AC 2010-343: NASA SENIOR DESIGN: MISSION ASSURANCE MANAGEMENT ENVIRONMENT

Jiang Guo, California State University Los Angeles

Jose Macias, Jet Propulsion Laboratory

James Conrad, University of North Carolina, Charlotte

Jimmy Hoo, California State University Los Angeles

Rony Velasquez, California State University Los Angeles

Maxim Fastovsky, California State University Los Angeles

SaeRom Hong, California State University Los Angeles

NASA Senior Design: Mission Assurance Management Environment

Abstract

This paper discusses the implementation experience of our NASA senior design project. Our NASA senior design project Mission Assurance Management Environment is to increase the reliability, availability, and safety of unmanned aircraft, by focusing on implementing the Jet Propulsion Laboratory, JPL, Flight Project Practices, FPPs, and Design Practices, DPs, in an integrated software environment. This project enables the students at California State University Los Angeles to understand the function and scope of the spacecraft mission assurance activities and to make contribution to NASA ESMD. During the senior design project implementation, students work with their advisor and NASA expert to conduct the research on mission assurance management and improve their related technical background of the project, including project requirement, design, and implementation related skills.

Introduction

Part of the NASA vision is to initially send robots and in the end humans to explore the Moon, Mars and possibly other celestial bodies. The Exploration Systems Mission Directorate (ESMD) is dedicated to creating a constellation of new capabilities, supporting technologies and foundational research that enables sustained and affordable human and robotic exploration of the solar system.

For these long duration human and robotic exploration missions that ESMD faces, the mission assurance management of the building, launching, and operating spacecraft is a crucial component. Both predictable and unpredictable risks exist in the lifetime of the spacecraft dispatched to explore our solar system. To address this challenge, we propose to improve spacecraft building and operating processes with integrated software support environments. Added components of this environment include systems engineering approaches with research focused on mission success of the flight projects and the development of all the requirements to make sure these projects minimize the unavoidable risks that such unique endeavors entail.

To achieve this goal, JPL has developed a set of practices called FPPs (Flight Project Practices) and DPs (Design Practices) [1]. However, it is difficult to ensure that the work being done by the typical “one-of-a-kind” JPL projects follow the FPPs and DPs in their entirety or at least to the extent that is necessary and sufficient to minimize risks. Therefore, we investigated and integrated approaches to implement a high level software system to support JPL’s FPPs and DPs. This will effect a possibly important change in the way JPL works at minimizing the risk involved in their flight missions. Our NASA senior design project Mission Assurance Management Environment is to increase the reliability, availability, and safety of space missions, by focusing on implementing the JPL’s FPPs and DPs in an integrated software environment.

During an academic year, the students complete the senior design project following a system engineering approach to implement the Mission Assurance Management Environment (MAME). The overall task includes preliminary design, design, implementation, system integration, validation and test.

Identification of System Requirements

NASA Jet Propulsion Laboratory (JPL) is leading NASA's efforts to construct and operate robotic planetary spacecraft. Some well known projects at JPL are the Phoenix Mars Lander, the Mars Exploration Rovers Spirit and Opportunity, the Voyager I and II, the Spitzer Space Telescope and many others.

Starting on June 1st and ending on July 10th 2009, invited by Dr. Macias, we visited the Mission Assurance Division and Directorate at NASA JPL and worked with personnel from the Directorate to identify the requirement of the Mission Assurance Management Environment (MAME). Our work at JPL was to investigate and integrate approaches to implement a high level software system to support JPL's mission assurance activities. Our work included components such as how to schedule and monitor the various activities regarding projects mission assurance. The main research goal of improving the performance of mission assurance management was studied thoroughly studied.

During the six weeks at JPL, we worked with many experts in the field of mission assurance management who significantly added to our research effort on mission assurance activities. These experts also helped us to understand the required characteristics of the tools that would allow them to improve the work in their technical field of expertise. Our work at JPL prepared us to enable our students to complete the senior design project and make potential contribution to NASA mission assurance management. Our work at JPL also provided us an opportunity to understand the function and scope of the mission assurance activities at JPL and advise our students to implement the related systems in their senior design project.

To identify the requirement, we started with interviews. We discussed the workflow and mission assurance managers' work with Dr. Macias and other NASA experts at JPL. These interviews were basically targeted to gather information regarding mission assurance managers' activities. We got a primary picture of how mission assurance managers complete their work, and what are contents of mission assurance, including Quality Assurance, Reliability Engineering, Electronic Parts Engineering and Environmental Engineering. From these interviews, we also understood how mission assurance mangers work with people from other fields of expertise.

During the interview process, we spent a lot of time on document analysis. Document analysis is was a very important step in gathering requirements. It helped to study the concepts, identify the requirements and how to describe the work activities of mission assurance managers. We read through many materials, including presentation, monthly reports, work plans, proposals, and mission assurance related handbooks prepared by JPL experts.

