CHAPTER 6

Validation

Practice proves more than theory, in any case.

Abraham Lincoln

While it is easy to propose a method, validation is required before the usefulness of the method may be determined. Software methods, such as the Goal-Based Requirements Analysis Method, need early validation while under development. The initial case studies which seek this validation, discussed in Chapter 3, are best characterized as the formative case studies due to their central role in shaping the method. In contrast, the case studies discussed in this chapter are best characterized as summative; their primary role in this thesis is to validate the method developed during the formative case studies.

Validation of the Goal-Based Requirements Analysis Method involved three applications of the method:

- use of the GBRAM to specify the requirements for tool support, followed by construction of a prototype based on the resulting requirements;
- a case study involving a large commercial application and multiple stakeholders to evaluate the scalability of the method; and
- an empirical evaluation whereby use of GBRAM by one group of subjects was compared to the use of alternative analysis methods by other subjects.

This chapter discusses the Goal-Based Requirements Analysis Method in the context of these three efforts. While theoretically different in research approach taken, each of these efforts seeks to validate the method. The GBRAM was used to specify the requirements for a tool to support the method*; consequently, a prototype, referred to as the Goal-Based Requirements Analysis Tool (GBRAT), described in Chapter 4.5, was developed. An overview of this validation effort is provided in Section 6.1. In order to investigate the scalability of the method, GBRAM was applied to a large commercial intranet web server application†, discussed in Section 6.2. A synopsis of an empirical evaluation performed to validate the Goal-Based Requirements Analysis Method is presented in Section 6.3‡. Table 6.1 summarizes the data availability for each of the validation studies discussed in this chapter.

Table 6.1. Availability of Case Study Data

Case Study	Raw Data	Requirements Document
GBRAT	√	√
Electronic Commerce WWW Server	**	**
Vacation/Sick Leave System	√	√
Key √ Available from author upon request ** Company Confidential		

*The data is available on the WWW at http://www.cc.gatech.edu/computing/SW_Eng/GBRAT/case.ps.

†This case study was sponsored by NTT Software Labs in Palo Alto, California, and is a company confidential project.

‡Available from the author upon request.

6.1 The Goal-Based Requirements Analysis Tool

The first summative effort to formally validate the efficacy of the Goal-Based Requirements Analysis Method involved using the GBRAM to identify the functional modules necessary for a tool to support the method. The objective of the Goal-Based Requirements Analysis Tool (GBRAT) is to provide analysts with the procedural support needed for the analysis and refinement of goals for software-based information systems, supporting and guiding analysts as they identify, capture, and structure requirements information in the form of goals.

Methodology and Case Study Artifacts

The Goal-Based Requirements Analysis Tool (GBRAT) case study was conducted for approximately 20 hours a week over a period of one month. One analyst was assigned to review the available documentation: a four-page description of the goal-based method (106 lines of text) and a one-page research abstract (20 lines of text). Both sources of information were analyzed using the GBRAM to identify system goals and translate those goals into operational requirements. Subsequently, a design document* was produced and a prototype† of the tool was built.

Goals were documented for the GBRAT using tables created in LaTeX for document preparation. The GBRAT requirements document was composed in Hyper Text Markup Language (HTML), which provided the ability to cross-reference between auxiliary notes

*http://www.cc.gatech.edu/computing/SW_Eng/GBRAT/design_doc.html
†http://www.cc.gatech.edu/computing/SW_Eng/GBRAT/

and requirements throughout the document. The hypertext links thus afforded a general level of requirements traceability. Since the subsequent design document was also written using HTML, traceability was ensured among the various document artifacts which led to the production of the final system.

Lessons Learned

This section summarizes the lessons learned from the GBRAT case study. In Chapter 3, the presentation of each case study includes a discussion of the lessons learned. The discussion of lessons learned in this chapter takes a different bent; the lessons learned during the summative case studies outlined in this chapter are much more validation oriented than development oriented, whereas the case studies detailed in Chapter 3 are primarily formative with a concurrent bent towards validation. Thus, the following discussion seeks to provide examples which confirm and augment the method presented in Chapter 3 and the heuristics presented in Chapter 5.

Redundant goals may be identified by looking for synonyms

During this case study, heuristic HRS 1. (page 168) was solidly confirmed by its application in numerous instances. For example, three synonymous maintenance goals were extracted from the research abstract: `strategies provided`, `techniques provided`, and `guidelines provided`. The three words *strategies*, *techniques*, and *guidelines* may imply different meanings. However, in the context of GBRAT, the three terms referred to the tool's provision of prescriptive advice. Since the word *strategies* encompassed the stakeholders'

intentions and was the most general of the three terms, and since one agent, GBRAT, was responsible for the three goals, these goals could be merged into one goal. This confluence is illustrated in HRS 1 (page 168): `strategies provided`.

The system is usually responsible for maintenance goals

Heuristic HCM 2. (page 164) suggests keywords which signal a continuous state within the system and indicate candidate maintenance goals. In this case study, the keywords 'provision of' and 'provide' facilitated the identification of several maintenance goals from the research abstract. This is of particular note in that the words 'provision of' and 'provide' in the research abstract directly correspond to the system as the responsible agent in this analysis. This suggests that maintenance goals may also be identified by considering who or what is the responsible agent. However, some goals had both the system and the user as shared agents; these goals with non-human and human agents were typically achievement goals. The following section further discusses the lessons learned from this occurrence.

Goals with two responsible agents may require two operational definitions

Heuristic HIA 3. (page 158) indicates that multiple agents may share responsibility for a goal. In the GBRAT case study, it was observed that viewpoints may be expressed via the operational definitions in the requirements document. Consider the GBRAT goal `Goals organized`. Two agents, the analyst, or user, and the system itself share responsibility for this goal, but the actions required of each agent for the organization of goals are different. Depending on the implementation, the analyst may need to specify formal dependency relations so that goals may be ordered, or use a drag and drop tool to order the goals (as

supported by the GBRAT goal hierarchy tool, discussed in Chapter 4.5). Alternatively, the tool may either parse some dependency rules or invoke sorting algorithms to order the goals based on the analyst's input. To address the different roles played by agents sharing responsibility for goals, two operational definitions were constructed and specified in the requirements document. Since there were only two agents in GBRAT, the user and the system, each GBRAT functional requirement has a *user-operational definition* and a *tool-operational definition*. Example 6.1 discusses the two operational definitions for the GBRAT goal `Specify agent`.

