Chapter 5 – G-Jigsaw Analysis and Design

This chapter presents the analysis and design of G-Jigsaw. The first part depicts the requirements analysis and object-oriented analysis of G-Jigsaw. G-Jigsaw architecture, design specification and user interface design are presented in the second part.

5.1 G-Jigsaw Analysis

Analysis is the process of extracting the needs of a system and what the system must do to satisfy the user's requirements (Bahrami, 1999). It is a very important phase to ensure the developed system will function according to users' needs. G-Jigsaw analysis includes both requirement analysis and object-oriented analysis.

5.1.1 Requirement Analysis

Requirement analysis is a process of discovery, refinement, modeling and specification (Pressman, 1992). It bridges the gap between system level software allocation and software design, enables software functions and performance to be specified, indicates software's interface with other system components and establishes design constraints that the software must meet.

Requirement analysis always encompasses functional requirements and non-functional requirements. Functional requirements describe how the system should behave given certain stimuli (Pfleeger, 1998). They illustrate interactions between the system and their environment. Conversely, non-functional requirements do not specifically focused on the system functionality. They place restrictions on the development process of the product being and specify external constraints that the product must meet. Non-

functional requirements include safety, security, usability, reliability and performance requirements (Kotonya and Sommerville, 1998).

5.1.1.1 G-Jigsaw Functional Requirements

G-Jigsaw functional requirements are based on the group jigsaw process model discussed in chapter 3. The process model is supported by G-Jigsaw modules named Jigsaw Task, Initial Group, Expert Group and Jigsaw Group. Besides that, G-Jigsaw supports two types of users with different responsibilities, which are teachers and students. Teachers are responsible for preparing the jigsaw tasks and evaluating students' answers. Students are engaged in three continuously stages of jigsaw collaborations, which are inter-member (group level) in Initial Group, inter-group (group level) in Expert Group and inter-member (group, classroom level) in Jigsaw Group.

Functional Requirements for Jigsaw Task Module

The Jigsaw Task Module supports both teachers and students. The functional requirements for these users are listed accordingly below:

1. Teachers are able to:
    i. create new jigsaw tasks for students' participation.
    ii. retrieve and use existing questions through question templates for the jigsaw task.
    iii. upload graphics for the jigsaw task.
    iv. modify created jigsaw task.
    v. extend jigsaw task submission date.
    vi. delete expired or unwanted jigsaw task.

2. Students are able to:
    i. view jigsaw tasks prepared by teachers.
    ii. participate in jigsaw tasks.
    iii. resume their uncompleted jigsaw task.

This module will automatically filter, categorize and save new questions in a shared repository.

Functional Requirements for Initial Group Module

The Initial Group Module supports student's inter-member (group level) collaboration. The functional requirements for this module are depicted below:

1. Students are allowed to:
    i. select their group and question assigned by their teacher.
    ii. view their selected question and other group members' questions.
    iii. respond to all group members.
    iv. view group members' responses towards their responsible question.
    v. create a summary for their responsible question.
    vi. retrieve and modify group members' responses into their summary.

2. Besides that, it is able to:
    i. create a profile for each student who starts a new jigsaw session.
    ii. keep track on the number of questions that students have responded.
    iii. inform students when there are no more questions to be responded.
    iv. control and automate the jigsaw activity.

3. Explicit controls are implemented for students to:
    i. respond to all their group members in any sequence.
    ii. create a summary only when they have read all their group members' responses.
    iii. proceed to the next stage (i.e. Expert Group) only when they have

submitted their summary.

Functional Requirements for Expert Group Module

The Expert Group Module supports student's inter-group (group level) collaboration. The functional requirements for this module are as follows:

1. Students are able to:
    i. view summaries of other groups.
    ii. respond to other groups members' summaries.
    iii. receive comments and feedback towards their own summary.
    iv. respond to feedbacks they received in order to discuss or clarify issues raised by other groups' members.
    v. create a report for their responsible question.
    vi. retrieve their summary as a part of their report answer.
    vii. view responses received towards their summary.
    viii. modify and improve their responsible question report.

2. Explicit controls are implemented for students to:
    i. create a report only when they have read all other groups members' summaries.
    ii. proceed to the next stage (i.e. Jigsaw Group) only when they have submitted their report.

