agentTool III
Validation

MSE Presentation One
by: Patrick Gallagher
Outline

- Project Overview
- Project Plan
- Cost Estimation
- Software Quality Assurance Plan
- Project Requirement w/ Alloy 1.1 Constraints
- Prototype Demonstration
- Question/Comments
Project Overview

- **Goal**
  Develop an extensible framework for agentTool III validation that is easily upgradeable

- **Motivation**
  The current version of agentTool III contains non-cohesive diagrams. We need a way to provide agentTool III’s users the ability to check consistency between diagrams through validation
Project Overview Diagram
# Project Plan, MS Project 2002

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Phase One - Inception Phase</td>
<td>57 days</td>
</tr>
<tr>
<td>2</td>
<td>Vision Document 1.0</td>
<td>15 days</td>
</tr>
<tr>
<td>3</td>
<td>Project Plan 1.0</td>
<td>10 days</td>
</tr>
<tr>
<td>4</td>
<td>Software Quality Assurance</td>
<td>10 days</td>
</tr>
<tr>
<td>5</td>
<td>1st Prototype</td>
<td>36 days</td>
</tr>
<tr>
<td>6</td>
<td>Formal Requirements Sp.</td>
<td>20.5 days</td>
</tr>
<tr>
<td>7</td>
<td>Presentation 1.0</td>
<td>0 days</td>
</tr>
<tr>
<td>8</td>
<td>Phase Two - Elaboration Phase</td>
<td>47 days</td>
</tr>
<tr>
<td>9</td>
<td>Action Items</td>
<td>5 days</td>
</tr>
<tr>
<td>10</td>
<td>Vision Document 2.0</td>
<td>5 days</td>
</tr>
<tr>
<td>11</td>
<td>Project Plan 2.0</td>
<td>5 days</td>
</tr>
<tr>
<td>12</td>
<td>Architectural Design 1.0</td>
<td>23 days</td>
</tr>
<tr>
<td>13</td>
<td>Test Plan 1.0</td>
<td>17.5 days</td>
</tr>
<tr>
<td>14</td>
<td>Inspection Checklist 1.0</td>
<td>5 days</td>
</tr>
<tr>
<td>15</td>
<td>Formal Technical Evaluation</td>
<td>5 days</td>
</tr>
<tr>
<td>16</td>
<td>Prototype 2.0</td>
<td>47 days</td>
</tr>
<tr>
<td>17</td>
<td>Presentation Two</td>
<td>0 days</td>
</tr>
<tr>
<td>18</td>
<td>Phase Three - Production Phase</td>
<td>55 days</td>
</tr>
<tr>
<td>19</td>
<td>Action Items</td>
<td>5 days</td>
</tr>
<tr>
<td>20</td>
<td>Component Design</td>
<td>5 days</td>
</tr>
<tr>
<td>21</td>
<td>Application Code</td>
<td>25 days</td>
</tr>
<tr>
<td>22</td>
<td>Testing</td>
<td>32.5 days</td>
</tr>
<tr>
<td>23</td>
<td>Functional Testing</td>
<td>25.5 days</td>
</tr>
<tr>
<td>24</td>
<td>Testing Evaluation</td>
<td>17.5 days</td>
</tr>
<tr>
<td>25</td>
<td>Documentation</td>
<td>27 days</td>
</tr>
<tr>
<td>26</td>
<td>Assessment Evaluation</td>
<td>10 days</td>
</tr>
<tr>
<td>27</td>
<td>User Manual</td>
<td>27 days</td>
</tr>
<tr>
<td>28</td>
<td>Project Evaluation</td>
<td>11.5 days</td>
</tr>
<tr>
<td>29</td>
<td>References</td>
<td>3 days</td>
</tr>
<tr>
<td>30</td>
<td>Presentation Three</td>
<td>0 days</td>
</tr>
</tbody>
</table>

![Project Plan Diagram](image)
Cost Estimation

- COCOMO
  \[ \text{Effort} = 3.2 \times \text{EAF}(\text{Size})^{1.05} \]
  \[ \text{Time (months)} = 2.5(\text{Effort})^{0.38} \]
  \( \text{EAF} = \text{Effort Adjustment Factor} \)

- My Calculations
  \( \text{EAF} = 1.8687 \)
  \( \text{Effort} = 12.38 \text{ staff months} \)
  \( \text{Time} = 6.5 \text{ months} \)
  \( \text{Size} = 2\text{KLOC} \)
## EAF Values

