Towards Self-managed Pervasive Middleware using OWL/SWRL ontologies

Weishan Zhang
Klaus Marius Hansen

University of Aarhus

IST-2005-034891
Plan

- Motivation
- Dynamic context modeling facilitating self-management
- Self-management Ontologies structure
  - Self-management rules with SWRL
- Architecture of semantic web based self-management
- Evaluations
Motivation

• Context awareness based self-management
• Dynamic contexts are critical for self-management

This leads to the design:
• SeMaPS: a set of Self-Management for Pervasive Service context ontologies, that considering dynamic contexts
• Three layered Self-management
Dynamicity reporting for self-management

- State change reporting. State machines are used to report device state changes as events through the Hydra Event Manager.
- Web service request/reply reporting. The requests and replies (and their associated data) can be used to analyse the runtime structure of the Hydra systems. Here the Flamenco/Probe (IPSniffer) is used.

Dynamic contexts should reflect this dynamicity of pervasive service systems at run time

- Limbo (a pervasive service compiler) is used to generate needed code to report invocations and state machines.
Web service call reporting
SeMaPS ontologies structure
Component ontology

Component
- factory: String
- enabled: Boolean
- immediate: Boolean
- componentName: String
- property: Instance* Property
- properties: Instance* ExternalProperties
- service: Instance* Service
- reference: Instance* Interface
- isOfComponentType: Instance* ComponentType
- implementation: Instance Implementation

Service
- serviceFactory: Boolean
- provide: Instance* Interface

Interface
- interfaceName: String

Implementation
- implementationClass: String
- reference: Instance Interface

Reference
- bind: String
- target: String
- cardinity: String
- policy: String
- referenceName: String
- unbind: String

MRC 2008
State machine and FlamencoProbe ontology
Self-management rule specification with SWRL

- A simple diagnosis example

```swrl
device:FlowMeter(?device) ∧
device:hasStateMachine(?device, ?statemachine) ∧
statemachine:hasStates(?statemachine, ?state) ∧
statemachine:doActivity(?state, ?action) ∧
statemachine:actionResult(?action, ?result) ∧
box:ishNumeric(?result) ∧
swrl:greaterThan(?result, 18.0)
→ device:currentMalfunction(device:Flowmeter, error:PumpBroken)
```
Self-management rule specification with SWRL

- A not simple example for checking current configuration

```swrl
ComponentBased(?con) ∧
hasComponent(?con, ?comp1) ∧
osgcomponent:reference(?comp1, ?ref1) ∧
 osgcomponent:cardinality(?ref1, ?card1) ∧
 svglib:containsIgnoreCase(?card1, "M") ∧
 osgcomponent:interface(?ref1, ?inter1) ∧
 osgcomponent:interfaceName(?inter1, ?name1) ∧
 hasComponent(?con, ?comp2) ∧
 osgcomponent:service(?comp2, ?ser2) ∧
 osgcomponent:provider(?ser2, ?inter2) ∧
 osgcomponent:interfaceName(?inter2, ?name2) ∧
 svglib:equal(?name1, ?name2) ∧
 → swrl:selectDistinct(?con, ?comp1, ?comp2, ?name1, ?name2) ∧
 swrl:select("validConfiguration")
```
Self-management rule specification with SWRL

• A complex example
Architecture of the semantic web based self-management

MRC 2008
Evaluations

• Using rule group
• Executing all rules

<table>
<thead>
<tr>
<th>Update</th>
<th>Inferring Time</th>
<th>AfterEvent TillInferred</th>
</tr>
</thead>
<tbody>
<tr>
<td>328</td>
<td>328</td>
<td>328</td>
</tr>
<tr>
<td>297</td>
<td>281</td>
<td>297</td>
</tr>
<tr>
<td>297</td>
<td>297</td>
<td>297</td>
</tr>
<tr>
<td>297</td>
<td>281</td>
<td>297</td>
</tr>
<tr>
<td>344</td>
<td>344</td>
<td>344</td>
</tr>
</tbody>
</table>

Performance after rule grouping

<table>
<thead>
<tr>
<th>Update</th>
<th>Inferring Time</th>
<th>AfterEvent TillInferred</th>
</tr>
</thead>
<tbody>
<tr>
<td>843</td>
<td>843</td>
<td>843</td>
</tr>
<tr>
<td>906</td>
<td>906</td>
<td>906</td>
</tr>
<tr>
<td>922</td>
<td>906</td>
<td>922</td>
</tr>
<tr>
<td>719</td>
<td>719</td>
<td>719</td>
</tr>
<tr>
<td>953</td>
<td>938</td>
<td>938</td>
</tr>
</tbody>
</table>

Protege 3.4 Build 130, JVM 1.6.02-b06, Heap memory is 266M, Windows XP SP3. The hardware platform is: Thinkpad T61P T7500 2.2G CPU, 7200rpm hard disk, 2G DDR2 RAM. The time measurement is in millisecond. The size of DeviceRule ontology is 238,824 bytes, and contains 20 rules, including 6 rules for the Pig system, 12 generic rules which can be used in a number of domains, 3 rules (2 are shared with Pig rules) for the Weather Station, and 1 rule for FlamencoProbe related rules which is the biggest rule in the DeviceRule ontology.
Thank you..