CHAPTER 5: GRAT Implementation and Execution

5.1 Introduction

This chapter looks into the implementation of and the execution of GRAT. The implementation environment is discussed in the early part of this chapter. Following that, the implementation phase is discussed and finally the execution of GRAT is discussed in detail.

5.2 Implementation

There are a number of tools and applications used to implement GRAT. The following sub sections discuss all of them generally.

5.2.1 Environment

5.2.1.1 Lotus Domino and Lotus Notes

Domino is a workgroup application that allows people to share information using networks. Domino servers and Lotus Notes workstations can communicate over local area networks (LANs) and wide area networks (WANs). For this research, Lotus Domino Release 5 was used because it brings messaging, Internet integration, and scalability in one system. The new Domino server includes the latest innovations in Internet messaging, with native support for all the major Internet standards, industry-leading support for Web applications, including CORBA support and integration with Microsoft Internet Information Server (IIS) and increased server reliability and scalability, including improvements in performance, capacity, availability, and maximum database size (Burch, 1999). In addition, the server has a new administration

interface, with a task-oriented approach that makes Domino easier to deploy, use, and manage. Domino Release 5 continues to support a wide variety of clients, in addition to the traditional Notes client.

Messaging features are available to Web browsers and Internet mail clients (such as POP3 and IMAPv4 clients) where else directory features are available to browsers and Lightweight Directory Access Protocol (LDAP) clients. Discussion features are available to browsers and NNTP newsreader clients and administration features are available to browsers as well as the Notes client. Plus, Domino continues to be the best platform for designing dynamic Web applications, and with the new Domino Designer R5 a single application that looks and runs the same for both the Web and Notes clients can be built (Burch, 1999).

In summary, specific Domino R5 features include:

- Internet messaging and directories
  - Provide full-fidelity messaging for your users, with native MIME and SMTP support
  - Use the new Directory Catalog to save space and provide quick name lookups
  - Use new LDAP features to authenticate users in external directories and customize the directory

- Expanded Web application services
  - Design applications with CORBA-standard distributed objects, Java, or JavaScript
  - Use Web clusters for high availability of Web services, expanded security options, and more

- Run Domino using the Microsoft Internet Information Server (IIS) HTTP services

- Database improvements
  - Use transactional logging for faster restarts and data recovery
  - Convert to the new on-disk structure (ODS) for better performance and data integrity

- Easier administration
  - Manage users, databases, and servers with the new Domino Administrator
  - Migrate users from cc:Mail, Microsoft (MS) Mail, Exchange, GroupWise, Netscape Mail, LDAP, or Windows NT with the redesigned user registration
  - Use new tools for server monitoring and message management

Domino Designer is an integrated application development environment in which lets developers and Web site designers create, manage, and deploy secure, interactive applications for the Domino Server. Domino applications let people share, collect, track, and organize information, using Lotus Notes or the Web. Domino applications can cover a wide range of business solutions. For this research, the following features of Domino applications have been implemented.

- *Workflow*. Applications that route information.
- *Tracking*. Applications that monitor processes, projects, performance, or tasks.
- *Collaboration*. Applications that create a forum for discussion and collaboration.
- *Personalization*. Applications that produce dynamic content based on, for example, user name, user profile, access rights, or time of day.

Every Domino application starts with a Domino database. All Domino applications contain one or more Domino databases. A database is the container for the data, logic, and design elements of Domino application. Design elements are building blocks you use to create your application. Design elements that were used for this research include pages, forms, outlines, views, framesets and shared resources.

5.2.1.2 Formula

Formulas are expressions that have program like attributes. For example, one can assign values to variables and use a limited control logic. The formula language interface to Domino is through calls to @functions. @Commands, a subset of the @functions, provide access to the user interface (Lotus, 2000).

Formula language provides syntax and @functions for evaluating constants and variables, and for performing simple logic. Variables can be fields in Notes documents or temporary variables (also called temporary fields) used only for the immediate formula.

