1. MELA Deployment

1.1 Compiling MELA

The “MELA” directory contains a MELA Maven "pom" project. The project has a set of Maven modules: MELA-AnalysisService, MELA-DataService, and MELA-Common. MELA-DataService is responsible of collecting monitoring information from existing monitoring systems, and storing it in a local monitoring repository. The MELA-AnalysisService queries monitoring data from the MELA-DataService structures, and analyzes it. MELA-Common contains the common data model used by the other two modules.

To compile the modules using Maven, from the MELA main directory, running "mvn clean package" will compile all MELA Modules. The MELA-DataService and MELA-Common compile in "jar" files, while MELA-AnalysisService compiles into a "war" file.

1.2 Configuring MELA

a. Configuring Monitoring Data Source

MELA-DataService can retrieve monitoring data from various data sources:

- Local Ganglia: monitoring data retrieved by connecting to a Ganglia running on the machine also running the MELA-DataService
- Remote Ganglia: monitoring data retrieved by issuing a Ganglia query trough SSH to a Ganglia installation running on a machine separate from the MELA-DataService
- JCatascopia
- Replay: in which monitoring data already collected is replayed to MELA.

To configure the data source, please set the appropriate options in the ”/MELA/MELA-AnalysisService/src/main/resources/config/Config.properties”. In future we plan to move this configuration to MELA- DataService, but currently it is located in MELA-AnalysisService.

Additional configurations are needed depending on the selected data source. For Local Ganglia or Remote Ganglia, the Ganglia port is requested, as Ganglia is queried by executing a “telnet localhost port”. For the “Remote Ganglia”, a SSH key is required, and the name of the OS user associated to the key, together with the IP of the machine hosting Ganglia (to which the SSH connection is opened). In case Replay is used, the monitoring sequence ID of the stored monitoring information needs to be specified.

b. Configuring Monitoring Data Collection Interval and Elasticity Analysis

Mela-AnalysisService has three configuration options: one for enabling elasticity analysis (in terms of elasticity space and pathway), and three for controlling the monitoring data access from the MELA-DataService. Please configure the appropriate properties in the ”/MELA/MELA-AnalysisService/src/main/resources/config/Config.properties” file.

1.3 Running MELA

a) The MELA-DataService must be started first, by running the MELA-DataService-1.0.jar from the maven "target" directory as a normal jar: "java -jar ./MELA-DataService-1.0.jar". This starts the data source back-end, to which the MELA-AnalysisService will connect.

b) Deploy the MELA-AnalysisService in an application server, by copying the MELA-AnalysisService-1.0.war from the "MELA-AnalysisService/target" directory. This starts the MELA RESTfull API, and MELA GUI. The MELA GUI is accessible at "applicationServerURL/MELA/" (example, for Tomcat, http://localhost:8080/MELA-AnalysisService-0.1-SNAPSHOT/). THE GUI was tested and is working on Google Chrome web browser, and does not work on Firefox. The MELA RESTful API endpoint will be "applicationServerURL /MELA-AnalysisService-0.1-SNAPSHOT/REST_WS".

2. Application Specific Configuration

MELA provides RESTful services for configuring it for a particular application, and thus all MELA configurations can be done through service calls.

2.1 Application Structure Description

MELA considers any cloud application as a service, and uses a representational model which divides Cloud Services in Service Topologies, which are composed of Service Units, which run in Virtual Machines members of Virtual Clusters.

In Listing 1 we show an application description example, which describes a CloudService having two topologies, a DataEndServiceTopology and an EventProcessingServiceTopology. The topologies are just logical concepts, and have no physical equivalent. Each topology has Service Units. For the detailed XML Schema of the application description format, check Appendix A.

