

The Use of Goals to Surface Requirements for Evolving Systems

Annie I. Antón
College of Business Administration
University of South Florida
4202 East Fowler, Ave. CIS 1040
Tampa, FL 33620-7800 USA
+1 813.974.6768
anton@burdell.bsn.usf.edu

Colin Potts
College of Computing
Georgia Institute of Technology
801 Atlantic Drive
Atlanta, GA 30332-0280 USA
+1 404.894.5551
potts@cc.gatech.edu

ABSTRACT

This paper addresses the use of goals to surface requirements for the redesign of existing or legacy systems. Goals are widely recognized as important precursors to system requirements, but the process of identifying and abstracting them has not been researched thoroughly. We present a summary of a goal-based method (GBRAM) for uncovering hidden issues, goals, and requirements and illustrate its application to a commercial system, an Intranet-based electronic commerce application, evaluating the method in the process. The core techniques comprising GBRAM are the systematic application of heuristics and inquiry questions for the analysis of goals, scenarios and obstacles. We conclude by discussing the lessons learned through applying goal refinement in the field and the implications for future research.

KEYWORDS

Requirements engineering, goals, obstacles, scenarios.

1 INTRODUCTION

Obtaining requirements for software systems is a conceptually and practically complex activity. In particular, the amount of communication that is required among stakeholders and analysts often leads to misunderstandings. Stakeholders may forget or not be aware of requirements, and analysts, without the deep domain knowledge of most stakeholders, may not be able to fill in the gaps [CKI88]. In addition, some requirements and constraints may be so obvious to an expert stakeholder that they do not seem worth mentioning, but the thin spread of domain expertise among analyst teams means that these unstated requirements may not be recognized or incorporated into the development process.

Stakeholders seldom understand their requirements fully at the beginning of a project. They often lack a clear vision of what the system should do and change their minds during development [PC96]. In general, stakeholders only become more sure about what they want when they see a system that does not exhibit the necessary features. The desire for early and frequent feedback about concrete system features has led to the adoption of process models that are based on incremental development and risk management [Boe88], participatory design [SN93], and rapid prototyping or scenario walkthroughs [AD93, Dar93, Luq89, Pot95].

All of these techniques address the completeness of requirements: the presence in a specification or other documentation of all the requirements that are intended by the stakeholders or which would reasonably be required after the system was delivered. Informally stated, the problem in increasing the completeness of requirements amounts to going beyond and behind the stakeholders' words to discover the goals that are driving the development process. Requirements that the stakeholders espouse may be detailed implementation plans for more general goals. These goals could also be fulfilled in ways that are inconsistent with the espoused requirements but which would satisfy the stakeholders more. Furthermore, an ascribed goal may necessitate detailed requirements in addition to those explicitly stated.

For these reasons, much recent research into requirements acquisition has investigated the formal refinement of goals [ALR96, AMP94, Ant97, DFv91, Pot95, vDM95]. Goal refinement is intended to reduce the risk of incomplete requirements by explicit consideration of the rationale for espoused requirements and the derivation of other requirements that are implied by this rationale. The emphasis of much of the work on goals has been on using them to derive formal specifications [DFv91, vDM95]. However, the practical consequences of adopting a goal-refinement approach have not been given the same degree of attention, and these consequences are significant.

First, many systems now developed are enhancements of existing, legacy systems. It is tempting when enhancing a system to do so in a piecemeal fashion, specifying a collection of desired features independently and grafting them onto the existing system one-by-one. As Lehman and Belady showed in the 1960s, such accretion of functionality leads to increasing disorder in the system architecture, which, once it crosses a threshold, makes the system unmaintainable without fundamental redesign (see [LB85]). *But is it practical to step back and specify the goals for a system that is already partly implemented?*

Second, stakeholders tend to express their requirements more in terms of operations and processes than goals or objectives [AMP94, Dar93, Ant97]. *Is it practical to go beyond the stakeholders' espoused requirements when these are often stated as concrete behaviors, and derive the requirements instead from more abstract goals? Do the stakeholders even recognize the correspondence between these goals and their espoused behavioral requirements?*

