CHAPTER 7

Conclusions

Let me tell you the secret that has led me to my goal.
My strength lies solely in my tenacity.

Louis Pasteur

The research in this thesis follows the "Industry as Laboratory" approach [62]. The Goal-Based Requirements Analysis Method (GBRAM) and the Goal-Based Requirements Analysis Tool (GBRAT) were developed by applying them to their own development as well as to other projects. The GBRAM evolved as a result of the refinements assembled from applying the method to the experiences detailed in Chapters 3 and 6 of this dissertation. In the first case study, goal decomposition and scenario analysis were applied in the context of Business Process Reengineering [7], demonstrating that both scenario and goal analysis are valuable for an effective analysis effort. The GBRAM was then evaluated by applying it to the requirements analysis of a continuing education system [4] and subsequently to the meeting scheduler system. Further validation of the method was sought by applying it to construct the requirements for a tool to support the method; this effort led to the design and implementation of the GBRAT prototype [6]. Through these various case studies, the method was evaluated and evolved based on the experiences and lessons learned. However, it was also necessary to reason about the scalability of GBRAM to larger commercial applications. This was accomplished by applying the GBRAM to the redesign and reorganization of the CommerceNet Web server, which involved the active participation of several analysts and stakeholders. Finally, in an effort to compare the method to other requirements analysis approaches, an empirical evaluation was conducted as an additional validation of the method.

This chapter provides a synopsis of each chapter in this dissertation, presents a summary of the contributions of this work, and provides an overview of future work.

7.1 Chapter Synopsis

Chapter 1 introduced and articulated the problem addressed in this work. One of the principal problems in software requirements engineering is the difficulty in transforming a set of incomplete, inconsistent, and nebulous ideas into a complete, consistent elaboration of the technical requirements for a software system which are comprehensible to the intended customers and stakeholders. This chapter presented the importance of a good software requirements process in the context of a discussion of the consequences of poorly interpreted software requirements and detailed the importance of the issue in terms of increased software costs and potential software failures. The notion of goals as a means of early requirements validation was also outlined, giving rise to the research that has been presented in this dissertation.

Chapter 2 presented a survey of the related work in the field of software engineering and discussed the influence of goals in other fields.

Chapter 3 discussed three case studies: the Financial Services Office business process reengineering project, the Career Track Training System, and the meeting scheduler system. The methodology employed for each case study was presented, followed by discussions of the lessons learned. The case studies presented in the chapter are significant in that they were formative in nature. The GBRAM was simultaneously developed and informally validated while being applied to real projects; thus, the GBRAM evolved as a result of its application to the case studies that were discussed in this chapter.

Chapter 4 introduced the Goal-Based Requirements Analysis Method (GBRAM). The chapter provided an overview of the method, discussing the activities an analyst is involved with while employing the method and differentiating between the goal analysis and goal refinement phases. The method was illustrated in detail via the presentation of a reasonable sequence of activities, progressing from the initial identification of goals to translations of those goals into operational requirements.

Chapter 5 presented four general types of heuristics employed by analysts using the GBRAM: identification heuristics, classification heuristics, refinement heuristics, and elaboration heuristics. Several techniques were investigated as background for this research: scenario analysis, identification of goal obstacles and constraints, and goal operationalization. The heuristics and guidelines provided in this chapter were shown to be useful for identifying and analyzing the specified goals and were beneficial for their elaboration and refinement. The heuristics and supporting inquiry included references to appropriate construction of scenarios and the process in which they should be discussed and analyzed. A set of recurring question types was presented to assist analysts in applying an inquiry-driven approach to goal-based analysis which discussed the inquiry process, examples, potential results, and final outcomes. Given a particular answer, analysts can determine the change which results from asking stimulative questions and the kind of refinement which may be made.

Chapter 6 presented the summative validation of the research discussed in this dissertation. Three specific validation efforts were conducted: two case studies and an empirical evaluation. In the first case study, GBRAM was employed to specify the requirements for tool support and was followed by construction of a prototype based on the resulting requirements. The second summative case study involved the reorganization of the CommerceNet Web Server. An empirical evaluation was also conducted in which the performance of one group of subjects using the GBRAM was compared with the performance of other groups of subjects using alternative analysis methods. Completion of these efforts conceptually validated the premise of this research.

7.2 Summary of Contributions

The principal contribution and focus of this work is the introduction of the Goal-Based Requirements Analysis Method for the identification and elaboration of goals and the refinement of those goals into operational requirements. Research for the method included the evaluation of existing goal-based approaches for the determination of the level of support and guidance they provide for the initial discovery of goals and requirements information. This, coupled with preliminary case studies, furnished the formative basis for the early development of the method. The GBRAM was developed to assist analysts in gathering software and enterprise goals from many sources and to support the process of elaborating and refining goals into operational requirements. The method's chief contribution is the provision of heuristics and procedural guidance for identifying and constructing goals. A catalog of heuristics and questions for guiding the inquiry process was constructed and later validated and updated during the summative case studies.