Example 6.1 (GBRAT functional requirement: Specify Agents) This goal requires a division of labor due to the shared responsibility between the user and the tool for its completion. The user must specify the agent responsible for the completion of each goal. The system must make the goals selectable to the user and display any previously specified agents associated with the selectable goals. The user, in turn, must select a goal and determine the responsible agent(s) for the selected goal. Thus, the user is responsible for selecting goals, determining the responsible agent(s), and specifying those agents. The tool is responsible for presenting the goals and agents, making them selectable, and then storing the responsibilities which the user specified for that goal. This separation of responsibilities in the operational definitions makes the role of each agent much more clear for the design and implementation of the system.

Inquiry surfaces implementation issues and alternatives

Several questions pertaining to implementation-specific issues were raised during the GBRAT case study. One of these issues addressed the need for a flag to indicate unresolved issues that require discussion among the analysis team and/or stakeholders. If there are several conflicting issues related to a specific goal, these issues must be managed by the tool. Possible alternatives were identified through an inquiry-driven approach. For example, one alternative posed during the investigation suggests that if one person has the authority to make the final modification based on a set of assumptions, a configuration management flag may not be needed since only one person/user may make modifications to the project repository. However, if multiple users will share responsibility for a project repository, the implementation must support auxiliary notes and provide a mechanism for both tracking revisions and unresolved issues. Due to the design and implementation ramifications, both of these options require decisions; the analyst must discuss the actual goals with the stakeholders since the decision will ultimately affect the systems' implementation.

Constraints point to questions which should be directed to the stakeholders

Constraints extracted from the method overview pointed to follow-up questions which the analysts should ask the stakeholder. Consider the constraint: `Goals are identified from two sources: textual statements of need and descriptions or models of current processes`. During the analysis of this GBRAT constraint, the goal was annotated with two questions: *Are you developing a completely new system?* and *Is a current system in place but in need of repair?* These questions may be generalized since they are not specific to one application. The answers to these questions will help the analyst determine

if there are, in fact, other sources of information available which the stakeholder(s) may have overlooked.

Discussion

Goal schemas were constructed for 45 GBRAT goals. However, the requirements document focused solely on the Goal Editor Module; thus, only the goal editor requirements and schemas appear in the requirements document. Nineteen functional requirements, 14 user operational definitions, and 14 tool operational definitions were identified for the Goal Editor module.

Requirements produced using the GBRAM enable effective development of a software system. The efficacy of a software method is best tested in practice; examination of the resulting artifact assist analysts in determining effectiveness and compliancy with the stated objectives. The requirements document produced using the GBRAM enabled the analyst to determine functional modules for the proposed system and translated the goals identified for these modules into operational requirements for a Goal Editor. Another analyst, the developer, was able to use the requirements document as the basis for the production of the GBRAT design document and for the construction of the GBRAT prototype. Thus, the GBRAM facilitated the development of the proposed system which was specified by employing the GBRAM.

While the requirements specified focused on *what* the system must do, the operational definitions which support the requirements favored *how* the system could work (e.g. design elements). While this facilitated the design phase, the analyst experienced the lack of

support GBRAM offers for information modeling. Since GBRAT is an information-intensive system, the ability to better model information requirements would have been beneficial to the study. Additionally, the lack of high-level organizational goals in this analysis was a limiting factor in this study; thus, while organizational issues and high level objectives played a role in the analysis of the FSO and CTTS studies, the GBRAT case study lacked the ability to develop more non-functional and organizational requirements.

The following case study was conducted due to the need for validation of the method's ability to handle high-level enterprise goals and the method's scalability for systems supporting a large organization comprised of multiple users holding conflicting goals.

6.2 CommerceNet Web Server

The objective of this case study was to further validate the GBRAM by broadening the base of practical experience through application of the method to an Intranet electronic commerce application. Application of the method to a large commercial application provided insights into how goals are used to identify and refine system requirements as well as the applicability of the method's strategies in reengineering efforts involving teams of analysts.

The electronic commerce World Wide Web (WWW) server case study involved the reengineering of a Web server which supports various consortium member organizations participating in electronic commerce over the Internet. The Web server, hereafter referred to as CommerceNet, must support secure payment and transactions, different access levels, membership and seminar registrations, and project and proposal status tracking. The

CommerceNet study also explores the issues which arise when applying the GBRAM in a collaborative effort. This section summarizes the knowledge garnered from using the GBRAM in the specification of requirements for a commercial web-based application.

Methodology and Case Study Artifacts

The CommerceNet case study was conducted by four analysts, the author, and various stakeholders, for approximately 30 hours a week over a period of four months. The author of this thesis served as the principle analyst conducting weekly video conference meetings over the MBone* with a group of three to four stakeholders in the EColabor room at NTT Multimedia Communications Laboratories in Palo Alto, California. These meetings were each one to three hours in duration. Some additional meetings were conducted with larger groups of stakeholders (CommerceNet consortium members). The meetings were primarily goal elicitation sessions during which the goals of the system were fleshed out and specified collaboratively using tools such as vic, vat, and whiteboard.

The principle analyst produced a requirements document based on the elicited goals; the document was comprised of six major sections pertaining to the six functional areas within the CommerceNet server. Each of these sections contains four subsections: Goals, Functional Requirements, Nonfunctional Requirements, and Organizational Requirements. The goals are the objectives of the system. The functional requirements are the behavioral descriptions of the system and thus define what the system does. The nonfunctional requirements define the attributes of a system, such as the levels of efficiency, reliability, se-

*MBone stands for the Multicast Backbone on the Internet and provides audio and video connectivity.

curity, maintainability, portability, visibility, capacity, and standards compliance. Although reporting procedures do not belong in a requirements document, they were included in the CommerceNet requirements document in the form of organizational requirements due to their affect on the continued analysis of the system. The main analyst requested elaborated scenarios from the CommerceNet analysts and stakeholders for those goals which were not readily understood. Goals were elaborated on a scenario-by-scenario basis.