Finally, stakeholders for most systems do not form a homogeneous population of opinion holders. They often find it difficult to agree on requirements, having different viewpoints and needs. Some recent work in requirements engineering has therefore emphasized the formal modeling of viewpoints and support for requirements negotiation [Eas93, FKG89, NKF94, BR89]. But this work assumes that viewpoints are structurally different representations and that the negotiations are negotiations about the priorities of atomic features. In goal refinement, requirements are not lists of features, but are derived from a tangled web of interrelated goals. *Can such goal networks be derived in the presence of multiple viewpoints and do they support the detection and resolution of stakeholder conflicts?*

This paper investigates these questions and demonstrates how goals can be used in the context of a real project with multiple stakeholders. Our contribution is to present the outline of a systematic method for inferring goals from espoused requirements, and deriving more complete requirements from the goals. The Goal-Based Requirements Analysis Method (GBRAM) involves the timely posing of systematic questions (so that requirements can be improved as early as possible), the relaxation of initial goals by considering obstacles (anything that can happen that could thwart a goal), and the exploration of scenarios. The GBRAM is presented via illustrations from a real project, the analysis of the requirements for enhancements to a virtual corporation's web server.

2 GOAL-BASED REQUIREMENTS ANALYSIS

The GBRAM addresses the critical nature of the discovery process in goal analysis. The process of identifying high-level goals is fundamental to the requirements analysis specification process. GBRAM assumes that goals have not been previously documented or explicitly elicited from stakeholders and that analysts must work from various sources of available information, each with its own scope

of knowledge, to determine the goals of the desired system. It also supports the elaboration of goals to represent the desired system. A detailed presentation of how to apply the method from the initial identification of goals to translation of those goals into operational requirements is available in [Ant97]. However, this paper provides a brief overview of the method, differentiating between the goal analysis and goal refinement activities. *Goal analysis* concerns the exploration of available information sources for goal identification followed by the organization and classification of goals. *Goal refinement* concerns the evolution of goals from the moment they are first identified to the moment they are translated into operational requirements for the system specification. These activities are described below.

2.1 GBRAM Analysis & Refinement Strategy

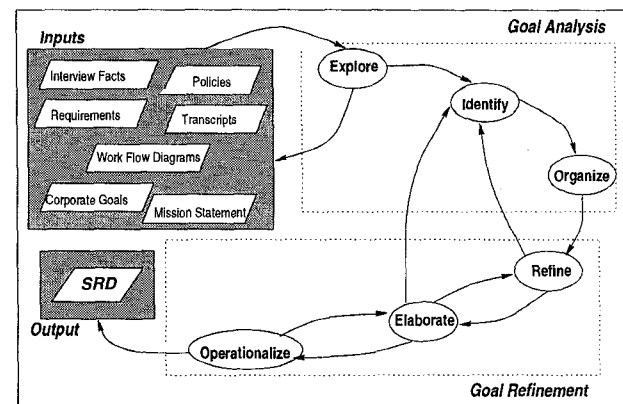
Figure 1 shows the activities which an analyst is intimately involved with when applying the GBRAM. The goal analysis activities may be summarized as follows:

- *Explore* activities entail the examination of available information.
- *Identify* activities entail extracting goals and their responsible agents from the information available to the analyst about the system.
- *Organize* activities involve the classification of goals and organization of those goals according to goal dependency relations.

The goal refinement activities may be summarized as follows:

- *Refine* activities entail the actual pruning of the goal set.
- *Elaborate* refers to the process of analyzing the goal set by considering possible goal obstacles and

Figure 1: GBRAM Activities



constructing scenarios to uncover hidden goals and requirements.

- *Operationalize* refers to translating goals into operational requirements for the final requirements specification.

2.2 Analysis and Refinement Principles

The high-level activities just described provide an overview of the GBRAM. In practice, we adopt several principles for identifying, analyzing and refining goals into requirements. We illustrate and evaluate these in Section 4.

Principle 1: It is possible and practical to infer goals from stakeholders' descriptions of their current processes and their very incomplete statements of behavioral requirements for the new system. Rather than insisting that the requirements acquisition process proceed as if it were a rational process of top-down refinement of abstract goals, our view is that the natural ways of describing existing and desired systems in terms of operational behavior should be built upon and exploited. In particular, we investigate the validation of derived requirements through the construction of scenarios, not the literal inspection of the inferred goals.