5.2.1.3 Lotus Script

LotusScript is an embedded, BASIC scripting language with a powerful set of language extensions that enable object-oriented application development within and across Lotus products (Lotus, 2000). LotusScript allows one to place more complex scripts in a greater variety of locations and events than traditional macros. LotusScript and its development toolset provide a common programming environment across

LotusScript offers a wide variety of features. Its interface to Lotus products is through predefined object classes. The products oversee the compilation and loading of

user scripts and automatically include class definitions to allow more efficient coding. LotusScript offers the following advantages (Lotus, 2000):

- Superset of BASIC

  Since LotusScript is a superset of the BASIC language, it is easy to learn, especially for Visual Basic users. Sophisticated scripts can be written using conditions, branches, subroutines, while loops, and other conventions

- Cross-platform

  LotusScript is a multi-platform BASIC-like scripting language. It works with platforms such as Windows, Macintosh, OS/2, UNIX, OS/390, and AS400. Scripts developed on Windows execute unchanged on any other supported platform. This portability is important as desktop applications become workgroup-enabled and documents are e-mailed to or shared by users.

- Object-oriented

  Lotus products provide Object Classes that are available to LotusScript. Scripts can be written to access and manipulate these objects. The scripts are event-driven, such as by an action, clicking the object or button, opening a document, or opening a view.

- Included in Lotus applications

  LotusScript is supported by Lotus products, so these products can access product classes using a product-supplied LotusScript extension.

5.2.1.4 Internet

Internet is a vast ocean of computers connected over network of cables, as is in Figure 5.1. As of 1996, the Internet was serving over 20 million users worldwide. A quick glance through the history of Internet reveals that Internet began from an obscure network known as ARPANET that was used by America's Department of Defense, its contractors and defense researchers (Honeycutt, 1997). In the late 1980s, this phenomenon wave started to spread its wing across academic organizations around the world and was greatly used by the academic community (Honeycutt, 1997). In 1991, the National Science Foundation of United States of America, who has been funding the Internet all this while, lifted the ban on commercial traffic (Moore, 1994). The results of it have basically changed the living and working culture for most of the people regardless of where they are.

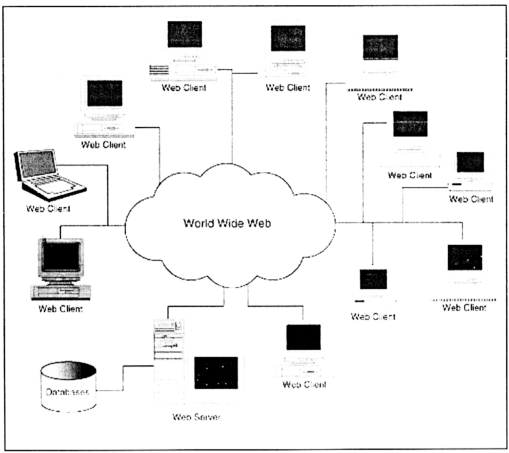

Figure 5.1: Internet in a glance.

The World Wide Web (web) is a network of information resources. The web relies on three mechanisms to make these resources readily available to the widest possible audience, which are (1) a uniform naming scheme for locating resources on the web, (2) protocols for access to named resources over the web and (3) hypertext for easy navigation among resources.

5.2.1.5 HyperText Markup Language (HTML)

To publish information for global distribution, one needs a universally understood language, a kind of publishing mother tongue that all computers may

potentially understand. The publishing language used by the World Wide Web is HTML. HTML gives authors the means to:

- Publish online documents with headings, text, tables, lists, photos and others.
- Retrieve online information via hypertext links or at the click of a button.
- Design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc.
- Include spreadsheets, video clips, sound clips, and other applications directly in their documents.

HTML was originally developed by Tim Berners-Lee while at European Laboratory for Particle Physics or CERN, and popularized by the Mosaic browser developed at NCSA (Reggett, 1999). During the course of the 1990s it has blossomed with the explosive growth of the Web. During this time, HTML has been extended in a number of ways. The Web depends on Web page authors and vendors sharing the same conventions for HTML. This has motivated joint work on specifications for HTML, which is headed by WorldWideWeb Consortium (W3C).

As it hearts, the WWW is nothing but a big collection of HTML files in the connected computers. Web browsers' takes the HTML codes and make it presentable on-screen to the user. HTML is not intended to be an all-encompassing, all-powerful page layout environment. HTML basically uses markup tags to indicate the relative position of elements on the page. However, there are limitations on HTML. Among them are type and size of font, color of the text and screen background. Figure 5.2 briefly depicts where HTML is during a web page is displayed on the web browser.