The following three subsections outline the experiences demonstrated in this body of work, the primary contributions of this dissertation, and the limitations of the Goal-Based Requirements Analysis Method.

Demonstrated Experiences

The experiences reported in this dissertation demonstrate that:

- It is possible to specify effectively the functional requirements for a software-based information system using the heuristics and guidelines presented in this dissertation.
- Information pertaining to the goals and system objectives which are often overlooked when using other analysis methods are readily uncovered by employing GBRAM's elaboration and refinement heuristics throughout the analysis process.
- The inquiry cycle has been effectively applied/instantiated in conjunction with the set of recurring question types GBRAM offers to guide analysts in applying the approach.
- Goals offer a rich structure for organizing requirements information (i.e., obstacles, scenarios, constraints, and auxiliary notes) in the form of goal topographies which aid analysts in finding information and sorting goals into naturally different functional requirements.

Primary Contributions

The primary contributions of this dissertation are:

- the Goal-Based Requirements Analysis Method which provides prescriptive advice to analysts for the initial discovery and identification of goals;

- four sets of heuristics to assist analysts in identifying, classifying, refining, and elaborating goals (the 51 heuristics are summarized in Appendix C);
- a set of recurring question types, following the inquiry cycle approach, with suggested applications and resolutions; and
- a method for converting goals into operational requirements.

In addition, as a result of the GBRAT case study, proof of concept tool support has been developed in the form of the Goal-Based Requirements Analysis Tool (GBRAT) which is a multi-user Web-based tool designed to support the GBRAM.

Limitations of the GBRAM

The main limitations of the GBRAM are:

- the method provides informal, as opposed to formal semantics, for goals and thus it does not support formal reasoning;
- while well suited for identifying functional requirements which represent specific behaviors the proposed system should exhibit, GBRAM has not been adequately proven and tested for nonfunctional requirements other than general maintenance requirements; and
- while this dissertation presents some useful key words for the identification of various elements (e.g. goals, constraints, etc.), the method does not offer a complete lexicon of key words which would greatly assist analysts.

7.3 Future Work

The work presented in this dissertation addresses some of the fundamental problems with requirements identification; however, work remains to be done in these areas. This section provides an overview of areas of future interest.

The rationale for Michael Jackson's problem frames is similar to the rationale for goals classes in the GBRAM. Jackson asserts that before analyzing a software problem, an appropriate method for solving the problem must be chosen [46, 47]. His problem frames* provide a way to characterize, classify, and analyze software problems before beginning to solve the problem. Given a software problem, it is helpful to determine whether there are other problems which fit the same problem frame so that the methods and techniques which work for other problems may be applied to the problem currently being solved. Problem frames address broad problem characteristics and their strategies for solution.

Three of Jackson's problem frames are the JSP frame, the workpieces frame, and the environmental-effect frame [46,47]. In the JSP frame, the machine to be built is the program. The frame is tightly constrained with input and output streams (sequential structures of elements). Since the real word domain in the JSP frame is assumed to be autonomous, it is inadequate for an embedded system. The workpieces frame regards the machine as a production tool for textual or graphic documents. While suitable for an application involving text processing, i.e., a word processor, the workpieces frame is not suitable for large or embedded systems. The environmental-effect frame is suitable for an embedded system which controls an external domain. Jackson points out the need for more problem frames since all systems do not adequately fit into his proposed problem frames. For example, he discusses the Simple IS frame in [?], exposing its limitations as being too general.

The systems addressed in this dissertation do not adequately fit into the Simple IS frame, as it is too general a frame for most aspects of IS problems; however, some components of the problems addressed in this dissertation do fit into his workpieces frame (e.g.

*A problem frame is a generalization of a class of problem [46, 47].

the GBRAT tool). This dissertation introduced the notion of goal classes as generalizable to different software systems. A generalizable class or subclass of goals which may be viable for more than one system is envisioned (i.e., security and access goals). As illustrated in the CommerceNet and Vacation/Sick Leave case studies, goal classes are a way of carving up the problem domain. The problem domain may be decomposed into goals about process, security, etc., so that the classes dictate both what the system goals are and what they mean.

There appears to be a substantial overlap between the concept of goal classes and Jackson's problem frames. Future work will examine the commonality which underlies both of these approaches for characterizing software problems. Further investigation will determine whether there are clusters of specific goal classes in certain systems, such as those characterized by Jackson's problem frames. This determination will impact the heuristics presented in this dissertation by indicating refinements for the heuristics which call for the addition of specific questions for analysts to ask regarding goals which fall into particular goal classes.

