Project Plan 2.0

for

Diagram Consistency and Validation in agentTool III

Submitted by Patrick Gallagher
CIS 895 MSE Project
Department of Computing and Information Sciences
Kansas State University
# Table of Contents

<table>
<thead>
<tr>
<th>Description</th>
<th>Page #</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Task Breakdown</td>
<td>1</td>
</tr>
<tr>
<td>1.1. Inception Phase</td>
<td>1</td>
</tr>
<tr>
<td>1.2. Elaboration Phase</td>
<td>1</td>
</tr>
<tr>
<td>1.3. Production Phase</td>
<td>2</td>
</tr>
<tr>
<td>2. Cost Estimate</td>
<td>2</td>
</tr>
<tr>
<td>2.1. COCOMO</td>
<td>2</td>
</tr>
<tr>
<td>3. Architecture Elaboration Plan Completion Criteria</td>
<td>4</td>
</tr>
<tr>
<td>3.1. Vision Document 1.0 Revision</td>
<td>4</td>
</tr>
<tr>
<td>3.2. Project Plan 1.0 Revision</td>
<td>5</td>
</tr>
<tr>
<td>3.3. Architectural Design 1.0</td>
<td>5</td>
</tr>
<tr>
<td>3.4. Prototype 2.0</td>
<td>5</td>
</tr>
<tr>
<td>3.5. Test Plan 1.0</td>
<td>5</td>
</tr>
<tr>
<td>3.6. Formal Technical Inspection 1.0</td>
<td>5</td>
</tr>
<tr>
<td>3.7. Formal Requirements Specification 1.0</td>
<td>5</td>
</tr>
</tbody>
</table>
## Summary of Changes

<table>
<thead>
<tr>
<th>Version #</th>
<th>Date</th>
<th>Changed by</th>
<th>Summary Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>September 21, 2005</td>
<td>Patrick Gallagher</td>
<td>Initial draft of document</td>
</tr>
<tr>
<td>1.0.1</td>
<td>September 24, 2005</td>
<td>Patrick Gallagher</td>
<td>Revision of initial draft</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Description</th>
<th>Page #</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Project Gantt Chart</td>
<td>4</td>
</tr>
</tbody>
</table>
**List of Tables**

<table>
<thead>
<tr>
<th>Description</th>
<th>Page #</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. COCOMO EAF Parameters</td>
<td>3</td>
</tr>
<tr>
<td>2. Project EAF Values</td>
<td>3</td>
</tr>
</tbody>
</table>
1. Task Breakdown
   1.1. Inception Phase

The inception phase will concentrate on the project’s overview and requirements. This phase will include the production of a Vision Document, a Project Plan, a Software Quality Assurance Plan, and a simple prototype of the software. Each of the four works will have to be approved by the committee before any further progress on the project.

1.1.1. Vision Document

This document will include the project’s critical requirements. It gives a listing of the current requirements and labels each requirement as Critical, Non-Critical, or Future. Requirements include validation rules and the purposed framework for the project. The introduction to this project includes its goals and purpose. It include graphical models that are used to describe the framework, and use cases to illustrate the functionality.

1.1.2. Project Plan

This document will include a timeline for the project and a cost estimate for completing this project. Cost estimates will be developed using the COCOMO estimating methodology. All activities involved with this project will be associated with milestones outlined within this document according to the modern software process. This process includes the following phases: Inception Phase, Elaboration Phase, Production Phase, and the Transition Phase.

1.1.3. Software Quality Assurance Plan

This document will include an outline of required documents. To ensure software quality, practiced standards and conventions will be monitored and followed. This document will outline project reviews and the responsibilities of those associated with the agentTool III validation.

1.1.4. Simple Prototype

A simple prototype will be developed to demonstrate at least one aspect of the software. It will demonstrate some of the critical requirements outlined in the Vision Document, such as implementation of critical validation rules. This is completed to demonstrate feasibility or to illuminate risky project requirements.

This phase will be complete once the supervisory committee has approved 1.1.1 through 1.1.4.

1.2. Elaboration Phase

The elaboration phase will concentrate on the project’s architecture. This phase will include the production of an Architecture Design Plan, revisions to the Project Plan and the Vision Document, a formal specification, Test Plan, and two architecture inspections.

1.2.1. Architecture Design Plan

The Architecture Design plan will include appropriate UML diagrams such as class diagrams, use-case diagrams, and sequence diagrams, but is not limited to those named here.

1.2.2. Revisions of Project Plan and Vision Document

Appropriate changes that were suggested by the committee at the end of phase one will be made to the Project Plan and Vision Document.
1.2.3. Formal Specification

The formal specification will include at least one function of the software to be specified using the Alloy Constraint Language 1.1.

1.2.4. Test Plan

The test plan will include a complete testing procedure for the project. This will include test suites and appropriate procedures for reporting and correcting failed test.

1.2.5. Architecture Inspections

Two architecture inspections will be performed by fellow MSE students here at Kansas State University. Their feedback will be documented and presented.

This phase will be complete once the supervisory committee has approved 1.2.1 through 1.2.5.

1.3. Production Phase

The production phase will include project implementation and testing. This phase includes project coding and documentation.