Attending JPL's monthly report meetings is was another powerful technique to gather requirements. We spent three days attending JPL's mission assurance managers' monthly report meetings, From these meetings, we collected information regarding mission assurance managers' work items, including work agreements, red flag or potential red flag failure/problem reports, the NASA 5 by 5 (5x5) risk assessment matrix, Incident/Surprise/Anomaly reports, medium and high risk waivers, etc.

Taking notes was a very important approach during the requirement eliciting phase. While working at JPL, we took more than 100 pages of notes. Every dad, we spent time re-organizing our notes, and wrote down new questions and topics to be discussed next day. In this way, our work is was very efficient.

Based on the notes, we created a Use-Case model for the Mission Assurance Management Environment (MAME). This model was a simplified representation of the MAME system that is intended to facilitate understanding of system functions and features. A Use-Case is a description how users interact with the system. We combined textual description and diagrams to describe the system requirements. The use-cases and step-based scenarios were very useful tools for us to communicate with JPL mission assurance managers. They provided a context for discussing the requirements for the MAME system.

We wrote the draft version of the requirements of the MAME system based on the use case model and submitted it to the JPL mission assurance managers for review. Then, we used a storyboard approach to collect the inputs and feedbacks of the first version of the MAME system requirements. We visited the reviewers individually and used a set of drawings to describe mission assurance managers' activities that occur in an interaction with the system. We started by drawing pictures of the screens, dialogs, toolbars, and other elements on papers to discuss the functions and features that the MAME system would provide. Storyboards provided an inexpensive and efficient communication approach.

After discussing with the reviewers, we finished the requirements of the MAME system. This document was the start point and guideline of the one year long senior design project that is to be finished at Cal State Los Angeles.

System Features of MAME

To increase the efficiency and thus Mission Assurance Manager's (MAM) work performance, this project aims at providing the following capabilities:

- Provide a web-based infrastructure to support MAM's work
  - Computers can do more than a spreadsheet
  - Reduce human errors
- Provide a Framework for MAM's work
  - Provide MAMs with a pre-defined process
- Provide MAMs with a Work Environment
  - Tailor work list to specific project's requirements

To implement MAME successfully, the following aspects and design considerations are very important:

- Dynamic – All items in MAME shall be editable and adaptable.

- Extensible – MAME shall provide a capability to add new functions.
- Traceable – Everything in MAME shall be recorded and tracked. (This enhances reliability).
- User friendly – MAME shall be able to provide default items (such as pre-defined work lists) so that MAMs can easily tailor the items according to the projects.
- Maintainability – Easy to fix bugs, update to new versions and different platforms

MAME's major features are discussed below:

- **Create Role Model**

MAME has two kind ~~s~~ of users: Administrators and Users. Users are organized according to pre-defined role model. This model will be used to control: Approval, Permission. The pre-defined role model shall be editable.

Administrator can add new roles, and new relationships. The relationships will be defined in a matrix. Current roles: OSMS engineer ~~s~~, specialist, consultant ~~s~~, mission assurance manager, chief mission assurance manager, section manager, division manager.

- **Create Process Definition**

MAME shall provide an editable process definition based on pre-defined work item lists. Users can edit process definitions using a graphical user interface. MAME will provide an ability to re-use a process definition. When creating a new project, the user can select previous process definitions. MAME will provide a schedule view. The scale of the schedule shall be adjustable. MAME will provide Process Definitions tailored to various project classifications such as Class A, B, C, D or E. Also it should consider long term or short term missions, Earth, moon or deep space environments, etc.

- **Execute Defined Process**

MAME shall provide different process definitions for different roles. Different role will see different parts of the work items. See different view of project progress. MAME will create reminders for work items that have not been completed when due. The reminders are automatically generated for overdue work items. Two reminders: email and red alert display on top screen. User can select to display later, or ignore. MAME will provide screen navigation driven by process definition execution. MAME will permit ~~s~~ users to view and complete work items (upload documents) that have been assigned to them.

- **Track Process Execution**

MAME shall provide upper level managers a "radar view" to allow them to monitor process definition progress, including: project status, pending work items, work item ownership, how long each work item has been waiting. MAME also will provide users the ability to track related work items and documents: (1) from one document to find out all its income documents and outcome documents. (2) from one user to find out all his/her assigned work items and completed work items.

- **Log Process Execution**

MAME will log all the changes to the work items. MAME will log all the work item execution status for later analysis. MAME will log ready time for work item. This is input available time. MAME will log end time for work item. This is output available time.

- **Analyze Process Execution**

MAME will help the managers to compare the defined process vs. actual process execution. It will also help managers to analyze where the bottle necks are. MAME will provide managers the ability to analyze the performance of teams and some statistic data, such as waiting time, number of changes, etc.

- **Monthly Report**

MAME will provide mission assurance managers' monthly report support. This includes graphic report generation of Waiver, Problem/Failure Report, Document Traceability, and Non-Conformance Report.

