

Chapter 6 – G-Jigsaw Implementation and Execution

This chapter describes the implementation and execution of G-Jigsaw. The first part discusses the implementation of G-Jigsaw, which covers the implementation of agents, its internal multi-agent architecture, communication that exist within the architecture and G-Jigsaw security features. The second part presents some scenarios of G-Jigsaw execution that cover all four modules of G-Jigsaw named Jigsaw Task, Initial Group, Expert Group and Jigsaw Group.

6.1 The implementation of G-Jigsaw's Agents

Collaboration always involves the process of sharing ideas and opinions. All these data are normally recorded separately either as files or documents and are stored in various repositories. When a group of users need to integrate their collaborative deliverables, they need to retrieve data and recompile them into the final outcome (e.g. copying the required data, pasting and modify them as a new document and save it) manually. This will definitely affect the collaborative learning process due to the time spent in searching and copying the required data. In order to overcome this limitation, G-Jigsaw deploys an Integration Agent to automatically obtain the information required for the integration and performs it on behalf of the users. By implementing this agent, the integration process can be done with a single mouse click. It certainly will save a lot of users' time and makes the integration process simpler and less troublesome.

In addition, knowledge sharing is a key requirement for an effective collaborative learning environment. To create a shared repository, users are usually required to explicitly provide such valuable data into the system. However, this process normally will consume a lot of time and efforts. Therefore, not many users are willing to spend

the extra time for such data entries task. G-Jigsaw attempts to create a knowledge shared repository without consuming a lot of the users' time. To achieve this, a Sharing Agent is developed to perform tasks such as data extraction, filtering and categorizing transparently without the user's knowledge. This agent also enables users to retrieve tasks from the shared repository. Therefore, the task questions are reusable and time spent to set new questions are minimize.

Although the jigsaw technique greatly promotes students' collaborative learning activities, however, its complex activity flows and processes are major drawbacks. The challenge became more demanding because G-Jigsaw activities are carried out via a computer-supported system. As a result, a Navigation Agent is implemented to control the flows, rules and processes of jigsaw activities automatically. This will not only help to simplify the user's participation, moreover, it also impacts user's learning experience. It assists user's learning materials navigation effectively, keeps track of user's status and prompts users into effective action. With the deployment of this agent, users will understand the concepts of jigsaw much easier and realize the benefits of collaborative learning.

Three agents were implemented in G-Jigsaw, which are named Integration Agent, Sharing Agent and Navigation Agent to support the jigsaw activities and processes. However, the execution of these agents needs to be monitored in a systematic manner. From the review carried out in chapter 4, Olguin and his colleagues proposed an Agent-based Architecture for group study (Olguin, 2000). In this architecture, each user of the Collaboration Framework has an associated agency. This agency is responsible for receiving requests and returning responses from and to its owner. The architecture consists of 4 agencies that interact together to support the collaborative learning


environment. Among the four agencies is the Activity Agency that is responsible to monitor activities of tasks and generates more information such as student's participation and collaboration levels using the information collected by Group Agency. This information is very important to evaluate the effectiveness of a group. Olguin and his colleagues have not implemented this agency, but the concept of an Activity Agency that monitors the participation and collaboration in this architecture can be applied in G-Jigsaw. It can serve as the agent manager to coordinate and monitor the execution of the three agents mentioned. Hence, the Activity Agency is used to manage agents of G-Jigsaw to achieve the following major objectives:

1. To speed up and simplify the integrating process in a group.
2. To simplify the process of jigsaw activities by automating the complex activity flows and processes of G-Jigsaw to ensure that students focus only on their learning process not the jigsaw process.
3. To perform back-end processing such as filtering, categorizing and extracting information transparently without the users' knowledge

In order for the agents to interact with each other to carry out specific tasks that support the jigsaw activity processes, a multi-agent architecture is required. Section 6.2 will describe further about the internal multi-agent architecture.

6.2 The Implementation of G-Jigsaw's Internal Multi-Agent Architecture

According to Fenton-Kerr and his colleagues (1998), although multiple agent tasks could be programmed into a single entity, but such multi-faceted agent would be difficult to construct and maintain. Moreover, due to its complexity, it would likely



affect the response time, especially if it is implemented across a busy network. To prevent this from occurring, an alternative approach is to implement a multi-agent system as described by (Bigus and Bugus, 1998) where each modular agent could almost carry out a specific task automatically. In this approach, a coordinating agent is responsible to manage various agents in its care.

Thus, this multi-agent approach is implemented in G-Jigsaw as its internal agent architecture. This architecture adopts the multi-agent architectural scheme system proposed by (Fenton-Kerr, et. al., 1998) as discussed in chapter 3, section 3.5.2. In G-Jigsaw, the Activity Agency fulfills the role of a coordinating agent. Fundamentally, Activity Agency monitors 3 web agents called Sharing Agent, Integration Agent and Navigation Agent.