Principle 2: Goals can be categorized in several ways, and these classifications are useful in posing questions during the requirements acquisition process. In particular, goals should be classified according to the type of target condition desired (an improvement in some metric, the achievement of a desired state, the avoidance of an undesired state, the preservation of a desired condition, etc.). In addition, goals may be classified according to their subject matter, a classification that has much in common with Jackson's [Jac94, Jac95] idea of "problem frames."

Principle 3: Goals exist at different levels of abstraction. Not only are lower level goals sub-goals of higher level goals; they also may introduce complications and retract simplifying assumptions (i.e., they prune the set of required system behaviors [Fea87]). Our mechanism for elaborating simplified goals is to explore the existence and consequences of "obstacles," conditions that are judged to be likely to thwart a stated goal [Pot95].

Principle 4: Requirements acquisition is a gradual, incremental process that necessarily involves the maintenance of large volumes of informal information [LPR93]. Goal refinement should therefore include methods for keeping track of open issues, reminders to consider recurring questions, illustrations of the rationale for previous decisions, etc. This objective is accomplished by merging goal refinement with the inquiry cycle model of design elaboration [PTA95] and the use of scenarios.

3 ANALYSIS OF THE COMMERCE NET WEB-SERVER REQUIREMENTS

The GBRAM was developed as the result of several action-research projects with real systems [Pot93]. These are documented further elsewhere [ALR96, AMP94, Ant96, Ant97, PTA94]. In this paper, we discuss the GBRAM

strategy and heuristics in the context of the largest of these projects, the analysis of the requirements for a web server for an industry consortium.

The CommerceNet Web server is used by a global consortium which consists of many international companies. The primary objectives of CommerceNet are 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 World Wide Web (WWW).

The requirements analysis effort discussed in this paper involved the reengineering of the CommerceNet Web server which must support secure payment and transactions, different access levels, membership and seminar registrations, and project and proposal status tracking. Application of the GBRAM 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 CommerceNet requirements analysis was conducted by four analysts, the authors, and various stakeholders, for approximately 30 hours a week over a period of four months. The principle analyst conducted weekly video conference meetings over the Mbone (Multicast Backbone on the Internet which provides audio and video connectivity) 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.

Figure 2 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 [PT93, PTA94] 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.

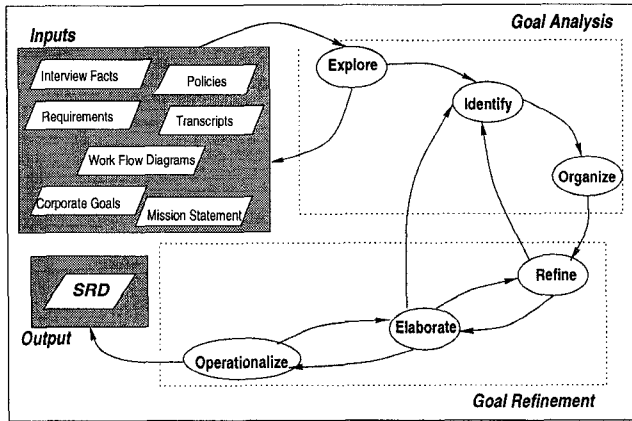


Figure 2: Evolution of the CommerceNet goals into operational requirements.

The principal 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 contain four subsections: Goals, Functional Requirements, Nonfunctional Requirements, and Organizational Requirements. The goals are the objectives of the system. Although reporting procedures do not belong in a requirements document, they were included in the CommerceNet requirements document as organizational requirements due to their effect on the continued analysis of the system. Scenarios for those goals which were not readily understood were provided by CommerceNet analysts and stakeholders. Goals were elaborated on a scenario-by-scenario basis.

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 mechanisms are 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.

The CommerceNet Web Server requirements document was made available to stakeholders via the WWW using HyperMail; hypertext links supported requirements discussions and captured auxiliary notations. The use of auxiliary notations is discussed below within the context of the CommerceNet requirements analysis project.

4 APPLICATION OF HEURISTICS

This section discusses learned during the CommerceNet requirements within the context of the principles asserted in Section 2.