The CommerceNet Web Server requirements document was made available to stakeholders via the WWW using HyperMail. In contrast to the GBRAT requirements document which utilized hypertext to support requirements traceability, the CommerceNet requirements document employed hypertext links to support requirements discussions and to capture auxiliary notations. The use of auxiliary notations is discussed below within the context of lessons learned in the CommerceNet case study.

Lessons Learned

This section summarizes the lessons learned from the electronic commerce web server case study.

Stakeholders who are well-trained analysts may express goals in terms of functions

Heuristic HIG 9. (page 149) discusses stakeholders' tendencies to express requirements in terms of operations and actions; during the CommerceNet case study, stakeholders expressed requirements in terms of functions that the system had to perform. During the first video conference meeting, which lasted for 1 hour and 45 minutes, four participants

(two analysts from Georgia Tech and two analysts representing CommerceNet) met for a brainstorming session. During this session the CommerceNet representatives expressed their goals for the system in terms of the functions the system must support; after some initial brainstorming, the functions were categorized into four broad categories (shown in Table 6.2). The function categories ultimately served as the four CommerceNet goal classes previously discussed in Chapter 3. Since functions correspond to actions or behaviors within a system, they were easily expressed as goals by the primary analyst at a later date.

Table 6.2. CommerceNet Function Categories

Function Category	Number of Functions Identified
Information Display & Organization	14
Process Support	11
Usage Levels (public or member)	9
Electronic Commerce	4

Several factors contributed to stakeholders' expression of system goals in terms of functions. The objective of the CommerceNet Web Server project reflected in the terminology used by stakeholders was the reorganization and redesign of an existing system. For the CommerceNet example, stakeholders were asked *how-to* questions as the requirements evolved. One of these questions was *"How is the public key entered on the application form?"*; the stakeholder responded *"Public keys are loaded in."* This resulted in a new goal `MAKE public key loaded in`. The use of computerese by the stakeholders was quite common in this case study because the stakeholders assumed the use of a medium/mechanism for the goals with which they were well versed and experienced. Had such a goal arisen in,

for example, the FSO, the analyst would have needed to abstract away from said terminology to ensure that the goal vocabulary employed was easily understood by all stakeholders. In another example of a *how-to* question which surfaced the need for new goals within the system, stakeholders expressed that membership kits are sent to CommerceNet users. The follow-up question asked by the analyst was *"How is the membership kit sent to the user?"* the stakeholders responded "Via email."

Stakeholders almost always refer to processes they want operationalized in the system as "automatic." In the CommerceNet case study, stakeholders identified several goals needing to be performed "automatically" by the system:

- `Web server history changed automatically`;
- `Web resources automatically rated by key word occurrences, accesses, and what's hot`; and
- `What's New automatically generated.`

Such goals imply "knowledge" on the part of the system. It was apparent throughout this project that the CommerceNet Web server was the responsible agent for all "automatic" goals as well as all "know" goals. "Know" goals pertain to states or conditions which the system must be able to distinguish or recognize (e.g. `KNOW user edit-authorization level`). This technique for the identification of automatic and know goals' responsible agents may simplify analysts' identification of those goals for which the system is responsible in other analysis efforts.

Stakeholders may be hesitant to focus on agents

During this study, participating stakeholders were initially hesitant to focus on agents due to their belief that the responsible agents would change as a result of the redesign and reengineering of the CommerceNet Web server. Of particular note, is the proximity of the specific/detailed goals to the operations which will be operationalized in the CommerceNet system; these goals will be performed by people, CommerceNet, or the server/system. As with most reengineering efforts, the primary focus is on the process(es), not on the people, since the employees involved with the system may change as a result of the newly designed process. However, since there were subtle policy issues which were only understood by the persons currently responsible for functions and goals, it was important to identify those persons in the event that follow-up interviews were needed later on. Also, if the newly designed system will alter stakeholders' work, it is important to track the people who will ultimately be affected.

Goals are named using a verb as the first word

Heuristic HIG 1. (page 144) states that goals may be named in a standard subset of natural language in which the first word is a verb describing the kind of goal being named. This naming convention, adopted during the CommerceNet study, facilitates the ordering of goals based on the knowledge that certain kinds of goals may have a pervasive affect on other goals. In this case study, all CommerceNet goals begin with verbs formed from the following set of words: `AVOID`, `ENSURE`, `IMPROVE`, `INCREASE`, `KEEP`, `KNOW`, `MAINTAIN`, `MAKE`, and `SPEEDUP`. Table 6.3 provides an example of each of these words which describe the kind of goal being named. `AVOID` implies a state which must be prevented within

the system. `ENSURE` refers to making certain that a particular state is achieved. `IMPROVE` is a 'quality' goal (discussed in Chapter 5, page 161) and which implies bettering the quality of some portion of the system and/or organization and increasing some level of productivity. `INCREASE` goals concern the amount or rate by which something is increased or made greater. `KEEP` implies the continuation of some state or event within the system at a steady level or pace, and may refer to saving information or maintaining some state within the system. `KNOW` implies the ability to distinguish or recognize some state within the system. `MAINTAIN` implies the provision for or sustainment of some existing condition. `MAKE` implies the formation or attainment of some state within the system. `SPEEDUP` implies the acceleration of production.

Table 6.3. Examples of CommerceNet Goal Verbs

Goals	Goal Type
AVOID obsolete information	Maintenance
ENSURE secure transactions	Maintenance
IMPROVE content maintenance and administration	Improvement
INCREASE profits from seminars	Improvement
KEEP soliciting participation	Maintenance
KNOW member access privileges	Achievement
MAINTAIN two servers	Maintenance
MAKE member registered	Achievement
SPEEDUP time required to procure approval for modifications/updates	Improvement

Maintenance and achievement goals need to be differentiated

In the CommerceNet server, maintenance goals are basically high-level goals; the associated achievement goals should comply with the maintenance goals. It was important

to differentiate between maintenance and achievement goals since a majority of the discussions which transpired with the stakeholders addressed "maintaining the server" and "content maintenance"; while the conversational references regarding server maintenance goals and meta-level goals were clear to the analyst, the were not clear to the participating stakeholders.

