Applying Broadcasting/Multicasting/Secured Communication to agentMom in Multi-Agent Systems

Project Plan

Version 1.1

This document is submitted in partial fulfillment of the requirements for the degree MSE.

Chairoj Mekprasertvit
CIS 895 – MSE Project
Kansas State University
Spring 2003

Table of Contents

1 Introduction

This section provides an overview of project plan

1.1 Purpose

The purpose of this document is to provide cost estimation and architecture elaboration plan for the project "Applying Broadcasting/Multicasting/Secured Communication to agentMom in Multi-Agent Systems". This document is intended to be viewed only by project advisor and committee members.

1.2 Scope

This document covers project plan for the project "Applying Broadcasting/Multicasting/Secured Communication to agentMom in Multi-Agent Systems", including time frame, cost estimation and architecture elaboration plan. Time frame provides the phases, iterations and milestones that will comprise the project. Cost estimation provides a detailed estimate on the size, cost and effort required for the project. Architecture elaboration plan provides details of activities and actions that must be accomplished prior to the Architecture presentation.

1.3 References

- AgentMom User's Manual, Air Force Institute of Technology, July 2000
- Multiagent Systems Engineering. *The International Journal of Software Engineering and Knowledge Engineering*
- Project Overview version 1.1, Kansas State University (http://www.cis.ksu.edu/~cme6556/project_overview_1.1.pdf)
- Software Requirements Specification, Kansas State University, March 2003, (http://www.cis.ksu.edu/~cme6556/software_requirements_specification_1.1.pdf)
- Fundamentals of Function Point Analysis (http://www.ifpug.com/fpafund.htm)

2 Time Frame

Deliverable	Estimated Date
Phase I: Objectives	March - April
Project Overview 1.0	March 24 – March 30
Software Requirements Specification	March 24 – March 30
Project Plan	March 31– April 06
Software Quality Assurance Plan	March 31– April 06
Prototype I	March 31– April 06
MSE homepage	April 07 – April 13
First presentation	April 14 – April 25
Phase II: Architecture	April - November
Update Documents	April 26 – April 30
Formal Requirement Specification 0.1	May 19 – May 25
Architecture Design 0.1	May 26 – May 31
Test Plan 0.1	June 01 – June 07
Formal Technical Inspection 0.1	June 08 – June 14
Executable Architecture Prototype 0.1	June 08 – June 14

Formal Requirement Specification 1.0	September 01 – September 07
Architecture Design 1.0	September 08 – September 14
Test Plan 1.0	September 15 – September 21
Formal Technical Inspection 1.0	September 22 – September 28
Executable Architecture Prototype 1.0	October 01 – October 15
Second Presentation	November 17 – November 21

Phase III: Implementation	November – February
Update Documents	November 22 – November 30
Component Design	December 01 – December 07
Final Product	December 08 – December 31
Javadoc	December 08 – December 31
Assessment Evaluation	January 01 – January 18
User Manual	January 19 – January 25
Project Evaluation	January 19 – January 25
References	January 26 – January 31
Formal Technical Inspection Letters	January 26 – January 31
Final Presentation	February 24 – February 28

For a graphical representation of the proposed project plan, consult the included *Gantt chart*.

3 Cost Estimation

3.1 Function Point

First, the different types of program features must be identified. These include the following:

a) Internal Logical Files – A file is a major logical group of user data or control information, which could be in a large database or a separate file. This is zero for the agentMom.

b) External Interfaces Files – Normally considered files passed or shared between systems. This is zero for the agentMom.

c) External Inputs – Unique user data or user control input that enters the external boundary of the system and adds or modifies a logical internal file. The inputs are unicast message, multicast message, broadcast message, secured multicast message and secured multicast message. Thus, there are five external inputs.

d) External Outputs – Each user data or control output type leaving the external boundary of the system is counted. The outputs are unicast message, multicast message, broadcast message, secured multicast message and secured multicast message. Thus, there are five external outputs.

e) External Inquiry – Each input-output combination is counted, when input causes an immediate output. This is zero for the agentMom.