There is insufficient space to describe all the heuristics in this paper (see [Ant97] for full details). There are four general types of heuristics used in GBRAM, which are used to guide the tasks of identification (25 heuristics in total), classification (six), refinement (eight) and elaboration (twelve). Some heuristics are straightforward and generic, not requiring the analyst to employ a specific inquiry technique. Others make sense only in conjunction with specific questions about the system. Table 1 includes a few identification heuristics for goals ("HIG" heuristics) and constraints (the "HIC" heuristic).

(A) Stakeholders may express goals in terms of activities

Heuristic HIG 9 (See Table 1) discusses stakeholders' tendencies to express requirements in terms of operations and actions; during the CommerceNet project, 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: process support, security, electronic commerce, and information display. Since functions correspond to actions or behaviors within a system, they were easily expressed as goals by the primary analyst at a later date.

Several factors contribute to stakeholders' expressing 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. Stakeholders were asked 'how-to' questions as the requirements evolved. When asked "How is the public key entered on the application form?" one stakeholder responded "Public keys are loaded in." This resulted in a new goal MAKE public key loaded in (The use of computerese is quite common because stakeholders assume the use of a medium or mechanism for the goals with which they are familiar.). In

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." The application of the questions which accompany GBRAM heuristics enables analysts to develop a deeper and more complete understanding of the requirements.

In the GBRAM, a system and its environment is represented as a collection of agents. The system is one agent among many, and it is seldom obvious at the beginning of a project what should be its responsibilities (or even, in some cases, whether it should exist). Nevertheless, in our experience, stakeholders almost always concentrate on what they think should be the system's responsibilities and frequently use the adjective "automatic." Because of this tendency, stakeholders may be hesitant to focus on other agents.

For example, in the CommerceNet requirements project, stakeholders identified several goals needing "automatic" implementation:

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

These goals illustrate another common observation for information systems: that such goals imply "knowledge" on the part of the system, not the bringing about of changes in the environment. Knowledge goals are a type of goal that have as their targets states or conditions that the system must be able to distinguish or recognize (e.g., KNOW user edit-authorization level) or in which information is available for use. This technique for the identification of the agents responsible for automatic and knowledge goals may simplify the identification of other goals.

The CommerceNet Web server is a tool which supports the activity of consortium members, serving as a centralized repository of information belonging to all of the participating companies. The self-contained nature of the system enabled stakeholders to regard the system as if it were a black box; they knew the activities to be performed by the system, but were aware that external agents would be introduced as the system evolved and indeed during its lifetime. The stakeholders treated the server as a 'boundary' often expressing responsibilities of the server in terms of what "it" is going to (or should) do. This is significant because while the stakeholders focused their attention on what "it" should do, it was clear that the responsibilities of the external agents is partially dependent on what the Web server expects these external agents to do. For example, before an electronic payment may be processed, an external agent must load their public key.

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. Some of these were included among the "organizational" requirements in the requirements document. Also, if the newly designed system will alter stakeholders' work, it is important to track the people who will ultimately be affected. There are a number of different ways to classify goals; as discussed below, goals are classified according to their target conditions and subject matter.

(B) Categories of goals need to be differentiated

We find it valuable to classify goals as they are introduced and named according to two orthogonal classification schemes. One differentiates among types of goals by noting the target conditions of the goals, classifying goals as either achievement or maintenance goals. An achievement goal is satisfied when the target condition is attained. A maintenance goal is satisfied as long as its target condition remains true. Maintenance goals are usually high-level goals with which associated achievement goals should comply. In the CommerceNet Web Server project, it was important to differentiate between maintenance and achievement goals since a majority of the discussions which transpired with the stakeholders addressed

Table 1: Example GBRAM heuristics for identifying and analyzing goals and constraints.

HIG ₁	Goals are named in a standardized subset of natural language, in which the first word is a verb that describes the target of the goal. For example, AVOID is used to name goals that are satisfied as long as the target condition is false.
HIG ₂	Abstraction may be used to extract goals from documentation by asking: What goal(s) does this statement exemplify or block?
HIG ₃	Stakeholders often express requirements in terms of desired operations rather than goals. Goals are inferred by matching the closest goal verb (see HIG ₁) to the operational term used by the stakeholder.
HIC ₂	Constraints can be identified by searching for temporal connectives. Restate as constraints statements that describe when a condition is true or when a goal can be completed.

