CHAPTER 5

Heuristics and Guidelines

The most important function that software builders do for their clients is the iterative extraction and refinement of the product requirements.

Frederick P. Brooks, Jr.

The objective of this chapter is to describe some typical heuristics used by analysts applying the Goal-Based Requirements Analysis Method. In the GBRAM, heuristics are rules which guide analysts towards a high probability of success while avoiding wasted efforts. The GBRAM provides sets of heuristics and guidelines for the identification of goals to be used in requirements specifications. The selection of heuristics depends upon the type of system desired and on the information available to the analyst.

There are four general types of heuristics employed by analysts using GBRAM:

- identification heuristics;
- classification heuristics;
- refinement heuristics; and
- elaboration heuristics.

Identification heuristics assist analysts in identifying goals, stakeholders, agents, and constraints from multiple sources. The objective of goal classification heuristics is to aid analysts in determining the type of each goal identified. Refinement heuristics employ a series of questions and techniques to reduce the size of the goal set. Elaboration heuristics address the need to acquire more detailed information by considering goal dependency relations, suggesting the goal obstacles for which scenarios should be constructed and which scenarios to elaborate. Although there is a distinction between refinement and elaboration, as discussed in Section 5.4, GBRAM recognizes the concurrent and overlapping nature of these activities. As discussed in Chapter 3, the lessons learned in the initial case studies served as the origin for the ideas which formulated the Goal-Based Requirements Analysis Method. The heuristics in this chapter were derived from these experiences and observations.

Section 5.1 presents a set of recurring question types to aid analysts in applying an inquiry-driven approach to goal-based analysis. Since the utilization of the heuristics depends upon the particular GBRAM activity with which the analyst is involved at any given time, each heuristic set (5.2, 5.3, 5.4, and 5.5) is presented in its own section, with discussion of the application of the heuristics to specific GBRAM activities.

5.1 Goal-Based Instantiation of Inquiry Cycle

The GBRAM heuristics detailed in this chapter are a set of rules. Some of these heuristics are straightforward, not requiring the analyst to employ any specific inquiry process. In contrast, other heuristics are meaningless without the accompanying questions to guide analysts in developing a deeper understanding of the system and uncovering hidden goals and requirements. Often a simple answer to a question is insufficient; for example, justification or rationale for a particular response may be needed to understand the system. Thus, GBRAM offers a set of recurring question types which follow the inquiry cycle approach [61, 64] instantiated for goal-based analysis, and guide analysts in applying an inquiry-driven approach as they comprehensively investigate the available artifacts for descriptive answers. This section discusses these question types; subsequent sections in this chapter suggest appropriate applications and resolutions based upon the answers derived from the questions asked.

Recall from the discussion in Chapter 2.2 that the Inquiry Cycle model is a formal structure for describing discussions about requirements [61]. The types of questions asked about a set of goals and requirements were investigated during the case studies discussed in Chapter 3. These recurring question types are summarized below and discussed throughout the remainder of this chapter.

- *What-is:* These questions request specific information regarding terminology which is unclear to someone with no knowledge of the application domain. For example, analysts may ask a stakeholder, *"What is the process for deciding how to organize and structure information in an intranet server?"* In any analysis effort, participants in the process have to develop a common understanding of the concepts and terms. What-is questions enable analysts (and stakeholders) to do precisely this. What-is questions clarify a situation or scenario enabling analysts to ensure that something is understood correctly. This type of question is most useful when the analyst is able to interact directly with the stakeholders.

- *Who-is:* These questions request specification of the agent responsible for the given task, process or goal. In Business Process Reengineering (BPR) cases it is often the case that one of the prevailing inefficiencies in an organization is the number of approvals and authorizations [levels] that are required. It is helpful to ask questions such as, *"Who is responsible for the ultimate decision?"* and *"Who is responsible for this task?"* By asking who-is questions, analysts can acquire information about the various agents and tie the information to the corresponding goal based on implied or explicit responsibility.

- *Why:* These questions request *reasons* which underlie work activities. This information is essential when redesigning processes. Analysts must ask questions such as *"Why is this information routed?"* While it is relatively simple for analysts to determine *what* information is required to route, the reason why the information is routed can be very difficult (if not impossible) to ascertain without direct stakeholder inquiry. These questions may be application specific, but may be generalized since it is critical in BPR efforts to ask *why?* questions.

- *What-if:* It is beneficial to ask questions that enable analysts to further examine cases in which an unexpected action occurred in order to explore how other system features may be affected. An example is *"What happens if an individual drops or cancels out of a class?"* This kind of question prompts consideration of whether this new opening in a class has any bearing on, for example, students who were previously turned away from the class because the course had reached its registration capacity. What-if questions mandate the consideration of the other agents and processes that would be affected in the event of such an unexpected cancellation.