Sharing Agent is responsible to extract, filter, categorize accordingly and store teachers' submitted jigsaw task questions in a shared repository. This is carried out to avoid redundancy questions in the shared repository. Other teachers will be able to retrieve these questions via the Sharing Agent. The Integration Agent enables students to integrate their group reports automatically. Upon activation, this agent will search the student group's information from the group profile. Based on the information gained, it will then search for the related group's reports and integrates all the individual report as one final report.

Navigation Agent is accountable to automate the complex flows and processes of jigsaw activity. Therefore, it will play the role of the tracker agent proposed by Fenton-Kerr, et. al.. This agent will keep track of every login user's profile. Based on the user's current

and the profile is updated accordingly. Figure 6-1 depicts the internal multi-agent architecture of G-Jigsaw which is implemented in the agent-tier of G-Jigsaw architecture as shown in figure 5-7 in chapter 5.

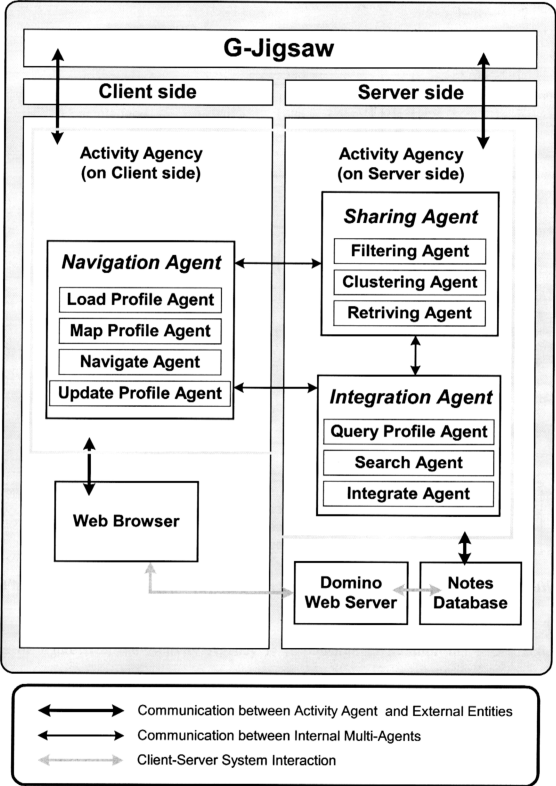


Figure 6-1 G-Jigsaw Internal Agent Architecture

Figure 6-2 further illustrates how the web agents work together in automating and simplifying the jigsaw session.

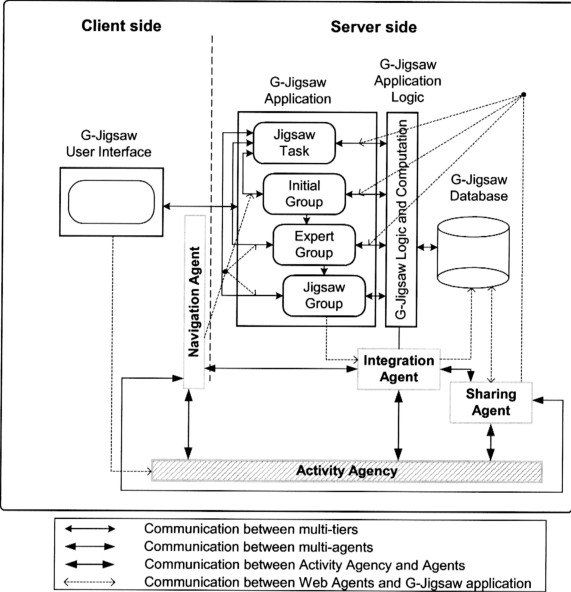


Figure 6-2 The Multi-agent Internal Architecture's Communication in G-Jigsaw

As of figure 6-2, the Activity Agency is responsible to monitor the user's action throughout the jigsaw session and identified the user's requests. Then, it activates the appropriate web agents and sends messages to the agents in carrying out the user's requests respectively. Based on the messages received, the web agents activate the internal agents under their supervision in turns to perform a more specific task. The

subsequent section discusses these communications, which includes the external communication of Activity Agent with its external entities (e.g. web server, web browser and applications) as well as its internal multi-agents communication in detail.

6.2.1 G-Jigsaw’s Multi-Agent Communication

As shown in figure 6-1 and figure 6-2, the architecture of G-Jigsaw internal agent serves as an additional layer for G-Jigsaw client-server architecture. Besides the common client-server interactions that occur between the client (browser), server and databases, this internal agent architecture enables agents to communicate internally with other agents and externally with other external entities.

Activity Agency communicates externally with the clients (browsers), server and application (G-Jigsaw). It receives requests from users via the web browser and triggers the appropriate agents to perform the user’s requests. Therefore, it is responsible for the execution of internal agents. Activity Agency acts as an agent manager that forward information from one agent to another agent, from user to agents and from agents to user. Table 6-1 depicts all the possible user’s requests throughout the entire jigsaw session as well as the messages and actions performed by the Activity Agency upon the user’s requests.