The application description must be submitted to MELA issuing a "PUT" request towards the MELA "/servicedescription" service.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MonitoredElement id="CloudService" level="SERVICE">
  <MonitoredElement id="DataEndServiceTopology" level="SERVICE_TOPOLOGY">
        <MonitoredElement id="DataControllerServiceUnit" level="SERVICE_UNIT"/>
        <MonitoredElement id="DataNodeServiceUnit" level="SERVICE_UNIT"/>
  </MonitoredElement>
  <MonitoredElement id="EventProcessingServiceTopology" level="SERVICE_TOPOLOGY">
    <MonitoredElement id="LoadBalancerServiceUnit" level="SERVICE_UNIT"/>
    <MonitoredElement id="EventProcessingServiceUnit" level="SERVICE_UNIT"/>
  </MonitoredElement>
</MonitoredElement>
```

Listing 1: Application Structure Description Example

2.2 Metric Composition Rules Description

The second important configuration step is the specification of metric composition rules, issuing a "PUT" request towards the MELA "/metricscompositionrules" service. In Listing 2 we show a composition rule example, which applies a SUM operation over the "numberOfVMs" metric collected from all the Virtual Machine instances belonging to any Service unit, and creates a new metric also called "numberOfVMs" which will be associated with the Service Unit for which is computed. Full XML Schema of the composition rules description can be found in Appendix B.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CompositionRulesConfiguration>
 <MetricsCompositionRules>
    <CompositionRule TargetMonitoredElementLevel="SERVICE_UNIT">
       <ResultingMetric type="RESOURCE" measurementUnit="number" name="numberOfVMs"/>
       <Operation MetricSourceMonitoredElementLevel="VM" type="SUM">
       <ReferenceMetric type="RESOURCE" name="numberOfVMs"/>
       </Operation>
    </CompositionRule>
  </MetricsCompositionRules>
</CompositionRulesConfiguration>
```

Listing 2: Metric Composition Rules Description Example

2.2.1 Writing Composition Rules

A composition rule applies a set of composition operations over a set of operands.

2.2.1.1 Composition rule

a. For a composition rule one must first specify the "ResultingMetric", i.e. the type (RESOURCE, COST, QUALITY), name and measurement unit of the Metric resulting from the composition rule.

```
Example: <ResultingMetric type="RESOURCE" measurementUnit="ms" name="avgResponseTime"/>
```

b. The "TargetMonitoredElementLevel" must be specified, which defines at WHICH level in the application structure level (SERVICE, SERVICE_TOPOLOGY, SERVICE_UNIT, VIRTUAL_CLUSTER, and VM) the composition rule is applied. If ONLY the "TargetMonitoredElementLevel" is specified, the composition rule will be applied to all instances having the specified level (e.g., to all SERVICE_UNIT instances)

```
Example: <CompositionRule TargetMonitoredElementLevel="SERVICE_UNIT">
```

c. If a list of "TargetMonitoredElementID" are specified, this indicates particular elements from the application structure on which the composition rule will be applied.

Example:

```
<TargetMonitoredElementID> DataControllerServiceUnit </TargetMonitoredElementID>
<TargetMonitoredElementID> LoadBalancerServiceUnit </TargetMonitoredElementID>
```

d. The last element is the list of “CompositionOperation” specification, where we define the composition operation to be applied at the specified “TargetMonitoredElementLevel” over the optionally specified “TargetMonitoredElementID”

e. The metric resulting from applying the composition rule will be added to the monitoring data at the specified “TargetMonitoredElementLevel” and optionally specified “TargetMonitoredElementID”. If no optionally specified “TargetMonitoredElementID” are present, the metric will be added to all elements belonging to the specified level.

2.2.1.2 Composition Operation

A Composition Operation applies an operation over one or more operands, which can be values or metrics. Depending on the operation type, each operation is processed differently.

a.

b. For a composition operation, an operand can be:

i. A “value”: any statically defined value
Example: <Operation value="100" type="ADD">

ii. A “ReferenceMetric” , which indicates that the operation will be applied on value of the ReferenceMetric

Example:

```
<Operation MetricSourceMonitoredElementLevel="VM" type="AVG">
        <ReferenceMetric type="RESOURCE" name="cpuIdle"/>
</Operation>
```

c. For an operation involving a “ReferenceMetric” , one must also specify the “MetricSourceMonitoredElementLevel”, which indicates at which application structure level (SERVICE, SERVICE_TOPOLOGY, SERVICE_UNIT, VIRTUAL_CLUSTER, and VM) the reference metric will be found UNDER the monitored element over which the rule is applied. The “MetricSourceMonitoredElementLevel” can equal the “TargetMonitoredElementLevel”, or indicate any level belonging to the children of the “TargetMonitoredElementLevel”.