1.3.1. Project Coding

Project coding will consist of all committee approved and designated tasks to be coded and developed. Both unit testing and integration testing will be performed throughout coding. Test cases will be developed according to the project requirements outlined in the project’s Vision Document.

1.3.2. Project Documentation

All aspects of this project will be documented. JavaDocs will be generated to exemplify coding structure and explanations. A post test document will be drafted with all results of the test plan. It will also include the solutions or corrections to the case of test failure. A user manual will be completed and will include a description of project installation, software usage requirements, and software usage.

This phase will be completed once the supervisory committee has approved 1.3.1 through 1.3.2.

2. Cost Estimate

2.1. COCOMO

Barry Boehm’s COCOMO cost modeling method will be used to estimate the software cost and development time of adding the diagram validation functionality to agentTool III. I consider this software to be an organic mode relationship, as opposed to an embedded or semi-detached mode, because of its lack of complexity and absence of the need for a highly rigorous process.
Table 1: COCOMO EAF Parameters

Effort will be estimated using the formula \( \text{Effort} = 3.2 \text{ EAF} \times \text{Size}^{1.05} \). EAF is the effort adjustment factor that allows for normalization of the formula. Table 1 gives a breakdown of the fifteen parameters that make up the EAF, while Table 2 gives my setting for each parameter and a brief explanation for my choice. The Size variable is the number of delivered source lines of code, or SLOC. This project will be measured using KSLOC, or thousand source lines of code. Once this value is found, we can then find a good estimate of time needed to complete this project.

Table 2: Project EAF Values

Time will be estimated using the formula \( \text{Time} = 2.5 \times (\text{Effort})^{0.38} \), where time is measured in months.
I have estimated the size variable to be 2KLOC. The best example of size for this project is the dynamic and static agentTool III diagram projects. Although these two projects differed in size, I found that the average KLOC for one diagram’s project was around 1.3 KLOC. I believe that this project will be roughly the same size and a diagram project. They both are Eclipse plug-ins and are directly related to each other. I have added about .7 estimated KLOC to this project because of the framework development. Below are the calculated values for both Effort and Time.

\[
\begin{align*}
\text{Effort} &= 12.38179899 \\
\text{Time} &= 6.504259102
\end{align*}
\]

Using these estimates, COCOMO concludes that it will take 12.38 month of development by one developer to complete this project. MSE projects span approximately 8 months. Using COCOMO, we could say that this timeframe is not enough to complete this project. I feel that this is not the case.

This project does not contain the need for complex communication channels between project members. There is not a deep hierarchy that artifacts must pass through in order to be approved. My interpretation is that COCOMO assumes there to be deeper levels of communication between team members. If this is so, it will take longer to communicate changes between project artifacts between project members. This is not the case in this project. There is only one level of separation between the project developer and the committee members, allowing less time to commit changes to the project. I believe that outlining the project according to Figure 1 shows this project is well capable of being completed in 8 months.

Figure 1 represents the project timeline using the estimated time from the COCOMO formula. Each minor milestone in the timeline is a requirement for the project and will be deliverable at each major milestone, also marked by the end of one phase and the giving of a presentation.

### Figure 1: Project Gantt Chart

3. **Architecture Elaboration Plan Completion Criteria**

The following is a listing of artifacts and tasks that are to be completed by the end of the project’s Inception phase and are to be completed before the second presentation.
3.1. Vision Document 1.0 Revision

Vision Document 1.0 will be revised and compiled into Vision Document 2.0. The revisions will contain a complete listing of project requirements and contain changes suggested by the committee members following presentation one. This revision will be approved by the major professor.

3.2. Project Plan 1.0 Revision

Project Plan 1.0 will be revised and compiled into Project Plan 2.0. The revisions will contain a complete listing of project requirements and contain changes suggested by the committee members following presentation one. This revision will be approved by the major professor.

3.3. Architectural Design 1.0

Architectural Design 1.0 will be documented using UML diagrams such as class, sequence, and state chart diagrams. All architectural components, including the data model, will be documented. This will be approved by the major professor.

3.4. Prototype 2.0

Prototype 1.0 will be enhanced to Prototype 2.0. This will include any improvements to existing functionality as recommended by the committee members. Additional functionality will also be implemented. This will be approved by the major professor.

3.5. Test Plan 1.0

Test Plan 1.0 will be developed. The document will follow the requirement listings found in Vision Document 2.0. It will also include evaluation criteria and testing data to complete acceptable testing methods. This will be approved by the major professor.

3.6. Formal Technical Inspection 1.0

Two fellow MSE students, who are yet to be determined, will provide input into the testing of this project by completing a formal technical inspection. The technical inspection will assess the project architecture and agentTool III framework. It will also include a function testing checklist so that the requirements outlined in the Vision Document may be evaluated. The formal technical inspection list will be prepared and will be approved by the major professor.

3.7. Formal Requirements Specification 1.0

At least one component of this project will be formally specified using Alloy 1.1. The specification will be completed by representing the formal requirements of the Agent Diagram in the Vision Document in the Alloy Constraint language. The Agent Diagram requirements are chosen because of their relationship to many different components in other diagrams and with in itself. This specification will be approved by the major professor.