System Design and Implementation of MAME

Object oriented design (OOD) was used in MAME's development. This approach has a natural structure for modular design. OOD supports objects, subsystems, and framework inherently. OOD systems are easier to modify and have high maintainability, which is important to MAME. As OOD systems support information encapsulation, so they have high extendibility. This is another important design quality attribute that MAME needs.

MAME shall be a web based system. Browser client and web server send and receive messages back and forth over TCP/IP based network using HTTP protocol. This provides MAMs a distributed and cooperative work environment, which is an essential aspect of MAME. MAME uses form and web page driven user interface. Forms are very efficient in creating or capturing information. They ensure that all the information needed for interactions between users and the MAME system is captured. MAME uses tree structure and GANTT Chart to display process structure and progress. This approach is informative, direct and natural, it can improve users' performance dramatically.

Reliable operation of a computing system depends on both error detection and error recovery. We use web page based form and page driven based interaction approach. It minimizes the users' error. As for system operation, our error detection will be based on programmed checks and assertions. The recovery mechanism aims at providing error recovery after any kind of detected error. Our recovery is transaction based. All operations of the MAME system are logged and these are the checkpoints of the system rollback mechanism. All data are recorded in a MySQL database, in case of system crash, the data can be recovered and all the documents that users prepared will still be in the system.

- **System Architecture**

Figure 1 illustrates the architecture of the MAME system. The MAME system is a web-based application. The client user interface consists of a web-browser. The user communicates with the MAME application logic and MySQL database through an Internet connection with a web server. The MySQL database is the data repository for the application.