It is possible that an achievement goal may share a relation with more than one maintenance goal. Consider the CommerceNet achievement goal: `What's New filtered according to member's personal preferences.` This goal shares a relation with two maintenance goals: `Personal preferences managed` and `User-level privileges enforced.` Thus, while it is beneficial to differentiate between achievement and maintenance goals, it is also important to consider the relationships which may exist among the goals.

Auxiliary notes help analysts track unresolved issues & requirements discussions

Within each subsection of the CommerceNet requirements document, various classifications of auxiliary notes appear, serving to document requirements discussions [61, 64]. *Questions* are reminders of unresolved issues pertaining to a particular requirement. *Answers* describe solutions or provide a clearer understanding of the requirements; when a question generates more than one answer, the answers are listed as *Alternatives*. *Reasons* provide justification for answers or requirements which are not immediately obvious. *Scenarios* serve to document issues and elaborate the requirements. At times a requirement or auxiliary note is followed by a parenthetical reference; some requirements address several functional areas and meet several goals, while the parenthetical cross-references imply that the same or substantive similar requirements are found elsewhere in the document. An

annotated requirements document allows analysts to list unresolved questions. The document is interspersed with unresolved issues and questions which serve as reminders for the analyst. Items requiring an answer or decision are explicitly flagged to expedite resolutions.

Discussion items assist in the development of requirements

An overriding objective of this thesis is to make the GBRAM and the inquiry process useful for many people by discussing the kinds of goals identified, the kinds of discussion items used, and how the inquiry cycle facilitated the process of formulating a system. This thesis does not attempt to impose unusable descriptive methods on analysts. Throughout the CommerceNet study, memos pointed to some content and subsequent actions (i.e., construct a scenario, answer a question, or explore a constraint). Each of these may appear in a scenario; by capturing and tracking memos, the analyst is essentially constructing a "memory" for the project. Another kind of memo used in the CommerceNet study is constraints*. During the early stages of analysis, constraints assume the form facts about how the system (CommerceNet) works and its possible relationship with other systems. Writing memos during the analysis process may be likened to incorporating a procedure into a system which reminds analysts to go back, check, and review.

Scenarios point to implementation alternatives

In the CommerceNet study, scenarios surfaced alternative implementation options which require a decision on the part of either the stakeholders or the analyst. For example, one scenario prompted the analyst to ask the question *"How are queries to be submitted?"*.

*Constraints are considered memos at the early stages of analysis because they have not yet been refined.

Stakeholders responded, "Via FORMS, but email is desirable as well; however, FORMS is a priority." This led to two possible scenarios: a query submitted via FORMS and a query submitted via email. Given these specifications, the system must eventually be able to handle these different negotiation protocols. If the system is driven by email protocols, the recipient is required to become a data entry user; this need for the recipient to enter data upon receipt of email would be by-passed by a FORMS implementation. Scenarios enable analysts to identify alternatives and consider the corresponding behaviors which the system must exhibit. In addition to surfacing implementation alternatives, scenarios point to policies which affect other goals; this was especially apparent when identifying policy-oriented scenarios affecting other goals for the goal `MAKE member registered`. For example, the scenario "Only sponsor members can vote" affects the goal `ENSURE voting supported`, which is a process support goal. Thus, the voting goal was elaborated with this scenario.

Scenarios facilitate the identification of new goals

Scenarios were used extensively in the CommerceNet case study. There are numerous examples of goals which were identified via scenario analysis. This form of identifying goals has been thoroughly discussed in this dissertation. Given the initial set of CommerceNet functions, the principle analyst identified several which were a bit vague, requiring clarification. The stakeholders were thus asked to elaborate the following five scenarios:

- Processing membership fees;
- Purchasing seminar video;
- Approving proposal;
- Finding information; and

- Making proposal.

The stakeholders elaborated each of the scenarios by listing the different activities for which agents are responsible. The first scenario "Processing membership fees," was comprised of the following 12 actions:

```
 1. User:                finds the membership application form page
 2. User:                fills out the membership application form
 3. User:                selects e-check as payment method
 4. User:                types in public key
 5. User:                submits the membership application form
 6. Certification        approves user payment
    Authority
 7. CN Server            responds to user with receipt
 8. CN Server            increases budget balance
 9. CN Server            creates user's entry in member database
10. CN Server            adds user to member mailing list
11. CN Server            adds user to member web page
12. CN Server            sends user membership kit
```

These actions were then stated using the naming conventions suggested by HIG 1. (page 144) and HIG 2. (page 144). Further analysis using the inquiry process suggested in the GBRAM's goal identification heuristics resulted in the identification of 12 goals. For example, consider the goal `MAKE payment method selected` in Table 6.4 which shows the obstacles and scenarios which correspond to this goal.

Table 6.4. CommerceNet Goal: MAKE payment method selected

Goal	Obstacles	Scenarios
MAKE payment method selected	1. Payment method not selected 2. Payment methods not clear	1. User selects e-check as payment method 2. George isn't sure if Burdell & Assoc. has an account set up yet & needs to know how to get one

Obstacles #1 and 2 indicate the users' need to select from various payment options, such as check, money order, or credit card. Additional goals were identified through the consideration of possible scenarios. For example, consider Scenario #2. George is an employee at Burdell & Associates. Before he selects a payment method, he must access his firm's CommerceNet Membership Web page to obtain the information he needs to select his firm's preferred payment method. This "walk through" approach was employed throughout the CommerceNet study and was helpful in the identification of goals.

When actions appear as instantiations (e.g. e-check vs. payment method), the goal is named in general terms (e.g. payment method) and the action is listed as a scenario. Having concrete scenarios aids analysts and stakeholders in considering other possible concrete scenarios. For example, given e-check as a possible payment method, other stakeholders are prompted to express their preferred payment method (e.g. credit card).

