CHAPTER 2: Review on Requirements Analysis and Groupware

2.1 Introduction

This chapter discusses various requirements analysis methods found in literature. The requirements analysis methods mentioned in this chapter are discussed based on its methodology and the tools that had been developed to support it. In addition, groupware functionalities and technologies are reviewed in order to shape the main features of GRAT. This chapter helped in finalizing the boundaries of GRAT.

2.2 Requirements Analysis

2.2.1 Goal Based Requirements Analysis

*Goal based requirements analysis method (GBRAM)* stresses the need to characterize, categorize, decompose and structure goals as requirements, but usually fail to offer strategies to identify goals, taking it for granted that the goals have already been documented (Antón, 1994; Dardenne, 1994; Yu, 1994). The method is useful in identifying, elaborating, refining and organizing goals for requirements specification.

There two important process in GBRAM, goal analysis and goal evolution. Goal analysis regards to the thorough studying on the documentation for organizing and classifying goals. Goal evolution meanwhile looks into goal changing from the moment it is first identified right up to the moment they are put in operation or implemented in a system specification. Further elaboration on the process,
• **Goal Analysis.** Goals are identified from process descriptions by searching for statements, which seem to guide design decisions at various levels within a system or organization. They are also sourced from various types of gathered information such as flow charts or Entity Relationship (ER) diagrams. However, process descriptions are insufficient for achieving thoroughness and completeness. This is due to the stakeholders tending to express their requirements in terms of operations and action rather than goals. Searching for action words is a useful way to extract goals from stakeholder descriptions. In addition to the goals, agents, stakeholders and constraints must also be identified. Identify the responsible agents as early as possible by determining what agents are ultimately responsible for the achievement or maintenance of a goal. Constraints are useful because they provide additional information regarding requirements that must be met in order for a given goal to be completed. Identify constraints by searching for temporal connectives, such as during, before and after, or any variants thereof. Figure 2.1 shows an example of an identified goal.

<table>
<thead>
<tr>
<th>Maintenance Goals</th>
<th>Agent</th>
<th>Stakeholders</th>
</tr>
</thead>
<tbody>
<tr>
<td>G₁: Career tracks provided</td>
<td>AFB</td>
<td>AFB</td>
</tr>
<tr>
<td>G₂: Tax payers money spent efficiently</td>
<td>AFB</td>
<td>AFB, DoD, empl</td>
</tr>
<tr>
<td>G₃: Training coordinated</td>
<td>AFB</td>
<td>AFB, empl</td>
</tr>
</tbody>
</table>

*Figure 2.1: Example of the identified Goal (Anton, 1996)*

• **Goal Evolution.** It is a known fact that during the development of software, stakeholders change their minds and refine and operationalize the goals into behavioral requirements. A stakeholder’s goals may change or, at a minimum,
their goal priorities are likely to change. Goal evolution is thus affected via goal elaboration and refinement. Useful techniques for goal elaboration are, (1) identifying goal obstacles, (2) analyzing scenarios [8] and constraints and (3) operationalizing goals.

In order to one to anticipate exception cases, goal obstacles need to be identified. Meaning, possible ways for goals to fail need to be considered. Goal refinement occurs when synonymous goals are reconciled, when goals are merged into a sub-goal categorization, when constraints are identified, and when goals are operationalized. Further consideration of goal and agent dependency relations yields deeper insights for conflict resolution. In other words, goals are refined by eliminating redundancies and reconciling with synonymous goals. Goals are also refined via elaboration. The operationalized goals, responsible agents, stakeholders, constraints and scenarios are ultimately consolidated into a set of goal schemas that can be easily translated into a requirements specification. The resulting artifact, while not formal in the strict sense, provides a textual representation of system requirements organized according to system goals. Figure 2.2 illustrates an example of a goal schema.
<table>
<thead>
<tr>
<th>Goal:</th>
<th>Available course slots announced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Achievement</td>
</tr>
<tr>
<td>Description:</td>
<td>HRD must announce the courses and the number of slots available so that qualified personnel can be identified, matched and notified.</td>
</tr>
<tr>
<td>Action:</td>
<td>Announce course slots</td>
</tr>
<tr>
<td>Agent:</td>
<td>HRD</td>
</tr>
<tr>
<td>Stakeholders:</td>
<td>HRD, employee, TSD</td>
</tr>
<tr>
<td>Obstacles:</td>
<td>1. No slots available&lt;br&gt;2. Employee prefs not available</td>
</tr>
<tr>
<td>Scenarios:</td>
<td>1.1 All courses closed (max capacity);&lt;br&gt;1.2 Courses cancelled (no slots avail)</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>1. Employee prefs ready&lt;br&gt;2. IPs made available</td>
</tr>
<tr>
<td>Postconditions:</td>
<td>Employee and course slot matched</td>
</tr>
<tr>
<td>Subgoals:</td>
<td>1. Qualifying training slots announced&lt;br&gt;2. Position training slots announced</td>
</tr>
</tbody>
</table>

Figure 2.2: Example of a goal schema (Antón, 1996)

**Goal Based Requirements Analysis Tool or GBRAT** supports the goal-based requirement analysis. The platform serves as a medium for project team members from different locations involve in the decision-making processes, which permeate requirements engineering. Ability to collaborative on new ideas, discusses, and decisions about goals despite their location.

GBRAT user is expected to be experienced requirements engineer in the goal-based method and web based applications. However the GBRAT developers have made certain assumption.