Table 7.1 provides an overview of three generalizable goal classes discussed in this dissertation. If analysts can characterize the software problem according to goal classes before attempting to solve the problem and thus tailor the GBRAM to their needs; the method would be both easier to apply and more effective in its results. Just as the JSP frame is not well suited to embedded systems, it may be that certain goal classes or subclasses may not be well suited for analysis using certain GBRAM heuristics. Consider that the GBRAM's performance in comparison to other analysis methods was superior for the identification of messaging requirements, but comparable with respect to the other three goal classes in the vacation/sick leave system. A goal class framework to characterize the types of problems which may be effectively analyzed using different heuristics and inquiry points in the GBRAM would enrich this work and increase usefulness for analysts who wish to employ the method.

Goal Class	Description
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.
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

Table 7.1. Generalizable Goal Classes

Future work will also involve extensions to the GBRAM. Two possible ways of structuring inquiry questions are under consideration: a process-based structure and a goal class structure. A process-based structure would offer analysts a set of questions structured according to when in the process the questions should be asked. Alternatively, questions may be organized and structured according to generalizable goal classes, as discussed earlier in this section.

This dissertation discussed the concept of employing a subset of natural language to comprise useful key words for the identification of the different elements which analysts work with when applying the GBRAM. Two extensions are under consideration to further this notion of a standard set of key words: a lexicon and a goal language. A lexicon which analysts can employ for the identification of goals, obstacles, constraints, etc., would afford a more systematic and standard stock of terms than are currently offered by the method. Similarly, a simple goal language composed of the key words presented in this dissertation would provide a standard goal naming convention which would provide a way of representing goals as clauses with associated rules for manipulation (e.g. `MAKE [state] true` and `KNOW [state] acquired`).

The concept of derived and synthesized goals was introduced in Chapter 6 of this dissertation. Derived goals are those goals directly identified or explicitly extracted from descriptions of the problem. Synthesized goals are those goals identified and constructed through inquiry or due to the realization of the need for a new goal while analyzing the problem. As discussed in Chapter 6, data has been collected for the investigation of this differentiation between derived and synthesized goals; analysis of this will facilitate consideration of the implications of goal origins. Analysis of the traceability between derived and synthesized goals may lead to the development of stopping criteria for analysts during inquiry-driven analysis.

The use of goals and scenarios to augment natural language descriptions has been explored throughout this dissertation. However, strategies are needed to improve requirements documents. Future work will focus on reducing the level of ambiguity of these documents without the need for formal models. Additionally, requirements documents are typically difficult to index and offer no clear organizational structure. This dissertation presented the concept of goal topographies for structuring requirements information in software requirements documents (SRD). If goal topographies provide a reasonable way of structuring

SRDs, it naturally follows that topographies may provide a direct way to structure hypertext documents using goal topographies for the automatic generation of Web pages.

7.4 Conclusions

Goal analysis is a useful approach for the early validation of requirements. Multiple stakeholders have multiple goals; this multiplicity often augments the difficulty of developing a clear understanding of those goals. In the Goal-Based Requirements Analysis Method, the stakeholders' goals are clarified by tracking associated rationale. Stakeholders frequently discard systems because they do not understand how the system assists in the achievement of their goals, or because they use only a portion of the system. Analysts are charged with addressing both the root of this dissatisfaction and the cause of this inefficiency and misuse of effort.

It is commonly held that the problem with requirements engineering stems from the need to develop an understanding of the requirements expressed by stakeholders who do not themselves understand the requirements. At the same time there is a tendency to take short cuts during the requirements analysis task leading to an unstable design and often analysts fail to consider alternatives before the software is specified. The GBRAM may be modeled as an instance of the Inquiry Cycle approach [61, 64]. Goal analysis improves the understandability of the requirements by enabling analysts to produce a set of requirements which may be easily understood, validated, and readily accepted by the stakeholders involved in the process. In sum, each of the stages of the method correspond to a set of related expression activities; the heuristics of the method correspond to guided discussion activities (i.e., standard questions and templates for reasonable answers and reasons); and detailed steps correspond to refinements.

The issues inherent in determining and specifying the requirements for software-based information systems are addressed by the Goal-Based Requirements Analysis Method. The strength of the GBRAM lies in its focus on goals and objectives and the derivation of operational requirements from those goals. The method offers a straightforward, methodical approach to identifying system and enterprise goals and requirements; it also suggests goal identification and refinement strategies and techniques through the inclusion of a set of heuristics, guidelines, and recurring question types. Use of this method produces a software specification of the functional requirements in the form of goal schemas depicting behaviors in terms of goals, relationships, constraints, obstacles, and agent responsibilities.