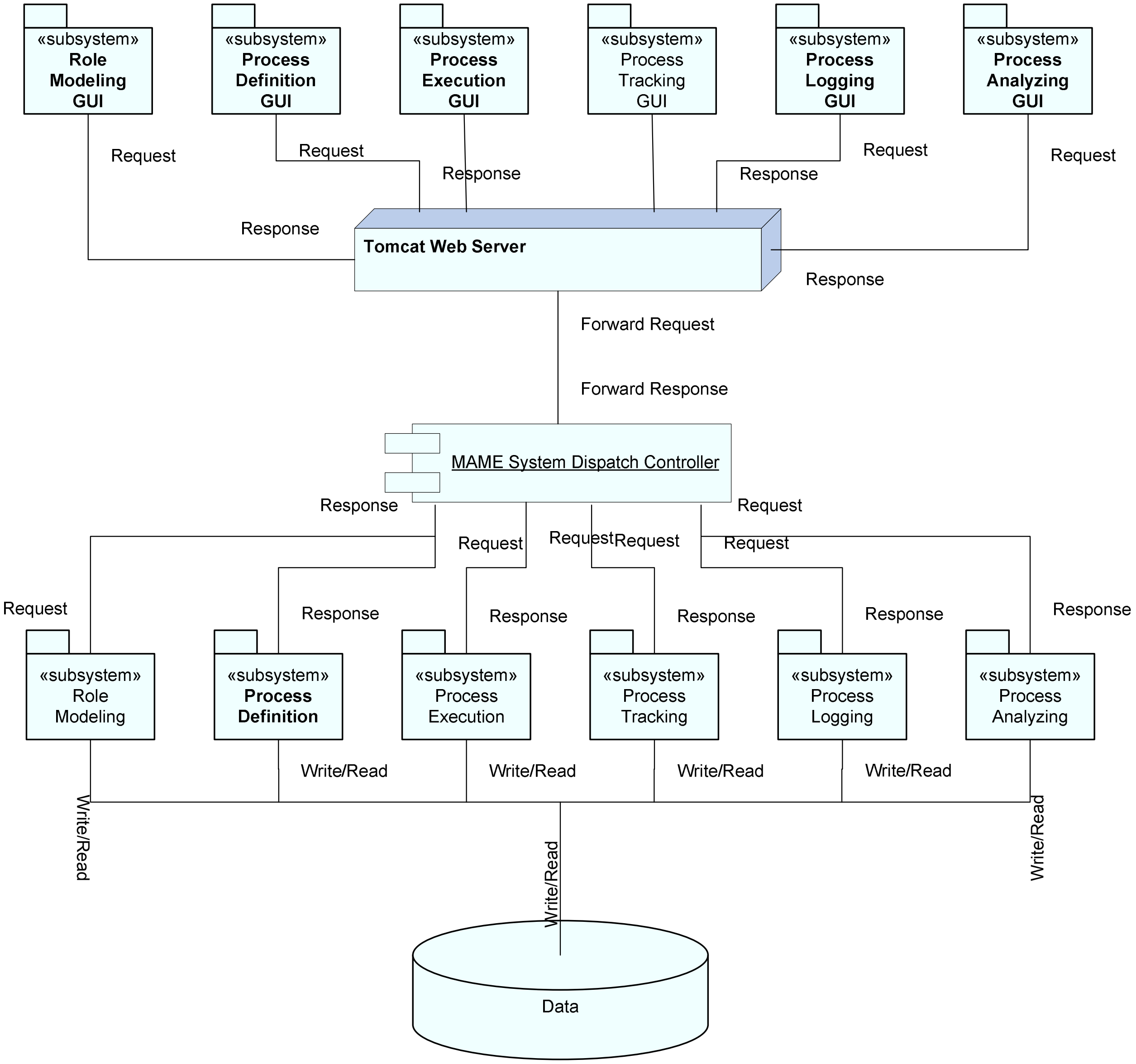

«subsystem» Role Modeling GUI
«subsystem» Process Definition GUI
«subsystem» Process Execution GUI
«subsystem» Process Tracking GUI
«subsystem» Process Logging GUI
«subsystem» Process Analyzing GUI
Request
Response
Request
Response
Response
Request
Request
Response
Tomcat Web Server
Forward Request
Forward Response
MAME System Dispatch Controller
Response
Request
Request
Request
Request
Request
Request
Response
Response
Response
Response
Response
«subsystem» Role Modeling
«subsystem» Process Definition
«subsystem» Process Execution
«subsystem» Process Tracking
«subsystem» Process Logging
«subsystem» Process Analyzing
Write/Read
Write/Read
Write/Read
Write/Read
Write/Read
Write/Read
Write/Read
Data

Figure 1. MAME System Architecture

(1) Presentation Layer

The presentation layer is the user interface of the MAME system. It comprises of a web browser and dynamic components to communicate with the application logic layer. This layer is responsible for accepting user input and actions. Then, it sends all the user requests to the application logic layer through an Apache Tomcat web server. This layer also is in charge of the display of feedback information received from the application logic layer.

HTML and JSP technologies are used in this layer. It provides a dynamic interactive environment for users.

(2) Web Server Layer

The web server layer is the communication layer between the presentation and the application logic on the backend server. It is a delegate layer. It receives users' requests and forwards them to the respective application components for processing.

In the MAME system, the Apache Tomcat is selected as Web server.

(3) Application Logic Layer

The application logic of the MAME application is the core part of the system. It handles all the related processing of the MAME system ~~s~~.

(4) MySQL Database Layer

The MySQL database provides a data repository for the MAME system. This contain all MAME related data including static data and runtime dynamic data. It is the core infrastructure of the MAME system.

- **Subsystems and Description**

MAME is composed of four major subsystems: Role Modeling, Process Definition, Process Execution, Process Tracking, Process Logging, and Process Analyzing. We focus on the first three subsystems below.