"We assume that GBRAT users will work from existing diagrams, textual statements of need and/or additional sources of information, such as transcripts of interviews with stakeholders to identify and specify the goals of the desired system. After the analyst/elicitor has gathered all available information about..."
the desired system he or she can then extract goals from these information sources and specify them using GBRAT as described below." (Antón, 1996)

Features of GBRAT enable users to create project repositories, create goals, trace goals, view goals from several perspectives and order goals. Among the features are

- **Project Repositories.** Goals concerning a project are kept or stored in a project repository. It is unique based on the specified project name and description as well as the name of the analysts working on a given project as shown in Figure 2.3. In a specific project repository, new goals can be created or previously specified goals can be viewed using three filters, (1) the maintenance and achievement goal filter, (2) the agent filter and (3) the total order filter.

![Figure 2.3: Project Repositories (Antón, 1996)](image_url)

Groupware Supported Requirements Analysis Tool
CHAPTER 2: Review on Requirements Analysis and Groupware

- **Creating Goals.** Users need to submit a form to create a goal, as illustrated in Figure 2.4, to specify the goal name, classification and responsible agents(s). The naming convention for goals is in a standardized subset of natural language. The first word is a verb that describes the kind of goal being named. The goals are classified either as an achievement or as a maintenance goal. Achievement goals are objectives of the system and are named by the verbs "MAKE" and "KNOW" where else, maintenance goals are those goals that are satisfied while their target condition remains true and are therefore named using the verbs "MAINTAIN", "KEEP", "AVOID" and "ENSURE".

![Figure 2.4: GBRAT Form to Create Goals (Antón, 1996)](image-url)
• **Goal Traceability.** The traceability is enabled through hypertext links. In creation of goal, the author must specify the name of the information source from which each goal was identified to ensure that each goal can be traced back to its author, place of origin. In addition, analysts are able to identify goals that have been extracted from more than one information source in order to reconcile any similarities and differences.

• **Viewing Goals.** GBRAT provides users viewing services such that various filters can be used to view goals. The goals are listed alphabetically and displayed in a tabular format, as shown in Figure 2.5. GBRAT allows users to view either achievement or maintenance goals by name. The goals in Figure 2.x are achievement goals.

![Figure 2.5: Viewing Goals by Name (Antón, 1996)](image)
CHAPTER 2: Review on Requirements Analysis and Groupware

In summary, GBRAT supports the two stages of the goal-based approach, (1) goal analysis and (2) goal refinement and decomposition. Looking into the facilities provided by GBRAT, it supports the collaborative nature of requirements engineering by implementing Internet technologies. It is done by simply allowing project members regardless of the place they are to work collaboratively. However, Annie I. Antón and her fellow colleagues (1996) of Georgia Institute of Technology highlighted that the use of the Internet introduces problems linked to cooperation, information sharing, different viewpoints and networked enterprises.

2.2.2 Win-Win Requirements

At the last two International Conferences on Software Engineering, three of the six keynote addresses identified negotiation techniques as the most critical success factor in improving the outcome of software projects. Tom DeMarco stated that

“How the requirements were negotiated is far more important than how the requirements were specified.” (DeMarco, 1996)

Mark Weiser concluded that

“Problems with reaching agreement were more critical to the projects’ success than such factors as tools, process maturity, and design methods.” (Weiser, 1997)

The original spiral model (Boehm, 1988) uses a cyclic approach to develop increasingly detailed elaborations of a software system’s definition, culminating in
CHAPTER 2: Review on Requirements Analysis and Groupware

incremental releases of the system’s operational capability. Each cycle involves four main activities:

- Elaborate the system or subsystem’s product and process objectives, constraints, and alternatives.
- Evaluate the alternatives with respect to the objectives and constraints.
- Identify and resolve major sources of product and process risk.
- Elaborate the definition of the product and process.
- Plan the next cycle, and update the life-cycle plan, including partition of the system into subsystems to be addressed in parallel cycles. This can include a plan to terminate the project if it is too risky or infeasible.
- Secure the management’s commitment to proceed as planned.

After putting into practice, some difficulties came about and to avoid these problems, the spiral model was further reviewed and three activities were to the front of each spiral cycle, as illustrated in Figure 2.6 (Boehm, 1994). The activities are

- Identify the system or subsystem’s key stakeholders.
- Identify the stakeholders’ win conditions for the system or subsystem.
- Negotiate win-win reconciliation of the stakeholders’ win conditions.
In Win-Win requirements model the decision space that gets collaboratively explored in a WinWin process is modeled using four main conceptual objects:

- Win Condition - capturing the desired objective of the individual.
- Issue - capturing the conflict between win conditions and their associated risks and uncertainties.
- Option - capturing a strategy for resolving an issue.
- Agreement - capturing the agreed upon set of conditions which satisfy stakeholder win conditions and also define the system objectives. The ontology also defines a set of relations between these objects.

Figure 2.7 shows a typical abstract structure of the decision rationale in terms of the above entities and the link types denoting the relations between them. As shown in the figure, an issue (I) is related to one or more win conditions (W x and W y) through the involved relation. An option (O) for resolving an issue (I) is related to the issue through the addresses relation. An agreement (Ag i) based on an option choice (Op) is related to Op through the adopts relation. The agreement (Ag i) has a covers relation with a win condition and a replaces relation to any previous agreement it substitutes.
CHAPTER 2: Review on Requirements Analysis and Groupware

Figure 2.8 provides a schema-based representation of each of the artifacts. Each object type is defined by a set of tuples, where each tuple consists of the slot name denoting the relation name with a single or multiple cardinality and a typed value field that ranges over an object or enumerated value set. For example, in the win condition object \( W \), the relation comment is an one-to-many function between win conditions and the set of all strings. Similarly the relation defined by the adopted by slot in an option is an one-to-one function between the options and agreements. The state slot of each object is an one-to-many function from the object to an enumerated set of unary predicate constants. For example, a win condition can be in both active and at issue state.

