Extending the OMG’s D&C specification for the design and analysis of real-time component-based applications

Patricia López Martínez, José M. Drake
Grupo de Computadores y Tiempo Real
Universidad de Cantabria, Spain
Objective: Component-based real-time applications

- Component-based approaches
  - Component
  - Application: Assembly of components
  - Isolation, Opaqueness, Composability

- Real-time systems
  - Real-time model required to analyse the temporal behaviour and/or to obtain the schedulability parameters configuration
  - Reactive model of real-time systems:
    - Applications conceived as a set of concurrent end-to-end flow transactions
    - Timing requirements defined as temporal constraints in the transactions
Solution: Real-time metadata

Component package
- Functional metadata
- Instantiation metadata
- Component Code
- Real-time metadata
- Parameterized Real-time model

Repository
- Platform Resource
- Parameterized Real-Time model

RT-Model Composer
- Execution Node Real-Time Model
- Application Real-Time Model

RT Analysis and Design Tools
- Application Schedulability Configuration
Our proposal: Extension of the D&C Specification

- The metadata included in a component package must follow a standard format to be easily managed by appropriate tools.
- The real-time design process must be consistent with the traditional component-based development process.

Extending the Deployment and Configuration of Component-based Distributed Applications Specification of the OMG

- with:
  - New tasks related to real-time design included in the development process
  - Support for RT-metadata about components and platforms

- for being able to:
  - Predict or analyse the temporal behaviour of component-based real-time applications based on a reactive model
  - Configure the platform and the component instances to guarantee that the applications meet their timing requirements during execution

- without requiring the application designers to be experts in the underlying real-time design and analysis methodology
Reactive Model

EndToEndFlow

Step

AnalysisContext

ResourcePlatform

ResourceUsage (from GRM)

ProcessingResource

SchedulableResource

SharedResource

WorkloadEvent

TimingObserver

Step

EndToEndFlow

Oper1

Oper2

Oper1

Oper4

Oper5

Oper6

Schedulable Resource

Shared Resource

Timing Observer

E1

E2

Oper2

Oper1

Oper1

Oper4

Oper5

Oper6

Step

Workload Event

WorkloadBehaviour

ResourcePlatform

Resources

Resource (from GRM)

Sequence Diagram

Behavior

Steps

Host

Concurrent Resources

EndToEnd Stimuli

EndToEndFlow

Oper2

Oper1

Oper1
RT-D&C Application Development Process

Installation phase
- Component package
  - Installer

Configuration phase
- Application requirements
  - Functional
  - Real-time
  - RT-D&C Workload Model
  - D&C Component Assembly Description
  - Planner

Planning phase
- D&C Domain Description
  - Schedulability configuration
  - RT-D&C Deployment plan
  - Executor

Preparation & Launching phase
- Executable code

Repository of the design environment
- Component Functional metadata
- Component Real-Time metadata
- Platform metadata
- Component Code

Prague, November 2009
Patricia López Martínez
Example Application: ScadaDemo

Application specification

- **T = samplingPeriod**
  - Read magnitudes
  - Register value for statistics

- **T = loggingPeriod**
  - Gather and pack data
  - Store data

- **T = monitoringPeriod**
  - Read last data
  - Refresh monitored value
  - Process Command

Physical magnitudes

External Environment

Monitor

Keyboard

Logger
Application Design

ScadaEngine package

- <<ComponentInterfaceDescription>>
  - ScadaEngine.ccd.xml
  - Connections Metadata (D&C)
  - Configuration Metadata (D&C)
  - RT Composability Metadata (RT-D&C)
  - Supported Reactivity Metadata (RT-D&C)

- <<ComponentImplementationDescription>>
  - AdaScadaEngine.cid.xml
  - AdaScadaEngine Code

Assembler

- Defines the application
- Defines the workload
- Describes the application structure
- Describes formally the reactive description

ScadaDemo.cad.xml

ScadaDemoWorkload.rtw.xml

Context Analysis: Workload
- Business transactions declaration
- Generation patterns of external events
- Frequencies of timed events
- Real-time requirements
Component Specification: D&C’s Component Interface Description

Specifier

<<ComponentInterfaceDescription>>
ScadaEngine.ccd.xml

<<describes>>

Domain required functionality

ScadaEngine [scada]

controlPort
ScadaControl

loggingPeriod:Duration
SamplingTrans (samplingDeadline:Deadline, supervisionOperList: UsageList)

loggingDeadline:Deadline
LoggingTrans

adqPort
AnalogIO

SuperviseVariable

samplingPeriod:Duration

Configuration properties

Required functionality

Required rt-Models:
- ao_ReadCode
- ao_ReadValue

Offered RTModels:
- getBufferedData

Real-time Composability Requirements

Reactivity: Supported
Transactions Declaration

Prague, November 2009
Patricia López Martínez
Extensions to the ComponentInterfaceDescription

