SLA Compliance Monitoring Through Semantic Processing

Luigi Coppolino, Danilo De Mari, Luigi Romano
(luigi.coppolino, danilo.demari, luigi.romano)@epsilonline.com

Valerio Vianello
(valerio.vianello@uniparthenope.it)

Third Service Level Agreements in Grids Workshop
(in conjunction with IEEE Grid 2010)

Bruxelles (Belgium) – October 26, 2010
Overview

Service level events

Network level messages

System level events

Specification

Third Service Level Agreements in Grids
Workshop
Bruxelles, October 26, 2010
Overview

Service level events

Network level messages

System level events

SLA

- System Logs
- Application logs
- Network probes
- ...

Specification

Third Service Level Agreements in Grids Workshop Bruxelles, October 26, 2010
“Even with network monitoring tools in place, a staggering 72.6% first learn about performance problems from end-user calls to the help desk, and another 82.3% said employee complaints usually are the first they hear of slowdowns on their networks.”

A use case from the telco domain 1/2
A use case from the telco domain 2/2

• What we get ...

---

Third Service Level Agreements in Grids
Workshop
Bruxelles, October 26, 2010

---

Enabling People to Share Information and to Learn over the Network
A use case from the telco domain 2/2

• What we get …

• What an SLA is related to …

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Goal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time of credit renewal</td>
<td>&lt;=10 minutes</td>
</tr>
<tr>
<td>% of transaction to be completed within 10 minutes</td>
<td>&gt;95%</td>
</tr>
</tbody>
</table>
Contribution

• We propose a solution to closing the gap between business process analysts and available low level data.

• The key idea is about addressing the problem by moving toward the concept of semantic data processing.

• Leveraging on business analyst knowledge about service requirements and on domain specialist knowledge about the system, the data are elaborated based on their semantic.
Main issues in semantic processing

• Processing an huge amount of data in real time.

• Providing an abstraction layer to business analysts

• Decoupling the knowledge of the system from the knowledge of the business.

• Providing evidence of breaches of the agreed QoS levels
An architecture for semantic processing
An architecture for semantic processing
System usage

• Configuration stage:
  
  – At this stage a team of domain specialists must provide the framework with a formalized description of the specific domain.

• Operational stage:
  
  – At this stage business analysts can query the system to retrieve business related measures such as the value of some Key Performance Indicators (PKIs) of the business.
Configuration

• The specific domain description must include:
  – A **concept hierarchy** that represents a static view of the domain under analysis.
    • the leaves of the hierarchy represent events that are actually monitorable into the system
    • the topmost elements are abstract business level concepts that could not be directly monitorable
  – The **relationships** among such concepts.
    • The occurrence of topmost elements can be inferred, using the defined relationships, by other monitorable events, typically described at lower levels into the hierarchy.
Operation

• The framework can be feed with:
  - Analyst query:
    • The framework provide the analyst with a list of concepts retrieved from the domain description and the analyst may graphically build the query by combining SQL-like predicates based on the offered concepts through a web based GUI.
  - Automatic query:
    • The framework can also be fed with meta queries directly extracted from SLAs (described in WS-agreement) of a service using a Query Extractor component.
The Query Translator Component

- The Query Translator component is in charge of converting high level META-Query, submitted by analysts, into query that can be executed by a Complex Event Processing Engine.

```
<?xml version="1.0" encoding="UTF-8"?>
<queryXML>
  <Query>
    <select>count(*) as Tot</select>
    <from>
      <!-- join or pattern -->
      <type>pattern</type>
      <!-- pattern -->
      <object>RechargeOK</object>
      <delay>10 m</delay>
      <!-- observe interval -->
      <timeInterval>1 day</timeInterval>
    </from>
  </Query>
</queryXML>
```

(a)

```
select count(*) as Tot from pattern [(every H80= msg(OP="H80")
  timer:interval(10 m) and H81OK = msg(OP="H81OK" and TID=H80.TID)
  timer:interval(10 m) and H82 = msg(OP="H82" and TID=H80.TID)
  timer:interval(10 m) and H83OK = msg(OP="H83OK" and TID=H80.TID)
  where timer:within(1 day))]
```

(b)

Figure 4. (a) Meta query: "number of successful recharges per day" (b) Executable query: "number of successful recharges per day"
Translation process

Figure 5. (a) Meta query: "number of recharge in a day" (b) Executable query: "number of recharge in a day"
Conclusions and Future Work

• In this work we have presented a system allowing SLA compliance monitoring.
• The system allows to express constraints and queries at an abstraction level close to the business process.
• The high level concepts are then translated to system level ones by inferring a domain ontology provided at configuration time.
• At the moment the monitoring entity is considered a trusted third party, in the future we plan to port it on the cloud and to make it trustworthy by design. …lots of conflicts (privacy, security, …)
Questions and Answers...

Q & A

Thank you!

Luigi Coppolino
Espsilon srl
luigi.coppolino@epsilononline.com