![Diagram](image)

*Figure 2.7: Win-Win decision objects and relation between them*
Given the above WinWin framework for collaboration, the elucidated Win-Win requirements model represents stakeholder oriented needs and constraints. Analyzing the Win-Win requirements model for understanding issues and guiding negotiation of tradeoffs requires a solution-oriented model at a level of abstraction relevant to understanding the Win-Win artifact interactions. Figure 2.9 shows an initial conceptualization of the extended decision rationale structure that builds on the viewpoint that the process of analysis is best facilitated by mapping the requirements model to an abstract design model. The extensions in the decision rationale structure involve use of a design model (D) that is motivated by the win conditions (W), considering design factors (Df) arising from D for explaining issues and introducing revisions (del-D and del-W) based on options.
Figure 2.9: An initial conceptualization of the decision structure supporting analysis of Win-Win requirements model.

The above conceptualization of the abstract design model and its relationship to the requirements model can be refined based on a more detailed understanding of the needs that must be met by the design representation. The refinement points would be

- Win conditions expressing functional objectives need to be mapped to functional elements and their task assignment. For instance, a representation, explicating the active information processing design elements and their task responsibility then serves as the basis for attributing constraints on design features of the function elements in order to achieve non-functional objectives.

- Win conditions expressing non-functional objectives need to be situated for relevant tasks and function elements and mapped to constraints on those elements - the constraints make explicit behavioral, communication, structural, and other architecture oriented characteristic features of function elements that strengthens quality properties.
• Issues expressing win condition conflicts have their basis in design factors - the design factors make explicit the negative ramifications of existing design feature constraints on functional and non-functional needs.

• Options expressing strategies for resolving issues need to be mapped to revision/tradeoff constraints - the revision constraints make explicit the necessary changes required to resolve issues pertaining interfering ramifications of design choices.

The point to note is that the abstraction being used in the above mapping of the WinWin requirements model must be adequate to capture multiple views. Such representational adequacy is required to map and understand different stakeholder win conditions that bear on different aspects of the underlying design model elements and their relationships.

**WinWin** is a computer program that aids in the capture, negotiation, and coordination of requirements for a large system. It assumes that a group of people, called stakeholders, has signed on with the express purpose of discussing and refining the requirements of their proposed system (Horowitz, 1999). The system can be of any type. WinWin contains facilities for:

• Capturing the desires (win conditions) of the stakeholders
• Organizing the terminology so that stakeholders are using the same terms in the same way
• Expressing disagreements or issues needing resolution
• Offering options as potential solutions
CHAPTER 2: Review on Requirements Analysis and Groupware

- Negotiating agreements which resolve the issues
- Using third party tools to enlighten or resolve issues
- Producing a requirements document that summarizes the current state of the proposed system
- Creating documents that support multimedia and hyperlinks
- Tracing the ways by which requirements decisions were reached
- Checking the completeness and consistency of requirements

According to the Ellis Horowitz (1999) and his colleagues at University of Southern California (USC), users are advised to follow this simple scenario to begin their work, which has been illustrated in Figure 2.10.

i. Identify the owner of the project. He/she identifies other members.

ii. Owner starts WinWin and registers the project and members, whom are the stakeholders of the system.

iii. Input the defined or tailored existing set of terms of the proposed system into WinWin.

iv. Input the defined or tailored existing taxonomy of the proposed system into WinWin.

v. Review and iterate the terms and taxonomy.

vi. Begin negotiation process, (a) Create Win Conditions and/or (b) create Issues and/or (c) create proposed Agreements.

vii. Review entered artifacts, (a) new Issue is created from new conflict, (b) Options developed to address Issues and (c) new Agreements are created.
Steps 6 and 7 are continued until all Win Conditions covered, all Issues are resolved and all Agreements are passed.

To support auxiliary tools that stakeholders desire to use during the course of negotiation, WinWin provides file attachments facility. For instance, the stakeholders need to use a spreadsheet to analyze the financial impacts of a given Option as shown in Figure 2.11.

Figure 2.10: WinWin Scenario

Figure 2.11: File attachment utility in WinWin
CHAPTER 2: Review on Requirements Analysis and Groupware

Since WinWin assumes that all the stakeholders are at different locations and working at different time, WinWin was made as a distributed and asynchronous mode of operation. In other words, a stakeholder can log into the system regardless of place or time.

2.2.3 Inquiry cycle

Results from several field studies and experiments that were conducted on work practice problems in requirements analysis clearly emphasizes the importance of three activities, namely (1) communication, (2) agreement and (3) traceability management (Curtis, 1988; Kaiya, 1993; Takashashi, 1994.). The problem is these activities are implemented ineffectively and are poorly supported.

Realizing this, Takahashi (1993) and his group of researches developed the Inquiry Cycle Model. It is basically a cyclic model of inquiry based requirements analysis. According to them, the model has been applied to several projects successfully. The Inquiry Cycle for requirements analysis consists of 3 activities

“(1) Expression is the proposing or preparing of requirements-related information including not only requirements documents, but also domain specific information, scenarios and enterprise goals. (2) Discussion includes discussing requirements in formal meetings and circulating of comments and individual requirements. (3) Commitment includes making decisions based on the discussions, such as charge requests, agreements about terminology and commitments seeking missing information.” (Takahashi, 1993)

The 3 activities identified as expression, discussion and commitment is done in loops under control to refine requirements until they become deliverables. Figure 2.12
briefly shows the Inquiry Cycle Model. Take note that the number and nature of the stages are controlled by the strategy. There are three possible views of the Inquiry Cycle to contribute to requirements analysis. It could act as rhetoric for explaining ideas, traceability and as a process model that coordinates and guides participants toward an agreed specification.

Figure 2.12: Inquiry Cycle Model.

