Representing real time semantics for distributed application integration

Polly M.S. Poon\(^1\), Tharam S Dillon\(^1\), Elizabeth Chang\(^2\), Ling Feng\(^3\)

\(^1\)Faculty of Information Technology, UTS, Sydney, Australia
{polly, tharam}@it.uts.edu.au

\(^2\)School of Information Systems, Curtin University of Technology, Australia
change@cbs.curtin.edu.au

\(^3\)Database Group, Department of Computer Science, University of Twente, Enschede, The Netherlands, ling@cs.utwente.nl

Abstract

Traditional real time system design and development are driven by technological requirements. With the ever growing complexity of requirements and the advances in software design, the alignment of focus has gradually been shifted to the perspective of business and industrial needs. This paper discusses a new paradigm in developing distributed real time system with the emphasis of knowledge representation and interoperability using XML. Real Time Markup Language (RTML) aims to provide a basis for interoperability.

1. Introduction

In the past, most of the real time applications run in specialized devices or hardware as a part of engineering solution to fulfill complex computational task. Over time, the application domain of real time systems has been extended to a wider spectrum. Today, soft real time system can be found running in enterprise server. Both of these types of systems share a common feature, timeliness. The only differences between these two are the way how the system responds to the overdue deadline. Hard real time systems, as the former type are most likely to be found, needs to face the absolutely correct timeliness, the past of deadline is not an option or the consequences could be catastrophic; whilst a soft real time system might accept the occasional misses of deadlines.

Today, businesses strive to become a winner in a competitive market. The urge of providing reliable, timely service is one of the main priorities, so timeliness plays an important role. For instances, an online ticketing systems are required to serve hundreds of customers in every seconds when a major event comes start the sales. During the peak time, the online ticket purchase system would need to handle the user loads as much as tens of thousands per seconds. Systems will need to provide a robust service for the ticket purchase transaction. In such a chaotic situation, the system needs to ensure the atomicity of a transaction (Ticket is booked and paid by customer) and the fairness of resource access.

2. Problem

The advances in software design technologies allow a more general level of abstraction. Software design models can be directly translated into implementation using computer-based automation. However, in many cases, the application of these tools are only limited at application code transformation. The knowledge capture during design fails to translate into useful data representation format, or in other words, there are no direct benefit resulted from the initial design to data semantics. The broken link between design and data definition could cause the loss of semantics. This could cause the problem in the long run as the change in design could not eventually be represented to the data layer until the application code is changed.
To address this, it requires a framework that can transform the semantics from the conceptual design and apply to data. In this paper we present Real Time Markup Language (RTML), a XML based solution for transforming the design time class model into XML instances. In this paper, we will discuss the importance of interoperability between real time systems, our model driven design and the details of RTML.

3. Interoperability between real time systems

The diversity in system architecture and component has resulted in heterogeneity. The key of success for the collaboration between distributed computing architecture is relied on consistent definition of interface and data definition/access, in particular of how the data is formatted, exchanged and stored. It is important to provide a consistent standard to ensure a smooth implementation of service collaboration, since data contains the abstract from the logic of the domain [1]. We are particularly concerned with a number of important features that characterize information dissemination in distributed systems and these include:

1. Collection and transmission of data from sensors to data concentrators
2. Storage of data at several levels and different databases that must interoperate with each other. Some systems, for example advanced systems which run in power plants [2], one requires integration of real time data, or historic data to make intelligent analysis and allow the system to run predictably. This could require the integration between archive data and operating real time data. Another example would be the implementation of communication between segments of networks with the use of management agent, where information is exchanged by agent on behalf of network stations.

4. Approach

In previous paper [3, 4, 5] we discussed the use of model transformation in obtaining data instances. In this section we will discuss a new approach in mapping conceptual model into RTML instances. As depicted in Figure 2.

4.1 OMG SPT and XMI

RTML is derived based on a number of specification including specification from Object Management Group, UML™ Profile for Schedulability, Performance, and Time Specification from Object Management Group (SPT hereafter) [5] and the UML™ Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms from Object Management Group[7] (QoS Profile hereafter). Our approach targets design that is based on the use of class models using these profiles. The widespread use of UML provides a solid ground for this approach to be practical. In particular, modeling in class diagram is extremely useful in modeling Object Oriented systems as the modeling notation can clearly visualize the relationship between entities. This importance is even highlighted when it applies to real time systems [8]

Models developed in SPT/QoSProfile can be exported in to XML Metadata Interchange (XMI) [9, 10], which represents metadata information in XML. It can be used for any metadata whose metamodel can be expressed in...
Meta-Object Facility (MOF). MOF-based models (such as Class Diagram in UML) can be transformed in XMI format. Once models are exported from diagram format to XMI, we need to extract the knowledge from the XMI document. Previous work [11, 12] has shown the model transformation from UML class model into XML data instances. This transformation will be the first step of RTML model transformation.