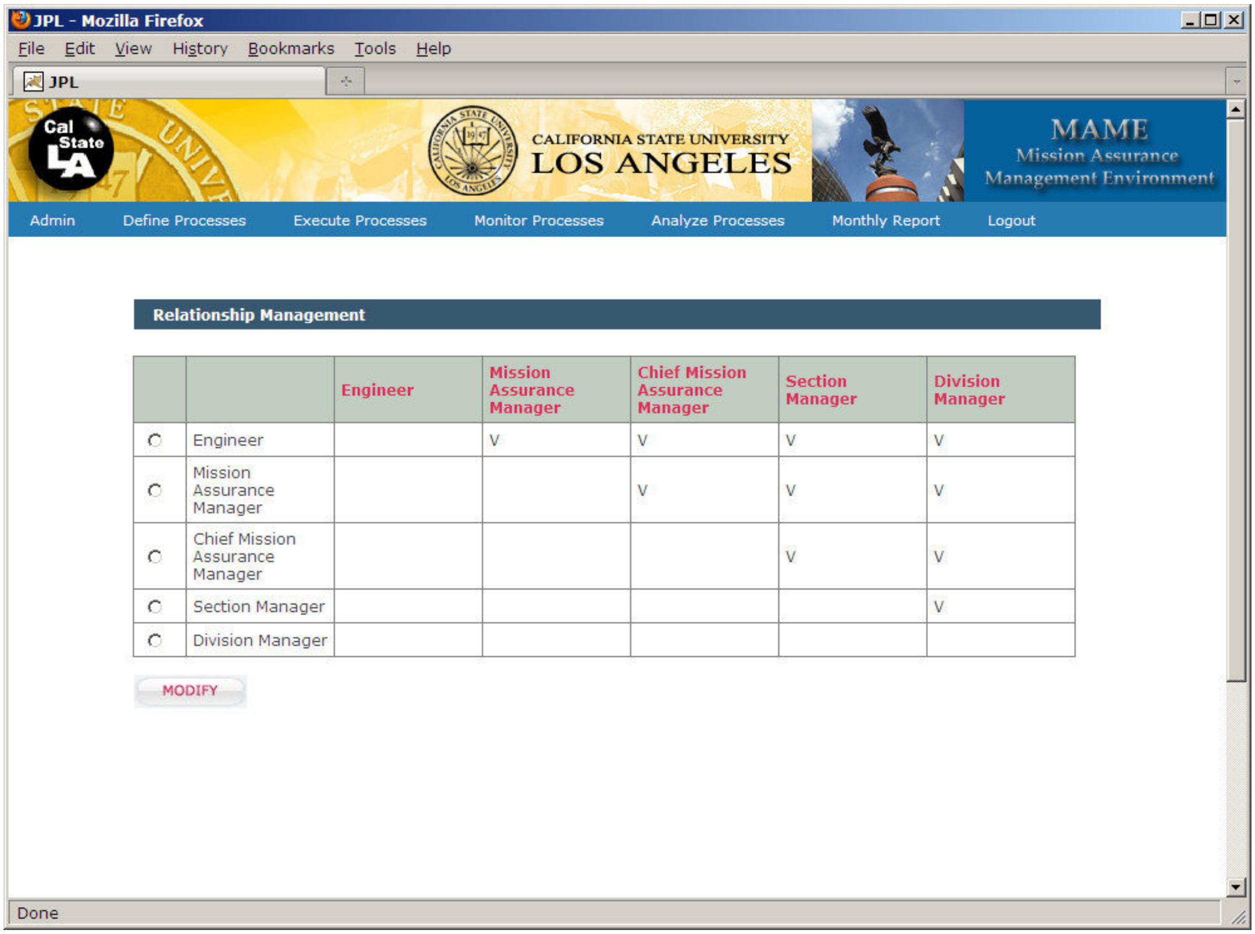
(1) Role Modeling

MAME users have the following attributes: name, role title, contact information, supervisor, appoint date and leave date. Overall, there are two types of people in MAME: Administrators and Users. Users are organized according to a pre-defined role model. A role controls "Approval" and "Permission" throughout the system. It affects what the user can and cannot do. An Administrator can add, edit, and delete these roles.

When an Administrator is adding a new user to the system, he/she inputs their information and selects their role title from a drop down menu. The default roles available are: Engineer, Mission Assurance Manager, Chief Assurance Manager, Section Manager, and Division Manager.

The system also supports user groups. An Administrator can assign specific people to a group. These groups are used to further control "Approval" and "Permission" throughout the system.

Essentially, users are limited by their role, title and additionally, by the group they are assigned to. (Figure 2)



Relationship Management

		Engineer	Mission Assurance Manager	Chief Mission Assurance Manager	Section Manager	Division Manager
○	Engineer		V	V	V	V
○	Mission Assurance Manager			V	V	V
○	Chief Mission Assurance Manager				V	V
○	Section Manager					V
○	Division Manager					

Figure 2. Role Modeling Matrix

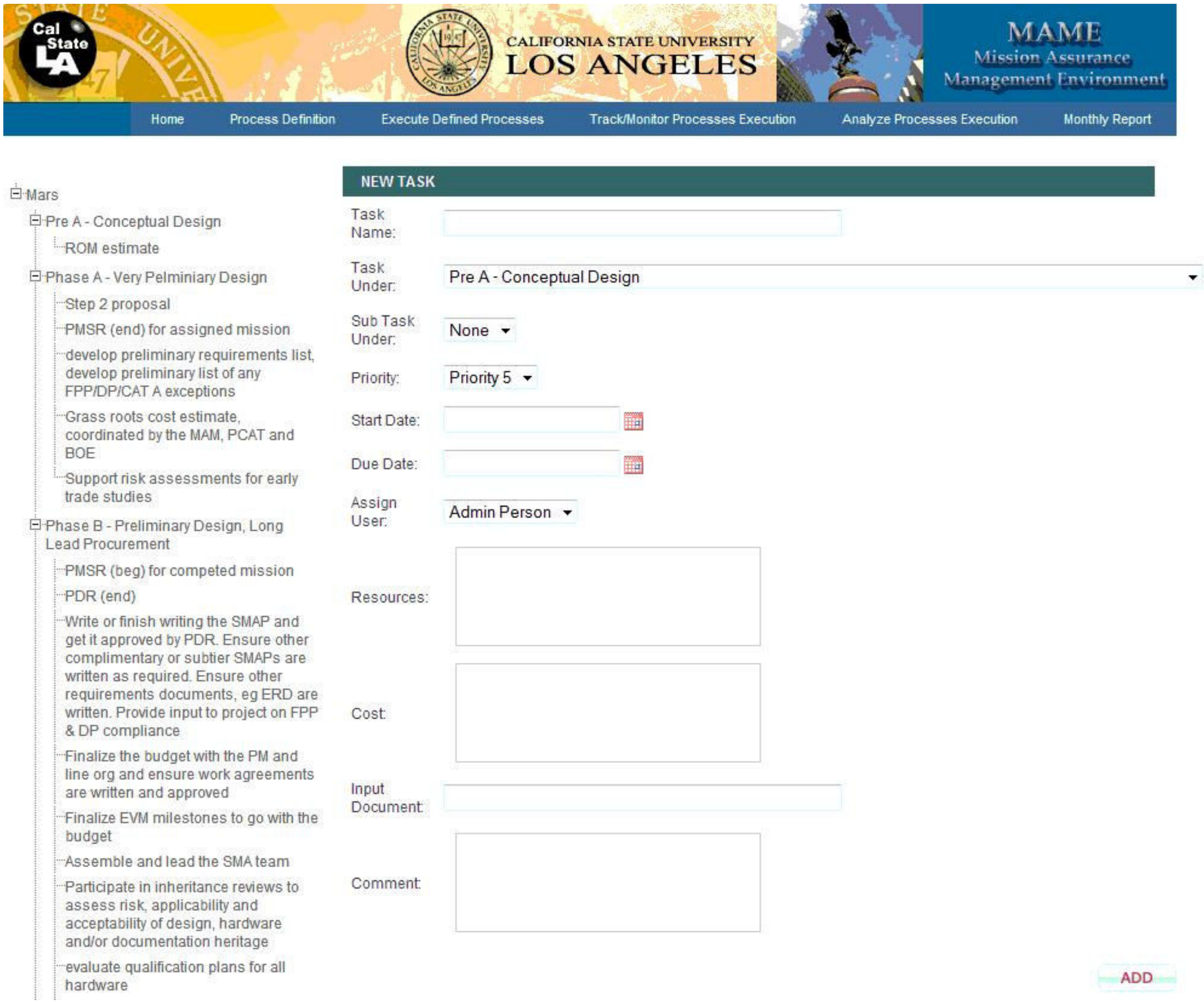
Every role is ranked in the system. In other words, each role is either above or below the other roles. These relationships are defined in a matrix. The relationship management feature defines the relationships among the different roles. Administrators can assign, view, or delete preconditions.