The advantages of Inquiry Cycle being an artifact-based model for collaborative work are the traceable changes and assisting participants share awareness by making the artifact being discussed visible and explicit.

EColabor is an active hypermedia for collaboration requirement analysis. It is designed to address problems in communication, agreement and change management in
requirements analysis. Based on the Inquiry Cycle model, EColabor supports analysts in systematically managing the requirements. EColabor records all the processes of elaborating requirements in shared hypermedia and provides comprehensive support for utilizing these records. According to Kenji Takahashi and his fellow Ecolabor developers,

"Ecolabor has client-server architecture that provides participant, even if distributed or working asynchronously, with transport access to the Ecolabor hypermedia information" (Takahashi, 1996).

Basic features of Ecolabor are,

- Assigning an URL to each element of the hypermedia information stored.
- Extending HTTP to handle check in/out of documents
- Sharing information
- Multicasting audio/video stream
- Shared electronic whiteboard
- Application sharing

Ecolabor provides traceability support through the assist of multimedia. It records an entire session in audio/video and provides support for segmenting the records on the fly. It requires participant to select elements of Ecolabor hypermedia information that they are currently discussing as specifically as possible on their computer displays to set a high level of awareness among other team members. It also supports taking snapshots of sketches and text on the shared whiteboard.

To support coordination, Ecolabor identifies and reports the status of requirements analysis process. It also lets the participant to control the access to the
working version and define a new version. It manages the status of discussions over revisions. In other word, Ecolabor provides status management and version controlling. However, many have opted out due to its complexity in customization. In addition, it requires a decision to be made by one person based on his/her justification without providing flexibility like voting to close a problem or issue.

In summary, Ecolabor manages the pieces of information generated and referred to during requirements analysis. It transmits information with the help of multimedia application. It keeps track of the evolving requirements documents along with the informal information regarding it. Finally, it monitors and navigates the requirements analysis process.

2.2.4 KJ

Repeatedly it has been emphasized that the most important step in system development is to analyze, understand and record the problem that the user, sponsor or client is trying to solve (Takeda, 1992). The functions, goals and constrains on the proposed system must be precisely specified and both the parties must agree on the specification, as they form the basis for the design of the system.

For such problem recognition stage, the KJ method is very effective. It was formalized by Kawakita (1982) for generating ideas in his ethno geographical works. The method has been well structured such that it is widely accepted by Japanese business community for usefulness of consensus making among participants of idea generation.

As mentioned earlier, KJ method was developed for new idea generation and claims to establish an orderly system, from chaos. It consists of 4 steps.

The initial stage is to write down on a card what has come to mind on the subject to be discussed. Only one thing must be written down on a card. No judgment
CHAPTER 2: Review on Requirements Analysis and Groupware

should be made on the importance of what has been written. According to Kawakita, the importance can only be valued after the stage is completed.

Next, is to associate several cards into one group. All the cards must be shuffled and spread on the table or floor. Each card is read several times. The classification or grouping is done subjectively rather than objectively or rationally by examining contents of the card. As quoted by Kawakita, the subject may reveal real desire hidden by the rationality. The grouped card may be labeled in a new card and further grouped in a hierarchical manner.

Following that, the cards are grouped on a large piece of paper and to enclose each group by an outline. Spatial relationships of arranged cards and groups must reflect semantic relationship and extreme prudence is required for this step. It is also necessary to make clear the mutual relationships between the cards and groups by drawing special lines. The relationships include opposition, causality and equality. At this stage, the internal structure of the matters written on the cards that is invisible in the first step becomes visible. The result is called A-type chart.

The final step would be to write down an essay on the subject according to the A-type chart. This step is called B-type writing. It should be noted that the A-type chart represents the subject spatially and that the B-type writing represents the same information in a sequential order. Because of the difference in representation, while doing B-type writing, oversight in the A-type chart may be found and some revision must then be made. However at this point if the cards are pasted on a piece of paper, this is not easy. This fact often makes one hesitant in revision the diagram. Figure 2.13 shows a brief outline of the KJ method while Figure 2.14 depicts the results of KJ Method in requirements analysis.
The system developed to support KJ method was called KJ Editor. KJ Editor stimulates index card arrangement on a desk is based on the KJ method that is widely used in the Japanese business community (Takeda, 1992). It is a new tool for getting panoramic view of the whole card arrangement on a computer display.

According to the KJ Editor developers, KJ Editor is a very good tool to record the thinking process of the designer. The thinking process is directly reflected on the
CHAPTER 2: Review on Requirements Analysis and Groupware

editing action and reviewing the whole process may easily be speeded up, because the record is machine readable and processable.

As the system is developed in Japan, most of the information regarding the functionality of the system is in Japanese. Therefore, there is not much a paper or manuals in English regarding the system that can be referred to. From what the developers have summarized, the KJ Editor is not just limited to KJ method but it is open-ended as it supports the development of flow chart, DFD and state transition diagram. Adding, they also quoted that the Editor ability of recording the thinking process makes the resultant chart produced from KJ editor a very good communication tool between the designer and the client.