<table>
<thead>
<tr>
<th>Parameter Identifier</th>
<th>My Value</th>
<th>Level</th>
<th>Reasoning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACAP</td>
<td>1.01</td>
<td>Moderate</td>
<td>Moderate experience in software analysis of developer</td>
</tr>
<tr>
<td>AEXP</td>
<td>1.102</td>
<td>Low</td>
<td>Low experience in major project applications development</td>
</tr>
<tr>
<td></td>
<td>1.27</td>
<td>Moderate</td>
<td>Project contains new concept, such as Eclipse plug-in development, but is</td>
</tr>
<tr>
<td>CPLX</td>
<td>0.984</td>
<td>Very Low</td>
<td>not overly complicated.</td>
</tr>
<tr>
<td>DATA</td>
<td>0.988</td>
<td>High</td>
<td>There is no DB access.</td>
</tr>
<tr>
<td>LEXP</td>
<td>0.988</td>
<td>High</td>
<td>Developer has profound experience in Java development.</td>
</tr>
<tr>
<td>MODP</td>
<td>0.964</td>
<td>High</td>
<td>Project follows modern practices closely.</td>
</tr>
<tr>
<td>PCAP</td>
<td>0.988</td>
<td>Moderate</td>
<td>Developer has adequate programming experience.</td>
</tr>
<tr>
<td>RELY</td>
<td>1.01</td>
<td>Low</td>
<td>Project is not safety critical and doesn’t need to be overly reliable.</td>
</tr>
<tr>
<td>SCED</td>
<td>1.152</td>
<td>Moderate</td>
<td>Project is on a firm schedule but time allowed is applicable.</td>
</tr>
<tr>
<td>STOR</td>
<td>1.224</td>
<td>Low</td>
<td>Only real main storage is for file I/O</td>
</tr>
<tr>
<td>TIME</td>
<td>1.396</td>
<td>Moderate</td>
<td>The project must respond to the user in a timely manner.</td>
</tr>
<tr>
<td>TOOL</td>
<td>0.912</td>
<td>High</td>
<td>The project will be implemented using many software tools and IDEs.</td>
</tr>
<tr>
<td>TURN</td>
<td>0.738</td>
<td>Very Low</td>
<td>Results are reported to the developer very quickly.</td>
</tr>
<tr>
<td>VEXP</td>
<td>1.024</td>
<td>Moderate</td>
<td>Developer has moderate experience with the most other project components,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>excluding software.</td>
</tr>
<tr>
<td>VIRT</td>
<td>1.042</td>
<td>Low</td>
<td>The complexity of hardware and software is not great.</td>
</tr>
</tbody>
</table>
Software Quality Assurance Plan

- Documentation
  - Available at http://www.cis.ksu.edu/~psg9999/classess/MSE/index.html
- Standards, Practices, Convention, and Metrics
- Reviews and Audits
- Problem Reporting and Corrective Action
- Tools, Techniques, and Methodologies
- Code Control, Media Control, and Supplier Control
- Risk Management
Project Requirements

- Consistency Driver Framework
  - Must be extensible
  - Must allow user to configure rules
  - Must be adaptable to new rules, new components, and new diagrams.
Validate Components
Requirements

1. Validated on save commands
2. Discrepancies are report to error log
3. Updates Eclipse’s ‘Problems’ tab
4. Previous errors that are still valid and classified as errors are still visible
5. (Future) The save command can be initiated by either the user or periodically by the system.
Validate Components Case Diagram
Each agent that is connected to a role by a “plays” relation must also be connected to every capability that the role is connected to by a “requires” relation in the role diagram by a “possess” relation. This is reported as an error.

all a : Agent | a.plays.requires in a.possess
Validate Agent Diagram

- Each role should exist in a role diagram present in the current directory. This is reported as a warning.

\[
\text{all } r : \text{Role} \mid \text{some } r.\text{existIn} \& \text{AgentDiagram} \rightarrow \text{some } r.\text{existIn} \& \text{RoleDiagram}
\]
Validate Agent Diagram

- A protocol must exist to at least one other agent or organization from each agent in agent diagram. This is reported as a warning.

all p : Player + Actor | some p.protocol & Player
Validate Agent Diagram

Each agent must be connected to a role with the “play” relation -- or be connected to a capability with the “possess” relation. This is reported as an error.

all a : Agent | (some a.plays) || (some a.possess)
Alloy Assertions

- assert AgentInAgentDiagramOnly { 
  no a : Agent | some a.existIn & Diagram - AgentDiagram} 
Analyzing AgentInAgentDiagramOnly ... 
Scopes: Name(10), Object(10) 
Conversion time: 6 seconds 
Solver time: 3 seconds 
No counterexamples exist in this scope

- assert PlayersHaveProtocol { all p : Player | some p.protocol } 
Analyzing PlayersHaveProtocol ... 
Scopes: Name(10), Object(10) 
Conversion time: 6 seconds 
Solver time: 3 seconds 
No counterexamples exist in this scope
Alloy Assertions cont.