- *When:* These questions request timing constraints for a given event or events. For example, it may be unclear when an electronic credit payment approval request is subsequently reviewed again after previously failing to gain approval. It may be important to know when requests for payment approvals are reviewed. To ascertain this type of information, analysts ask questions such as, *"When is an electronic credit payment approval request reviewed?"* which may be followed up with a clarification question such as: "Is it reviewed immediately or at the beginning of the next working day?" In Business Process Reengineering efforts, this type of information is critical because it enables analysts to identify candidates for redesign so that unnecessary delays may be eliminated.

- *Relationship:* These questions ask how one agent is related to another or how one goal is related to another goal so that the dependency relations may be established. For example, analysts may consider each goal and ask: *"What goals are prerequisites for this goal?"*, *"What goals must follow this goal?"*, and *"What agent depends on this goal for completion of their responsibilities?"*

These question types assist analysts in knowing when and how to apply the GBRAM heuristics by providing a guide as to how much detail is needed before one can be reasonably confident that the goals have been fully elaborated and that any hidden goals or requirements have been uncovered.

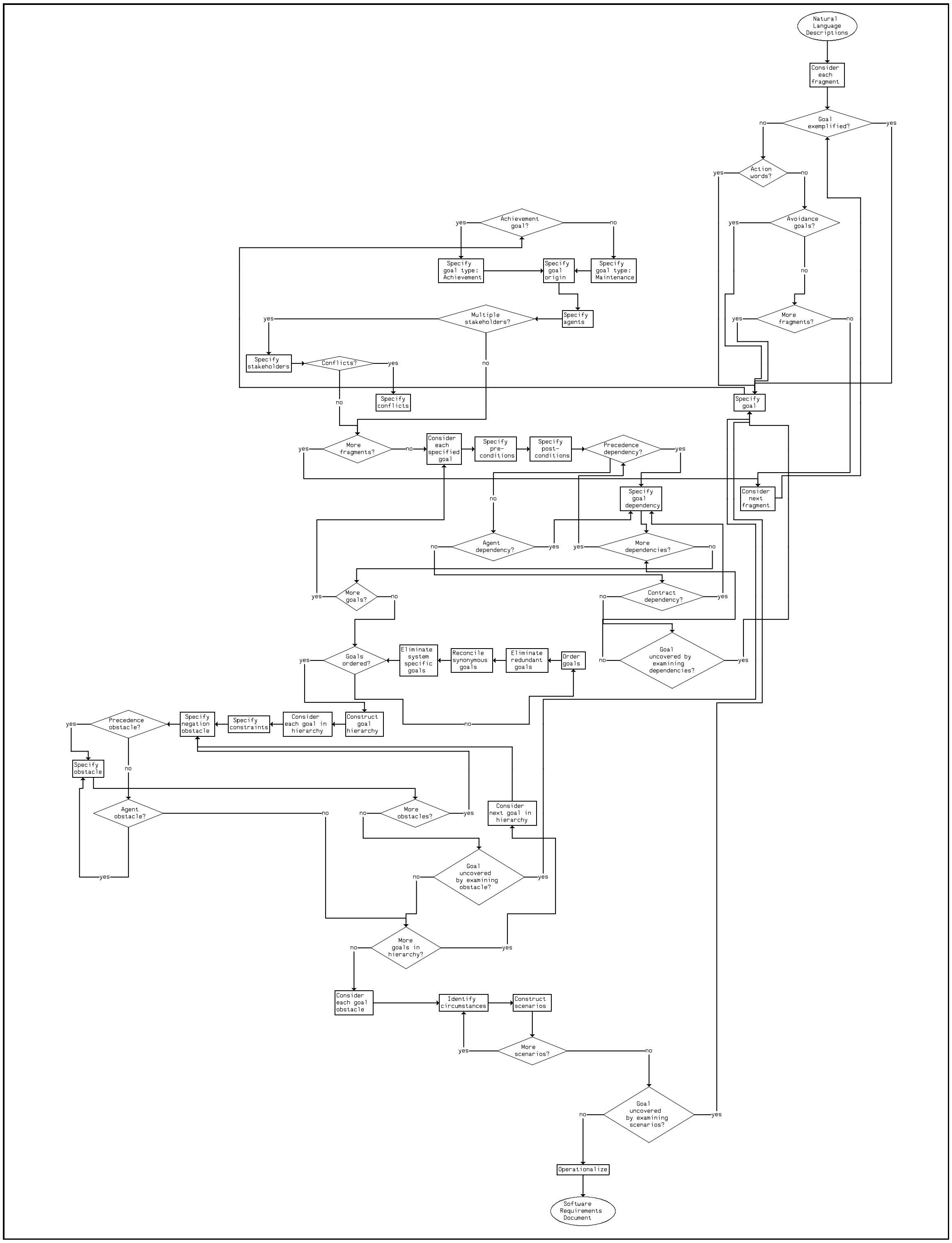

Figure 5.1. GBRAM Control Flow Chart

Figure 5.1 provides a reduced overview of the Goal-Based Requirements Analysis Method, represented in a control flow chart. The flow chart is an approximation of the method; it is presented here to show the overall scope of the method. Each part of this flow chart is presented in expanded form throughout the remainder of the chapter and is discussed within the context of the specific inquiry points for each set of GBRAM heuristics. Each of the activities shown in Figure 4.1 on page 69 are represented in this control flow chart. The boxes in Figure 5.1 represent the activities that an analyst must perform. The diamonds represent the various inquiry points for an inquiry-driven approach throughout the GBRAM process, while the ellipses represent the input (natural language descriptions) and output (a SRD) of the method. The arrows denote the flow throughout the process as well as the iterative nature of goal-based analysis. Boxes and ellipses have only one output edge; however, since the diamonds represent inquiry points, a decision is required of the analyst. Thus, diamonds have two outgoing edges; one edge represents an answer of 'yes,' and the other edge indicates an answer of 'no'. A detailed discussion of the types of questions analysts ask during inquiry is provided in earlier in this section. The decision points, as shown by the diamonds in Figure 5.1, are discussed throughout the remainder of this chapter. Since Figure 5.1 is compressed to allow for an overview of the complete process, the discussion of each set of heuristics is accompanied by the relevant portions of Figure 5.1 for the elucidation of the possible sequences of heuristic and inquiry application detailed in this chart. Again, it must be emphasized that the flow charts throughout this chapter provide an approximate representation of the method; a more detailed description of the GBRAM activities is available in Chapter 4. Since many of the activities may be performed concurrently, the flow charts simply serve to demonstrate how the GBRAM instantiates the inquiry cycle [64].