3. Besides that, this module is able to control and automate the jigsaw activity.

Functional Requirements for Jigsaw Group Module

The Jigsaw Group Module supports student's inter-member (group, classroom level) collaboration. The functional requirements for this module include:

1. Students are able to:
    i. view all their group members' reports.
    ii. modify their own report if necessary.
    iii. create an integrated report (only by the group leader).
    iv. view their group and other groups' individual and integrated reports.

2. Besides that, it is able to perform the integration process automatically.

3. The explicit control implemented in this module is only the group leader is allowed to perform the integration report process.

5.1.1.2 G-Jigsaw Non-functional requirements

Non-functional requirements restrict how the functional requirements should be implemented. The non-functional requirements for G-Jigsaw include:

1. The system should provide an easy user interface for users to understand how to use the system.
2. The system should be expendable for future enhancement.
3. The system should be able to provide good security control for all documents and pages in the database.

These non-functional requirements define the overall qualities of G-Jigsaw.

5.1.2 Object-Oriented Analysis

"Object-oriented Analysis (OOA) is a method of analysis that examines requirements from the perspective of the classes and objects found in the vocabulary of the problem domain." - Grady Booch (Verma, 2000). Unified Modeling Language (UML) is one of the modeling languages used in OOA and it is selected to produce the G-Jigsaw analysis

model. Use Case Diagrams and Class Diagrams are used in the analysis process.

5.1.2.1 G-Jigsaw's Use Case Diagrams

A use case is a typical interaction between user and the system in order to achieve some goals (Verma, 2000). It provides a communication basis between users and developers in planning a project. Use case diagrams are used to visualize use cases as primary elements in software development. These diagrams consist of actors, a set of use cases enclosed by a system boundary and relationship among the use cases. An actor is a role of an external object or a user plays with respect to the system. There are two types of relationships (i.e. uses and extends) among use case diagrams. The 'uses' relationship is used to avoid repetition when there are some general behavior that are similar across more than one use case while the 'extends' relationship is used to describe a variation on a normal behavior.

There are 26 use cases in the G-Jigsaw analysis model. These use cases are identified and categorized into 5 modules. All use cases in a module can be represented in one use case diagram. Each of these use case can be described by using a scenario. Interaction diagrams illustrate the details of these scenarios in the object-oriented design (OOD) phase. Only two actors (i.e. teacher and student) are involved throughout the entire process. This chapter only shows 3 use case diagrams and the other use case diagrams are attached in Appendix A-1.

Jigsaw Task Module

There are 8 use cases in the Jigsaw Task module. Figure 5-1 shows the use case diagram for this module. In this module, only teacher can create new jigsaw task for

students' participation. Teachers can retrieve existing questions from a shared repository using available task templates. Teacher can edit or delete the questions if necessary. After the jigsaw task is created, both teachers and students can view it. They can also navigate other modules.

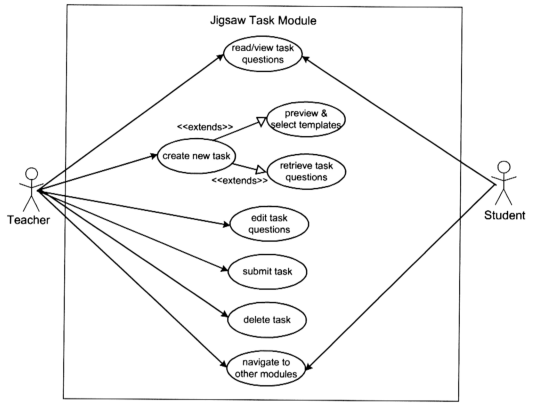

Figure 5-1 Use Case Diagram For Jigsaw Task Module

Initial Group Module

There are 7 use cases in the Initial Group module as shown in figure 5-2. In this module, only students are involved. When they start to participate in the jigsaw activity, students set their own profile, post responses to their group members, read responses from their group members and compose a summary for their selected question to be used in the Expert Group discussion. Students can retrieve and use their group members' responses as part of their summary.

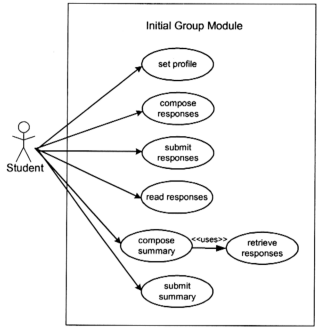

Figure 5-2 Use Cases Diagram For Initial Group Module

Expert Group Module

Figure 5-3 presents the Expert Group module's use case diagram. There are 5 use cases in this module. Similar to the Initial Group module, this module involves only the students. Students can read other groups' summaries and post and receive comments from other groups members. After the expert group discussion, students need to create a report. Their can retrieve and use their previous summary to be apart of their report. These individual reports will be integrated later in the Jigsaw Group.