Table 6-1 Activity Agency’s Reactions Towards User’s Requests

User’s Requests (User’s Actions)	Messages Sent	Actions Performed
Retrieve jigsaw questions	1	Activate Sharing Agent
Submit jigsaw task	2	Activate Sharing Agent
Start answering new task	3	Activate Navigation Agent
Submit a profile	4	Activate Navigation Agent
Resume previous task session	5	Activate Navigation Agent
Compose response	6	Activate Navigation Agent
Submit Response	7	Activate Navigation Agent
Read Response	8	Activate Navigation Agent
Compose Summary	9	Activate Navigation Agent
Retrieve Responses	10	Activate Sharing Agent
Submit Summary	11	Activate Navigation Agent

Read Summary	12	Activate Navigation Agent
Compose Report	13	Activate Navigation Agent
Retrieve previous summary	14	Activate Sharing Agent
Submit Report	15	Activate Navigation Agent
Read Report	16	Activate Navigation Agent
Integrate Group Reports	17	Activate Integration Agent
Read Integrated Report	18	Activate Navigation Agent

There are 3 web agents modular included in the Activity Agency. Each modular is responsible to invoke, monitor and control a set of more specific agents that will perform specific tasks. In a multi-agent structure, the communication within each agent is very important to ensure the success of the system. This can be achieved through the internal agent communication. Table 6-2 shows the communication of Sharing Agent with Activity Agency as well as its internal agents.

Table 6-2 Sharing Agent's Communication

Messages Received	Messages Sent	Actions Performed
1, 10, 14	1 – Retrieve Document [Username, task_id]	Activate Retrieving Agent
19	2 – Retrieving task completed	Report task completion to Activity Agency
2	3 – New Task Submission	Activate Filtering Agent
20	4 – Cluster Questions	Activate Clustering Agent
21	5 – Submission process completed	Report task completion to Activity Agency
Note: 19 = Message from Retrieving Agent indicating the completion of retrieving task 20 = Message from Filtering Agent indicating the completion of filtering task 21 = Message from Clustering Agent indicating the completion of clustering task For messages 1, 2, 10 and 14, refer to table 6-1		

Based on table 6-2, the sharing agent will perform its tasks accordingly. Table 6-3 depicts the pseudo-codes for the agents' algorithms to describe the mechanisms on how each agents work and communicate with one another. The algorithms are attached in Appendix D.

Table 6-3 Pseudo-codes for Sharing Agent's Algorithm

<p>For Sharing Agent:</p> <p>When activated by Activity Agency, Check the message received If message = 1 or message = 10 or message = 14 Then Activate Retrieving Agent [message, type, username, task id] Else If message = 2 Then Activate Filtering Agent [message] Else If message = 20 Then Activate Clustering Agent [message] Else If message = 19 Then Send message 2 to Activity Agency Else If message = 21 Then Send message 5 to Activity Agency End If End If Sharing Agent Quit</p>
<p>For Retrieving Agent:</p> <p>When activated by Sharing Agent, Check the message received Based on the type received Use the arguments (username, task id) as keys Search through the shared repository for documents that math the keys Retrieving the related documents accordingly Report the task completion message 19 to Sharing Agent Retrieving Agent Quit</p>
<p>For Filtering Agent:</p> <p>When activated by Sharing Agent, Check the message received Extract submitted new questions into an array For each question in the array Loop through the share repository's questions If question already exists Then Delete question in array Else Continue End If End Loop Increase array index by 1 End For Report the task completion message 20 to Sharing Agent Filtering Agent Quit</p>

For Clustering Agent:

