A POLICY-BASED DIALOGUE SYSTEM FOR PHYSICAL ACCESS CONTROL

STIDS 2012
GEORGE MASON UNIVERSITY
10/25/2012

MOHAMMAD ABABNEH (George Mason University)
DUMINDA WIJESEKERA (George Mason University)
JAMES BRET MICHAEL (Naval Postgraduate School)
Problem Definition

- Interactive voice (i.e. Dialog) systems are becoming popular, such as
  - Apple’s Siri, Google’s Android S-Voice, Microsoft Windows Speech Recognition
- Interactive voice dialogs are hardcoded using technologies like VoiceXML
- Dialogs eventually access or update some information – they are governed by policies
- **Research objective:** To develop a means for using policies to govern interactive voice-based dialogs
Outline of Presentation

- Requirements
- Potential Applications
- A Sample Dialog
- System Architecture
- Background
  - VoiceXML
  - XACML
- Creating Policy Controlled Dialogs
- Architectural Integration of Dialogs and Policies
- Future Work and Framework Improvement
- Summary
Requirements

- In order to govern dialogs, we need 3 components:
  - A Dialog System, Policy Framework and Use Cases
- **Dialog System:** VoiceXML
- **Policy Framework:** XACML (eXtensible Access Control Markup Language).
- **Use Cases:** Physical Access Control to Smart Buildings
- **What would we like to do?**
  - Input: Policy and user’s presence information
  - Runtime: Dialog system automatically produces a series of questions and answers that, if correctly answered, permit the requester to enter the secure facility or smart building.
- **Novelty:** Generate questions appropriate for the requester and the operating environment.
Potential Applications

- Mobile computing
- Disaster response
- Weapon systems
- Electronic commerce
- Controlling equipment
- Physical access control
## A Sample Dialogue for Physical Access

<table>
<thead>
<tr>
<th></th>
<th>User/System</th>
<th>Hello</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S</td>
<td>Welcome, Please Say who you are?</td>
</tr>
<tr>
<td>2</td>
<td>U</td>
<td>I am User01 (Alice)</td>
</tr>
<tr>
<td>3</td>
<td>S</td>
<td>Please say your Password or enter it using the key pad</td>
</tr>
<tr>
<td>4</td>
<td>U</td>
<td>Pass01</td>
</tr>
<tr>
<td>5</td>
<td>S</td>
<td>Ok, I got that. Now tell me Why do you need to access the building (Role)</td>
</tr>
<tr>
<td>6</td>
<td>U</td>
<td>Professor</td>
</tr>
<tr>
<td>7</td>
<td>S</td>
<td>What is your office number</td>
</tr>
<tr>
<td>8</td>
<td>U</td>
<td>4429</td>
</tr>
<tr>
<td>9</td>
<td>S</td>
<td>In which floor is the dean’s meeting room</td>
</tr>
<tr>
<td>10</td>
<td>U</td>
<td>5th</td>
</tr>
<tr>
<td>11</td>
<td>S</td>
<td>What is the time on the clock to your right</td>
</tr>
<tr>
<td>12</td>
<td>U</td>
<td>7:30</td>
</tr>
<tr>
<td>13</td>
<td>S</td>
<td>Are you sure</td>
</tr>
<tr>
<td>14</td>
<td>U</td>
<td>Yes</td>
</tr>
<tr>
<td>15</td>
<td>S</td>
<td>Is it raining outside?</td>
</tr>
<tr>
<td>16</td>
<td>U</td>
<td>Yes</td>
</tr>
<tr>
<td>17</td>
<td>S</td>
<td>Is it cold or warm?</td>
</tr>
<tr>
<td>18</td>
<td>U</td>
<td>Cold</td>
</tr>
<tr>
<td>19</td>
<td>S</td>
<td>Is the temperature below 40 degrees?</td>
</tr>
<tr>
<td>20</td>
<td>U</td>
<td>No</td>
</tr>
<tr>
<td>21</td>
<td>S</td>
<td>Can you estimate the range that the temperature might be in?</td>
</tr>
<tr>
<td>22</td>
<td>U</td>
<td>50 to 60</td>
</tr>
<tr>
<td>23</td>
<td>S</td>
<td>Ok, granted access</td>
</tr>
</tbody>
</table>
System Architecture

- Intended for audio dialogues with
  - Synthesized speech, digitized audio, recognition of spoken and
  - Dual Tone Multi-Frequency (DTMF) key inputs
  - Recording of spoken input, etc.

- VXML interfaces between user and Web, using a voice interface.