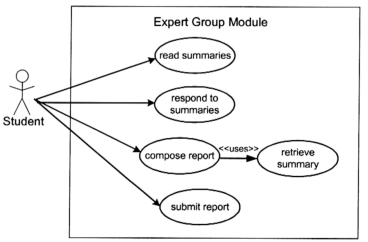

Figure 5-3 Use Cases Diagram For Expert Group Module

5.1.2.2 G-Jigsaw Class Modeling

UML class diagrams show the classes of the system, their inter-relationships, the operations and attributes of the classes (Ambler, 2003). In this section, class diagrams are used to analyze requirements in the form of an analysis model. Thus, the focus is to identify the classes' responsibilities instead of focusing on specific attributes or operations. A class model comprised one or more class diagrams and the supporting specifications that describe the model elements include classes, relationships between classes and interfaces (Ambler, 2003). The G-Jigsaw class model is depicted in figure 5-4. This model serves as an analysis model that shows a general view on how various classes in the system collaborate with each other. The design details for G-Jigsaw are discussed in the OOD section.

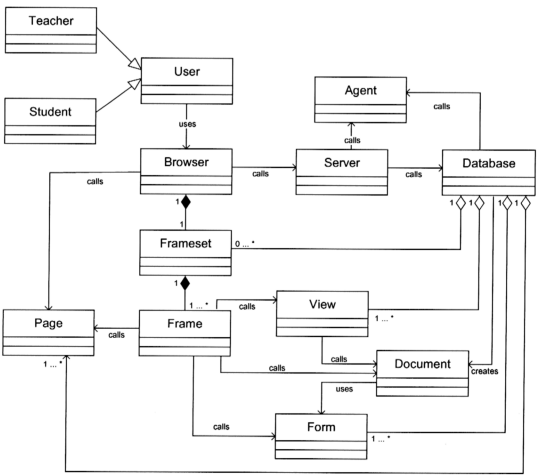

Figure 5-4 G-Jigsaw Class Model

5.1.3 Summary of Web Agents Requirements in G-Jigsaw

This section summarizes the functional requirements in G-Jigsaw that involve web agents.

In Jigsaw Task Module:

1. The Sharing Agent enables the teachers to retrieve and use existing questions through question templates for the jigsaw task.
2. The Sharing Agent will automatically filter, categorize and save new questions in a shared repository.
3. The Navigation Agent enables the students to resume their uncompleted jigsaw task.

In Initial Group Module:

1. The Sharing Agent enables the students to retrieve and modify group members' responses into their summary.
2. The Navigation Agent:
    i. creates a profile for each student who starts a new jigsaw session.
    ii. keeps track on the number of questions that students have responded.
    iii. informs students when there are no more questions to be responded.
    iv. controls and automate the jigsaw activity.
3. The Navigation Agent enforces some explicit controls for students to:
    i. create a summary only when they have read all their group members' responses.
    ii. proceed to the next stage (i.e. Expert Group) only when they have submitted their summary.

In Expert Group Module:

1. The Sharing Agent enables the students to retrieve their summary as a part of their report answer.
2. The Navigation Agent enforces some explicit controls for students to:
    i. create a report only when they have read all other groups members' summaries.
    ii. proceed to the next stage (i.e. Jigsaw Group) only when they have submitted their report.

In Jigsaw Group Module:

1. The Integration Agent enables the students to perform the report integration automatically.
2. The Navigation Agent allows only the group leader to perform the integration report process.

5.2 G-Jigsaw Design

Design is a creative process of transforming a problem into a solution; the description of a solution is also called design (Pfleeger, 1998). It is a phase that emphasizes on how the system should perform in order to fulfill the requirements identified in the analysis phase. G-Jigsaw Design includes the Architecture Design, Object-oriented Design as well as Interface Design.