(2) Process Definition

Authorized Users are able to create a project from the following three options: Blank Template, Pre-Defined Template, and Templates from Previous Projects. If the user select Blank template the system will take the user to the process create page in which the user will define each process manually. If the user selects Pre-Defined Template or Templates from previous projects the system will create an exact copy of the template and show all the processes which the user selected. Then the system will ask the user to reset all the dates of each process created. The user will have the option to edit the name or delete the process.

After the user finished reset the dates according to their project MAME will take the user to a page in which the user can add any additional process and see the final process tree. In that page the user will be able to modify any process. After completion of the process definitions MAME will take the user to the task definition page in which the user is able to create all tasks under each process and assign to a corresponding user in which he/she will be responsible of the task.

EditProcess, AddProcess, EditTask and AddTask pages are separately created in Java Server Pages (JSP) just as regular pages. Then in the parent page when the user click on a link to either edit or add it will call the JQuery Modal function in which will build the frame and set the window dimension of the calling page and shadow the parent page and display the window. Tree view is used to display the overall static structure of the process definition. It locates on the left side of the working area. (Figure 3)


Cal State LA
CALIFORNIA STATE UNIVERSITY
LOS ANGELES
MAME
Mission Assurance
Management Environment
Home
Process Definition
Execute Defined Processes
Track/Monitor Processes Execution
Analyze Processes Execution
Monthly Report
ADD

⊟Mars
- ⊟Pre A - Conceptual Design
  - ROM estimate
- ⊟Phase A - Very Pelminiary Design
  - Step 2 proposal
  - PMSR (end) for assigned mission
  - develop preliminary requirements list, develop preliminary list of any FPP/DP/CAT A exceptions
  - Grass roots cost estimate, coordinated by the MAM, PCAT and BOE
  - Support risk assessments for early trade studies
- ⊟Phase B - Preliminary Design, Long Lead Procurement
  - PMSR (beg) for competed mission
  - PDR (end)
  - Write or finish writing the SMAP and get it approved by PDR. Ensure other complimentary or subtier SMAPs are written as required. Ensure other requirements documents, eg ERD are written. Provide input to project on FPP & DP compliance
  - Finalize the budget with the PM and line org and ensure work agreements are written and approved
  - Finalize EVM milestones to go with the budget
  - Assemble and lead the SMA team
  - Participate in inheritance reviews to assess risk, applicability and acceptability of design, hardware and/or documentation heritage
  - evaluate qualification plans for all hardware

NEW TASK

Task Name:	
Task Under:	Pre A - Conceptual Design
Sub Task Under:	None
Priority:	Priority 5
Start Date:	
Due Date:	
Assign User:	Admin Person
Resources:	
Cost:	
Input Document:	
Comment:	