Constraints are often preconditions which may be expressed as goals

During the CommerceNet case study, conditions which could prevent a goal from being achieved were initially expressed as constraints. However, during goal elaboration, it became apparent that some of these constraints are actually preconditions. For example, consider goals G_{54} and G_{55} in Table 6.5. Both of their respective constraints are actually preconditions; they represent or illustrate states that must be achieved before the goal can be completed. Therefore, these constraints served as pointers to new goals for the system. In contrast, consider the constraint for goal G_{56}. This constraint restricts achievement of the goal by limiting the ability to purchase a videotape to a selected population of authorized persons. Thus, the constraint associated with G_{56} remains a constraint in the CommerceNet

Server goal set. Heuristic HIC 2. (page 159) suggests searching for temporal connectives to identify constraints. In CommerceNet, it was observed that `Register for seminar before deadline` indicates a constraints on the goal `MAKE member registered`.

Table 6.5. Constraints, Preconditions, and Goals

Goals	Constraints
G_{54} : MAKE on-line statement available for each organization	Member must be authorized to access on-line statement
G_{55} : AVOID duplicate purchases	Member must be able to ascertain whether his organization previously purchased the desired item
G_{56} : MAKE video tapes purchased	Only CN members & seminar participants may purchase seminar video tape

Goal Refinement

Goal refinement concerns the way goals change from the moment they are first identified to the moment they are operationalized in a system specification. Figure 6.1 shows the evolution of the CommerceNet goals into operational requirements. As shown in the figure, 18 high level goals were initially elicited from the stakeholders during the meetings conducted over the MBone. Subsequently, 54 goals were derived from the initial set of 18 by applying the GBRAM. The ovals in the figure represent the elaboration phase in which the inquiry cycle [61,64] was employed. The resulting requirements document for the electronic commerce web server contained a total of 79 requirements. The oval on the left side of the figure illustrates the number of questions, scenarios, alternatives, and answers generated which led to the specification of 52 functional requirements. The oval on the right side of

the figure corresponds to the inquiry process which led to the specification of 27 nonfunctional requirements. As illustrated in the figure, the inquiry cycle led to the specification of nearly twice as many functional requirements as nonfunctional requirements. This is due mainly to the techniques employed in GBRAM, their strength lies in forcing analysts to systematically consider the behavioral aspects of systems by focusing on obstacles and scenarios.

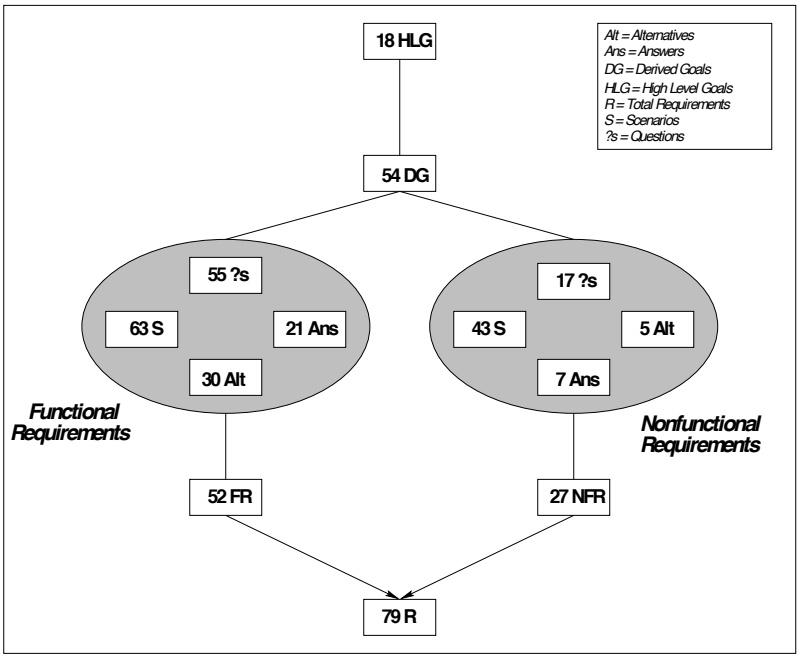

Figure 6.1. Evolution of Electronic Commerce Requirements

The CommerceNet requirements document was not traditional in that it was not composed solely of requirements. Instead, each requirement in the document was annotated with the relevant questions, answers, alternatives, and scenarios that arose during application of

the inquiry-driven approach. Moreover, scenarios pertaining to processes and issues that were nebulous or not well understood were elaborated by specifying the possible sequences of actions for that scenario. This was beneficial in that the requirements document became a "living" document which could be easily annotated by the stakeholders. Any item (i.e., a requirement or a scenario) may be the target of several annotations. While annotation support via HyperMail in this case study was certainly helpful for tracking the discussion elements, paper is adequate if annotation mechanismsare not available. The critical issue is tracking the rationale associated with specific decisions and flagging any unresolved issues so that they may be discussed among analysts and stakeholders and targeted for resolution.

Summary and Discussion

The CommerceNet Web server is used by a global consortium which consists of many international companies. The primary objective of CommerceNet is to change the way in which customers, service providers, and developers participate in business transactions and to facilitate interactions and collaborations between these parties. Users are provided access to all CommerceNet information and applications via the WWW.

In the CommerceNet case study, goal tables supported the actual production of requirements. At the early stages of analysis, scenarios provided glimpses of current processes and future iterations. The goal tables initially identified scenarios primarily as notes to the analyst indicating possible scenarios. For example, the goal `Different entrance to server supported for each user level` has two subgoals: `Public entrance to server supported` and `Member entrance to server supported`. The corresponding scenario pro-

vided by one of the stakeholders was: "As a Working Group chair, Kenji has access to more content," which implied that there were issues involving access to information. By constructing this scenario, it was possible to reason about what constrains this behavior and, as analysis ensued, the scenarios which were elaborated were those for which the analyst had not yet developed a thorough understanding. The goals, obstacles, scenarios, and associated annotations were captured in tables for the analysis by utilizing both Microsoft Excel spreadsheets and GBRAT. The initial brainstorming results were analyzed using GBRAT, while subsequent analyses employed Excel due to the ability to expedite ordering goals, formatting and printing the goal tables, and scrolling the tables.