The next step will be transforming a SPT XML document into a RTML document. SPT models describe the design time details involve in a component. Since the details in a SPT model are too abstract to be directly usable at in the application, it requires further details to be included (E.g. identifier for an instance) for data exchange over distributed applications. This process involves the mapping from SPT syntax to RTML syntax. In the following section, we will describe some of the important concepts in RTML.

5. RTML Structure

RTML provides interoperability between real time systems. Among the diversity of concerns for real time systems, RTML must capture and represent of 4 major aspects. These include:

**Time** --- RTML provides a detailed definition of time. It provides a metric based and time based representation. A metric based time instant or duration representation can be described in a list of integer or double. Apart from the time granularity extension from the XML Schema, we have also included the illustration of timing mechanisms in our schema which was represented in [3]

**Resources** --- In order to illustrate the resource representation, RTML provides a wide range of primitives for resource description including physical resources such as device, CPU or communication devices; as well as resource service such as application or services description. Resources have a complementary relationship with QoS in real time systems.

**QoS** --- In [4, 5] we have presented the XML Profile for QoS. At this stage we have provided a profile for illustration of QoS concepts for Resource Instances description.

**Causality** --- The description of action execution and resource service instance are described using the CausalityModel. Essentially the CausalityModel acts as a container to illustrate the actions involved in a message interaction instance between two application domains.

6. The anatomy of RTML

6.1 Global Timing Mechanism

The timing mechanism is defined in RTML as a global invariant in a RTML instance. The Timing Mechanism in RTML provides the reference to the location of a physical clock, and some of the offered QoS attributes for the characterization of the timing device. This allows the elements that are defined locally within the RTML structure to reference it through the entire hierarchy.

```xml
<GlobalTimingDevice id="ID000107">
  <Clock id="ID000108">
    <CurrentValue>
      <TimeInstant>
        <CurrentValue>5:00</CurrentValue> ClockRef="/8AA"0505-12-11T09:30:47-05:00</TimeInstant>
      </CurrentValue>
    </CurrentValue>
    <ReferenceClock>
      <ExternalClockURI>http://localhost/rtml/time</ExternalClockURI>
      <ReferenceClock/>
      <Drift unit='ns'/>0.00001 </Drift>
    </ReferenceClock>
    <Resolution unit="ns">1</Resolution>
    <Clock/>
  </Clock>
</GlobalTimingDevice>
```

Figure 3  Global Timing Mechanism

6.3 Extended time types

As mentioned previously, many of the existing XML profiles do not cater for general timing properties. The primitive built-in types from XML Schema are normally adequate to handle most business transactions. However, the timing constraints could be different for real time systems depending on the application domain. In some cases the time granularity could be as fine as nanoseconds whilst in others it could be minutes or hours. In RTML, we have proposed a collection of timing standards in order to provide an extension that captures a more specific characterization of timing description for XML elements.

Generally the timing properties of RTML can be classified as absolute time (Time Instant) and relative time (duration). Time can be described using metric based measurement with the augmentation of Time Unit to specify its granularity. Each of the Instant is associated with a Clock reference, as specified by the Global Clock defined in the header of the RTML instance.

```xml
<xs:choice>
  <xs:element name="TimeInstantType">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:dateTime">
          <xs:attribute name="ClockRef" type="ttml:xPathReferenceType"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="MetricTimeInstant">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:dateTime">
          <xs:attribute name="ClockRef" type="ttml:xPathReferenceType"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
</xs:choice>
```

Figure 4  Time Instant in RTML

Time duration can be represented in primitive XML Schema duration time or using a metric based recurring time interval.

```xml
<xs:complexType name="RecurringMetricDurationType">
  <xs:sequence>
    <xs:element name="Interval" maxOccurs="unbounded">
      <xs:complexType name="RecurringMetricDurationType">
        <xs:sequence>
          <xs:attribute name="ClockRef" type="ttml:xPathReferenceType"/>
        </xs:extension>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Figure 4  Time Instant in RTML
An event is executed when the particular resource instances has completed all its process. In some cases, for example a DB transaction, one would want to ensure the atomicity of the process and therefore the poll only executes when the transaction is completed.

This could also represent in RTML as follows:

