CHAPTER 3: Groupware Support for a Requirements Analysis Model

3.1 Introduction

This chapter introduces the processes and features of GRAT. It discusses how Ian Sommerville’s requirements analysis model, discussed in the previous chapter, is supported by the architecture of GRAT.

3.2 GRAT Architecture

This section outlines groupware architecture to support the requirements analysis model described earlier. It exemplifies a conceptual design found in a generic requirements analysis process as proposed by Ian Sommerville (1996). This architecture was developed using Lotus Notes. Figure 3.1 shows the requirements analysis process and Table 3.1 describes the phases of the requirements analysis process as defined by the Ian Sommerville (1996). Later, the architecture of GRAT is discussed based on the different phases identified.
Requirements Analysis Process

Figure 3.1: Requirements Analysis Process (Sommerville, 1996)

Table 3.1: Phases in Requirements Analysis.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Understanding</td>
<td>Analyst must develop their understanding of the application domain. Therefore if a system for a supermarket is required, the analyst must find out as much as possible about supermarkets</td>
</tr>
<tr>
<td>Requirements Collection</td>
<td>This is a process of interacting with stakeholders in the system to discover their requirements. Obviously domain understanding also develops during this activity.</td>
</tr>
<tr>
<td>Classification</td>
<td>This activity takes the unstructured collection of requirements and organizes them into coherent clusters</td>
</tr>
<tr>
<td>Conflict Resolution</td>
<td>Inevitably, where multiple stakeholders are involved, requirements conflict. This activity is concerned with finding and resolving these conflicts.</td>
</tr>
<tr>
<td>Prioritization</td>
<td>In any set of requirements, some are more important than others. This stage involves interaction with stakeholders to discover the most important requirements</td>
</tr>
<tr>
<td>Requirements Validation</td>
<td>The identified requirements are checked to discover if they are complete, consistent and in accordance with what stakeholders really want from the system.</td>
</tr>
</tbody>
</table>
The phases mentioned in Table 3.1 are further described in terms of functionality and architecture in the following sections.

### 3.2.1 Domain Understanding

When analysts have some information to be shared with other team members, they submit the forms via web. They can submit the forms text basis, attach a file, or create a hyperlink. Once they have submitted the form, other users can look at the information in the View. They can choose to read the information by double clicking on the view or choose to add more information regarding the domain. This is shown in Figure 3.2.

![Diagram of Domain Understanding in GRAT]

**Figure 3.2: Domain Understanding in GRAT**

### 3.2.2 Requirements Collection

Rapid generation of a single pool of ideas by individual group members with evaluation and criticism is being discouraged at this stage (Macaulay, 1997). Therefore the interface for brainstorming needs to permit the members of a distributed team with a means of inputting a suggested object at their terminal and sending it to update a common area which is seen by all members as depict in Figure 3.3.
Figure 3.3: Requirements Collection in GRAT

Team members post the requirements that are feasible for the domain. All the team members are able to see the whole set of requirements submitted to the server. The list of posted requirement refreshes every 5 seconds or whenever a new requirement is posted. There will not be any communication facilities provided between team members.

3.2.3 Classification

Classify with lists according to functional, nonfunctional, environment and also design constrains; and also according to partitions defined by domain models and development paradigm (Christel, 1992). For this phase, it is divided into two sub-processes. One is to collect the list of categories and the other to classify the requirements into appropriate categories.

3.2.3.1 Categories Collection

While looking and browsing into the list of requirements, participants can start posting the appropriate categories they think suits for the requirements. The list of
categories refreshes every 15 seconds or when the participant proposes a new category.

This is pictured in Figure 3.4.

![Figure 3.4: Categories Collection in GRAT](image)

### 3.2.3.2 Classification

Participants have to match each requirement with appropriate category. The list of categories is provided in the list box. The System checks if all the requirements have been matched with an appropriate Category. Only then participants are allowed to preview their classification based on the categories. If they are not satisfied they can re-categorize the requirement. If they are satisfied, then they can submit to the server for further actions. Figure 3.5 shows the classification process in GRAT system.

![Figure 3.5: Classification in GRAT](image)
After users have submitted their respective classification result, the result are processed and calculated. The calculation produces a summary of all the requirements based on the respective category. Figure 3.6 shows how GRAT handles the situation.