Real-time composability requirements

Reactivity:
Supported Transactions Declaration
(transactions can be parameterized)
ScadaEngine’s Component Interface Description

ScadaEngine

controlPort

samplingPeriod:Duration
loggingPeriod:Duration

SamplingTrans
LoggingTrans
SuperviseVariable

ScadaControl

loggingPort

Logging

adqPort

AnalogIO

<port name="adqPort"
specificType="interfaces/io/AnalogIOCard.idl::io::I_Analog"
provider="false">
<offeredRTPortModel>
<rtUsage name="aiReadCode"/>
<rtUsage name="aiWriteCode"/>
</requiredRTPortModel>

<property name="loggingDeadline"
type="DEADLINE"
label="Deadline for the data gathering activity">
</property>

<property name="loggingPeriod"
type="DURATION"/>

<rtWorkloadEntity name="LoggingTrans"
label="Periodic activity for storing permanently statistical values about the magnitudes">
</rtWorkloadEntity>
Application Design

Context Analysis: Workload
- Business transactions declaration
- Generation patterns of external events
- Frequencies of timed events
- Real-time requirements

Application: Component Assembly
- Component instances declaration
- Components interconnection
- Business Configuration

Reactive description of the application

Assembler
- Defines the workload
- Defines the application

Repository

Component Package

ScadaDemoWorkload.rtw.xml

ScadaDemo.cad.xml

Describes formally the set of transactions

ScadaManager

Logger

IOCard

TextConsole

ScadaEngine
ScadaDemo’s Components architecture

```
theManager: ScadaManager
displayPeriod = 1.0

theConsole: TextConsole
rwPort

TextRW

ScadaControl
scadaPort
consolePort
controlPort

theEngine: ScadaEngine
samplingPeriod = 0.01
loggingPeriod = 2.0

Logging
logPort
regPort

remoteSensor: IOCard
remoteSensor: IOCard

innerSensor: IOCard
innerSensor: IOCard
cardID = 0
cardID = 0

AnalogIO
analogPort

theManager: ScadaManager

theManager: ScadaManager

theConsole: TextConsole
rwPort

TextRW

ScadaControl
scadaPort
consolePort
controlPort

theEngine: ScadaEngine
samplingPeriod = 0.01
loggingPeriod = 2.0

Logging
logPort
regPort

remoteSensor: IOCard
remoteSensor: IOCard

innerSensor: IOCard
innerSensor: IOCard
cardID = 0
cardID = 0

AnalogIO
analogPort
```
New DnC descriptor: Workload Data Model
Application deployment

ScadaEngine package

- ComponentInterfaceDescription
  ScadaEngine.ccd.xml

- ComponentImplementation
  AdaScadaEngine.cid.xml

- ComponentImplementation
  Description
  AdaScadaEngine.cid.xml

Implementation
- Configuration
  Metadata (RT-D&C)

Implementation
- RT Configuration
  Metadata (RT-D&C)

<references>
AdaScadaEngine Code

<describes>
AdaScadaEngine Real-Time Model

Application description

- ScadaDemo.cad.xml
- ScadaDemoWorkload.rtw.xml

Platform description

- ScadaDemoPlatform.rtw.xml

Planner

Defines the deployment

DemoScada.cdp.xml

Application: Deployment plan
- Component instances assigned to nodes
- Selection of communication mechanisms
- Assignment of schedulability configuration properties
Component Implementation: D&C’s Component Implementation Description

- Example: AdaScadaEngine, an implementation of ScadaEngine for Ada-CCM
Extensions to the Component Implementation Description

Default values for:
- Real-Time configuration properties
- Implementation configuration properties

Real-time Configuration Properties

Implementation Configuration Properties

Real-Time Model reference
RT-CCM Components Technology

- Based on the Container/Component model => Schedulability features managed by the container to keep the opacity and reusability of components.