5.2.1 Architecture Design

G-Jigsaw is a web-based application that resided on the Lotus Domino platform. Figure 5-5 depicts the G-Jigsaw's environment that comprises of both hardware and software components. The hardware components of G-Jigsaw consist of client computers, a domino server computer and a network that connects them. The software components of G-Jigsaw reside on the client and server computers. There are three-levels of software architecture for its client and server, which are the client/server level, Notes Object Services (NOS) level and Database/File level.

The client and server programs are located in the client/server level of the software architecture. The Domino Server program supports the connection between clients and server and manages a set of server tasks.

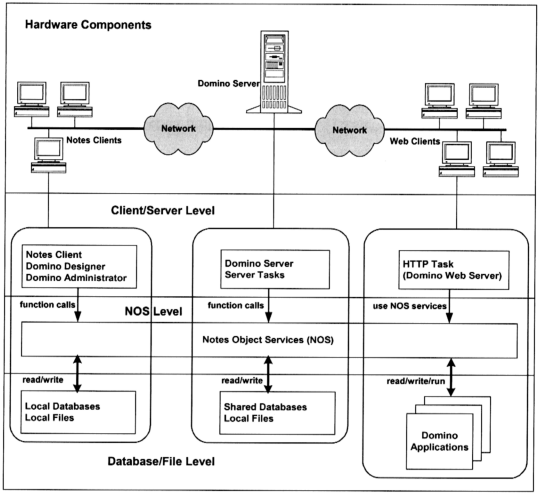

Figure 5-5 G-Jigsaw Environment

G-Jigsaw supports two types of clients, the Notes Clients and Web Clients. The Notes Clients comprises of Lotus Notes Client, Domino Designer and Domino Administrator. The Notes Clients are mainly restricted to the developers and administrators only. Developers use Domino Designer to modify and customize G-Jigsaw design elements and to control users' access to the G-Jigsaw database. Administrators use Domino Administrator to register new users into the Domino server and manage the Domino Web Server configuration. On the other hand, the G-jigsaw Web Clients are the Internet Browser that connects to the Domino server. Teacher and students can use the Web Clients to carry out various collaborative learning activities supported by G-Jigsaw.

The second level in G-Jigsaw architecture is the NOS level. The NOS is a set of portable C/C++ functions that create and access information in databases and files, compile and interpret formulas and scripts and as an interface for the operating system services. The client and server programs use NOS to create, modify, read and maintain databases and files. The third level of G-Jigsaw architecture is the database/file level. This level serves as repositories for all the data stored in databases or files. G-Jigsaw is a shared database that can be access over the network.

Since G-Jigsaw is designed as a web application, therefore, it is important to know the architecture of the Domino Web server. Lotus Domino Server can be a Web application server that provides an integrated set of services to create more secure interactive Internet and Intranet applications. In this context, the HTTP (Hyper Text Transfer Protocol) server task of the Domino Server plays an important role in handling requests from web clients. Figure 5-6 illustrates the Domino Web Server architecture.

As shown in figure 5-6, the heart of the Domino Web application server is the HTTP server task. Requests from Web clients go directly to the HTTP task to be processed. When the HTTP task receives a request from a Web browser, a connection is made to the HTTP stack. The HTTP stack includes all codes that deal with both inbound and outbound HTTP communications. It manages the connection between Web clients and server. The default action for HTTP stack is to send the request directly to the URL (Uniform Resource Locater) parser. The URL parser handles incoming Domino URL calls. It determines if it is a standard URL or a special Domino URL.

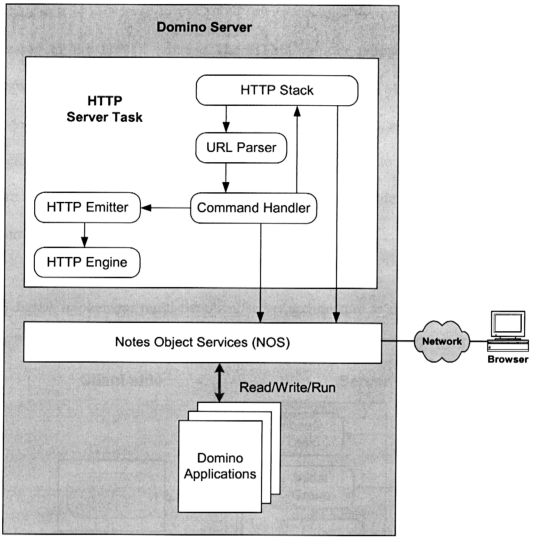

Figure 5-6 Domino Web Server Architecture

If it is a standard URL, the parser sends the information to the HTTP stack, which processes it as simple HTTP commands. If it is a Domino URL, the parser breaks the URL into different parts, performs a series of checks, provides implicit commands if necessary and invokes appropriate command handlers. These command handlers are direct links between HTTP task and NOS. They manage all details associated to each command by establishing the correct identity for security purposes, accessing NOS, executing formulas and scripts and retrieving information.