2.2.5 Summary of the reviewed methodologies and tools

The above review reveals the characteristics of some of the widely used tools currently available in market. The two tables below summarize the basic characteristics of the tools and the advantages and disadvantages of the tools.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Requirements Analysis Methods</th>
<th>Web-Based</th>
<th>Distributed</th>
<th>Stand Alone</th>
</tr>
</thead>
<tbody>
<tr>
<td>GBRAT</td>
<td>Goal-based Requirements Analysis</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>WinWin</td>
<td>Win-Win</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Ecolabor</td>
<td>Inquiry Cycle</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>KJ Editor</td>
<td>KJ</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 2.1: Comparison of the reviewed tools
### Table 2.2: Advantage and disadvantage of the reviewed tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Advantage</th>
<th>Disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>GBRAT</td>
<td>Creates repository for projects, which enables the building of knowledge network or future reference within an organization. Traceability supported through hypertext links. Supports collaborative nature by implementing Internet Technologies</td>
<td>The users are required to know about the goals, which are documented elsewhere before using the system. No room for discussion or information sharing.</td>
</tr>
<tr>
<td>WinWin</td>
<td>Proper project kick-off with the selection of team members. Capturing the win conditions of the stakeholders Ability to express disagreements or issues Negotiating to resolve issues Producing a requirements document Traceability of requirements decisions Checking the completeness and consistency</td>
<td>Abstraction for Win-Win mapping must be adequate to capture multiple views.</td>
</tr>
<tr>
<td>Ecolabor</td>
<td>Sharing information Multicasting audio/video stream Semi-structured email Shared electronic whiteboard Application sharing</td>
<td>Complexity in customization Decision-making is based on one's own justification.</td>
</tr>
<tr>
<td>KJ Editor</td>
<td>Supports brainstorming session Supports classifications of the requirements</td>
<td>Stand alone system Does not support collaboration related activities.</td>
</tr>
</tbody>
</table>

2.3 Groupware

Groupware as a collaborative tool provides an easily accessible, widespread platform for gathering and sharing information and for capturing ideas. Groupware is an umbrella term describing the electronic technologies that support person-to-person collaboration. Technologies that support collaboration are in greater demand today than ever before, and, in recognition of that fact, vendors are integrating collaboration technologies into their products. Distributed workforces, information overload, and
getting products to market as quickly as possible are just a few of the motivations pushing collaboration technology development. Additional, as strongly supported by Herd Krasner (1991) and his colleagues, the development of collaborative related technologies depends on the identification of specific human enterprises where the new technologies might prove useful.

2.3.1 Definition of Groupware

Groupware is a relatively new term or concept. Peter and Trudy Johnson-Lenz originally used it in between 1978 to 1980 (Baecker, 1993) to refer to,

“Intentional group processes plus software to support them”

Industry leaders present the following definitions, the most commonly used,

“Computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment.” (Ellis, 1991)

“Groupware is computer-based technology that (1) actively facilitates two or more users working on a common task, possibly simultaneously, using a shared environment and (2) provides synergistic mechanisms for coordinating each user’s actions with respect to the rest of the group and system.” (Krasner, 1991).

“Computer-mediated collaboration that increases the productivity or functionality of person-to-person processes.” (Coleman, 1996)
"Computer mediated culture." (Baeker, 1993)

As shown in Figure 2.15, groupware lies on a network infrastructure that includes PCs, cabling, network operating systems and administration utilities, or phone lines for a wide area network (WAN). Although groupware is part of the networked applications environment, not all networked applications constitute groupware. For example, access to a corporate database through a network is not necessarily groupware (Coleman, 1997).

Figure 2.15: Groupware's position in IT architecture (Collaborative Strategies, 1996)

In software process the terms like teamwork, cooperation, coordination and communication that occur within and among groups is synchronous through out the life of software project currently. As mentioned by Alan Kay (cited McLaughlin, 1989) in the final plenary session of the 11th International Conference on Software Engineering, the current examples of automated tools aimed at collaborating groups from the notion of groupware, which is oriented toward the "Tomorrow" column in Table 2.3. The
systems of today as reported by Kenneth L. Kraemer and John Leslie King that support working groups such as automated scheduling applications and electronic mail do not support group as such but rather individuals working in a group (Kramer, 1988).

<table>
<thead>
<tr>
<th>Function</th>
<th>Yesterday</th>
<th>Today</th>
<th>Tomorrow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where</td>
<td>Machine Room</td>
<td>Desktop</td>
<td>Wherever you are</td>
</tr>
<tr>
<td>Who</td>
<td>Expert</td>
<td>Individuals</td>
<td>Collaborative Tools</td>
</tr>
<tr>
<td>What</td>
<td>Edit</td>
<td>Layout</td>
<td>Orchestrate</td>
</tr>
<tr>
<td>How</td>
<td>Remember and Type</td>
<td>See and tell</td>
<td>Ask and tell</td>
</tr>
<tr>
<td>Style</td>
<td>Data/procedures</td>
<td>Object Oriented</td>
<td>Rules</td>
</tr>
</tbody>
</table>

To summarize, groupware is computer technology that actively facilitates groups of collaborating users. The goal of groupware is to improve group process, bringing about more effective performance of the group task.

2.3.2 Groupware Parameters

Systems that support groupware activities can be characterized by the way they deal with the set of key parameters. The parameters that were taken into considerations and its definitions are listed below as guided by Krasner (1984).

- Time is the period over which task-related interactions may occur, either synchronous or asynchronous. The occurrences may be for a short or a long period of time.

- Place basically refers to the physical locations of the members of a group, which could be face to face or from one corner to another corner of earth.
CHAPTER 2: Review on Requirements Analysis and Groupware

- Task meanwhile refers the group’s objective and nature of the tasks involved. For instance making decision or voting.

- Human-machine allocation of work refers to the extent to which both the task to be performed and the coordination of the work are performed by humans versus the extent to which they are automated.

Other parameters as directly mention by Herb Krasner and his colleagues are

"Group context includes the groups environment, the incentives and the social conventions present in the cultural infrastructure of the working group, including the group’s physical surroundings, its adopted style (pecking order), the level of formality in its communication, its willingness to try new things and so forth.

Group composition includes the characteristics of the individuals who make up the group and the relationship among those characteristics, for example, the distribution of task-related expertise among group members.

Artifact and process focus is the extent to which the groupware focuses on products of a process versus the extent to which it focuses on processes."