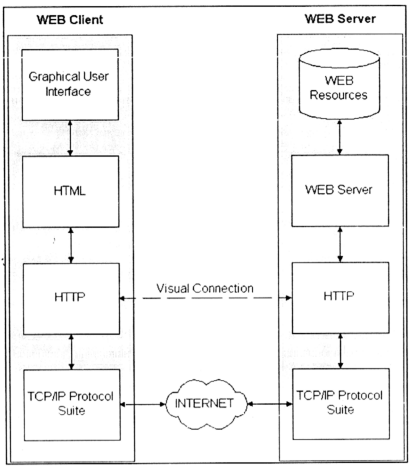

Figure 5.2 : HTML during a web page is displayed

HTML was first developed to maintain the compatibility of the files with any platform or any browsers. However, there are three major defects that make it impossible. The defects are:

- HTML is an evolving standard. Not all the browsers are compatible with the latest HTML standard set by the World Wide Web Consortium (W3C).

- Competition to dominate the browser market has caused two major players, Microsoft and Netscape, to introduce their very own browser-specific tags.

- The operating system and the display characteristics of a computer cause a variation in displaying of pages in the web.

From the research done, HTML's advantage is that it is easy to learn and easy to use. HTML is meant to be open-platform, meaning one could run it from text-only UNIX terminals to the flashiest Silicon Graphics workstation. Although, there certainly be some differences in the same Web page viewed on different machines with different browsers, the results are acceptable enough to convey the information on the page.

5.2.1.6 JavaScript

JavaScript is Netscape's cross-platform, object-oriented scripting language. Client-side JavaScript extends the core language by supplying objects to control a web browser and its Document Object Model (DOM) (JavaScript, 1999). For example, client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks, form input, and page navigation. JavaScript allows applications run over the Internet. Using JavaScript, dynamic HTML pages can be created that processes user input and maintain persistent data using special objects, files, and relational databases.

Web browsers can interpret client-side JavaScript statements embedded in an HTML page. When the browser requests such a page, the server sends the full content of the document, including HTML and JavaScript statements, over the network to the client. The browser reads the page from top to bottom, displaying the results of the HTML and executing JavaScript statements as they are encountered. The components of JavaScript are shown in the figure below.

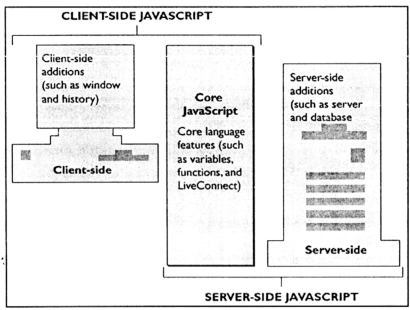

Figure 5.3: Components of JavaScript

Client-side JavaScript statements embedded in an HTML page can respond to user events such as mouse clicks, form input, and page navigation. For example, a JavaScript function could be used to verify that users enter valid information into a form requesting a telephone number or zip code. Without any network transmission, the embedded JavaScript on the HTML page can check the entered data and display a dialog box if the user enters invalid data.

5.2.1.7 Java Applets

Java applets provide a fascinating layer on top of the already dynamic Java language that extends far beyond traditional programming architecture and methodology (Weber, 1997). An applet is a program that is not only runs on any computer but also can be included in a standard HTML page. An applet is a program written in the Java programming language that can be included in an HTML page, much in the same way an image is included. When a Java technology-enabled browser is used

to view a page that contains an applet, the applet's code is transferred to local system and executed by the browser's Java Virtual Machine (JVM).

The advantages of integrating Java applets with web browsers are,

- Applets can usually make network connections to the host they came from.
- Applets running within a Web browser can easily cause HTML documents to be displayed.
- Applets can invoke public methods of other applets on the same page.
- Applets that are loaded from the local file system have none of the restrictions that applets loaded over the network do.
- Although most applets stop running once one leaves the page, they don't have to.

However, there are some setbacks using applets. Current browsers impose the following restrictions on any applet that is loaded over the network(Sun, 2000).

- An applet cannot load libraries or define native methods.
- It cannot ordinarily read or write files on the host that's executing it.
- It cannot make network connections except to the host that it came from.
- It cannot start any program on the host that's executing it.
- It cannot read certain system properties.
- Windows that an applet brings up look different than windows that an application brings up

5.2.1.8 Web Browser

Web browsers are software used by web surfers to view information within World Wide Web. It is used to view pages on and navigate the World Wide Web. Internet surfers are not limited to a specific kind of machine or platform. The browser takes the information it gets from the web server, formats and displays it in screen. Different browsers depending on the capabilities of system and the browsers might display same files differently. Figure 5.4 describes the role of web server in the Internet.