“maintaining the server” and “content maintenance.” While the conversational references regarding server maintenance goals and meta-level goals were clear to the analyst, they were not clear to the participating stakeholders.

An achievement goal may share a relation with more than one maintenance goal. Consider the achievement goal: MAKE What’s New filtered according to member’s personal preferences. This goal shares a relation with two maintenance goals: MAINTAIN personal preferences managed and MAINTAIN 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. Maintenance goals are helpful when operationalizing achievement goals because they can point to previously overlooked goals. In a previous case study [Ant96], analysis of information providing additional descriptions of maintenance goals resulted in refinements to the set of achievement goals, thus yielding more meaningful operationalizations. By categorizing goals, analysts may also begin bridging the communication gap between stakeholders and developers.

The second goal classification scheme is based on subject matter. Whereas the distinction between achievement and maintenance goals is structural and domain-independent, the subject matter of goals is domain-specific. Nevertheless, there are goal classes that occur in many software systems. The goal classes identified for the CommerceNet Web Server project are given in Table 2. Security and access goals are by no means unique to this domain, and may be observed in most multi-user systems.

Table 2: Categories of goals organized by subject matter.

Goal class	Description
Process support	The environmental processes monitored and enacted by or with the support of the system.
Security and access control	Restriction of access to system functions to authorized users
Electronic commerce	The core functionality for the system in question.
Information display and organization	Organization and presentation of information.

The goal MAKE Proposal voted on is a *process support goal*, ENSURE Read/write access controlled is a *security and access goal*; MAKE Member billed for membership fees is an *electronic commerce goal*; and MAKE Search results Web page displayed is an *information display and organization goal*. The organization provided by the goals facilitated their organization into naturally different functional requirements.

(C) *The posing of systematic questions enables the derivation of a more complete set of goals and requirements*

The integration of GBRAM [Ant96, Ant97] and the inquiry cycle [PT93, PTA94] facilitates the process of formulating a system by raising standard questions about the evolving requirements. To keep track of the status of these questions, the analyst keeps semi-structured auxiliary notes that do not document the requirements so much as the process of elaborating them. Throughout the CommerceNet project, 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 (Constraints are considered memos at the early stages of analysis because they have not yet been refined). During the early stages of analysis, constraints assume the form of facts about how the system works and its possible relationship with other systems.

Conditions that can prevent a goal from being achieved are initially expressed as loose constraints. However, during goal elaboration, they are reformulated as preconditions for their goals. For example, in Table 3, goals G54 and G55 have constraints or preconditions that suggest further goals that must be achieved before the goals in question can be satisfied. Constraints thus serve as pointers to unrecognized goals for the system. In contrast, the constraint for goal G56 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 G56 remains a constraint in the resulting goal set. The reason that heuristic HIC2 (see Table 1) suggests searching for temporal connectives to identify constraints is the preference many stakeholders exhibit for describing goals in terms of specific actions. If a system behavior B2 desired by a stakeholder is an implementation of an implicit goal G2, the phrase “B1 and then B2” suggests that there is a goal, G1, that is implemented by B1 and that constrains the achievement of G2.

Table 3: Example goals and constraints for the CommerceNet Web Server.

Goals	Constraints
G ₅₄ : MAKE online statement available for each organization	Member must be authorized to access online statement.
G ₅₅ : AVOID duplicate purchases	Member must be able to find out whether his or her organization previously purchased the desired item.
G ₅₆ : MAKE videotapes purchased	Only consortium members and seminar participants may buy a seminar videotape.

Within each subsection of the resulting CommerceNet requirements document, various classifications of auxiliary notes appear, serving to document requirements discussions

[PT93, PTA94]. 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. 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 issues and questions which serve as reminders for the analyst. Items requiring an answer or decision are explicitly flagged to expedite resolutions.