Example:

```
 <CompositionRule TargetMonitoredElementLevel="SERVICE_UNIT">
            <Operation MetricSourceMonitoredElementLevel="VM" type="AVG">
```

d. For refining which monitored element from the “*MetricSourceMonitoredElementLevel* “ is considered in extracting metric values, any number of “SourceMonitoredElementID” can be defined.

Example:

```
<Operation MetricSourceMonitoredElementLevel="SERVICE_UNIT" type="KEEP">
          <ReferenceMetric type="RESOURCE" measurementUnit="req/s" name="troughput"/>
          <SourceMonitoredElementID>LoadGenerator</SourceMonitoredElementID>
</Operation>
```

e. An operation can contain sub-operations, and the values returned by those sub-operations are used by the current operation, in a hierarchical manner.

Example:

```
<Operation value="100" type="ADD">
   <Operation value="-1" type="MUL">
            <Operation MetricSourceMonitoredElementLevel="VM" type="AVG">
                    <ReferenceMetric type="RESOURCE" name="cpuIdle"/>
             </Operation>
   </Operation>
 </Operation>
```

f. For any operation, it is important to specify the type, as the operations are processed differently depending on their type

- SUM
  - Intended to sum the values retrieved from a “ReferenceMetric”
  - Example:

```
<CompositionRule TargetMonitoredElementLevel="SERVICE_UNIT">
      <ResultingMetric type="RESOURCE" measurementUnit="numberOf"
name="numberOfVMs"/>
        <Operation MetricSourceMonitoredElementLevel="VM" type="SUM">
            <ReferenceMetric type="RESOURCE" name="numberOfVMs"/>
         </Operation>
       </CompositionRule>
```

- AVG
  - Applied similar to SUM, extracts the average.
- MAX / MIN
  - Applied similar to SUM, extracts the maximum or minimum value.
- DIV/MUL/ADD/SUB
  - Divide, Multiply, Subtract From, Add To, the value of the first operand the value of the second operand
  - *Example:*

```
<Operation value="-1" type="MUL">
      <Operation MetricSourceMonitoredElementLevel="VM" type="AVG">
      <ReferenceMetric type="RESOURCE" name="cpuIdle"/>
</Operation>
```

- SET_VALUE

    - Used in creating metric values
    - *Example:*

```
<CompositionRule TargetMonitoredElementLevel="VM">
      <ResultingMetric type="RESOURCE" measurementUnit="ms"
name="numberOfVMs"/>
       <Operation value="1" type="SET_VALUE"/>
 </CompositionRule>
```

- CONCAT
- UNION
- KEEP/ KEEP_LAST/ KEEP_FIRST

2.2.2 Historical Metrics Composition Rules

As MELA can be configured to aggregate also metrics over time (e.g., collect metrics every 5 seconds, but aggregate every 5 monitoring frames), Historical Metric Aggregation rules need to be defined for such scenario in the CompositionRuleConfiguration. They MUST be defined ONLY for the VM level metrics. So, for each SERVICE_UNIT historical composition rules need to be defined for each metric retrieved from the VM level.

```
Example:
<CompositionRule TargetMonitoredElementLevel="SERVICE_UNIT">
        <TargetMonitoredElementID>EventProcessingServiceUnit</TargetMonitoredElementID>
        <ResultingMetric type="RESOURCE" measurementUnit="no" name="responseTime"/>
        <Operation MetricSourceMonitoredElementLevel="VM" type="AVG">
           <ReferenceMetric type="RESOURCE" name="responseTime"/>
        </Operation>
</CompositionRule>
```

2.3 Application Structure Update after Scaling Action

As MELA is designed to support controllers of elastic services, it does not know when a new Virtual Machine has been allocated/deallocated for a particular Service Unit. To keep MELA up to date and ensure it collects monitoring information also for the newly added Virtual Machines, the application structure needs to be kept updated by the MELA user, by submitting to MELA an updated application description after each scaling in/out action. The updated description must contain the Virtual Machine and Virtual Cluster instances which are currently active, and is to be submitted issuing a “POST” request towards the MELA “/servicedescription” service.

