpoints, which may lead to really complex models. Thus, without a specific set of restrictions or modeling guidelines, it may be difficult to automatically obtain a schedulability analysis model from the description of the application. With the purpose of simplifying this process, a tool and a modeling methodology to generate a MAST-compliant model from a UML + MARTE model have been recently proposed [15].

SymTA/S [8] is a modeling and analysis professional tool that uses formal analysis techniques based on the busy window technique by Lehoczky [12], and supports local analysis techniques for fixed priorities (preemptive and non-preemptive), TDMA (time division multiplexed access), Round Robin, or EDF (earliest deadline first) scheduling. AFDX has been included as a target in the SymTA/S 3.0 version [20], which also supports CAN (Controller Area Network [10]), TTP (Time-Triggered Protocol [19]) and Flexray [6]. The model defined in SymTA/S is proprietary.

Furthermore, most of the works that include modeling of networks and distribution are more centered around the analysis or simulation of a particular network, rather than on providing general modeling capabilities. In particular, modeling of AFDX networks and switches appears in works on the calculation of latencies or simulation, which usually do not model the whole distributed system but treat the network in an isolated way. For instance, Zhang et al. [24] present a model based on Deterministic and Stochastic Petri Nets to simulate the behavior of the AFDX network. The work by Bauer et al. [4] is specialized in the calculation of worst-case end-to-end latencies of messages in an AFDX network, but messages are treated as anonymous and messages composed by more than one packet are also not considered.

Anand et al. [2] developed a formal model of the AFDX frame management to ascertain the reliability properties of the design. They model the system as a network of timed automata and use UPPAAL [5] as a model-checker. The timed automata model allows the management of the temporal aspects of the frame management such as maximum latency, skew and BAG (bandwidth allocation gap).

In comparison with these modeling techniques, MAST 2 provides detailed modeling capabilities that are designed to support the schedulability analysis of large complex distributed systems. With the additions described in this paper, MAST 2 will also support switched networks and scheduling policies requiring a global notion of time.

3. Overview of the MAST 2 model

A real-time system is modeled in MAST 2 using four different independent views (see Fig. 1) for describing: the execution platform, the software modules and messages exchanged through the networks, the concurrent architecture, and the workload and flow of events for a particular configuration of the application. This independence among the various elements of the model is ideal for building a full model through the composition of partial models developed independently. We will now review the main elements included in these views in the following subsections. These elements are also shown in Fig. 8 through a simple example described at the end of this section. For a more extensive description of the MAST 2 model refer to the metamodel that can be found in the MAST home page [14].

3.1. Platform view

The execution platform view contains *Processing_Resources* such as *Computing_Resources* and *Networks*, together with their *Schedulers* and associated *Scheduling_Policy* elements. Each of these elements contains attributes that describe their timing behavior including overheads such as context switching, interrupt service, or system timers. *Processors* have a model of the interrupt mechanism and *Networks* have a bandwidth expressed in bits per time unit. Fig. 2 shows the main processing resources in the platform view, together with their main attributes.

The *Schedulers* are specialized in two classes: on the one hand, *Primary_Schedulers* are hosted on a *Processing_Resource*. On the other hand, *Secondary_Schedulers* are hosted on a schedulable resource and are used to model hierarchical scheduling.

The scheduling policies that are currently defined are fixed priorities, EDF, the new AFDX policy that will be described in Section 8, and the timetable driven policy. The latter can be used to model partitioned systems designed under the ARINC 653 standard [3] used in avionics, or systems developed with the Time-Triggered Protocol (TTP) [19]. Among fixed priorities, different scheduling strategies are allowed: preemptive and non preemptive, interrupt service routines, sporadic servers, and polling servers.

MAST 2 is adding the network switches and routers as new processing resources. They will be described in Sections 6 and 7.

3.2. Operations view

The operations view contains the elements that describe the usage of the processing resources, through *Operations*. Fig. 3 shows the class diagram of the main elements involved in the operations view.

Code_Operations model the execution of sequential code in a processor, with a given execution time distribution. The most important attribute for the schedulability analysis is the worst case execution time (WCET). *Message_Operations* represent data of a given size that are sent through a network.

The *Code_Operation* abstract class is specialized with *Simple_Operation* and *Composite_Operation*, while *Message_Operation* is specialized with *Message* and *Composite_Message*. The composite versions allow for composition of operations or messages to produce longer more complex operations and messages.