When activated by Sharing Agent,

- Check the message received

- Extract task title from submission task and make it as new category

- Loop through the share repository's categories

 - If category already exists Then

 - Save the filtered questions in the array in this category

 - Sort the question collection in ascending order

 - Else

 - Create a new category

 - Save the filtered questions in the array in this category

 - Sort the question collection in ascending order

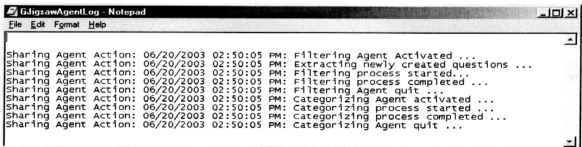
 - End If

- End Loop

- Report the task completion message 21 to Sharing Agent

Clustering Agent Quit

Interactions among teachers, Activity Agency and Sharing Agent are best described using a scenario where a teacher is creating a new jigsaw task for student's participation. When a teacher submits a jigsaw task, the Activity Agency will invoke the Sharing Agent to call the Filtering Agent. The Filtering Agent will extract the submitted questions and filter the existing questions in the shared repository. Upon completion, the Filtering Agent will report back to the Sharing Agent. The Sharing Agent then invokes the Categorizing Agent to categorize and sort the questions into respective categories. Then, it reports back to the Sharing Agent. All these tasks are performed transparently without the teacher's knowledge. The teacher will only be notified that his/her submission is successfully entered in the shared repository. Figure 6-3 shows the execution log of Sharing Agent. Similarly, when a teacher intends to retrieve questions from the shared repository, the Activity Agency will notify the Sharing Agent to activate the Retrieving Agent in order to perform the retrieving task.



```
GJigsawAgentLog - Notepad
File Edit Format Help

Sharing Agent Action: 06/20/2003 02:50:05 PM: Filtering Agent Activated ...
Sharing Agent Action: 06/20/2003 02:50:05 PM: Extracting newly created questions ...
Sharing Agent Action: 06/20/2003 02:50:05 PM: Filtering process started...
Sharing Agent Action: 06/20/2003 02:50:05 PM: Filtering process completed ...
Sharing Agent Action: 06/20/2003 02:50:05 PM: Filtering Agent quit ...
Sharing Agent Action: 06/20/2003 02:50:05 PM: Categorizing Agent activated ...
Sharing Agent Action: 06/20/2003 02:50:05 PM: Categorizing process started ...
Sharing Agent Action: 06/20/2003 02:50:05 PM: Categorizing process completed ...
Sharing Agent Action: 06/20/2003 02:50:05 PM: Categorizing Agent quit ...
```

Figure 6-3 The Sharing Agent Execution

Table 6-4 shows the communication of Navigation Agent with Activity Agency as well as its internal agents. Table 6-5 lists the locations returned by the Map Profile agent based on the current user’s level and task status in the user profile. The location returned is then passed to Navigate agent in order to automate the entire jigsaw session.

Table 6-4 Navigation Agent’s Communication

Messages Received	Messages Sent	Actions Performed
5, 25	1 – Load user’s profile	Activate Load Profile Agent
22	2 – Map profile values [username, level, status]	Activate Map Profile Agent
23	3 – Navigate to current location [location]	Activate Navigate Agent
24	4 – Navigation completed	none
3, 4, 6, 7, 8, 9, 11, 12, 13, 15, 16, 18	5 – Update user’s profile [username]	Activate Update Profile Agent
Note: 22 = Message from Load Profile Agent indicating the completion of loading task 23 = Message from Map Profile Agent indicating the completion of mapping task 24 = Message from Navigate Agent indicating the completion of navigation task 25 = Message from Update Profile Agent indicating the completion of updating task For messages 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 15, 16 and 18, refer to table 6-1		

Table 6-5 Map Profile Agent’s mapping table

Level	Status	Location Returned
0 – Jigsaw Task module	0 – new task	Jigsaw Task
0	1 – create profile	Profile Form
1 – Initial Group module	2 – view questions	Questions View
1	3 – create response	Response Form
1	4 – read responses	Responses View
1	5 – continue to create summary	Responses View
1	6 – create summary	Summary Form
2 – Expert Group module	7 – read summaries	Summaries View
2	8 – continue to create report	Summaries View
2	9 – create report	Report Form
3 – Jigsaw Group module	10 – read reports	Report View
3	11 – integrate report	Report View
3	12 – read integrated report	Integrated Report View

Table 6-6 depicts the pseudo-codes for the agents’ algorithms in Navigation to describe the mechanisms on how each agents work and communicate with one another. The algorithms are attached in Appendix D.

Table 6-6 Pseudo-codes for Navigation Agent's Algorithm

<p>For Navigation Agent:</p> <p>When activated by Activity Agency, Check the message received If message = 5 Or 25 Then Activate Load Profile Agent Else If message = 22 Then Activate Map Profile Agent [username, level, status] Else If message = 23 Then Activate Navigate Agent [location] Else If message = 3 Or message = 4 Or message = 6 Or message = 7 Or message = 8 Or message = 9 Or message = 11 Or message = 12 Or message = 13 Or message = 15 Or message = 16 Or message = 18 Then Activate Update Profile Agent [username] Else If message = 24 Then Send message 4 to Activity Agency End If Navigation Agent Quit</p>
<p>For Load Profile Agent:</p> <p>When activated by Navigation Agent, Check the message received Extract the current user's username as a key Search through the database for user profile document that matches the key Load the values from the profile document to the memory Report the task completion message 22 to Navigation Agent Load Profile Agent Quit</p>
<p>For Map Profile Agent:</p> <p>When activated by Navigation Agent, Check the message received Based on the user's current level and status received Map them to the mapping table Return mapped location Report the task completion message 23 to Navigation Agent Map Profile Agent Quit</p>
<p>For Navigate Agent:</p> <p>When activated by Navigation Agent, Check the message received Based on the location received Redirect the user to the location accordingly Display the instructions and enables the functions accordingly Report the task completion message 24 to Navigation Agent Navigate Agent Quit</p>

For Update Profile Agent:

```
When activated by Navigation Agent,
  Check the message received
  Extract the current user's username as a key
  Loop through the database for user profile document that match the key
    If profile document exists Then
      Update the status in profile document
      Exit Loop
    Else
      Create a profile document for the user
      Insert and save the values in the profile document
    End If
  End Loop
  Report the task completion message 25 to Navigation Agent
Update Profile agent quit
```