The result is sent back to the HTTP server through the command handlers and then passed to the HTML emitter. The HTML emitter prepares the outgoing flow of information based on the URL calls results. It uses HTML engine to translate the information into HTML pages. The HTML engine acts as a standards source that defines the proper format of any information translated into HTML. Finally, the pages are sent to HTTP stack, which establishes a connection to the client and display the data correctly.

G-Jigsaw implements multi-tier client/server architecture as shown in figure 5-7.

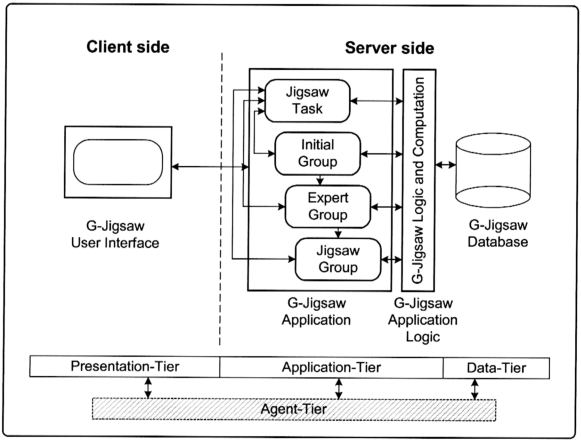

Figure 5-7 G-Jigsaw's Multi-tier Client/Server Architecture

In this architecture, G-Jigsaw's user interface is located in the presentation-tier at the client side, which receives users' inputs and display information to users through a web browser. G-Jigsaw's modules and its application logic reside on the application-tier at

the server side. The application-tier is the core component that supports the jigsaw activities. The data-tier provides access to G-Jigsaw database. The database can only be accessed through the application-tier. In addition, an agent-tier is embedded in this architecture to support the deployment of web agents. The agent-tier has its own unique internal multi-agent architecture that enables the communication within the web agents as well as its external entities located in other tiers. This internal multi-agent architecture will be discussed in more details in chapter 6.

5.2.2 Object-Oriented Design

"OOD is a method of design encompassing the process of object-oriented decomposition and a notation for depicting both logical and physical as well as static and dynamic models of the system under design" - Grady Booch (Verma, 2000). The analysis model presented in section 5.1.2 serves as a core basis for the design models, while a mature design model provides better guidelines for programming activities. Sequential Diagrams and Activity Diagrams are used in the design process.

5.2.2.1 G-Jigsaw's Sequential Diagrams

A Sequence Diagram provides a diagrammatic representation of a specific instance of a use case (a scenario) (Verma, 2000). It provides a way to visually step through invocation of the operations defined by the classes. The G-Jigsaw design model uses sequential diagrams to validate and flesh out the logic of a usage scenario. A usage scenario is the description of a potential way that the system is used (Ambler, 2003). This chapter shows only 5 sequential diagrams and the other diagrams are enclosed in Appendix A-3.