![Figure 3.6: Computation of Classification Results in GRAT](image)

### 3.2.4 Conflict Resolution

Perform abstraction to answer “Why do you need X?”. This in effect moves statements of “how” to statements of “what” (Christel, 1992). Capture rationale to support future requirement evolution. The rationale behind the information collected in previous stages should be examined to determine whether the true requirements are hidden instead being explicitly expressed as depicted in Figure 3.7. After the classification result has been submitted to the server, project team members are able to see the requirements sorted out according to the categories. From here, if the participants feel there is a problem with the requirements, they can double click the row of the requirements and get to see the form. If they choose to post a question, then just by clicking an option, the web page directs it to another form where they can post their conflicts. Other members can view the conflicts just by double clicking them. They can...
respond to the conflict by simply clicking a button. The conflicts and the responds to the conflicts are arranged in a hierarchy form.

![Diagram of conflict resolution process in GRAT]

**Figure 3.7: Conflict Resolution in GRAT**

### 3.2.5 Prioritization

Here is where the criticality of the requirement is determined. The requirements are prioritize based on importance, cost and dependency (Christel, 1992). Participants have to prioritize each requirement in their respective categories. The values of prioritize is numbered. The range depends on the number of requirement in each category. For example, if a category contains 6 requirements, then the range would be 1 to 6. The values are provided in list box. 1 represents most prioritize and 6 represents the least prioritize. The System checks if all the requirements in a category have a unique value. Only then participants are allowed to preview their prioritization. The requirements are arranged according to the priority given by participant. If they are not
CHAPTER 3: Groupware Support for a Requirements Analysis Model

satisfied they can re-prioritize the requirement. If they are satisfied, then they can submit to the server. The process is clearly shown in Figure 3.8.

![Diagram showing prioritization process]

**Figure 3.8: Prioritization in GRAT**

After users have submitted their respective classification result, the result are processed and calculated. The calculation produces a summary of all the requirements based on the respective category as depict in Figure 3.9.

![Diagram showing prioritization process]

**Figure 3.9: Prioritization in GRAT**

### 3.2.6 Requirements Validation

Consistency checking, validating the requirements that are in agreement with originally stated goals and obtain authorization or verification to move to the next step of development is the routine conducted in this phase (Christel, 1992). Participants who
feel that particular requirement have some problem, double clicks on the requirement, and it shall prompt user to the requirement validation form. The validating options are based from Ms Ow's (1998) book. Once the form has been submitted, the submitted result is displayed below the respective requirement as shown in Figure 3.10.

![Figure 3.10: Requirements Validation in GRAT](image)

3.3 The Role of Project Manager

In order for the requirements analysis to be conducted in a effective manner, there is a need for a person in the project team to initiates the project, orchestrate the activities, have a final say when there is a conflict arises and unable to be solved by the team members. This role is mostly played by the Project Manager. Similarly, in GRAT the Project Manager is needed to ensure that the processes and product of the different phases mentioned above is as closely coupled with the objectives of the project.

The role of a facilitator or Project Manager is undeniable. He/she is the core for the success of the project (Viller, 1991). The Shorter Oxford English Dictionary clearly defines that facilitate is “To render easier; to promote, help forward”. Therefore it is concerned with assisting the group members in performing their collective task as a group (Macaulay, 1997).
CHAPTER 3: Groupware Support for a Requirements Analysis Model

This research takes into account the role of the facilitator seriously as mentioned by Linda A. Macaulay (1997) in her PhD thesis. She mentioned that the function of the facilitator is very important at the initial stage. With the knowledge of the group process, the facilitator can utilize this position to improve group cohesion and for setting of group norms. However as the group process moves on to the middle stage the facilitator becomes less important where they are mere enabler and intervening when necessary. When the process reaches the final stages, the role becomes important. The precise role played by the facilitator at this stage depends upon the circumstances in which the group is breaking up. Meaning, whether or not the group has fully achieved its purpose (Douglas, 1970).

According to Marsh (1991) who introduced Quality Function Deployment, the facilitator has to be,

- Planner
  - Help team establish objectives.
  - Develop agenda.
  - Establish dates, times and places for meetings.
- Guide
  - Explain the process
  - Regulate the flow of the process
  - Monitor participation
- Cheerleader
- Coach
  - Develop the team.
  - Facilitate consensus decision-making.
- Arbitrator
CHAPTER 3: Groupware Support for a Requirements Analysis Model

- Helps settles disputes and conflicts- collaborative effort
- Keep the team spirit alive
- Ensure problems are being solved

For GRAT, the project manager will be the planner, the guide, the coach and the arbitrator.

3.4 Summary

This chapter has introduced the architecture to support GRAT. Each phase of the requirements analysis has been identified as modules of GRAT. The modules are discussed in detail on its architecture. The role of project manager for GRAT was also introduced in this chapter. The following chapters will discuss the analysis, design and implementation of GRAT.