(D) Scenarios facilitate the identification of goals and the evaluation of implementation alternatives

Scenarios also serve to document issues, facilitating the evaluation of implementation alternatives, the identification of new goals, and the elaboration of the requirements. They offer a natural and concrete way to describe the circumstances in which a goal may fail or be blocked, facilitating the discovery of new goals and the consideration of alternative mappings from goals to operations. Scenarios were used extensively and there are numerous examples of goals which were identified via scenario analysis. 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 these scenarios by listing the different activities for which agents are responsible. For example, one scenario "Processing membership fees," was comprised of 12 actions, to be performed in sequence by a responsible agent, as shown in Table 4.

These actions, or behaviors, can be mapped to the goals by stating the actions as prescribed by the naming conventions suggested by HIG1 and HIG2 (See Table 1).

Further analysis using GBRAM's goal identification heuristics resulted in the identification of 12 goals in the CommerceNet Web Server project that had not previously been identified. In Table 5, the goal MAKE payment method selected is shown together with two identified obstacles and exploratory scenarios. Obstacles #1 and 2 indicate the users' need to select from various payment options, such as check, money order, or credit card to participate in the electronic commerce activities of the consortium. Additional goals were identified through the consideration of possible scenarios. For example, consider Scenario #2 (see Table 5). George is an employee at Burdell & Associates. Before George 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 helpful in the identification of

goals because it forced the consideration of possible exceptions within the system.

Table 4: Example scenario from CommerceNet Web Server project.

Step #	Agent	Action
1	User	Find the membership application form page.
2	User	Fill out membership application form.
3	User	Select "e-check" as payment method.
4	User	Enter public key.
5	User	Submit membership application form.
6	Certification authority	Approve user payment.
7	Web Server	Give user receipt.
8	Web Server	Increase budget balance.
9	Web Server	Create user entry in membership database.
10	Web Server	Add user to membership mailing list.
11	Web Server	Add user to member web page.
12	Web Server	Send user membership kit.

Obstacles are an effective mechanism for the anticipation of exception cases that must be handled by system operations. Some requirements arise from analysis of obstacles and are thus not obvious to stakeholders. Another benefit of obstacle analysis is the ability to more easily identify scenarios to determine why a goal could be blocked and when the relationship between goals and scenarios deserves further research. This is useful in that obstacles indicate which scenarios, if elaborated, would ensure coverage of exception cases.

Table 5: Example goal, obstacles and scenarios from CommerceNet Web Server project.

Goal	Obstacles	Scenarios
MAKE payment method selected	(1) Payment method not selected. (2) Payment method choices not clear.	(1) User selects "e-check" method as payment method. (2) George isn't sure whether Burdell & Associates has an account set up yet and needs to know how to get one.

GBRAM heuristics aid analysts by providing prescriptive guidance for managing varying levels of detail in the information available to analysts. For example, when actions appear as instantiations (e.g., e-check is an instantiation of 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).

Scenarios also 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?" 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 bypassed 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, providing a more complete set of requirements.

(E) Goals are named using a standardized set of verbs as the first word

Heuristic HIG1 (see Table 1) 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, facilitates the ordering of goals based on the knowledge that certain kinds of goals may have a pervasive affect on other goals. All CommerceNet goals begin with verbs selected by the analyst and formed from the following set of words: AVOID, ENSURE, IMPROVE, INCREASE, KEEP, KNOW, MAINTAIN, MAKE, and SPEEDUP. These verbs are generic to ensure applicability across multiple domains. Table 6 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' 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. REDUCE implies the acceleration of production.

Table 6: Example goals from the CommerceNet Web Server project, and their type.

Goal	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
REDUCE time required to secure approval for modification updates	Improvement

If stakeholders discuss goals and objectives in terms of behaviors, then analysts must be able to map those behaviors back onto the goals. The use of a standardized set of verbs, serving as a naming convention for goals, allows analysts to easily map behaviors and goals. Distinguishing between maintenance and achievement goals facilitates this process.

One way to capture and document the goals is via goal tables which may be represented using spreadsheets. Goal tables supported the actual production and tracking of requirements in the CommerceNet project. 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 MAKE different entrance to server supported for each user level has two subgoals: ENSURE public entrance to server supported and ENSURE member entrance to server supported. The corresponding scenario provided 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 spreadsheets.