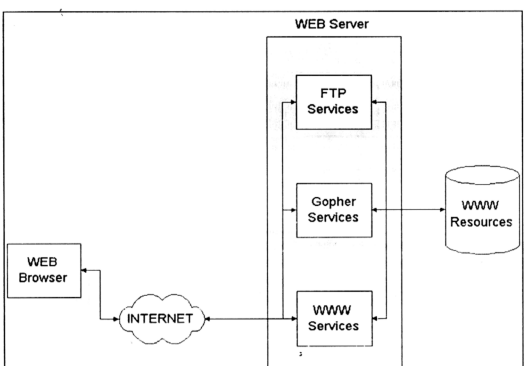

Figure 5.4: Web browser role in the Internet (Abstracted from Walther S.)

For this research, Microsoft Internet Explorer 5.5 or later is recommended due to its capabilities to support Java applets developed by Lotus, JavaScript codes and HTML 4.01.

5.2.1.9 Windows NT

Windows NT 4.0 is the fourth generation of Windows NT. Windows NT is a cross-platform product with identical versions is available for Intel X86, Digital Alpha, Silicon Graphics MIPS, and Apple/IBM/Motorola PowerPC computers (NTServer, 2000). Major hardware manufacturers such as Hewlett-Packard, IBM, Digital, Tandem, Amdahl, and Unisys offer high-end servers designed specifically to run A simple understanding on server and clients can be viewed in Figure 5.5.

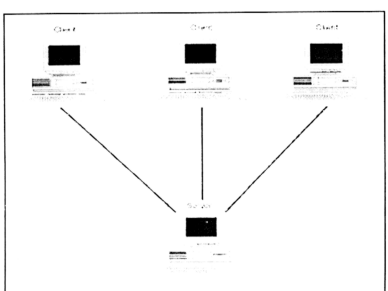

Figure 5.5: Client computer connected to a Server.

Windows NT 4.0 adopts Windows 95's user interface (UI) and operating system shell. Beneath the interface improvements, Windows NT Server 4.0 provides several new networking features, the most important of which for networking is the Distributed Common Object Model (DCOM) and a substantial improvement in the Domain Name Service (DNS) for TCP/IP networks. Figure 5.6 shows how Windows NT complies with Open System Interface(OSI) (NTServer, 2000).

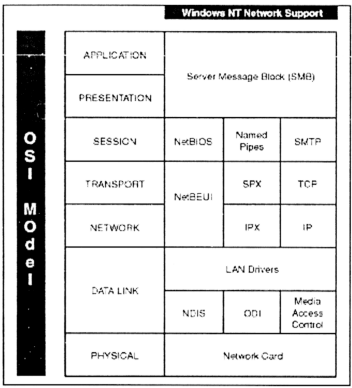

Figure 5.6: Windows NT and OSI (Abstracted from Wolters V.)

Ever-increasing network traffic and expansion of LANs and WANs to accommodate a larger number of domains requires commensurate enhancement of server capabilities, and especially the performance of servers used as domain controllers. Microsoft claims up to double the throughput over 100BaseT networks compared with Windows NT Server 3.51, based on tests conducted by National Software Testing Laboratories (NTSL) (NTServer, 2000).

Basically, in this project, in order to for Domino and its services to run, Windows NT Server 4.0 was used although Windows 2000 was released in the mid of the project development.

5.2.2 GRAT Phases

This section describes how the various phases identified in earlier chapters are implemented. A more detailed description along with the screen capture is presented in Appendix B.

5.2.2.1 Project Repository

Figure 5.7 shows the design and the layout of the Project Repository which lists all project that is currently active. Users can either choose to create a new project by clicking on raised button on the picture or view the project details by double clicking on the project's name itself. Only the Domino Administrator is able to view and see the delete button. The project list is shown by using an applet.