- Purpose: To bring the advantages of web-based development and content delivery to interactive voice response (IVR) applications
VoiceXML Functionality
<?xml version="1.0" encoding="UTF-8"?>
<vxml xmlns="http://www.w3.org/2001/vxml"
xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
xsi:schemaLocation="http://www.w3.org/2001/vxml
http://www.w3.org/TR/voicexml20/vxml.xsd"
version="2.0">
<form>
<field name="e-mail">
<prompt>What is your e-mail address?</prompt>
<br/>&lt;grammar src="email.grxml"
type="application/srgs+xml"/&gt;
</field>
</form>
</vxml>
XACML is an OASIS standard XML-based language for specifying access control policies.

Access decisions to resources are made based on requester’s (Subject) and environment attributes.

Data-Flow Diagram
<?xml version="1.0" encoding="UTF-8"?>
  <Description>Med Example Corp access control policy</Description>
  <Target/>
  <Rule Effect="Permit" RuleId="urn:oasis:names:tc:xacml:2.0:example:SimpleRule1">
    <Description>Any subject with an e-mail name in the med.example.com domain can perform any action on any resource.</Description>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-match">
            <AttributeValue DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name">@med.example.com</AttributeValue>
            <SubjectAttributeDesignator DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name" AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"/>
          </SubjectMatch>
        </Subject>
      </Subjects>
    </Target>
  </Rule>
</Policy>

<Request>
  <Subject>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id" DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name">
      <AttributeValue>mababneh@med.example.com</AttributeValue>
    </Attribute>
    <Attribute AttributeId="group" DataType="http://www.w3.org/2001/XMLSchema#string" Issuer="admin@gmu.edu">
      <AttributeValue>Developers</AttributeValue>
    </Attribute>
  </Subject>
  <Resource>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" DataType="http://www.w3.org/2001/XMLSchema#anyURI">
      <AttributeValue>http://server.example.com/code/docs/developer-guide.html</AttributeValue>
    </Attribute>
  </Resource>
  <Action>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>read</AttributeValue>
    </Attribute>
  </Action>
</Request>
Creating Policy Controlled Dialogs

- Transform an access control policy into a voice dialog

**Method:**
- Read and transformed rules of the policy are into VoiceXML blocks and forms.

**Details:**
- The entire policy is parsed using a DOM parser
- Every relevant rule is converted into a question
  - How: translate the rule into a VXML block
  - Voice user interface translating text to speech (TTS), posing the question to the user.
  - Waiting for the user’s response through voice recognition.
  - Translate voice back to text
  - Check against the XACML policy for accuracy
Dialogue-Policy Integration

Voice Recognition

Subject Attributes

Dialog Management

XACML Request

XACML to VoiceXML

Voice Recognition Enhancement

XACML Architecture
Policy Controlled Dialogs using Voxeo VoiceXML Designer
`Policy.jsp`<%@page

<% DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
DocumentBuilder        builder = factory.newDocumentBuilder();
Document doc = builder.parse("http://127.0.0.1:9990/xacml/Example_1/Policy_1.xml");    %>

<% Element  root        = doc.getDocumentElement(); // policy node
NodeList ruleNodes = root.getChildNodes();     // rule nodes
int numberOfChildren;
numberOfChildren = ruleNodes.getLength();
String textRule = new String();
String subjectString = new String();
String subjectDataTypeString = new String();
int numberOfSubjectAttributes = doc.getElementsByTagName("SubjectMatch").getLength(); //AttributeValue
for (int h=0; h<numberOfSubjectAttributes; h++){
    subjectDataTypeString = doc.getElementsByTagName("AttributeValue").item(h).getAttributes().item(0).getNodeValue();
    subjectString = doc.getElementsByTagName("AttributeValue").item(h).getFirstChild().getNodeValue();
    out.println(subjectString);
} //end for
%>`
Dialogue-Policy integration
Client Side

Integration

Step Name: Rule_1

Integration Type
Select the type of data integration you would like to perform.
Integration Type: HTTP Request

HTTP Request
Click Configure to set up your HTTP request.
URL: http://127.0.0.1:9990/xacml/Example_1/Rule_1.jsp
Method: GET
Response Type: STRING

Configure
Test
Remove
Save
Cancel
Ongoing Work

- Improving the generation of VoiceXML from XACML
- Integrating the system with a physical access device such as an NFC lock
- Integrating presence information with the dialog access control system
Future Research Directions

- Making efficient and effective dialogs
- Addressing scalability issues, such as generating dialogs from policies with a large number of hierarchical rules
- Attempting to generate policies from dialogs
- Exploring privacy issues that arise with the use of dialogues
Summary

- Presented a novel approach to generate dialogues from policies
  - Example: To control physical access to facilities
- Policy language driven interaction with the user or authorization requester is generated at runtime and implemented using standards-based languages
- Lots of open research questions