The second scenario illustrates interaction between the students, the Activity Agency and the Navigation Agent. When a student joint a jigsaw session, the Activity Agency will activates the Navigation Agent. First, the Navigation Agent checks the student's profile based on the login information. If there is no profile available, a new profile will be generated otherwise, the Navigation Agent will load the student's profile via the Load Profile Agent. Based on this information, the Navigation Agent calls the Map Profile Agent to perform the mapping process. The Map Profile Agent will map the student's current level and status with its mapping table as shown in table 6-5. Then, it will return the mapped location and report back to the Navigation Agent. This immediately followed by the activation of Navigate Agent to position the student in the proper stage to continue with his/her jigsaw session.

Each time a student completed an activity (e.g. post a response, compose a summary or create a report), the Activity Agent will notify the Navigation Agent. Based on the types of activity, the Navigation Agent will update the student's profile through the Update Profile Agent. The Update Profile Agent is responsible to update the student's profile status. Once the profile is updated, the Navigation Agent will redirect the student to the proper stage to continue with his/her jigsaw session through its Load Profile Agent,

Map Profile Agent and Navigate Agent. Figure 6-4 shows the execution log of the Navigation Agent.

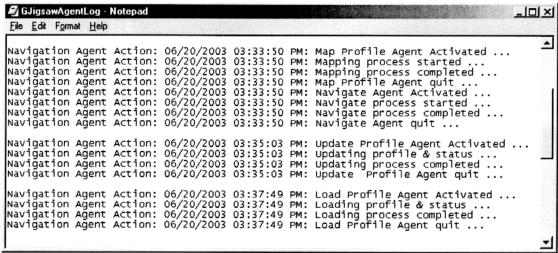


Figure 6-4 The Navigation Agent Execution

Table 6-7 shows the communication of Integration Agent with Activity Agency as well as its internal agents.

Table 6-7 Integration Agent’s Communication

Messages Received	Messages Sent	Actions Performed
17	1 – Get profile values [username]	Activate Query Profile Agent
26	2 – Create integrated report form	Activate Integrate Agent
27	3 – Search and retrieve documents	Suspend Integrate Agent, Activate Search Agent
28	4 – Integrate group reports	Resume Integrate Agent
29	5 – Integration completed	Report task completion to Activity Agency
Note: 26 = Message from Query Profile Agent indicating the completion of querying task 27 = Message from Integrate Agent indicating the completion of form creation 28 = Message from Search Agent indicating the completion of searching task 29 = Message from Integrate Agent indication the completion of integrating task For message 17, refer to table 6-1		

Table 6-8 presents the pseudo-code for the agents’ algorithms to show the mechanisms on how the agents work and communicate with each another in Integration Agent. . The algorithms are attached in Appendix D.

Table 6-8 Pseudo-codes for Integration Agent's Algorithm

For Integration Agent:

```

When activated by Activity Agency,
  Check the message received
  If message = 17 Then
    Activate Query Profile Agent [username]
  Else If message = 26 Then
    Activate Integrate Agent [group, document]
  Else If message = 27 Then
    Suspend Integrate Agent
    Activate Search Agent [group, task id]
  Else If message = 28 Then
    Resume Integrate Agent
  Else If message = 29 Then
    Send message 5 to Activity Agency
  End If
Integrating Agent Quit
  
```

For Query Profile Agent:

```

When activated by Integration Agent,
  Check the message received
  Extract username as a key
  Loop through the user profile documents
    If profile document exists Then
      Get the values of task id and user's group name
      Exit Loop
    End If
  End Loop
  Return the values to Integration Agent
  Report the task completion message 26 to Integration Agent
Query Profile Agent Quit
  
```

For Integrate Agent:

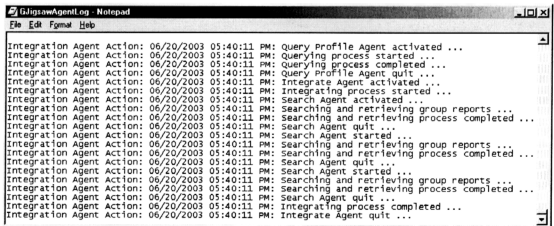
```

When activated by Integration Agent,
  Check message received
  If message = 2 Then
    Create a new Integrated Report Form
    Extract the report's task title value and insert to the task title field
    Insert the group name field with the provided value
    Report form created message 27 to Integration Agent
    Integrate Agent Wait
  Else If message = 4 Then
    Loop through the document collection
      Extract the task question, report content field values from the document collection
      Insert the values to the Integrated Report form
    End loop
    Save the integrated report document in repository
  End If
  Report the task completion message 29 to Integration Agent
Integrate Agent Quit
  