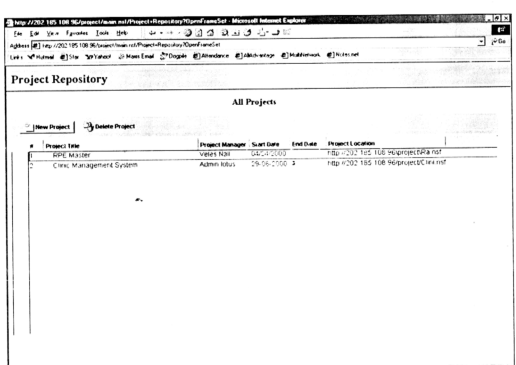

Figure 5.7: Project Repository

5.2.2.2 Domain Understanding

Figure 5.8 shows the list of submitted documents in a applet. Same as Project repository, to submit a new material, simply click a button and to view it, double click on the material's information on the applet.

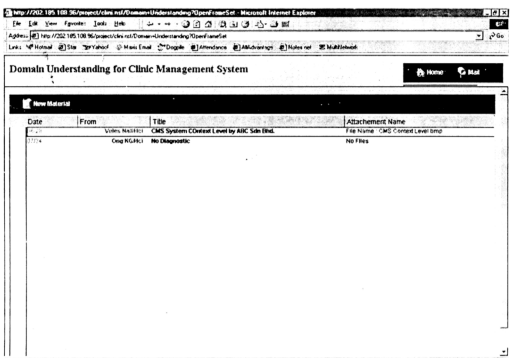

Figure 5.8: Domain Understanding

5.2.2.3 Requirements Collection

Figure 5.9 shows the platform to collect the requirements and to view the requirements submitted. The view is updated every time a requirements is submitted or every 15 seconds, whichever is sooner. The frame below the view is the place where team members are required to type the requirements.

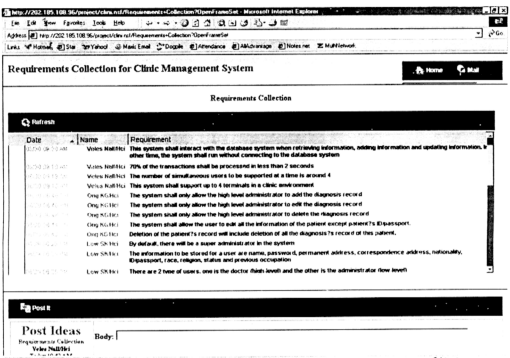

Figure 5.9: Requirements Collection

5.2.2.4 Categories Collection

Figure 5.10 depicts the categories collection phase. The picture shows the list of requirements and categories, which team members think is suitable for the requirements listed. To submit a category, team members have to click a button to add category and submit the category. Again applets are used to show the list of requirements and categories

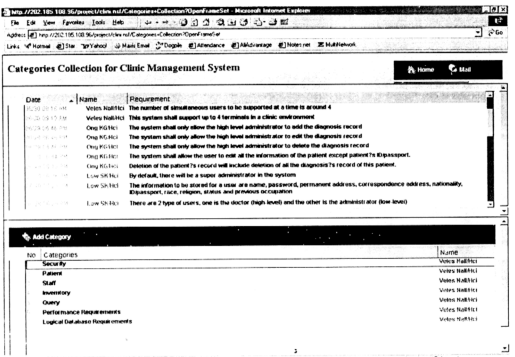

Figure 5.10: Project Repository

5.2.2.5 Classification

Figure 5.11 shows how team members do the classification process. They need to match the appropriate requirements with the category they feel suitable. The categories are listed in a list box. Since tables were used to list the boxes and the list box and since both of number of requirements and categories are not limited, meaning that they are dynamic, JavaScript was used to create the dynamic HTML as seen in the picture.

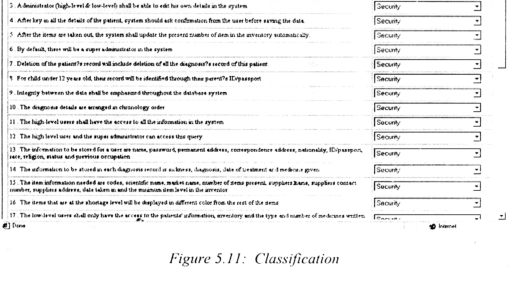

Figure 5.11: Classification

5.2.2.6 Conflict Resolution

The team members are able to view the conflicts and the response to the conflict in a hierarchy manner as shown in Figure 5.11. In this manner, viewers are able to look at the requirements according to the categories and the conflicts according to the requirements and the responds according to the conflicts. The title of the conflicts and responds are shown on the view and for more detailed description of the conflicts and responds, team members have to double-click on the selection.