Create New Jigsaw Task Scenario

Teacher clicks on the create new task link

Browser requests question template form from server

Server retrieves template form from server database

Server database searches for requested template form

Template form is sent to browser

Browser displays template form to teacher

Teacher previews and selects template

Teacher set questions

Teacher retrieves questions

Teacher modifies questions

Teacher submits task questions to server database

Server database activates the web agent

Web agent filters the submitted questions

Web agent categorizes the task questions

Web agent updates the server database

Server database sends confirmation message to browser

Browser displays message to teacher

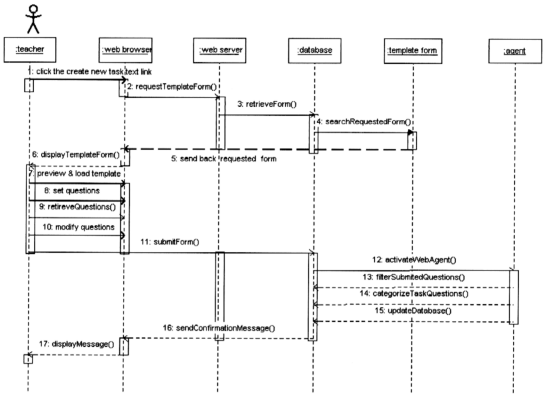

Figure 5-9 Jigsaw Task Module's Create New Jigsaw Task Scenario

Compose Response Scenario

Student clicks the Post Response button

Browser requests the response form

Server retrieves the requested form from server database

Server database searches for the requested form

Response form is sent to the browser

Browser displays response form to the student

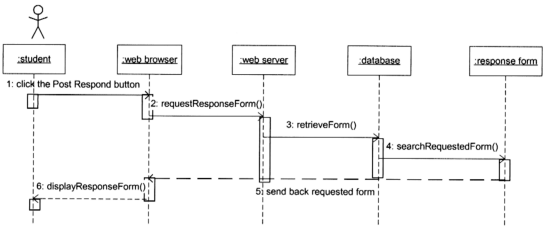

Figure 5-10 Initial Group Module's Compose Response Scenario

Read or View Summaries Scenario

Student clicks the link to read or view summaries

Browser requests for the required document from server

Server retrieves requested document from server database

Server database sends the requested document to browser

Browser displays the requested document to user

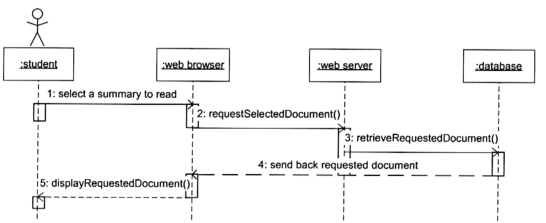

Figure 5-11 Expert Group Module's View Summaries Scenario

Integrate All Reports Scenario

Student clicks the Create Integrated Report button

Browser requests for report integration from server

Server activates web agent (integration agent)

Web agent integrates all related reports in server database

Server database sends integrated report to browser

Browser displays the integrated report to student

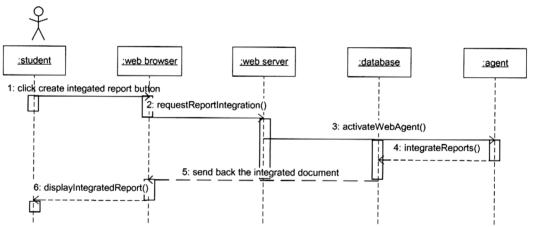

Figure 5.12 Jigsaw Group Module's Integrate All Reports Scenario

5.2.2.2 G-Jigsaw's Activity Diagrams

An Activity Diagram shows the flow from an activity to an activity within a system. It shows a set of activities, the sequential or branching flow from an activity to an activity and objects that act and acted upon (Booch et. al., 1999). The G-Jigsaw design model uses activity diagrams to illustrate the system dynamic view. This chapter only shows one activity diagram in figure 5-13 which emphasize on the control flow of students' participation in G-Jigsaw activities. The complete activity diagrams for each module are depicted in Appendix A-4.