The CommerceNet project involved reengineering an existing Web server. The stakeholders were experienced with the existing system and were also experienced analysts. This introduced new challenges in applying the GBRAM. Experienced analysts and stakeholders were eager to jump to discussions surrounding functions; thus it was important to ensure that everyone objectively analyzed the relevant goals. This was used as a form of early validation (e.g. "Are we building the right product"). Further studies with experienced analysts may show that some people are sufficiently comfortable with the system and its functionality to warrant by-passing goals. However, for this analysis, it was imperative that the analyst always keep an eye on the overall objectives and goals to allow for the cohesive integration of functions. Additionally, the stakeholders worded the goals using computerese which often referred to actual functions within the system. The analyst is charged with managing these facets of the projects.

In constructing the CommerceNet goal hierarchy, the goals were organized according to the four goal classes shown in Table 6.6.

Table 6.6. Electronic Commerce Web Server Goal Classes

Goal Class	Description
Process Support	Process support goals involve the underlying functionality of the system as needed by the users. This class of goal was observed in the Electronic Commerce System.
Security and Access Goals	Security and access goals restrict access to certain parts of the system to authorized users. This goal class was observed in both the Vacation/Sick Leave System and the Electronic Commerce System
Electronic Commerce Goals	Electronic commerce goals involve transactions that take place within an Intranet and over the Internet.
Information Display and Organization Goals	These goals pertain to how information is organized throughout an information system and how the information is displayed for the users.

`Proposal voted on` is a *process support goal*; `Read/write access controlled` is a *security and access goal*; `Member billed for membership fees` is an *electronic commerce goal*; and `Search results Web page displayed` is an *information display and organization goal*. All of the identified CommerceNet goals fit into one of the four classes detailed in Table 6.6. The organization provided by the goals facilitated the organization of goals into naturally different functional requirements.

The CommerceNet case study further validates the GBRAM by demonstrating its successful application to a large commercial internet application. The method facilitated the identification and refinement of goals into operational requirements which guided a team of analysts throughout the process. The following section discusses further validation of the GBRAM; the validation process discussed sought to determine whether GBRAM allows for better identification of system requirements as compared to a commonly used method.

6.3 An Empirical Evaluation of the GBRAM

In this investigation, the Financial Services Office (FSO) is a real organization and the requirements are for a vacation/sick leave hour tracking system. The hypothesis for this empirical evaluation is that GBRAM allows analysts to better identify requirements than does another commonly used method or the lack of a method. Additionally, interest is also expressed in the difference in the nature of the requirements. This section provides an overview of the experimental method employed and discusses the results of the investigation.

Experimental Method

This experiment contrasts the use of GBRAM for the specification of information system requirements with one other analysis method, OMT [75], and a control condition in which no method was stipulated. This section explains the experimental method employed for this validation effort.

Design

The subjects were subdivided into three groups, each of which was asked to use one of the following methods:

- the Goal-Based Requirements Analysis Method (GBRAM);
- the Object Modeling Technique (OMT) [75]; or
- an analysis method of their choosing.

The independent measures involve two groups using different analysis methods and a third control group. The assignment materials supplied were analogous, but the subject groups differed in the assigned methodology.

Subjects

The subject population was a group of Georgia Tech undergraduate students enrolled in a senior level Management Information Systems course who were given the opportunity to voluntarily participate in an experiment for which they would be compensated with extra course credit. The subjects performed a problem analysis followed by a requirements analysis for the vacation/sick leave hour tracking system. Subjects were informed of the purpose of the study and assured that their identities would be concealed in the final report.

Measures were taken to ensure that the experts in a particular analysis method were not over-represented in any one condition. Questionnaires were administered to the students regarding their expertise level with different analysis methods; students were then assigned to the three conditions in a balanced fashion. Allocation was random within the expertise level. Although the students were initially subdivided into three groups with 8 students each, different participants dropped out of the study, resulting in condition groups with varying numbers of subjects (7 GBRAM subjects, 6 OMT subjects, and 4 CONTROL subjects). A total of seventeen subjects completed the analysis for this research.

Materials

Since this experiment entailed a two-part study, the materials* used for the experiment include two sets of detailed instructions. The first set consists of instructions for the problem analysis assignment for each of the three conditions; the second set consists of instructions for the requirements analysis assignment for each of the three conditions.

The instructions for Part One explain the objective of the study, provide a description of the problem, and present a step-by-step overview of the analysis method, providing a detailed example of how to apply the respective methods. The instructions for part two explain how to write a requirements document and how to translate the results of the problem analysis into requirements. The instructions for both parts explain several condition-dependent† ways to construct and represent software requirements.

The vacation/sick leave hour tracking system involves the process of monthly submission and tracking of employee vacation, sick leave, and consulting hours at an academic institution. Currently, all absences are either submitted in writing or via email to the Financial Services Office (FSO), which in turn generates various monthly and yearly reports using the submitted data. A new system is desired which supports employee hour submission and report generation for the FSO office.

This problem was well suited for our empirical analysis because the system involves various types of stakeholders with conflicting needs (e.g. faculty, full-time and part-time staff, and FSO employees responsible for processing the data and records responsibly) and it displays properties that are characteristic of information systems such as security, messaging,

*A copy of the materials is available from the author upon request.
†Condition: GBRAM, OMT, or CONTROL.

and report generation. Additionally, the scope of the system was sufficiently self-contained for a controlled experiment, not requiring a major time commitment of the participants in the study.

The GBRAM subjects were provided with blank spreadsheets and worksheets to facilitate capturing and tracking the number of identified goals. The OMT subjects were provided with blank worksheets to create a data dictionary, object models, and dynamic models. The control subjects were provided with blank worksheets for a table of definitions, flow charts, and a two-page handwritten description of the properties of the problem.

Procedure

The experimenter introduced the assignment in class and explained how to use the provided worksheets. The assignment was self-paced; subjects were free to work as much as they wished, completing the study as a homework assignment without supervision during a one-week period. For reasons beyond the control of the experimenter, some subjects were not able to start the assignment immediately; thus, the experiment continued for several weeks. Subjects were instructed to avoid discussing the assignment with other participants in this study.