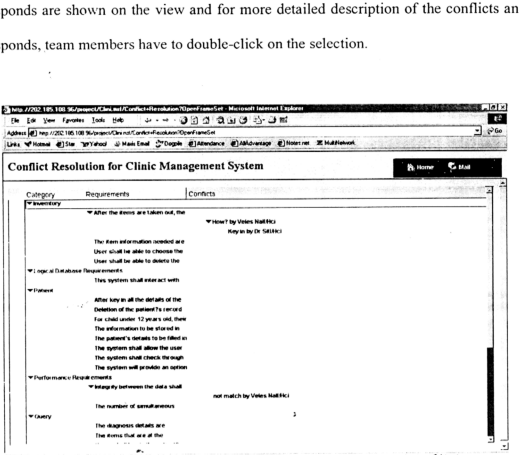

Figure 5.12: Conflict Resolution

5.2.2.7 Prioritization

Figure 5.13 depicts the process of prioritization. The concept of prioritization is same as classification as mentioned before. Team members need to match the requirements with the priority, which are listed in a list box. The scale starts from 1 means the highest priority. Supposing a category has 6 requirements, than the scale are from 1 to 6 with 1 being the highest priority and 6 as the lowest priority.

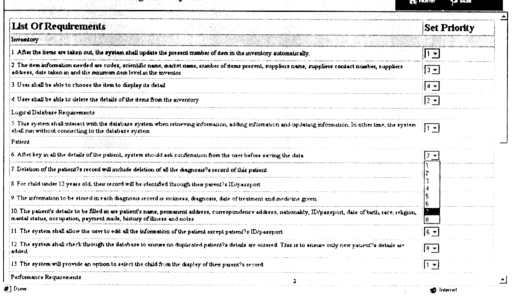

Figure 5.13: Prioritization

5.2.2.8 Requirements Validation

The team members are able to view the invalid characteristics of the requirements in a hierarchy manner as shown in Figure 5.14. In this manner, viewers are able to look at the prioritized requirements according to the categories and the invalidities according to the requirements. If team members feel a requirements is invalid, than he or she have to double-click the requirements and then submit a requirements invalid form.

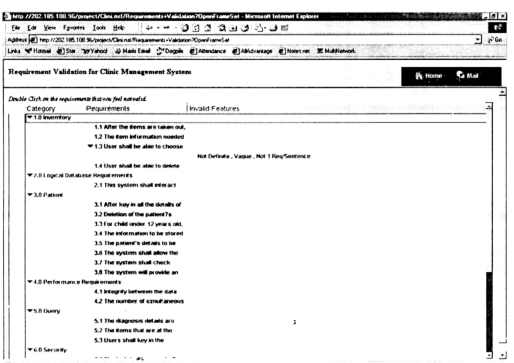

Figure 5.14: Requirements Validation

5.2.2.9 Change Phases

This function is only available to the project manager. The project manager is supposed to control the flow of the requirements analysis process by setting and changing the phases of the process. This is done as shown in Figure 5.15. Project Manager needs to double click on the phase to set the phases.

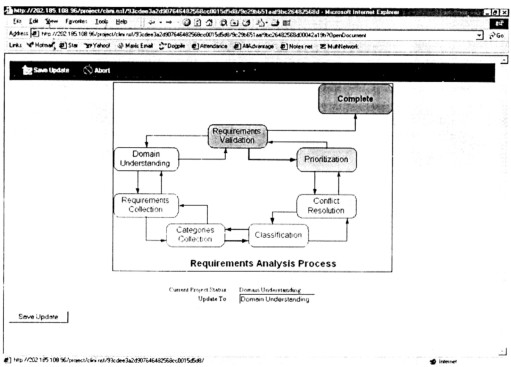

Figure 5.15: Change Phase

5.2.2.10 Activity Scheduling

This service is available to all members. By simply filling and submitting the form as shown in Figure 5.16, the scheduled activities are able to be seen by other members when they login to the project.