```

For Search Agent:

```
When activated by Integration Agent,  
  Using the arguments (group, task id) passed by Integration Agent as keys  
  Loop through the report documents  
    If document exists then  
      Assign the document in a document collection  
    Else  
      Continue  
    End If  
  End Loop  
  Return the document collection to Integration Agent  
  Report the task completion message 28 to Integration Agent  
Search Agent Quit
```

When all the group members of a group completed their report in the Jigsaw Stage, the group leader is required to integrate all the individual reports. This integration process will be carried out by the Activity Agency through the Integration Agent that will coordinate the entire integration process as described in table 6-8. First, The Integration Agent activates the Query Profile Agent to obtain the group's information. Then, it will invoke the Integration Agent to create a new document to place the integrated report. Once the document is created, Search Agent is called. The Search Agent will use information from the Query Profile Agent to repeatedly search and retrieve the correct group reports. Upon completion, the Integration Agent will integrate all these reports into an integrated report. Figure 6-5 shows the execution log of the Integration Agent.



```
JigsawAgentLog - Notepad  
File Edit Format Help  
Integration Agent Action: 06/20/2003 05:40:11 PM: Query Profile Agent activated ...  
Integration Agent Action: 06/20/2003 05:40:11 PM: Querying process started ...  
Integration Agent Action: 06/20/2003 05:40:11 PM: Querying process completed ...  
Integration Agent Action: 06/20/2003 05:40:11 PM: Query Profile Agent quit ...  
Integration Agent Action: 06/20/2003 05:40:11 PM: Integrate Agent activated ...  
Integration Agent Action: 06/20/2003 05:40:11 PM: Integrating process started ...  
Integration Agent Action: 06/20/2003 05:40:11 PM: Search Agent activated ...  
Integration Agent Action: 06/20/2003 05:40:11 PM: Searching and retrieving group reports ...  
Integration Agent Action: 06/20/2003 05:40:11 PM: Searching and retrieving process completed ...  
Integration Agent Action: 06/20/2003 05:40:11 PM: Search Agent quit ...  
Integration Agent Action: 06/20/2003 05:40:11 PM: Search Agent started ...  
Integration Agent Action: 06/20/2003 05:40:11 PM: Searching and retrieving group reports ...  
Integration Agent Action: 06/20/2003 05:40:11 PM: Searching and retrieving process completed ...  
Integration Agent Action: 06/20/2003 05:40:11 PM: Search Agent quit ...  
Integration Agent Action: 06/20/2003 05:40:11 PM: Search Agent started ...  
Integration Agent Action: 06/20/2003 05:40:11 PM: Searching and retrieving group reports ...  
Integration Agent Action: 06/20/2003 05:40:11 PM: Searching and retrieving process completed ...  
Integration Agent Action: 06/20/2003 05:40:11 PM: Search Agent quit ...  
Integration Agent Action: 06/20/2003 05:40:11 PM: Integrating process completed ...  
Integration Agent Action: 06/20/2003 05:40:11 PM: Integrate Agent quit ...
```

Figure 6-5 The Integration Agent Execution

6.3 The Implementation of G-Jigsaw Security Aspects

Generally, the entire process of G-Jigsaw is supported by Activity Agency through its multi agents. Besides the execution of agents, the security aspects are also very important to ensure the success of G-Jigsaw. Since G-Jigsaw is implemented using Lotus Notes, therefore it will inherit the Notes database security features, which includes Manager, Designer, Editor, Author, Reader, Depositor and No Access.

Lotus Notes Domino server will handle the security of the G-Jigsaw database efficiently if the access control is assigned properly. Therefore, G-Jigsaw has 4 different user roles for different group of user. Users with Administrator role are assigned with Manager Access. This group of users is responsible to administer and maintain the database of G-Jigsaw. Users with the role of Teacher are assigned with the Designer Access. They are able to create new tasks, delete existing task in the database or modify the Lotus Notes formula of G-Jigsaw.

The third group of users is the Students, which are assigned with Editor Access. They are able to create new documents such as summary and reports and delete only their own documents. However, they are not allowed to modify the Lotus Notes formula of G-Jigsaw. The default access control of G-Jigsaw is set to No Access, which prevents the public users from accessing G-Jigsaw.

6.4 Scenarios of G-Jigsaw Execution

Group Jigsaw aims to enhance student's skills and capabilities in contributing valuable ideas and opinions collaboratively in groups. It also supports an integration feature for students to produce a complete group outcome. These can be achieved through modules

of G-Jigsaw that are named Jigsaw Task, Initial Group, Expert Group and Jigsaw Group. Each of these modules supports different roles of users in different ways. Figure 6-6 presents the homepage of G-Jigsaw.



Figure 6-6 G-Jigsaw Homepage

6.4.1 Jigsaw Task Module

The Jigsaw Task module is meant for teachers to create jigsaw task questions for the students’ participation. When a teacher creates a new jigsaw task, he/she is provided with 4 different question templates. Figure 6-7 depicts the preview of Template 1.