The assigned task was a week-long effort. The assignment was presented in two parts: a problem analysis assignment and a requirements analysis assignment. Students were asked to sign a consent form and were provided with a five-page summary of the problem and the method to be employed. The assignment included two interview transcripts, one with a professor and the other with an FSO staff member. Thus, subjects had three sources of information available to two them: two interview transcripts and the problem description

in the assignment. Table 6.7 summarizes what subjects were required to produce for each of their respective conditions. The problem analysis assignment required the subjects to produce different method-dependent artifacts. The requirements analysis assignment resulted in the production of a requirements document, not to exceed 3 pages in length.

Table 6.7. Artifacts Produced by Subjects in Each Condition

Condition	Artifacts (Part 1)	Artifacts (Part 2)
GBRAM	Goal tables (goals, scenarios, obstacles) Elaborated scenarios (3) Goal hierarchy	Requirements document
OMT	Data dictionary Object model Dynamic models (3)	Requirements Document
CONTROL	Table of definitions Two-page description of the problem properties	Requirements Document

Subjects in all three conditions first read an introduction to and explanation of the vacation/sick leave system. They then read instructions for how to perform the problem analysis for the system. Results of their analysis were recorded on the provided spreadsheets and/or worksheets (e.g. GBRAM: goal tables, goal hierarchy; OMT: data dictionary, object model, dynamic model; and control: table of definitions, flow charts) and the assignment turned in at the beginning of class. By the end of that class period, the submitted problem analysis worksheets were copied and the original worksheets returned to the subjects with part two of the study, the requirements analysis assignment. Subjects were allotted a half-week to complete the requirements analysis assignment. For this second part, subjects were

asked to use a word-processor to compose their requirements documents.

A pilot study was initially run in a graduate level software engineering course. The requirements produced during this initial test were given to an independent judge* to rank the requirements on the basis of critical importance. Critical requirements are those requirements that cause a system to fail if not met; they established a threshold for a minimal set of requirements for the system. Of the 185 requirements identified by the three condition groups in the pilot study, 81 were ranked as critical.

Measurements

Some aspects of the experiment that may be compared across the three groups of subjects are:

- effort;
- total number of requirements and number of critical requirements expressed; and
- total number of requirements of different classes and number of critical requirements of those classes identified (detailed in Table 6.8 as the goal classes of interest in this study).

The effort expended cannot be analyzed for several reasons. First, the time required to use a method with which one is unfamiliar will vary greatly in comparison with a subject using a method with which they are an expert. Second, although the subjects were asked to keep track of the amount of time they spent on each phase of the analysis, informal interviews with subjects and an analysis of the submitted time data indicates that many

*The judge is a major stakeholder, not one of the previously interviewed stakeholders, as well as an expert in the application domain.

subjects merely jotted down a time for each phase as an after-thought, not taking the need for this information seriously.

Table 6.8. Vacation/Sick Leave System Goal Classes (See Chapter 4.2 for rationale)

Goal Class	Description
Calculation Goals	Calculation goals involve mathematical calculations. These may be in the form of accounting, accrual rates, balance, etc., as seen in the Vacation/Sick Leave Hour Tracking System.
Messaging Goals	Messaging goals pertain to notifications and reminders sent by and/or within the system. This goal class was observed in the Vacation/Sick Leave System.
Report Generation Goals	Report generation goals pertain to the generation of reports for an organization or enterprise. In the Vacation/Sick Leave System, the report generation goals involved the generation of internal departmental reports as well as external institute summary reports.
Security and Access Goals	Security and access goals restrict access to certain parts of the system to authorized users.

Results

This section discusses the results of the empirical evaluation of the Goal-Based Requirements Analysis Method. We employed the Mann Whitney U non-parametric test [68,77] because of the small sample sizes and the large and possibly non-normal inter-subject variances. The resulting computation of the significance tests are shown in Table 6.9.

Figure 6.2. Total Number of Requirements Identified (Ordered by subjects in the three conditions). $GBRAM : OMT$ ($U = 11$, $p > .05$) and $GBRAM > CONTROL$ ($U = 4$, $p < .05$).

Figure 6.3. Number of Critical Requirements Identified (Ordered by subjects in the three conditions). $GBRAM > OMT$ (Mann Whitney $U = 6$, $p < .02$).

Of particular note are the results concerning the messaging requirements, shown in Figure 6.4. GBRAM subjects identified significantly more messaging requirements than the subjects in either of the two other conditions (GBRAM:OMT Mann Whitney $U = 6$, $p < .02$ and GBRAM:CONTROL $U = .5$, $p < .01$). An example of what is meant by "messaging requirements" is illustrated in Figure 6.5. Two obstacles are shown for the goal `Vacation/Sick leave hours collected`: `Hours not submitted` and `Employee never`

Figure 6.4. Number of Total Messaging Requirements Identified (Ordered by subjects in the three conditions). $GBRAM > OMT\ U = 6$, $p < .02$ and $GBRAM > CONTROL\ U = .5$, $p < .01$.

In order to better evaluate the performance of the GBRAM subjects who were using the method for the first time, an expert (the author) also performed the problem and requirements analysis using the exact same materials utilized during the controlled study. Figure 6.6 illustrates a comparison between the GBRAM experts' performance compared to the average performance of the subjects in the experiment. As may be expected, there is a great performance difference between the expermental sets; the expert out-performed the experiment subjects in all points of comparison.