Note:

For any research work in which MELA is used, please cite the following articles:

- Daniel Moldovan, Georgiana Copil, Hong-Linh Truong, Schahram Dustdar, MELA: Monitoring and Analyzing Elasticity of Cloud Services), 5'th International Conference on Cloud Computing,

CloudCom. Bristol, UK, 2-5 December, 2013 Accepted
http://www.infosys.tuwien.ac.at/research/viecom/publications.html

For more details and other prototypes check
http://www.infosys.tuwien.ac.at/research/viecom/prototypes.html

3. Appendix A – Cloud Application Description XML Schema

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="MonitoredElement" type="monitoredElement"/>

  <xs:element name="MonitoredElementLevel" type="monitoredElementLevel"/>

  <xs:complexType name="monitoredElement">
    <xs:sequence>
      <xs:element ref="MonitoredElement" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required"/>
    <xs:attribute name="name" type="xs:string" use="required"/>
    <xs:attribute name="level" type="monitoredElementLevel" use="required"/>
  </xs:complexType>

  <xs:simpleType name="monitoredElementLevel">
    <xs:restriction base="xs:string">
      <xs:enumeration value="SERVICE"/>
      <xs:enumeration value="SERVICE_TOPOLOGY"/>
      <xs:enumeration value="SERVICE_UNIT"/>
      <xs:enumeration value="VM"/>
      <xs:enumeration value="VIRTUAL_CLUSTER"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

4. Appendix B – Composition Rules Description XML Schema

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="CompositionOperationType" type="compositionOperationType"/>

  <xs:element name="CompositionRule" type="compositionRule"/>

  <xs:element name="CompositionRulesConfiguration" type="compositionRulesConfiguration"/>
  <xs:element name="Metric" nillable="true" type="xs:anyType"/>

  <xs:element name="MetricsCompositionRules" type="compositionRulesBlock"/>

  <xs:element name="MonitoredElementLevel" type="monitoredElementLevel"/>

  <xs:element name="Operation" type="compositionOperation"/>

  <xs:complexType name="compositionRulesConfiguration">
    <xs:sequence>
      <xs:element ref="MetricsCompositionRules" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="TargetServiceID" type="xs:string" use="required"/>
  </xs:complexType>

  <xs:complexType name="compositionRulesBlock">
    <xs:sequence>
      <xs:element ref="CompositionRule" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="compositionRule">
    <xs:sequence>
      <xs:element name="TargetMonitoredElementID" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="ResultingMetric" type="metric"/>
      <xs:element ref="Operation"/>
    </xs:sequence>
    <xs:attribute name="TargetMonitoredElementLevel" type="monitoredElementLevel"
use="required"/>
  </xs:complexType>

  <xs:complexType name="metric">
    <xs:sequence/>
    <xs:attribute name="name" type="xs:string" use="required"/>
    <xs:attribute name="measurementUnit" type="xs:string" use="required"/>
    <xs:attribute name="type" type="metricType" use="required"/>
  </xs:complexType>

  <xs:complexType name="compositionOperation">
    <xs:sequence>
      <xs:element name="ReferenceMetric" type="metric" minOccurs="0"/>
      <xs:element name="SourceMonitoredElementID" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element ref="Operation" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="type" type="compositionOperationType" use="required"/>
    <xs:attribute name="value" type="xs:string"/>
    <xs:attribute name="MetricSourceMonitoredElementLevel" type="monitoredElementLevel"
use="required"/>
  </xs:complexType>

  <xs:simpleType name="monitoredElementLevel">
    <xs:restriction base="xs:string">
      <xs:enumeration value="SERVICE"/>
      <xs:enumeration value="SERVICE_TOPOLOGY"/>
      <xs:enumeration value="SERVICE_UNIT"/>
      <xs:enumeration value="VM"/>
      <xs:enumeration value="VIRTUAL_CLUSTER"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="metricType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="RESOURCE"/>
      <xs:enumeration value="COST"/>
      <xs:enumeration value="QUALITY"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="compositionOperationType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="SUM"/>
      <xs:enumeration value="MAX"/>
      <xs:enumeration value="MIN"/>
      <xs:enumeration value="AVG"/>
      <xs:enumeration value="DIV"/>
      <xs:enumeration value="ADD"/>
      <xs:enumeration value="SUB"/>
      <xs:enumeration value="MUL"/>
      <xs:enumeration value="CONCAT"/>
      <xs:enumeration value="UNION"/>
      <xs:enumeration value="KEEP"/>
      <xs:enumeration value="KEEP_LAST"/>
      <xs:enumeration value="KEEP_FIRST"/>
      <xs:enumeration value="SET_VALUE"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