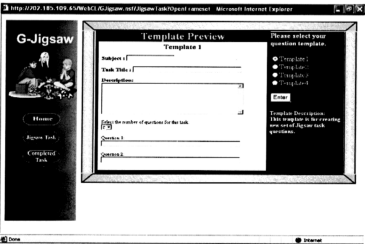


Figure 6-7 G-Jigsaw’s Question Templates

Template 1 enables teachers to create a new set of jigsaw task questions and in Template 2 teachers are able to retrieve existing questions from the shared repository as their new jigsaw task questions without any modifications. However, in Template 3 and Template 4 teachers are allowed to modify the retrieved questions. Template 3 is used for short questions while Template 4 is more suitable for long questions.

Once a template is selected, a form is displayed for the teacher to enter the jigsaw task questions. Figure 6-8 presents a sample of a science subject jigsaw task entitled “The Food Chain” for standard 5 students.

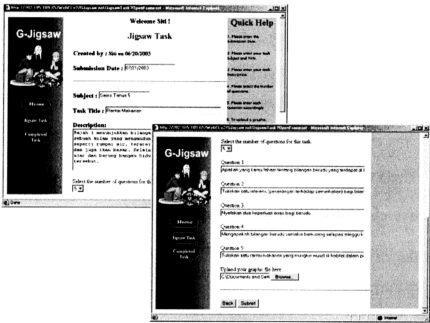


Figure 6-8 Creating a Jigsaw Task

As of figure 6-8, a teacher is required to set submission date, subject, task title and description of the jigsaw task. Then, the teacher must enter the number of questions he/she requires, enter the details for each question and submit the jigsaw task questions. Upon completion, the new jigsaw task will be displayed in the Jigsaw Task view as shown in figure 6-9. Here, students are able to read and start to participate in the new jigsaw task.

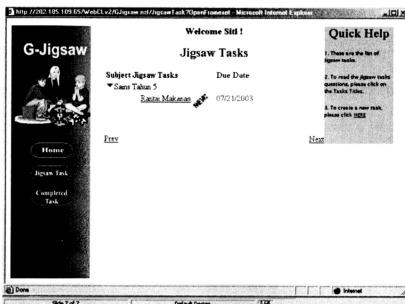


Figure 6-9 Jigsaw Tasks View

6.4.2 Initial Group Module

When a student starts to participate in a new jigsaw session, first he/she is required to select his/her group then select his/her responsible question assigned by the teacher. These selections will generate the student's profile and the student is now in the Initial Group, which is the first level of the jigsaw collaboration. In Initial Group, the student is required to give responses to his/her group members' responsible questions. Figure 6-10 illustrates an example of questions that the student needs to respond.

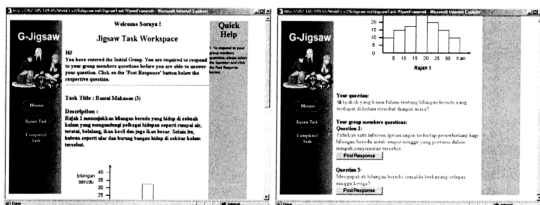


Figure 6-10 Questions to be responded in Initial Group

Figure 6-11 below shows a student response towards his/her member's question. G-Jigsaw keep tracks the number of questions that have been responded and how many are left. Once the student has responded to all his/her group member's questions, he/she can now read their responses towards his/her question and then, create a summary (answer). This summary will be used during the discussion in the Expert Group.