(Kramer, 1991)

Table 2.4 refers to the examples based on the discussion and the conclusion by Johansen (1988) on the role of time and place observing that most groupware supports either one of the four conceptual quadrants according to the needs of the group. He also
asserted that the future groupware to overcome barriers of time and place in order to provide smooth transitions between synchronous and asynchronous work and between collocated and dispersed work.

Table 2.4: Time and Place Dimensions of Groupware Examples.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Same time</th>
<th>Different time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same place</td>
<td>Meeting Room</td>
<td>Hospital nurses’ records</td>
</tr>
<tr>
<td>Different Place</td>
<td>Teleconferencing</td>
<td>Wide area computer conferencing</td>
</tr>
</tbody>
</table>

2.3.3 Groupware Taxonomy

According to David Coleman (1997), there are twelve functional categories, which form a logical taxonomy. He added 3 new categories from the original, which is a separate category for groupware services, a new category for groupware applications and a special category for the emerging Internet-based collaborative applications and products. The twelve categories are,

- Electronic Mail and Messaging

Email is by far the most common groupware application. While the basic technology is designed to pass simple messages between 2 people, even relatively basic email systems today typically include interesting features for forwarding messages, filing messages, creating mailing groups, and attaching files with a message. Other features that have been explored include automatic sorting and processing of messages, automatic routing, and structured communication.
CHAPTER 2: Review on Requirements Analysis and Groupware

- Group Calendaring and Scheduling

Allows scheduling, project management, and coordination among many people, and may provide support for scheduling equipment as well. Typical features detect when schedules conflict or find meeting times that works for everyone. Group calendars also help to locate people. Typical concerns are privacy, completeness and accuracy.

- Electronic Meeting Systems

This is a real-time conferencing system as well as collaborative presentation systems. It integrates with calendaring/scheduling systems. It also allows post-meeting follow through the posted information on action items, goals and commitments. Affordability of desktop videoconferencing and Availability of multi-point conferencing make it a suitable groupware application. Addition, they provide tools for brainstorming, critiquing ideas, putting weights and probabilities on events and alternatives, and voting. Such systems enable presumably more rational and even-handed decisions they can encourage equal participation by, for instance, providing anonymity or enforcing turn taking.

- Desktop Video and Real-time Data Conferencing (Synchronous)

Video communications systems allow two-way or multi-way calling with live video, essentially a telephone system with an additional visual component. Cost and compatibility issues limited early use of video systems to scheduled videoconference meeting rooms. Video is advantageous when visual information is being discussed, but may not provide substantial benefit in most cases where conventional audio telephones are adequate. In addition to
supporting conversations, video may also be used in less direct collaborative situations, such as by providing a view of activities at a remote location.

- Non Real-time Data Conferencing (Asynchronous)
Contrary to desktop video and real-time data conferencing, this is an asynchronous conferencing. It is mostly like bulletin boards, where one can carry on a conversation over time, leave a message for someone and they answer it, and your can respond back to them later. These messages can be left public or private basis. A clear example would be newsgroups and mailing lists. In practice the main difference between newsgroups and mailing lists is that newsgroups only show messages to a user when they are explicitly requested, sort of an on-demand service, while mailing lists deliver messages as they become available, an interrupt-driven service.

- Group Document Handling
At a wider definition, group editing, shared screen editing work, group document/image management and document databases falls into this category. A famous feature of group document handling would be hypertext. Hypertext is a system for linking text documents to each other, with the Web being an obvious example. Whenever multiple people author and link documents, the system becomes group work, constantly evolving and responding to others' work. Some hypertext systems include capabilities for seeing who else has visited a certain page or link, or at least seeing how often a link has been followed, thus giving users a basic awareness of what other people are doing in the system.

- Workflow
CHAPTER 2: Review on Requirements Analysis and Groupware

Workflow systems allow documents to be routed through organizations through a relatively fixed process. A simple example of a workflow application is an expense report in an organization: an employee enters an expense report and submits it, a copy is archived then routed to the employee's manager for approval, the manager receives the document, electronically approves it and sends it on and the expense is registered to the group's account and forwarded to the accounting department for payment. Workflow systems may provide features such as routing, development of forms, and support for differing roles and privileges.

- Collaborative - Internet-based Applications and Products

Many collaborative functions are moving to the WWW and use the Internet as the input and output while still using traditional groupware on the LAN. It is concentrated on application customization for seamless collaboration on the WWW, data or information storage and balance between security and collaboration. However, there is a very significant limitation of Web applications relative to traditional groupware.

2.3.4 Importance of Groupware

Collaborative Strategies (1996) was commissioned to conduct research in examining how large companies are using IP networks to support electronic collaboration. Their findings were

- 56% felt collaboration was necessary to support a distributed workforce by increasing communication, coordination, facilitation and planning. Sales force
automation and distributed project management were two key functions where
collaboration was identified as critical.

- 13% use collaboration strategies to reduce cycle times in product development,
  thereby realizing an increase in competitive advantage.

- 6% believe that collaboration technologies increase productivity and coordinate
  or facilitate complex processes

### 2.3.5 Groupware Design

Those who practice traditional software design have a particular way of thinking
about computer. At the core of their mindset is a notion that the computer is a
mechanism for manipulating and exchanging data. Meanwhile effective groupware
design sees the computer as a medium for people to collaborate (Bock, 1995). Table 2.5
looks at different mindsets, how they differ, and discover how each influences the
design of the computer briefly.