5 DISCUSSION AND FUTURE WORK

In GBRAM, several principles are assumed for identifying and refining goals into operational requirements. First, the process of acquiring the requirements involves an integrative approach, focusing on both abstract goals and concrete behaviors that stakeholders expect the system to exhibit. In the case of the CommerceNet project, goals and requirements were validated via the elaboration of scenarios and the analysis of possible exceptions and

constraints. Second, the inquiry process is simplified by the categorization of goals according to their target conditions and subject matter. These categorizations yield a more focused set of questions, which analysts selectively employ depending on the nature of the proposed system. For example, certain classes of goals are found in various types of systems, such as security and access goals. The process of identifying goals exhibiting such recurring themes is greatly improved by categorizing goals in such a manner since there are clearly generalizable questions specific to those classes. Third, since dependencies exist between goals, a mechanism for identifying goal conflicts and for determining pre- and post-conditions is needed so that goals may be operationalized correctly. Analyzing obstacles is an effective way to address such conflicts and to identify conditions likely to thwart goals. Finally, since requirements continually evolve, it is imperative to manage and track open issues, as well as the rationale associated with the refinement of goals into requirements. Semi-structured auxiliary notes and memos support this process. In the CommerceNet project, the quality and completeness of the resulting requirements were improved as a result.

To reason about possible extensions to GBRAM, we are exploring Michael Jackson's concept of problem frames. The rationale for goal classes in GBRAM is similar to that for Jackson's problem frames [Jac94, Jac95], which provide a way to characterize, classify, and analyze software problems before beginning to solve them. And there seems to be substantial overlap between the concepts of goal classes and problem frames. Further research is investigating whether clusters of subject matter-related goals recur across domains.

One area we have not considered here is the interaction between goals and quantitative non-functional requirements, such as performance and reliability. In this case, improvement and maintenance goals do not become operationalized directly as achievement goals, but rather constrain the fulfillment of other achievement goals within bounded constraints. We are investigating how to integrate the NFR approach [CNY95] into GBRAM.

Finally, in the case of evolving systems, the current system may provide opportunities for identifying agents at a finer level of architectural structure than the system as a whole. We are currently developing a GBRAM-like method, ScenIC, specifically to address the mission-oriented enhancement of legacy systems [AGJ97], in which scenario-based analysis of requirements is carried out in conjunction with similar analyses of architectural feasibility [AKB95].

ACKNOWLEDGMENTS

The CommerceNet case study was sponsored by NTT Multimedia Communication Laboratories in Palo Alto, California. In particular, the authors wish to thank Dr. Kenji Takahashi, Jeff Smith, and Erik Battaler.

REFERENCES

- [AD93] Anderson, J. & B. Durney, "Using Scenarios in Deficiency-Driven Requirements Engineering," Int. Symp. Requirements Engineering (RE'93), San Diego, CA, pp. 134-141, 1993.
- [AGJ97] Abowd, G., A. Goel, D. Jerding, M. McCracken, M. Moore, J.W. Murdock, C. Potts, S. Rugaber & L. Wills, "MORALE: Mission-Oriented Architectural Legacy Evolution," Int. Conf. Software Maintenance'97, Bari, Italy, Sep 29-Oct 3 1997.
- [AKB95] Abowd, G., R. Kazman, L. Bass & J. Pitkow, "Scenario-based evaluation of software architectures: A case study on Internet information systems," SIGSOFT'95: 3rd Ann. Symp. on the Foundations of Software Eng., 1995.
- [ALR96] Antón, A.I., E. Liang & R. Rodenstein, "A Web-based Requirements Analysis Tool," 5th Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, Stanford, CA, pp. 238-243, 19-21 Jun 1996.
- [AMP94] Antón, A.I. W.M. McCracken & C. Potts, "Goal Decomposition and Scenario Analysis in Business Process Reengineering," Adv. Info. System Eng.: 6th Int. Conf. Utrecht, The Netherlands, pp. 94-104, Jun 1994.
- [Ant96] Antón, A.I. "Goal-Based Requirements Analysis," Int. Conf. on Requirements Engineering (ICRE '96), Colorado Springs, Colorado, USA, pp. 136-144, Apr 1996.
- [Ant97] Antón, A.I. Goal Identification and Refinement in the Specification of Software-Based Information Systems, Ph.D. Dissertation, Georgia Institute of Technology, Atlanta GA, 1997.
- [Boe88] Boehm, B. "A Spiral Model for Software Development and Enhancement," IEEE Computer, 21(5) pp. 61-72, May 1988.
- [BR89] Boehm, B. & R. Ross, "Theory-W Software Project Management: Principles and Examples," IEEE Trans. Software Eng., Jul 1989.
- [CKI88] Curtis, B., H. Krasner & N. Iscoe, "A Field Study of the Software Design Process for Large Systems," Comm. ACM 31(11), pp. 1268-1297, 1988.
- [CNY95] Chung, L., B. Nixon & E. Yu, "Using Non-functional Requirements to Systematically Support Change," Int. Symp. Requirements Engineering (RE'95), York, UK: Mar 27-29, 1995.
- [Dar93] Dardenne, A. On the Use of Scenarios in Requirements Acquisition. University of Oregon, Technical Report CIS-TR-93-17, Eugene, OR, 1993.
- [DFv91] Dardenne, A., S. Fickas & A. van Lamsweerde, "Goal-directed Concept Acquisition in Requirements Elicitation," 6th Int'l. Workshop on Software Specification and Design, Como, Italy, pp. 14-21, Oct 1991.