- assert NoAgentToRoleProtocol {
    all a : Agent | all r : Role | no a.protocol & Role && no r.protocol & Agent } 
Analyzing NoAgentToRoleProtocol ... 
Scopes: Name(10), Object(10) 
Conversion time: 6 seconds 
Solver time: 64 seconds 
No counterexamples exist in this scope 

- assert PlaysSameDiagram {all c : Player | c.existIn in c.plays.existIn} 
Analyzing PlaysSameDiagram ... 
Scopes: Name(10), Object(10) 
Conversion time: 6 seconds 
Solver time: 127 seconds 
No counterexamples exist in this scope
Alloy Conditions

- cond SR41Correct {
  some a : Agent | some r : Role | r in a.plays}

- cond SR41Incorrect {
  some a : Agent | some r : Role | r in a.plays && r.requires !in a.possess }  

- cond SR42Correct{
  some r : Role | some ad : AgentDiagram | some rd : RoleDiagram | ad in r.existIn}

- cond SR42Incorrect{
  some r : Role | some r.existIn & AgentDiagram && no r.existIn & RoleDiagram }
Condition SR4.1

- Analyzing SR41Correct ...
- Scopes: Name(10), Object(10)
- Conversion time: 14 seconds
- Solver time: 388 seconds
- Instance found:

  **Domains:**
  - Name = \{N2,N3,N4,N5,N6,N7,N8,N9\}
  - Object = \{O0,O1,O2,O3,O4,O5,O6,O7,O8,O9\}

  **Sets:**
  - ActivityDiagram = \{O2\}
  - Actor = \{O4,O5\}
  - Agent = \{O9\}
  - AgentDiagram = \{O7\}
  - Capability = \{O0\}
  - Component = \{O0,O1,O4,O5,O6,O8,O9\}
  - Diagram = \{O2,O3,O7\}
  - Element = \{O4,O5,O8,O9\}
  - Goal = \{O1\}
  - Organization = {}  
  - Player = \{O9\}
  - Role = \{O8\}
  - RoleDiagram = \{O3\}
  - Service = \{O6\}

  **Relations:**
  - achieve = \{O8 \rightarrow \{O1\}\}
  - existIn = \{O0 \rightarrow \{O3,O7\}, O1 \rightarrow \{O3\}, O4 \rightarrow \{O3\}, O5 \rightarrow \{O3,O7\}, O6 \rightarrow \{O7\}, O8 \rightarrow \{O3,O7\}, O9 \rightarrow \{O7\}\}
  - name = \{O0 \rightarrow N9, O1 \rightarrow N8, O2 \rightarrow N7, O3 \rightarrow N7, O4 \rightarrow N7, O5 \rightarrow N7, O6 \rightarrow N7, O7 \rightarrow N7, O8 \rightarrow N6, O9 \rightarrow N8\}
  - plays = \{O9 \rightarrow \{O8\}\}
  - possess = \{O9 \rightarrow \{O0\}\}
  - protocol = \{O4 \rightarrow \{O4,O5,O8,O9\}, O5 \rightarrow \{O4,O8,O9\}, O8 \rightarrow \{O4,O5,O8\}, O9 \rightarrow \{O4,O5,O9\}\}
  - provides = \{O9 \rightarrow \{O6\}\}
  - requires = \{O8 \rightarrow \{O0\}\}

  **Skolem constants:**
  - a = O9
  - r = O8
Validate Activity Diagram

- The name of the activity diagram must be the same as a service within a role or agent diagram. This is reported as an error.
- An incoming event within the activity diagram must be a message in a sequence diagram. This is reported as an error.
- An outgoing event within the activity diagram must be a message in a sequence diagram. This is reported as an error.
Validate Goal Diagram

- Goals within the goal diagram must be connected to all other goals in the diagram in tree form. Tree form consists of both nodes and leaves. A node is a goal that is connected to at least one child leaf or node. A leaf has no children nodes or leaves. This is reported as an error.

- Leaf nodes should be connected to at least one role in a role diagram by the “achieved” relation. This is reported as a warning.
Validate Organization Diagram

- Each goal must be also be in a goal diagram present in the current directory. This is reported as an error.
- Each goal must be connected to an organization by a “achieves” relation. This is reported as an error.
- Each service must be connected to the organization element by a “provide” relation. This is reported as a warning.
- Each service must be associated with an activity diagram with the same name. The activity diagram must present in the current directory. This is reported as a warning.
Validate Role Diagram

- Each role must be connected to at least one other role by the “protocol” relation. This is reported as a warning.
- Each role should be connected to at least one agent in an agent diagram by the “played” relation. This is reported as a warning.
Validate Sequence Diagram

- There must be exactly two roles within a sequence diagram. These roles must also exist within role diagrams. This is reported as an error.
Adjust Validation Settings

- All validation rules will have the ability to be either active or inactive.
- Only active validation rules are used in validation.
- Only the chosen validation level errors will be reported to the user.
- Validation types are adjustable and are representative of related diagrams.
Adjust Validation Case Diagram
Phase 2 Deliverables

- Vision Document Revision
- Project Plan Revision
- Architectural Design
- Test Plan
- Technical Inspection Plan
- Executable Architecture Prototype
Prototype Demo
Current Obstacles

- XML File inconsistencies
- Eclipse plug-in development
Questions & Comments