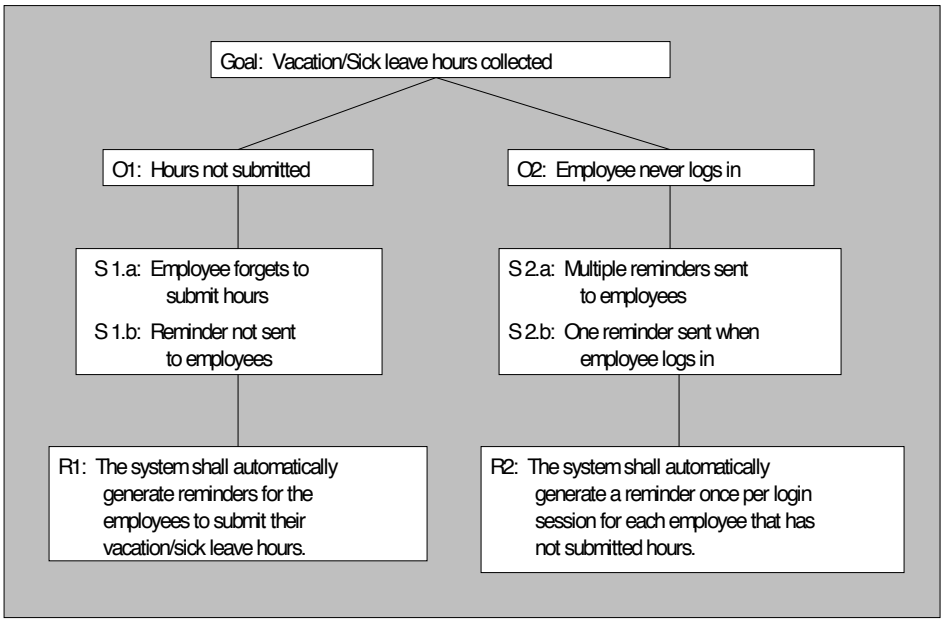

Figure 6.5. Messaging Requirements Example for Vacation/Sick Leave Hours Tracking System ($O = obstacle$, $S = scenario$, and $R = requirement$)

Figure 6.6. GBRAM Expert vs. GBRAM Subjects (Average)

Since the difference in performance observed between the GBRAM expert and the average of the subjects was so vast, an additional comparison was made between the GBRAM

*The concept of derived and synthesized goals was introduced by W. Michael McCracken during a discussion with the author of this thesis at Georgia Tech.

Figure 6.7. GBRAM Expert vs. Best Performing GBRAM Subject

The interest in tracking whether a goal is derived or synthesized is two-fold. This differentiation enables those goals which were simply extracted from available information sources to be distinguished from those goals which were uncovered as a result of an in-depth inquiry-driven analysis. Further investigation of the possibility of forcing traceability

from synthesized goals to derived goals thereby establishing reasonable stopping criteria for analysts is also an objective. Although collection of data for such an analysis has begun, analysis and further collection of data is left for future work, as discussed in Chapter 7.

Summary and Discussion

This study demonstrated the feasibility of employing the GBRAM as a reasonable analysis method, and is supported by similar identification of requirements for the proposed system across the GBRAM and OMT conditions.

Since the critical requirements are considered 'important', we can conclude that from this study that GBRAM performed as well as or better than OMT and the CONTROL conditions in identifying the important requirements. The most meaningful results in these studies are observed for the messaging requirements. GBRAM subjects out-performed both OMT and CONTROL subjects in the total number of identified messaging requirements, and out-performed OMT in the number of identified critical messaging requirements.

GBRAM relies on obstacle and scenario analysis to uncover exceptional behaviors in systems. Since scenarios are descriptions of sequences of behaviors, it naturally follows that they would yield behavioral requirements such as the messaging requirements in the vacation/sick leave system. GBRAM performed at least as well as the other two conditions in identifying goals and requirements for the other three goal classes.

A significant limitation of this evaluation of the method is the size of the subject population. Additionally, further comparisons are needed with other analysis methods which employ the use of scenarios, such as Jacobson's use case approach [48].

6.4 Summary

This chapter has presented three efforts (two case studies and an empirical evaluation) which served as the summative validation for the Goal-Based Requirements Analysis Method. Each case study detailed in this chapter involves a particular system:

- the Goal-Based Requirements Analysis Tool (GBRAT);
- the CommerceNet Web Server; and
- the Vacation/Sick Leave Hour Tracking System.

The Goal-Based Requirements Analysis Tool provides analysts with the procedural support needed for the analysis and refinement of goals for software-based information systems, supporting and guiding analysts as they identify, capture and structure information in the form of goals. This case study validated the efficacy of the GBRAM to acquire specific requirements for a tool to support the method itself, but limited in size and scope of the problem to which the GBRAM was applied. The CommerceNet Web Server case study validated the GBRAM's ability to handle enterprise and functional goals and the method's scalability for systems supporting a large organization comprised of multiple users holding conflicting goals. The empirical evaluation involving the Vacation/Sick Leave Hour Tracking System demonstrated the feasibility of employing the GBRAM as a reasonable analysis method, and was supported by similar identification of requirements for the proposed system across the GBRAM and Object Modeling Technique conditions. Table 6.10 summarizes the findings of each of the validation efforts discussed in this chapter.

Table 6.10. Summary of GBRAM Validation Efforts

GBRAT	CommerceNet	Vacation/Sick Leave
+ Viewpoints expressed using operational definitions + Implementation issues & alternatives raised using inquiry + Follow-up questions for stakeholder identified via constraints + Straightforward design & implementation by the developer as a result of requirements documentation + Redundant goals easily identified + GBRAM facilitated GBRAT's development	+ Goals easily named using standard subset of natural language terms + Auxiliary notes track unresolved issues and requirements discussions + Discussion items assist in development of requirements + Scenarios surface implementation alternatives + Obstacles and scenario analysis facilitate identification of new goals + Constraints may be refined into goals	+ GBRAM subjects identified more requirements than OMT & control subjects + GBRAM provides better coverage of requirements than does the control condition + GBRAM subjects identified more critical requirements than subjects in other two conditions + GBRAM subjects identified more messaging requirements than subjects in other two conditions + Obstacle & scenario analysis surfaced exceptional behaviors
• Operational definitions which supported the requirements leaned towards design • Requirements focused on system operations		• GBRAM performed at least at same level as other conditions in identifying calculation, report generation, & security/access goals
− An experienced analyst who was also the primary stakeholder & user of the proposed system tested GBRAM − GBRAM does not support information modeling − High level/organizational level goals were not part of this analysis − Only one stakeholder/user was actively involved in the analysis process, despite GBRAT's multi-user capabilities	− Stakeholders may be hesitant to focus on agents	− Size of subject population − Need comparisons with other scenario-intensive methods

A synopsis of each case study in this chapter was followed by a discussion of the methodology and the lessons learned. Discussion of the empirical evaluation encompassed the experimental method, the results, and a discussion of the findings. These case studies served as the source of summative validation, confirming the lessons learned about the GBRAM from the earlier case studies. The next chapter presents the conclusions of this dissertation and discusses future work.