- [Eas93] Easterbrook, S. "Domain Modelling with Hierarchies of Alternative Viewpoints," Intl. Symposium On Requirements Engineering (RE '93), San Diego, CA, pp. 65-72, Jan 1993.
- [Fea87] Feather, M. "Language Support for the Specification and Development of Composite Systems," ACM Trans. Programming Languages & Systems, (9)2, pp. 198-234, Apr 1987.
- [FKG89] Finkelstein, A., J. Kramer & M. Goedicke, "Viewpoint Oriented Software Development: Methods and Viewpoints in Requirements Engineering," Algebraic Methods II: Theory, Tools & Applications, Mierlo, Netherlands, pp. 29-54, Sep 1989.
- [Jac94] Jackson, M. "Problems, Methods and Specialisation," Software Eng. J. (9)6, pp. 249-255, Nov 1994.
- [Jac95] Jackson, M. Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices, Addison-Wesley, 1995.
- [LB85] Lehman, M. & L. Belady, Program Evolution: Processes of Software Change, Academic Press, 1985.
- [LPR93] Lubars, M., C. Potts & C. Richter, "A Review of the State of Practice in Requirements Modeling," Intl. Symposium On Requirements Engineering (RE '93), San Diego, CA, pp. 2-14, 4-6 Jan 1993.
- [Luq89] Luqi, Software Evolution through Rapid Prototyping, IEEE Software, pp. 13-25, May 1989.
- [NKF94] Nuseibeh, B., J. Kramer & A. Finkelstein, "A Framework for Expressing the Relationships Between Multiple Views in Requirements Specification," IEEE Trans. on Software Eng., 20(10), pp. 760-773, Oct 1994.
- [PC96] Potts, C. & L. Catledge, "Collaborative Conceptual Design: A Large Software Project Case Study," Computer-Supported Cooperative Work, 5(4), pp. 415-445, 1996.
- [Pot93] Potts, C. "Software Engineering Research Revisited," IEEE Software, pp. 19-28, Sep 1993.
- [Pot95] Potts, C. "Using Schematic Scenarios to Understand User Needs," DIS'95: Designing Interactive Systems, Ann Arbor: MI, pp. 247-256, 23-25 Aug 1995.
- [PT93] Potts, C. & Takahashi, K. "An Active Hypertext Model for System Requirements." 7th Int. Workshop on Software Specification and Design, Redondo Beach, California, pp.62-68, Dec 6-7 1993.
- [PTA94] Potts, C., K. Takahashi & A. Antón, "Inquiry-Based Requirements Analysis," IEEE Software, 11(2), pp. 21-32, March 1994.
- [SN93] Schuler, D. & A. Namioka (eds.), Participatory Design: Principles and Practices, Erlbaum, 1993.
- [vDM95] Van Lamsweerde, A. and Darimont, R. & Massonet, P. "Goal-Directed Elaboration of Requirements for a Meeting Scheduler: Problems and Lessons Learnt," 2nd Int. Symposium on Requirements Engineering (RE '95), York, UK, pp. 194-203, March 1995.