Table 1. Total Unadjusted Function Points

Type	Complexity			Function Points
	Low	Average	High	
Internal Logical Files				0
External Interfaces Files				0
External Inputs	5x3			15
External Outputs	5x4			20
External Inquiry				0
Total				**35**

3.2 COCOMO I

Estimation is based upon the Organic mode in the Constructive Cost Model (COCOMO) cost model developed by Barry Boehm. Since this project is fairly simple and very flexible, we can assume using the Organic mode. Also, original COCOMO model is used since COCOMO II is more appropriate with large team development project with large number of developers.

The COCOMO estimating equations follow this simple form:

$\text{Effort} = C1*EAF*(\text{Size})^{P1}$

$\text{Time} = C2*(\text{Effort})^{P2}$

where:

Effort = number of person-months

C1 = constant scaling coefficient for effort

C2 = a constant scaling coefficient for schedule

P1 = an exponent that characterizes the economics of scale inherent in the process used to produce the end product

P2 = an exponent that characterizes the inherent inertia and parallelism in managing a software development effort

EAF = an effort adjustment factor that characterizes the domain, personnel, environment, and tools used to produce the artifacts of the process

Size = size of the end product (in human-generated source code), measured by the number of delivered source instructions

Time = total number of months

As in Organic mode,

$C1 = 3.2$

$C2 = 2.5$

$P1 = 1.05$

$P2 = 0.38$

Since EAF value is difficult to determine, the EAF effect is not considered at this point. For instance, the EAF of Programmer capability is range from 1.42 – 0.70. It is hard to specify the value when there is no database that refers to the number of years in programming experience for each value. The estimation of size is defined as human-generated source line of code, excluding comments. The SLOC per Function Point for java is 46, so SLOC is 35 x 46 = 1610.

Therefore, the total effort and time are:

Effort = $3.2*(1.610)^{1.05}$ = 5.3 person-months (4.9 previously)
Time = $2.5*(5.3)^{0.38}$ = 4.7 months (4.6 previously)
Productivity = 1610/4.7 = 343 LOC-month (330 previously)
Staff = 5.3/4.7 = 1.13 person (1.07 previously)

As the number shown above, this project requires one person to complete in 4.7 months with 343 SLOC per month, or one person works 4.7*152 = 715 hours

Note: As described by Boehm, there are 152 working hours in a month. Therefore, time to complete this project may vary depend on number of working hours in a month.

4 Architecture Elaboration Plan

4.1 Vision Document (revision)

After the first presentation, suggestions shall be provided by the committee and these shall be used to revise the Vision document. The revised document shall be approved by the major professor. Vision document consists of Project Overview document and Software Requirements Specification document.

4.2 Project Plan (revision)

After the first presentation, suggestions shall be provided by the committee and these shall be used to revise the Project Plan document. The cost estimation shall be updated as appropriate. Also, the implementation plan shall be included. The revised document shall be approved by the major professor.

4.3 Formal Requirement Specification

The class diagram from architecture design shall formally be specified using UML/OCL methodology. The tool USE, a UML-based Specification Environment, shall be used. For more information about USE, please refer to "www.db.informatik.uni-bremen.de/projects/USE/ "

4.4 Architecture Design

The completed class diagrams and use cases diagrams shall be produced and well document. This design shall be implemented based upon the class diagram and use cases presented in Vision document. Also, this architecture design shall be undergo formal technical inspection.

4.5 Test Plan

Test plan shall be produced to show that all requirements specified in vision document are satisfied. Unit testing, integration testing, and system testing shall be conducted. Unit testing shall be class-based. Two or more related classes shall be used for integration testing. Finally, the whole system shall be used for system testing. Reliability shall involve in the testing to measure successful rate of message delivery.

4.6 Formal Technical Inspection

The architecture design shall be undergo formal technical inspection. The group of inspector consists of Ravikanth Athipatla and Acharaporn Pattaravanichanon. The Developer shall develop a formal checklist and provide it to inspectors. The inspectors shall provide a formal report on the result of their inspection during Phase III.

4.7 Executable Architecture Prototype

All driving requirements identified in vision document shall be implemented. This prototype shall be implemented based on the first prototype from phase I. Specifically, The executable architecture prototype shall be integrated into agentMom, and it shall have all driving requirements capabilities.