<table>
<thead>
<tr>
<th>Design Mindset</th>
<th>What is collaboration</th>
<th>How is groupware designed</th>
<th>What does the design represent</th>
</tr>
</thead>
<tbody>
<tr>
<td>From Understanding</td>
<td>Complex and integrated human interactions</td>
<td>Detached study. Note complexity of interactions.</td>
<td>Purpose, goals, behavior, equilibrium</td>
</tr>
<tr>
<td>According to form</td>
<td>Stereotyped work. Tasks that can be labeled and ordered.</td>
<td>Detached study. Focus on form. Identify patterns.</td>
<td>Types of objects, object relationships, object states.</td>
</tr>
</tbody>
</table>
2.3.6 Performance and Acceptance of Groupware Application

One form or other sort of electronic communities were existence for almost as long there has been reliable electronic communication, at least since the mid 1970s (Hiltz, 1984, Hiltz, 1978 and Valee, 1978). Mainly used for proprietary conferencing system. The evolution of Internet and internal network spawned email, listservs and usenet newsgroup to came into action. Their advantage was the universality characteristics compared to the proprietary conferencing systems. Since conferencing systems, email, listservs and newsgroup are generally categorized as different forms of groupware, groupware was defined as a set of hardware and software designed to help groups work together regardless of time and place (Nunamaker, 1991).

The web presents a new opportunity. Rather than relying on proprietary software and architecture, which by their very nature are limiting, new groupware systems and architectures that take advantage of the web, a widely accepted open standard. Due to that, more than 75 groupware systems were made available in 1996, the same year such system made a debut (Woolley, 1996).

Based on the survey conducted on understanding the ways web based groupware was being used by Alan R.Dennis and Bradley C.Wheeler (1997), the major categories are supporting project teams, educational use and special interest group. A further breakdown on the result of the survey is shown in Figure 2.16.
Figure 2.16: Survey on web based groupware

The most mentioned advantage of web groupware was the use of open network standard of Internet and the use of open client standard to enable any-place-any-time interaction. The other advantages are the ability of transformation such as to structure, analyze and sort discussion, specific features, common tools, inexpensive setup costs and minimal learning. The advantages of and its influence are shown in Figure 2.17.
However, it is a common nature that when there are advantages, disadvantages tends to appear. For this web groupware, the disadvantages can be categorized into three groups. The first and most prominent set of disadvantage dealt with network technologies. The second group is centered on the features of current web group and final group includes operating costs, changing workgroup skills and others. Figure 2.18 depicts the result of the survey conducted on disadvantages of web groupware (Dennis, 1997).

![Disadvantages of Web Groupware](image)

*Figure 2.18: Survey results on Groupware Disadvantages*

### 2.3.7 Groupware for Requirements Analysis

Groupware use collaborative technologies to either enhance processes or support collaboration in a specific work environment. With the development of web in terms of architecture and infrastructure has definitely created a boom in groupware applications. The web revolutionizes groupware and the way in which organizations choose to use it. It was expected in 1997 that the web and groupware has transformed from electronic publishing to building and maintaining virtual organizations and enhancing the work of project teams within two years (Dennis, 1997). Overall, groupware can be an important
productivity tool, but the current state of the art is better suited for small team environments rather than large groups. However, active, intelligent, adaptive, orchestrated groupware that manages multiple and possibly concurrent accesses to shared workspaces is the desirable goal for future technology (Krasner, 1991). It is a common believe now that the one who get to the web first, learn its intricacies and push its limits to have a distinct advantage.

Looking at the various requirements analysis frameworks and mentioned in section 2.2, all the requirements analysis methods require some level of teamwork if not at all level. However, the question is to what extend does the tools that supports the requirements analysis methods are supported by groupware technology. The table below looks into the tools and the various groupware technology implemented.

Table 2.6: Comparison of the requirements analysis tool against the groupware support

<table>
<thead>
<tr>
<th></th>
<th>GBRAT</th>
<th>WinWin</th>
<th>Ecolabor</th>
<th>KJ Editor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronic Mail and Messaging</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Group Calendaring and Scheduling</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Electronic Meeting Systems</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Desktop Video and Real-time Data Conferencing (Synchronous)</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Non Real-time Data Conferencing (Asynchronous)</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Group Document Handling</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Workflow</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Collaborative – Internet-based Applications and Products</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
</tbody>
</table>

Although requirements analysis is a process where teamwork or collaborative exercises are basically in high priority in successful projects, but the tools except for Ecolabor does not seem to be able to perform those task as clearly shown in Table 2.6.
2.4 Research Framework

After reviewing various requirement analysis methods and tools that complement them, in addition to the groupware reviews, a framework for this research has been defined and described in the following sub-sections. The tool to be developed was named as Groupware Supported Requirement Analysis Tool (GRAT).

2.4.1 Requirements Analysis Method

According to Ian Sommerville (1996), requirements analysis or elicitation is the major stage of a generic requirements engineering process as shown in Figure 2.19.

![Figure 2.19: Generic Requirements Engineering Process](image)

In order to execute requirements analysis, the problem domain must be well understood by the analysts, as the process used is often domain dependent. Most of the models of the process are inevitably simplifications. The process model shown in Figure 2.20 illustrates a number of important requirements analysis activities.
As depicted in Figure 2.20, the processes in requirements analysis are highly iterative with feedback from each activity to other activities. As Ian Sommerville quoted in his book,

"The process can be viewed as a cycle starting with domain understanding and ending with requirements validation. The analyst’s understanding of the requirements improves with each round of cycle." (Sommerville 1996)

The process activities are:

- **Domain Understanding**. In this process, the domain must be clearly understood by the analyst. For instance if a clinic management system is being developed, participants need to know as much as possible on the operational of a clinic management.
CHAPTER 2: Review on Requirements Analysis and Groupware

- **Requirements Collection.** All stakeholders interact and the requirements for a system are identified or discovered.

- **Classification.** As the name of the process states, it is basically grouping of requirements into the relevant classes or categories.