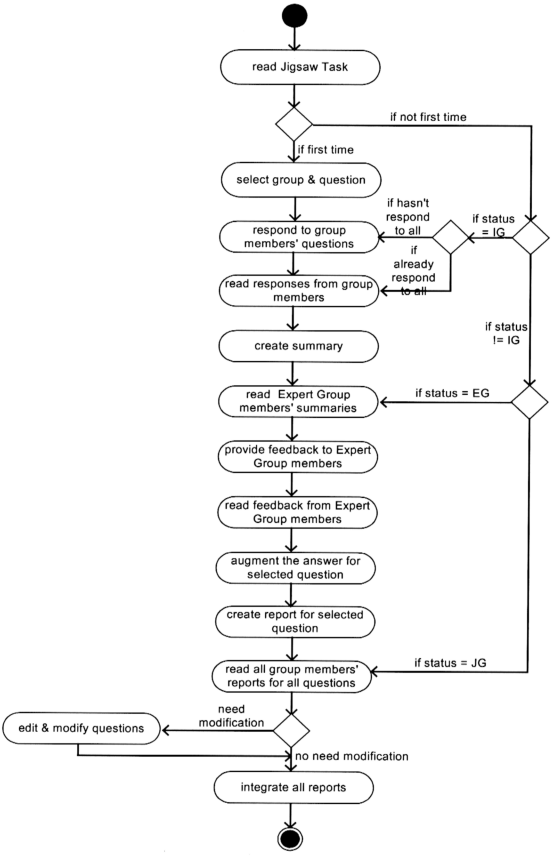

Figure 5-13 Activity Diagram Showing Students' Jigsaw Activity Process

5.2.3 G-Jigsaw Graphical User Interface Design

G-Jigsaw is implemented as one of WebCL modules to support the jigsaw-type collaborative learning. Therefore, Graphical User Interface (GUI) designs of G-Jigsaw are consistent with WebCL GUI. Figure 5-14 presents the common layout for WebCL GUI design for its homepage module.

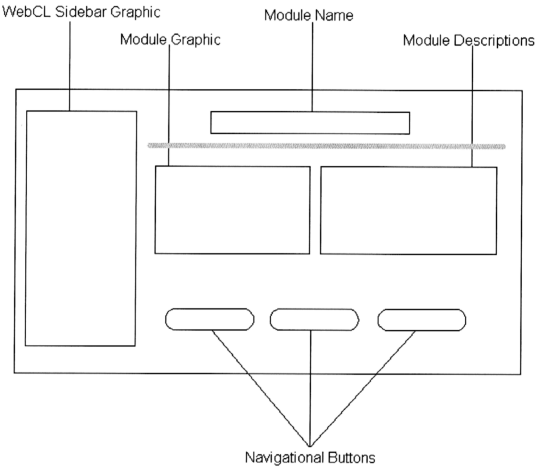

Figure 5-14 G-Jigsaw's Homepage GUI Design

Based on the GUI layout shown in figure 5-14, G-Jigsaw homepage generally consists of five major sections. The WebCL Sidebar Graphic section on the left side is the same for all WebCL modules. This GUI graphic is obtained directly from the WebCL. The G-Jigsaw name, graphic and descriptions sections make it differ from WebCL homepage module. The navigational buttons are placed at the bottom of the homepage. G-Jigsaw

homepage comprises of 3 navigational buttons that provides access to the WebCL homepage, Jigsaw Task and Completed Task. Figure 5-15 displays other G-Jigsaw pages layouts. These pages are also designed to be consistent with WebCL.

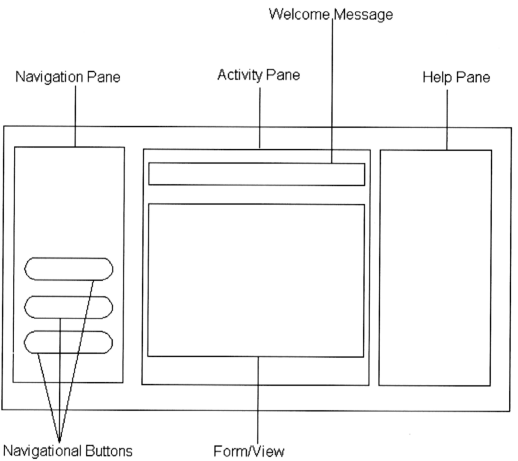

Figure 5-15 G-Jigsaw's Page GUI Design

As illustrated in figure 5-15, all G-Jigsaw pages consist of 3 panes namely Navigational Pane, Activity Pane and Help Pane. The Navigational Pane provides access to G-Jigsaw internal modules, which are Jigsaw Task and Completed Task. This pane is always placed on the left side of the page. The Activity Pane is located at the center. This is where different jigsaw activities are carried out. This pane always has a welcome message on top of it and it embeds forms and views that support the jigsaw activities. On the right hand side is the Help pane. This pane displays Quick Helps for user to

perform various tasks and operations. The G-Jigsaw's GUI design screenshots will be depicted in the next chapter.

5.3 Chapter Summary

This chapter has presented G-Jigsaw analysis and design. Both of the requirements and object-oriented analysis are identified in the first part of this chapter. G-Jigsaw architecture design, object-oriented design and GUI design are also discussed in this chapter. Most of the UML diagrams such as use case diagrams, interaction diagrams, activities diagrams and class diagrams are used in G-Jigsaw object-oriented analysis and design process. Chapter 6 will discuss the implementation of G-Jigsaw agents, its internal multi-agent architecture and the execution aspects of G-Jigsaw.