- LwCCM with added aspects for real-time support:
  - Activation ports used to require threads to the container
  - Synchronization ports used to required synchronization mechanisms to the container
  - Remote invocations implemented by means of connectors
  - Scheduling parameters managed by interceptors and services of the environment
AdaScadaEngine: Ada-CCM implementation of ScadaEngine

Schedulability configuration in RT-CCM: Activation and Synchronization Ports

Real-time configuration properties:
- For each periodic activation port
  - \(<activationPortName>Prty\)
  - \(<activationPortName>Period\)
- For each Mutex synchronization port:
  - \(<synchPortName>Ceiling\)
AdaScadaEngine’s real-time model

**Offered services**
- **<<SimpleOperation>>**
  - controlPort.getBufferedData
    - wcet=1.8E-06
    - acet=1.3E-06
    - bcet=1.0E-06
    - sharedResources=dataMtx

**Internal elements**
- **<<SharedResource>>**
  - dataMtx
    - ceiling=@dataMtxCeiling@
- **<<SimpleOperation>>**
  - PackStatisticDataAndReset
    - wcet=1.5E-06
    - acet=1.3E-06
    - bcet=1.1E-06
    - sharedResources = dataMtx

**Supported Transactions**
- **<<RegularTransaction>>**
  - LoggingTrans
    - params:
      - loggingDeadline: TIME_INTERVAL
    - Aggregated Resource
      - <<RegularSchedulingServer>>
        - loggingThServer
          - Priority=@loggingThPrty@
          - Preassigned = NO
    - CreateLoggingMsg_act
      - usage = PackStatisticDataAndReset
      - Log_act
        - usage = @logPort@.log
      - LoggingTrans
        - refEvent=loggingTrigger
        - period=@loggingThPeriod@
        - @loggingDeadline@

- **<<periodicPattern>>**
  - loggingTrigger
    - period=@loggingThPeriod@

- **<<HardGlobalDeadline>>**
  - refEvent=loggingTrigger
    - deadline=@loggingDeadline@
Extensions to the TargetDataModel

- RT-D&C defines descriptors of platform elements which are used to store their corresponding real-time models.
- Each element of the domain references and configures its corresponding descriptor.

- Descriptors of connection mechanisms can be stored also in the TargetDataModel.
- They are used to generate and configure the code and the real-time models of connectors.
Predictable Remote Invocations: Connectors RT Models

Diagram showing the interaction between Processor_A, Network, and Processor_B, highlighting the threads, activation modes, and processing times involved in remote method invocations.
ScadaDemo deployment

LocalProc: Node_MarteOS_PC_2_2GH

manager:: AdaScadaManager
displayPeriod = 1.0
displayThPeriod = 1.0
displayThPrty = default
displayMtxCeiling = default

close:: MaRTEConsole

engine:: AdaScadaEngine
samplingPeriod: 0.01
loggingPeriod: 2.0
samplingThPeriod: 0.01
loggingThPeriod: 2.0
loggingThPrty: default
dataMtxCeiling = default

localSensor:: PCI9111IOCard

cardID = 0
aiMutexCeiling = default
doMutexCeiling = default

Net: Ethernet_100MHz

RemoteProc: Node_MarteOS_PC_2_2GH

logger:: RAMLogger
notifyThPrty = default
logMtxCeiling = default

remoteSensor:: PCI9111IOCard

cardID = 0
aiMutexCeiling = default
doMutexCeiling = default
Domain description of the ScadaDemo platform

<domain name="scadaPlatform: Domain">
  <node name="LocalProc" source="platforms/MaRTEBaseProcessorDescription">
    <rtConfigProperty name="speedFactor">
      <value><factor>0.5</factor></value>
    </rtConfigProperty>
    <connection>net</connection>
  </node>
  <node name="RemoteProc" source="platforms/MaRTEBaseProcessorDescription">
    <rtConfigProperty name="speedFactor">
      <value><float>1.0</float></value>
    </rtConfigProperty>
    <connection>net</connection>
  </node>
  <interconnect name="Net" source="platforms/MaRTEBaseProcessorDescription">
    <connection>net</connection>
  </interconnect>
</domain>
Real-time design of the application

Analysis Context

ScadaDemoWorkload.rtw.xml
ScadaDemoPlatform.rtw.xml
DemoScada.cdp.xml

RT-Model Composer

Planner

Schedulability configuration (automatically assigned to the deployment plan)

Schedulability config. data

Schedulability report

ScadaEngine package

<<ComponentInterfaceDescription>>
ScadaEngine.ccd.xml

<<ComponentImplementation

<<ComponentImplementation

<<ComponentImplementation Description>>
AdaScadaEngine.cid.xml

<<references>>

AdaScadaEngine Code

<<describes>>

AdaScadaEngine Real-Time Model

ScadaDemo.mmd.xml

Mast Tools
Extensions to the Deployment Plan

- Assignment of real-time configuration properties
- Assignment of schedulability configuration properties
- Configuration of each connection between components
- Configuration of the domain elements