- **Conflict Resolution.** In any requirements analysis, requirements conflict especially when it involves more than one participant. So, at this stage, the conflicts are identified and resolved.

- **Prioritization.** Approaching this stage, a set of requirements and its relevant categories or classes is prepared. Some requirements have higher precedence than others. Here is where the important requirements are identified.

- **Requirements Validation** Finally, the requirements are checked to see if they are complete, consistent and in parallel of users need.

As a final result of this stage would be the functional and non-functional requirements of the system being developed. This is a part of the software requirements specification in which could be used as an agreement between user and the developer before proceeding to the later stages of the implementation.

For this research, requirement analysis introduced by Ian Sommerville is selected. Among the reason implementing Ian Sommerville’s requirement analysis, as deemed by DeMarco (1996) as foundations for a good requirements analysis methodology.
CHAPTER 2: Review on Requirements Analysis and Groupware

- Supports collaborative. As can be seen in the diagram above, almost all the activities require some level of co-operation and collaboration.

- Ability to express disagreements or issues and negotiating to resolve issues as done in conflict resolution.

- Producing a requirements document as a result of a successful requirement validation.

- Checking the completeness and consistency as done in requirements validation.

- Supports classifications of the requirements.

Table 2.7 compares the characteristics of the Ian Sommerviles method against the ones reviewed earlier

Table 2.7: Comparison of Ian Sommerville’s methodology based on the advantage of the GBRAM, Win-Win, Inquiry Cycle and KJ

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>GBRAM</th>
<th>Win-Win</th>
<th>Inquiry Cycle</th>
<th>KJ</th>
<th>ISRA*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repository for projects</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Traceability</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Supports collaborative</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Ability to express disagreements or issues</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Negotiating to resolve issues</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
</tbody>
</table>
2.4.2 Groupware Supported

Based on the methods mentioned briefly earlier, a tool with groupware-supported facilities was developed. In other words, the aim is to develop groupware supported requirements analysis tool and at the same time reduce constrains on time and space. The model is discussed in detail later in Chapter 3. Tables 2.7 shows how GRAT fares when compared to other tools in terms of groupware applications.

Table 2.8: Comparison of the groupware features of GRAT against other tools

<table>
<thead>
<tr>
<th>Feature</th>
<th>GBRAT</th>
<th>WinWin</th>
<th>Ecolabor</th>
<th>KJ Editor</th>
<th>GRAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronic Mail and Messaging</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Group Calendaring and Scheduling</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Electronic Meeting Systems</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Desktop Video and Real-time Data Conferencing (Synchronous)</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>
CHAPTER 2: Review on Requirements Analysis and Groupware

<table>
<thead>
<tr>
<th>Non Real-time Data Conferencing (Asynchronous)</th>
<th>x</th>
<th>x</th>
<th>x</th>
<th>x</th>
<th>✓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Document Handling</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Workflow</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Collaborative – Internet-based Applications and Products</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
</tbody>
</table>

As can be analyzed in the table above, GRAT provides a total groupware support for the requirement analysis activities. It provides almost all the integrated groupware facilities of the other 4 tools reviewed earlier except for desktop video conferencing. The reason for not being able to use desktop video conferencing is due to the development environment or platform does not support synchronous activities and also due to limitation on bandwidth.

2.4.3 Implementing a web-based tool

As mentioned earlier, a tool named GRAT or Groupware Supported Requirements Analysis Tool was developed. Making the tool web-based has given the tool an competitive and usability edge compared to other non-web based tools in the market. The advantages are:

- Open Network and Client Standard

GRAT is running on the web. Meaning, anyone with an Internet browser and valid login information are able to use and benefit from GRAT.
CHAPTER 2: Review on Requirements Analysis and Groupware

• Setup Costs

GRAT has significantly reduced setup costs as the installation of the system is only done at the server and does not require any installation on client. Purely representing thin client with some applets running is some of the process.

• Awareness

GRAT provides the facility where email notification can be executed for meetings as well as discussions.

• Repository

GRAT provides a repository for all the projects that were executed using that system which could be used for future reference, which indirectly feeds to the improvement of the process.

Other features mentioned in 2.4.2 that are included in GRAT are electronic mail and messaging, group calendaring and scheduling, group document handling, workflow, traceability of requirements, discussion forum and information sharing which are totally accessible from the web.

Table 2.9 summarizes the features of GRAT when compared to other tools mentioned earlier in the chapter.

Table 2.9: Comparison of GRAT against other tools based on the requirements analysis methods and supported architecture

<table>
<thead>
<tr>
<th>Tool</th>
<th>Requirements Analysis Methods</th>
<th>Web-Based</th>
<th>Distributed</th>
<th>Stand Alone</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRAT</td>
<td>Goal-based Requirements Analysis</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Groupware Supported Requirements Analysis Tool
CHAPTER 2: Review on Requirements Analysis and Groupware

<table>
<thead>
<tr>
<th>WinWin</th>
<th>Win-Win</th>
<th>No</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ecolabor</td>
<td>Inquiry Cycle</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>KJ Editor</td>
<td>KJ</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>GRAT</td>
<td>Ian Sommerville’s requirements analysis</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

2.4.4 Evaluating GRAT

GRAT was evaluated based on experimental materials that are mentioned in the later chapters. The evaluation is done to validate GRAT and not the issues on requirements analysis method.

2.5 Summary

This chapter reviewed some of the widely used requirements analysis methods and tools in the first section of this chapter. Next, it looked at the groupware from a general point of view and narrowed the view into requirements analysis by comparing groupware features in the requirements analysis tools. Finally, from the understanding on requirement analysis and how groupware complements the process, a general framework and features for GRAT was derived and identified.