Figure 3. Process Definition

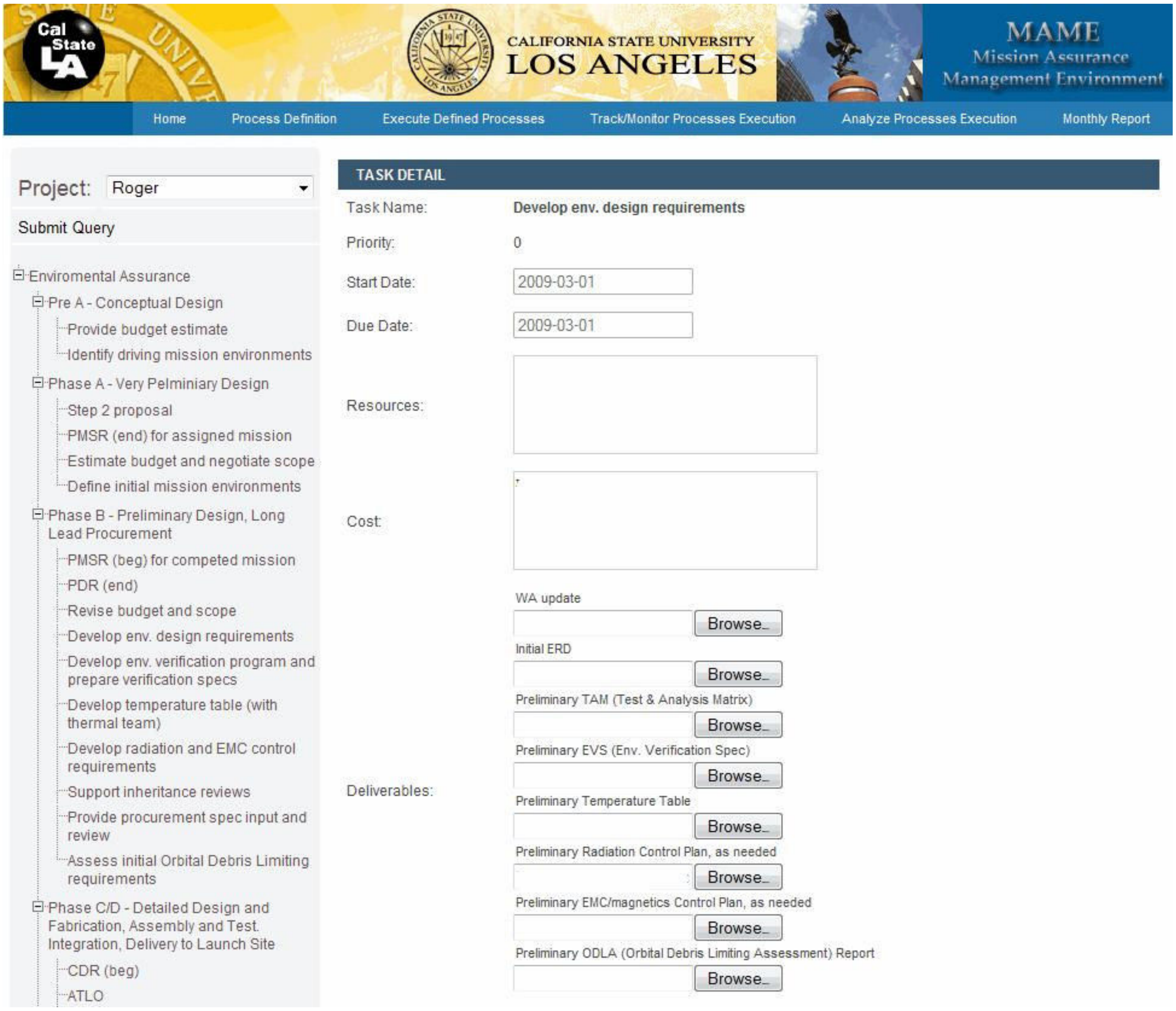
In Process Definition module, an important function is Calendar support to provide start and end dates of each process and task. The MAME calendar in JSP is instantiated by using the JQuery plug-in datePicker library in which it is attached to an html textbox. The html textbox is then associated with one distinctive datePicker object so that in the case when there are many datePicker objects it will automatically fill-in the corresponding textbox with the user selected date. The JQuery DatePicker library can be configured to have previous dates or set a limit in dates to be selected by the user. All it is a done in the DatePicker.js, the look and feel of the Calendar is done using Cascading Style Sheets (CSS) file associated with the library.

In MAME, JQuery is used to construct the tree view structure. Right-click JQuery library is call Context Menu. The library uses a set of unordered list in which each list item is referenced to a JavaScript function to call a different function. After the unordered list and the respective function are developed, the JQuery context Menu is associated to a section of the page in which the right click will take effect [2]. Another option that JQuery Context Menu has is the ability to display and hide options [3].

JQuery Content change is an elegant way to dynamically change the content on several MAME pages such as Process View and Task View without reloading the page. These pages have the tree view structure in which a process or a task has a reference to call the Content change function. The function itself calls another JSP file and passing the caller id. After that the calling page retrieves the data from the database and display as a form [4].