PlanConnectionDescription

DeploymentPlan

ConnectionConfiguration

QoSConfiguration

InstanceDeploymentDescription
Deployment Plan for ScadaDemo (before analysis)
MAST analysis and design tools

MAST System Description

Parser

Restrictions & Consistency Checks

Priority Ceilings

Blocking Times

Analysis:
- Offset-Based
- Offset-Based Unoptimized
- Holistic
- Varying Priorities
- Classic RM
- Parse

User Options:
- Verbose
- Calculate Ceilings
- Assign Priorities
- Analysis Technique
- Calculate Slacks
- Input and Output Files
- Priority Assignment Parameters

Priority Assignment:
- Monoprocessor
- HOPA
- Simulated Annealing

Print Results

MAST Results Description

New MAST System Description

HOPA Parameters

Annealing Parameters

Prague, November 2009

Patricia López Martínez
MAST results

-- GMAST Results --

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Slack</th>
</tr>
</thead>
<tbody>
<tr>
<td>manager.changetrans</td>
<td>Regular_Transaction</td>
<td>5.47%</td>
</tr>
<tr>
<td>manager.checkingtrans</td>
<td>Regular_Transaction</td>
<td>5.47%</td>
</tr>
<tr>
<td>manager.pollingtrans</td>
<td>Regular_Transaction</td>
<td>5.47%</td>
</tr>
<tr>
<td>controller.samplingtrans</td>
<td>Regular_Transaction</td>
<td>207.03%</td>
</tr>
</tbody>
</table>

-- Timing Results --

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Event</th>
<th>Referenced Event</th>
<th>Best Response</th>
<th>Worst Response</th>
<th>Hard Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>manager.changetrans</td>
<td>manager.changetrans.endchange</td>
<td>manager.changetrans.changetrigger</td>
<td>3.458E-04</td>
<td>0.019746</td>
<td>0.050000</td>
</tr>
<tr>
<td>manager.checkingtrans</td>
<td>manager.checkingtrans.endchecking</td>
<td>manager.checkingtrans.checkingtriger</td>
<td>0.006657</td>
<td>0.018857</td>
<td>0.020000</td>
</tr>
<tr>
<td>manager.pollingtrans</td>
<td>manager.pollingtrans.endpolling</td>
<td>manager.pollingtrans.pollingtrigger</td>
<td>5.216E-04</td>
<td>0.019756</td>
<td>0.100000</td>
</tr>
<tr>
<td>controller.samplingtrans</td>
<td>controller.samplingtrans.endsampling</td>
<td>controller.samplingtrans.samplingtrigger</td>
<td>1.098E-04</td>
<td>5.846E-04</td>
<td>0.001000</td>
</tr>
</tbody>
</table>
RT-D&C Application Development Process

Installation phase
- Component package
- Installer

Configuration phase
- Application requirements
  - Functional
  - Real-time
- RT-D&C Workload Model
- Component Component Assembly Description
- Planner

Planning phase
- D&C Domain Description
- Schedulability configuration
- RT-D&C Deployment plan

Preparation & Launching phase
- Executable code
- Executor

Repository of the development suite
- Component Functional metadata
- Component Real-Time metadata
- Platform metadata
- Component Code
D&C as roadmap for CBSE Tools

D&C

Installation tools

Planning tools

Assembling tools

Code Generation tools

Launching tools

Artifact

RT-Workload model

RT-Application model

RT-Platform model

RT-Analysis tools

RT-Design tools

Deployment criteria

Functional & Non-Functional requirements

Execution code

Platform schedulability Configuration data

Component schedulability Configuration data

Adapter and container code

Executable code partitions
Conclusions and Future Work

- Real-time design and analysis has been included in the process of development of component-based applications
  - Based on a reactive model of the system (like the one proposed in the SAM subprofile of MARTE)
- Supported by means of an extension of the D&C Specification.
  - Real-time metadata has been included in the components and applications descriptors. They can be used to:
    - Analyse the temporal behaviour of real-time component-based applications during the design phase
    - Configure automatically the application to guarantee its timing requirements
  - The extension has been defined at PIM level
    - A PSM has been defined for the Ada-CCM technology
    - Currently it is formulated by means of W3C-Schemas (XML technology)
  - The extension has been defined at the D&C Data Models dimension
- As future work
  - Formalizing the extension for the D&C Management models
  - Finishing the transformation tools
  - Trying to use MARTE SAM modelling concepts to describe the temporal behaviour of components