5. Appendix C – Cloud Application Description Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<MonitoredElement id="CloudService" level="SERVICE">
```

```
  <MonitoredElement id="DataEndServiceTopology" level="SERVICE_TOPOLOGY">
    <MonitoredElement id="DataControllerServiceUnit" level="SERVICE_UNIT">
      <MonitoredElement id="10.99.0.147" level="VM"/>
    </MonitoredElement>
    <MonitoredElement id="DataNodeServiceUnit" level="SERVICE_UNIT">
      <MonitoredElement id="10.99.0.83" level="VM"/>
    </MonitoredElement>
  </MonitoredElement>
  <MonitoredElement id="EventProcessingServiceTopology" level="SERVICE_TOPOLOGY">
    <MonitoredElement id="LoadBalancerServiceUnit" level="SERVICE_UNIT">
      <MonitoredElement id="10.99.0.151" level="VM"/>
    </MonitoredElement>
    <MonitoredElement id="EventProcessingServiceUnit" level="SERVICE_UNIT">
      <MonitoredElement id="10.99.0.62" level="VM"/>
    </MonitoredElement>
  </MonitoredElement>
</MonitoredElement>
```

6. Appendix D – Composition Rules Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CompositionRulesConfiguration>
  <MetricsCompositionRules>

    <CompositionRule TargetMonitoredElementLevel="SERVICE_UNIT">
      <TargetMonitoredElementID>LoadBalancerServiceUnit</TargetMonitoredElementID>
      <ResultingMetric type="RESOURCE" measurementUnit="no" name="numberOfClients"/>
      <Operation MetricSourceMonitoredElementLevel="VM" type="AVG">
        <ReferenceMetric type="RESOURCE" name="activeConnections"/>
      </Operation>
    </CompositionRule>

    <CompositionRule TargetMonitoredElementLevel="SERVICE_UNIT">
      <TargetMonitoredElementID>EventProcessingServiceUnit</TargetMonitoredElementID>
      <ResultingMetric type="RESOURCE" measurementUnit="ms" name="responseTime"/>
      <Operation MetricSourceMonitoredElementLevel="VM" type="AVG">
        <ReferenceMetric type="RESOURCE" name="responseTime"/>
      </Operation>
    </CompositionRule>

    <CompositionRule TargetMonitoredElementLevel="SERVICE_UNIT">
      <TargetMonitoredElementID>EventProcessingServiceUnit</TargetMonitoredElementID>
      <ResultingMetric type="RESOURCE" measurementUnit="operations/s" name="throughput"/>
      <Operation MetricSourceMonitoredElementLevel="VM" type="AVG">
        <ReferenceMetric type="RESOURCE" name="throughput"/>
      </Operation>
    </CompositionRule>
```

```xml
<CompositionRule TargetMonitoredElementLevel="VM">
   <ResultingMetric type="RESOURCE" measurementUnit="ms" name="numberOfVMs"/>
   <Operation value="1" type="SET_VALUE"/>
</CompositionRule>

 <CompositionRule TargetMonitoredElementLevel="SERVICE_UNIT">
   <ResultingMetric type="RESOURCE" measurementUnit="ms" name="numberOfVMs"/>
   <Operation MetricSourceMonitoredElementLevel="VM" type="SUM">
      <ReferenceMetric type="RESOURCE" name="numberOfVMs"/>
   </Operation>
</CompositionRule>

<CompositionRule TargetMonitoredElementLevel="SERVICE_TOPOLOGY">
   <TargetMonitoredElementID>EventProcessingServiceTopology</TargetMonitoredElementID>
   <ResultingMetric type="RESOURCE" measurementUnit="no" name="numberOfClients"/>
   <Operation MetricSourceMonitoredElementLevel="SERVICE_UNIT" type="KEEP">
      <ReferenceMetric type="RESOURCE" measurementUnit="no" name="numberOfClients"/>
      <SourceMonitoredElementID>LoadBalancerServiceUnit</SourceMonitoredElementID>
   </Operation>
</CompositionRule>

<CompositionRule TargetMonitoredElementLevel="SERVICE_TOPOLOGY">