(3) Process Execution

MAME will get the user credential and query the database for data corresponding to the user and display a tree view and only show the project(s) and task(s) associated with it. The user then will be able to see what task(s) he/she must finish and upload/download files of each corresponding task. (Figure 4)



Project: Roger

- Enviromental Assurance
  - Pre A - Conceptual Design
    - Provide budget estimate
    - Identify driving mission environments
  - Phase A - Very Pelminiary Design
    - Step 2 proposal
    - PMSR (end) for assigned mission
    - Estimate budget and negotiate scope
    - Define initial mission environments
  - Phase B - Preliminary Design, Long Lead Procurement
    - PMSR (beg) for competed mission
    - PDR (end)
    - Revise budget and scope
    - Develop env. design requirements
    - Develop env. verification program and prepare verification specs
    - Develop temperature table (with thermal team)
    - Develop radiation and EMC control requirements
    - Support inheritance reviews
    - Provide procurement spec input and review
    - Assess initial Orbital Debris Limiting requirements
  - Phase C/D - Detailed Design and Fabrication, Assembly and Test. Integration, Delivery to Launch Site
    - CDR (beg)
    - ATLO

TASK DETAIL

Task Name:	**Develop env. design requirements**
Priority:	0
Start Date:	2009-03-01
Due Date:	2009-03-01
Resources:	
Cost:	
Deliverables:	WA update Initial ERD Preliminary TAM (Test & Analysis Matrix) Preliminary EVS (Env. Verification Spec) Preliminary Temperature Table Preliminary Radiation Control Plan, as needed Preliminary EMC/magnetics Control Plan, as needed Preliminary ODLA (Orbital Debris Limiting Assessment) Report

Figure 4. Process Execution

Email is an important support in process execution of MAME. The email form was created by first querying the projects database and displaying all projects in a drop box, after a project has been chosen the user will have the option of sending an email to all project participants or choose certain ones. Once the button has been pushed the JSP sends the choice to a servlet and the servlet decides which path to choose. If the choice is send email to all then using SQL queries all participants of the project are entered into the "To" field of the email form, this is done by inputting all emails into a variable and then displaying that variable using a parameter in the input field.

If the user chooses to send email to several users then the servlet takes out all the people that are associated with the project using SQL queries and then shows all of them using checkboxes, those who are checked are being processed and input also into the variable and shown using a parameter in the input field.

After everything is inputted into the “To” field of the form the user writes the subject and the email and the servlet sends it.

MAME creates reminders for work items that have not been completed when due. The email reminders are automatically generated for overdue work items. The email system has the option of checking all the tasks and emailing to someone if his/her task is close to being due or overdue. This is done by a scheduling library called Quartz [5]. Using such library a task is created, and using the CronoTrigger function the program is set to run the task every day at 6 am. The task is a simple java script that sends the program into a java function, the function is going through all the tasks using a SQL script and compares the date that it is due with the current date. If the task is due tomorrow, today, or late due it sends the corresponding email to the user and informs him/her of that. After the task is found the user id is sent to a function which sends the email. The process repeats every day at 6 am.

Conclusion

This senior design project trains and stimulates undergraduate research on current NASA ESMD activities and brings out innovative ideas that can be used to complement those currently under development at JPL. This project not only aids students and faculty participate in ESMD activities at JPL, but also enables better matches between the ESMD work required and background of senior students and faculty. It enables students to meet current and future NASA needs, especially for ESMD areas.

Our students follow the system engineering process to implement the MAME project. In the concept development phase, students use prototyping and visualization technologies to build a concept model and use case models. These models will serve as the fundamental basis of project development. After the team and NASA technical expert review the requirements, students work on the preliminary design phase of the project. For the preliminary design, students choose the design platforms and finish the design of data schema. They also need to complete workflow model and interoperability analysis. NASA technical expert review the preliminary design report. Based on review results, students improve their preliminary design. Next, in the final design and fabrication phase, students divide the system into subsystems, and design and implement subsystem modules and components. At this phase, students conduct unit testing and interface verification. Then, at system assembly and integration phase, students assemble and integrate individual modules and components to achieve specified system requirements. Next, in the validation and test phase, students need to complete system testing and validation testing and demo system functions to NASA technical experts. In addition to that, students are required to document the full project in a high-quality final report and make a final presentation and demonstration to NASA technical experts.

This project helps our students to pursue their careers in aeronautics, space and related fields and to participate in cooperative programs among universities, NASA, and aerospace industry. This project also helps our students to get interdisciplinary training, research and public-service programs related to NASA and aerospace, and to improve their problem solving and critical thinking skills in ESMD areas.

Acknowledgement

This paper is published thanks to the support of Gloria Murphy, the NASA ESMD Space Grant Manager.

References

[1] JPL, MAM Handbook, Technical Report, Jet Propulsion Laboratory, California Institute of Technology.

[2] JQuery, http://www.jquery.com.

[3] JQuery Context Menu, http://www.trendskitchens.co.nz/jquery/contextmenu/.

[4] JQuery Content Change, http://www.thecreativeoutfit.com/index.php?view=Simple-AJAX-Content-Changer-with-EZJax.

[5] Quartz, http://www.quartz-scheduler.org/.