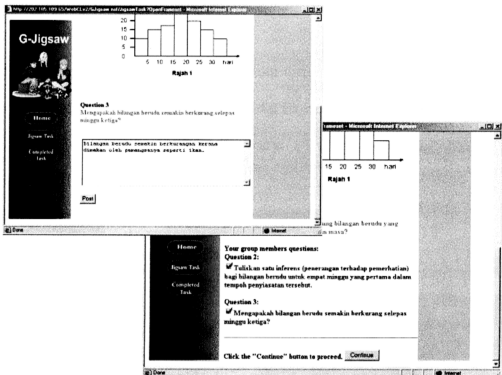


Figure 6-11 Giving Responses in Initial Group

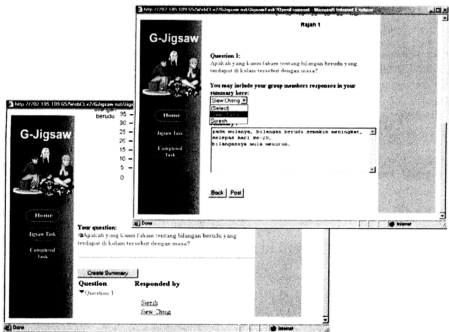


Figure 6-12 The Summary View and Form

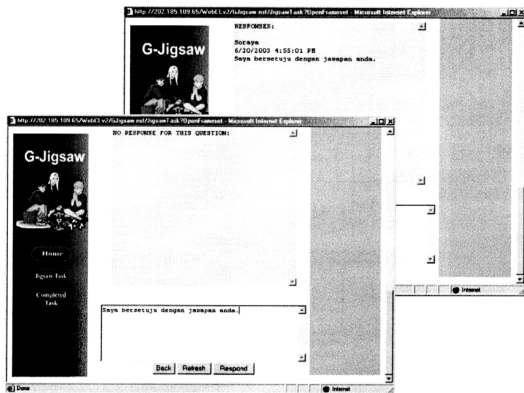


Figure 6-14 Providing comments and feedback to the Summary

Based on these comments, the students need to compose a new improved report. Figure 6-15 displays the Report View with the create report feature. Figure 6-16 shows the Report Form where students are able to retrieve and modify their previous summary as part of their report. This is the last step of Expert Group collaboration and the students will now proceed to the Jigsaw Group.

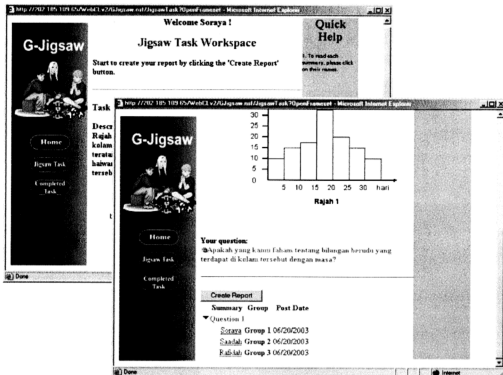


Figure 6-15 The Report View

G-Jigsaw

Home

Jigsaw Task

Completed Task

Soraya

6/20/2003 4:47:08 PM

Saya berasa jawapan ini adalah lengkap.
Bentuk graf adalah berbentuk "Bukit".

Rafidah

6/20/2003 4:42:46 PM

Saya sependapat dengan anda.

Report:

Pada peringkat awal, bilangan berudu semakin meningkat, keadaan ini berterusan selama 20 hari. Selepas hari yang ke-20, bilangan berudu semakin berkurangan.

Copy

Back Post

Figure 6-16 The Report Form

6.4.4 Jigsaw Group Module

Students will be grouped back to their original group to form the Jigsaw Group. In Jigsaw Group students are able to view their group member's report as shown in figure 6-17.

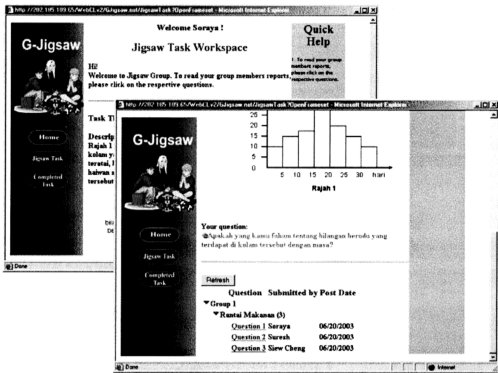


Figure 6-17 Jigsaw Group Report View

By reading their group member's report, they are equipped with all the answers for the jigsaw task questions and not only their responsible question. The group leader is responsible to integrate his/her group members' reports by clicking the Create Integrated Report button as depicted in figure 6-18. The integration process is automatically managed by G-Jigsaw. Students' integrated reports can be viewed in the Completed Task view as shown in figure 6-19.

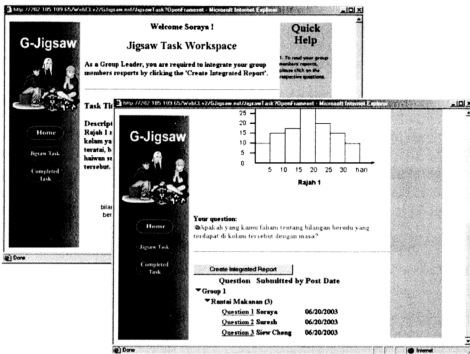


Figure 6-18 Integrate Report View

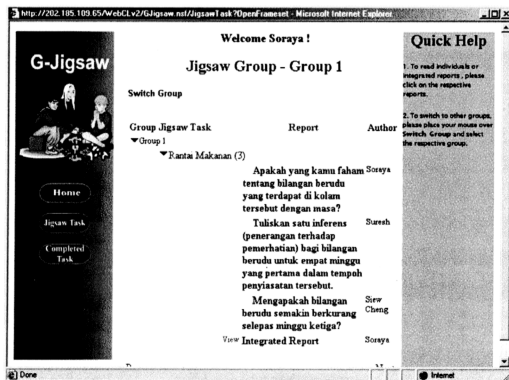


Figure 6-19 Completed Task View

6.5 Chapter Summary

This chapter has presented the implementation and execution of G-Jigsaw. The web agents that are essential for G-Jigsaw have been identified and the implementation of Activity Agency that coordinates these agents in a multi-agent architecture was discussed. In addition, the communication and the interaction of these agents are also described. Furthermore, this chapter presented the security features of G-Jigsaw that are inherited from the Lotus Notes and the scenarios of G-Jigsaw execution.