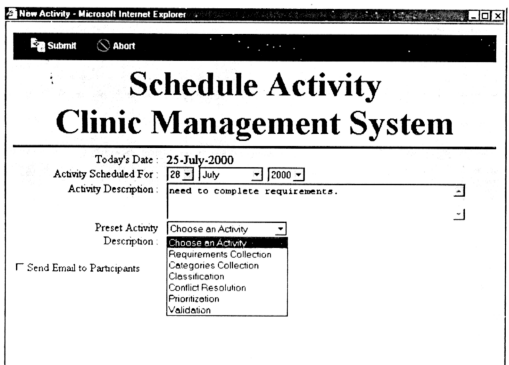

Figure 5.16: Activity Scheduling

5.2.2.11 Completion

At the end of the requirements analysis process, the completed list of requirements for the project are available to users as shown in Figure 5.17.

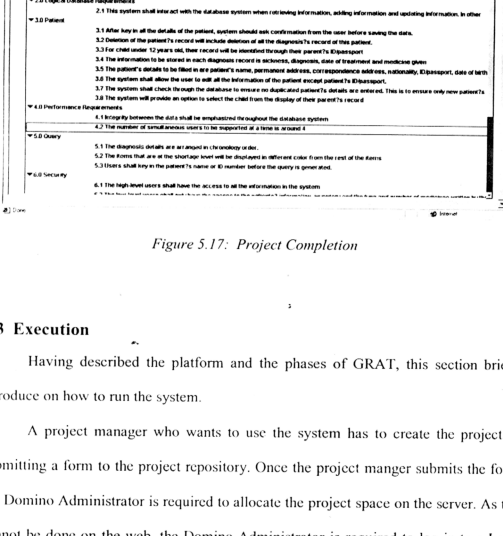

Figure 5.17: Project Completion

5.3 Execution

Having described the platform and the phases of GRAT, this section briefly introduce on how to run the system.

A project manager who wants to use the system has to create the project by submitting a form to the project repository. Once the project manger submits the form, the Domino Administrator is required to allocate the project space on the server. As this cannot be done on the web, the Domino Administrator is required to log in to a Lotus Client machine and allocate the project space and the web site. This can be done at a

click of a button. Once completed, the details of that project are updated and the project location is saved for other users to visit the page.

The project starts with the Domain Understanding phase. This is where team members share their knowledge, experience and materials. All they have to do is submit a form and the materials they submitted are available to others. The materials could range from opinion to web links to files.

Once the project manager feels that this phase is complete, he or she could change the phase by simply clicking at the appropriate phase. Next time the user logs in, it directly updates with the updated phase. Anyone could also post an activity so that others know what is going on that particular day.

The next phase is Requirements Collection. Here all the team members have to do is simply brainstorm the requirements they feel suitable for the project. There would not be any discussion among team members. The requirements list are updated when a requirements is posted or 15 seconds, whichever is sooner.

Following that would be the Categories Collection. The concept is same as requirements collection. The categories are collected. For the ease of users, the lists of requirements are also shown so that categories appropriate to the requirements can be derived.

Next is the Classification. This is a matter of matching the requirements and the categories collected earlier. Once all the requirements have been matched, users can preview and edit the classification. Once satisfied, they can submit. Members who have not submitted are identified and listed.

Moving to the Conflict Resolution phase, here team members can raise any doubts and conflicts they find in the requirements. Those who feel can clarify the conflict can choose to respond to the conflict. At the end of the phase, the Project Manager has to delete the requirements which he or she finds it conflict.

The Prioritization phase is also conceptually same as Classification. The team members are required to match the requirements with the right level of priority. The level of priority starts from 1 representing the highest level of priority and lowest priority level is equivalent to the number of requirements in a category. Each requirement in a category must have a unique value. Once this condition is satisfied, users preview the result and reprioritize. They can choose to submit and as the Classification process, the names of members who have not completed this phase are listed.

Now, the Requirements Validation is a phase, which requires team members to identify requirements that they feel not valid. The identified requirements are shown in a form with different characteristics of invalidity. Team members need to select the appropriate checkboxes and submit the form. Again the project manager justification comes into play. If the Project Manager feels a requirement is not valid based on the invalidity identified by the user, he or she has to delete the requirements.

Finally, once all the process is completed, the Completion phase simply would the final list of requirements.

5.4 Summary

This chapter has described the implementation and the execution of GRAT. The implementation looked into developmental environment of GRAT and the manners of the GRAT phases were implemented. Each phase was briefly described with a general screen capture presented. A more detailed description of the system with all the images is presented in Appendix B. The second part, execution discusses about the flow and the actions required for the GRAT system.