   <TargetMonitoredElementID>EventProcessingServiceTopology</TargetMonitoredElementID>
   <ResultingMetric type="RESOURCE" measurementUnit="ms" name="responseTime"/>
   <Operation MetricSourceMonitoredElementLevel="SERVICE_UNIT" type="AVG">
      <ReferenceMetric type="RESOURCE" name="responseTime"/>
      <SourceMonitoredElementID>EventProcessingServiceUnit</SourceMonitoredElementID>
   </Operation>
</CompositionRule>

<CompositionRule TargetMonitoredElementLevel="SERVICE_TOPOLOGY">
   <TargetMonitoredElementID>EventProcessingServiceTopology</TargetMonitoredElementID>
   <ResultingMetric type="RESOURCE" measurementUnit="operations/s" name="throughput"/>
   <Operation MetricSourceMonitoredElementLevel="SERVICE_UNIT" type="SUM">
      <ReferenceMetric type="RESOURCE" name="throughput"/>
      <SourceMonitoredElementID>EventProcessingServiceUnit</SourceMonitoredElementID>
   </Operation>
</CompositionRule>

<CompositionRule TargetMonitoredElementLevel="SERVICE_TOPOLOGY">
   <ResultingMetric type="RESOURCE" measurementUnit="$" name="cost"/>
   <Operation value="0.12" type="MUL">
      <Operation MetricSourceMonitoredElementLevel="SERVICE_UNIT" type="SUM">
         <ReferenceMetric type="RESOURCE" name="numberOfVMs"/>
      </Operation>
   </Operation>
</CompositionRule>
```

```
      <CompositionRule TargetMonitoredElementLevel="SERVICE">
         <ResultingMetric type="RESOURCE" measurementUnit="$" name="cost/client/h"/>
         <Operation MetricSourceMonitoredElementLevel="SERVICE" type="DIV">
            <Operation MetricSourceMonitoredElementLevel="SERVICE_TOPOLOGY" type="SUM">
               <ReferenceMetric type="RESOURCE" name="cost"/>
            </Operation>
            <Operation MetricSourceMonitoredElementLevel="SERVICE_TOPOLOGY" type="KEEP">
               <ReferenceMetric type="RESOURCE" name="numberOfClients"/>

<SourceMonitoredElementID>EventProcessingServiceTopology</SourceMonitoredElementID>
            </Operation>
         </Operation>
      </CompositionRule>

   </MetricsCompositionRules>

   <HistoricalMetricsCompositionRules>
      <CompositionRule TargetMonitoredElementLevel="SERVICE_UNIT">
         <TargetMonitoredElementID>LoadBalancerServiceUnit</TargetMonitoredElementID>
         <ResultingMetric type="RESOURCE" measurementUnit="no" name="activeConnections"/>
         <Operation MetricSourceMonitoredElementLevel="VM" type="AVG">
            <ReferenceMetric type="RESOURCE" name="activeConnections"/>
         </Operation>
      </CompositionRule>
      <CompositionRule TargetMonitoredElementLevel="SERVICE_UNIT">
         <TargetMonitoredElementID>EventProcessingServiceUnit</TargetMonitoredElementID>
         <ResultingMetric type="RESOURCE" measurementUnit="no" name="throughput"/>
         <Operation MetricSourceMonitoredElementLevel="VM" type="AVG">
            <ReferenceMetric type="RESOURCE" name="throughput"/>
         </Operation>
      </CompositionRule>
      <CompositionRule TargetMonitoredElementLevel="SERVICE_UNIT">
         <TargetMonitoredElementID>EventProcessingServiceUnit</TargetMonitoredElementID>
         <ResultingMetric type="RESOURCE" measurementUnit="no" name="responseTime"/>
         <Operation MetricSourceMonitoredElementLevel="VM" type="AVG">
            <ReferenceMetric type="RESOURCE" name="responseTime"/>
         </Operation>
      </CompositionRule>
   </HistoricalMetricsCompositionRules>
</CompositionRulesConfiguration>
```