```
<MemPoll id="ID002345" name="RegularSpaceCheck" isInterrupt="1">
  <FiringInterval ClockRef="/GlobalClock">
    <RecurringMetricDuration unit="s">
      <FiringInterval ClockRef="/GlobalClock">
        <RecurringMetricDuration unit="s">
          <FiringInterval ClockRef="/GlobalClock">
            <MemPoll id="ID002345" name="RegularSpaceCheck" isInterrupt="1">
              <FiringInterval ClockRef="/GlobalClock">
                <RecurringMetricDuration unit="s">
                  <FiringInterval ClockRef="/GlobalClock">
                    <MemPoll id="ID002345" name="RegularSpaceCheck" isInterrupt="1">
                      <FiringInterval ClockRef="/GlobalClock">
                        <RecurringMetricDuration unit="s">
                          <FiringInterval ClockRef="/GlobalClock">
                            <MemPoll id="ID002345" name="RegularSpaceCheck" isInterrupt="1">
                              <FiringInterval ClockRef="/GlobalClock">
                                <RecurringMetricDuration unit="s">
                                  <FiringInterval ClockRef="/GlobalClock">
                                    <MemPoll id="ID002345" name="RegularSpaceCheck" isInterrupt="1">
                                      <FiringInterval ClockRef="/GlobalClock">
                                        <RecurringMetricDuration unit="s">
                                          <FiringInterval ClockRef="/GlobalClock">
                                            <MemPoll id="ID002345" name="RegularSpaceCheck" isInterrupt="1">
                                              <FiringInterval ClockRef="/GlobalClock">
                                                <RecurringMetricDuration unit="s">
                                                  <FiringInterval ClockRef="/GlobalClock">
                                                    <MemPoll id="ID002345" name="RegularSpaceCheck" isInterrupt="1">
                                                      <FiringInterval ClockRef="/GlobalClock">
                                                        <RecurringMetricDuration unit="s">
                                                          <FiringInterval ClockRef="/GlobalClock">
                                                            <MemPoll id="ID002345" name="RegularSpaceCheck" isInterrupt="1">
                                                              <FiringInterval ClockRef="/GlobalClock">
                                                                <RecurringMetricDuration unit="s">
                                                                  <FiringInterval ClockRef="/GlobalClock">
                                                                    <MemPoll id="ID002345" name="RegularSpaceCheck" isInterrupt="1">
                                                                      <FiringInterval ClockRef="/GlobalClock">
                                                                        <RecurringMetricDuration unit="s">
                                                                          <FiringInterval ClockRef="/GlobalClock">
                                                                            <MemPoll id="ID002345" name="RegularSpaceCheck" isInterrupt="1">
                                                                              <FiringInterval ClockRef="/GlobalClock">
                                                                                <RecurringMetricDuration unit="s">
                                                                                  <FiringInterval ClockRef="/GlobalClock">
                                                                                    <MemPoll id="ID002345" name="RegularSpaceCheck" isInterrupt="1">
                                                                                      <FiringInterval ClockRef="/GlobalClock">
                                                                                        <RecurringMetricDuration unit="s">
                                                                                          <FiringInterval ClockRef="/GlobalClock">
                                                                                           <MemPoll id="ID002345" name="RegularSpaceCheck" isInterrupt="1">
                                                                                            <FiringInterval ClockRef="/GlobalClock">
                                                                                             <RecurringMetricDuration unit="s">
                                                                                               <FiringInterval ClockRef="/GlobalClock">
                                                                                                 <MemPoll id="ID002345" name="RegularSpaceCheck" isInterrupt="1">
                                                                                                   <FiringInterval ClockRef="/GlobalClock">
                                                                                                    <RecurringMetricDuration unit="s">
                                                                                                      <FiringInterval ClockRef="/GlobalClock">
                                                                                                        <MemPoll id="ID002345" name="RegularSpaceCheck" isInterrupt="1">
                                                                                                          <FiringInterval ClockRef="/GlobalClock">
                                                                                                            <RecurringMetricDuration unit="s">
                                                                                                              <FiringInterval ClockRef="/GlobalClock">
                                                                                                                <MemPoll id="ID002345" name="RegularSpaceCheck" isInterrupt="1">
                                                                                                                  <FiringInterval ClockRef="/GlobalClock">
                                                                                                                    <RecurringMetricDuration unit="s">
                                                                                                                      <FiringInterval ClockRef="/GlobalClock">
                                                                                                                        <MemPoll id="ID002345" name="RegularSpaceCheck" isInterrupt="1">
                                                                                                                          <FiringInterval ClockRef="/GlobalClock">
                                                                                                                            <RecurringMetricDuration unit="s">
                                                                                                                              <FiringInterval ClockRef="/GlobalClock">
                                                                                                                                <MemPoll id="ID002345" name="RegularSpaceCheck" isInterrupt="1">
                                                                                                                                  <FiringInterval ClockRef="/GlobalClock">
                                                                                                                                   <RecurringMetricDuration unit="s">
                                                                                                                                     <FiringInterval ClockRef="/GlobalClock">
                                                                                                                                       <MemPoll id="ID002345" name="RegularSpaceCheck" isInterrupt="1">
                                                                                                                                         <FiringInterval ClockRef="/GlobalClock">
                                                                                                                                             <RecurringMetricDuration unit="s">
allows models develop in UML CASE based tools be directly translated into RTML based solution using model transformation method and rule mapping mechanism. We have also explained the structure of RTML in this paper. RTML is derived from OMG SPT since UML is widespread and it has become one of the common standards in software design. RTML aims to provide interoperability for real time systems. By providing such a XML profile, it establishes a knowledge based for organizations to exchange data using XML messaging.

Reference


