Micis - A Model-Integrated Clinical Information System
Janos Mathe, Jan Werner, Yonghwan Lee, Akos Ledeczi, Brad Malin, Janos Sztipanovits
Institute for Software Integrated Systems, Vanderbilt University

Goal
Develop a Modeling and Simulation Platform for Health Information Systems
Platform is suitable for
• Modeling and model-based integration of Health Information Systems (HIS) - including Patient Portals (PP) - providing access to Electronic Medical Records (EMR)
• Performing security and privacy analysis using model verification and simulation-based testing
• Providing mapping and deployment to standard SOA execution platforms

Approach
1) Development of abstractions in Domain-Specific Modeling Languages (DSMLs)
2) Construction of the models: capturing the key elements of operation
3) Translation (interpretation) of models
4) Execution and simulation of models

Results
• Built a testbed for Health Information Systems, which includes a SOA-based execution environment, and is extended with policy evaluation and enforcement capabilities using an embedded PROLOG engine.
• Developed and evaluated experimental modeling languages for three different CIS applications.
• Developed model translators for translating architecture models into PROLOG clauses to enable context dependent evaluation of privacy and security policies.
• Built a component library of web services for prototyping experimental HISs.

Architecture of MICIS

Applications Layer
• Integration of medical applications that use sensitive medical information
• Applications communicate by using web services standards (OASIS, WC3)
• Utilization of already available services is possible

Deployment Architecture

<table>
<thead>
<tr>
<th>SOA Execution (A)</th>
<th>SOA Execution (B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache ODE v1.1 (or higher)</td>
<td>Oracle BPM Process Manager v10g</td>
</tr>
<tr>
<td>Axis2 v1.3 (or higher)</td>
<td>SunXACML Axis2 handler</td>
</tr>
<tr>
<td>Tomcat Apache v5.5 (or higher)</td>
<td>none</td>
</tr>
</tbody>
</table>

Policy Enforcement

<table>
<thead>
<tr>
<th>Policy Enforcement (A1)</th>
<th>Policy Enforcement (A2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWI-Prolog Axis2 handler</td>
<td>SunXACML Axis2 handler</td>
</tr>
</tbody>
</table>

Operating System
Windows XP (or higher) or Linux (e.g. Fedora Core 8 / RHEL5)

Hardware
Regular desktop PC

Policies
Policy representation, decision and enforcement
• Modular Policy Enforcement Point integrated into a Web Service container
• PEP supports XACML and Prolog based Policy Decision Points
• Policies and policy descriptions generated from the models
• Support of past events using stored facts
• Support of future events using obligation mechanism