Each heuristic set addresses a problem space, as discussed in the text preceding each set of heuristics in this chapter. Many examples have already been presented in Chapter 4; however, some additional supporting examples for the heuristics not previously discussed throughout this thesis are provided in this Chapter. The remainder of this chapter presents the four sets of GBRAM heuristics and provides guidelines for the appropriate time and manner for application. Table 5.1 provides a glossary for the labels which serve as tags for the identification of each of the four heuristic sets.

Table 5.1. Glossary of Heuristic Identifier Codes

Code	Definition
HIG	Heuristic for Identifying Goals
HIS	Heuristic for Identifying Stakeholders
HIA	Heuristic for Identifying Agents
HIC	Heuristic for Identifying Constraints
HCM	Heuristic for Classifying Maintenance
HCA	Heuristic for Classifying Achievement
HRR	Heuristic for Refining Redundancies
HRS	Heuristic for Refining Synonymous
HRSS	Heuristic for Refining System-Specific
HED	Heuristic for Elaborating Dependencies
HEO	Heuristic for Elaborating Obstacles
HES	Heuristic for Elaborating Scenarios

5.2 Identification Heuristics

This section presents four sets of identification heuristics in GBRAM which may be summarized as follows:

- the *goal identification heuristics* provide strategies for identifying and extracting goals from various information sources;
- the *stakeholder identification heuristics* guide the process of identifying all parties claiming an interest in the proposed system;
- the *agent identification heuristics* provide strategies for allocation of goal responsibilities; and
- the *constraint identification heuristics* offer strategies for identifying conditions that must exist or be met for a goal to be realized or completed.

As discussed in Chapter 4, the GBRAM identification activities (Figure 4.1, page 69) involve the identification of goals, stakeholders, and agents. Another GBRAM identification activity involves constraints. Although constraint identification was introduced late in Chapter 3, this activity is a recurring task for the analyst. As such, analysts may identify constraints as early as during the initial exploration of documentation or as late as during the operationalization of goals into requirements. The constraint identification heuristics are included in this section of the chapter because this activity is not limited to any one phase of GBRAM.

The following subsections provide a discussion of these identification heuristics within the context of the associated activities shown in Figure 4.1.

Heuristics for Goal Identification

The goal identification heuristics address the problem of how to extract goals from the documentation and resources available to the analyst. As discussed in Chapter 4, goal analysis is the process of exploring gathered documentation, ranging from information about the organization (i.e., enterprise goals) to system-specific information (i.e., requirements), for the purpose of identifying, classifying, and organizing goals. It is often assumed that software systems are constructed with a purpose or goal(s) in mind [27]; the heuristics in this section address the origin of goals when their origin is initially unknown and what happens when the goal or purpose is not clear to the analyst.

As previously discussed in Chapter 3 on page 74, analysts must work with various information sources. When reengineering legacy systems, an existing requirements document may be used as a starting point for extracting goals and objectives. However, such a document may not be available, requiring the goals to be deduced from the obtainable sources. Goals may be extracted or identified from a number of different sources, as shown in Figure 5.2.

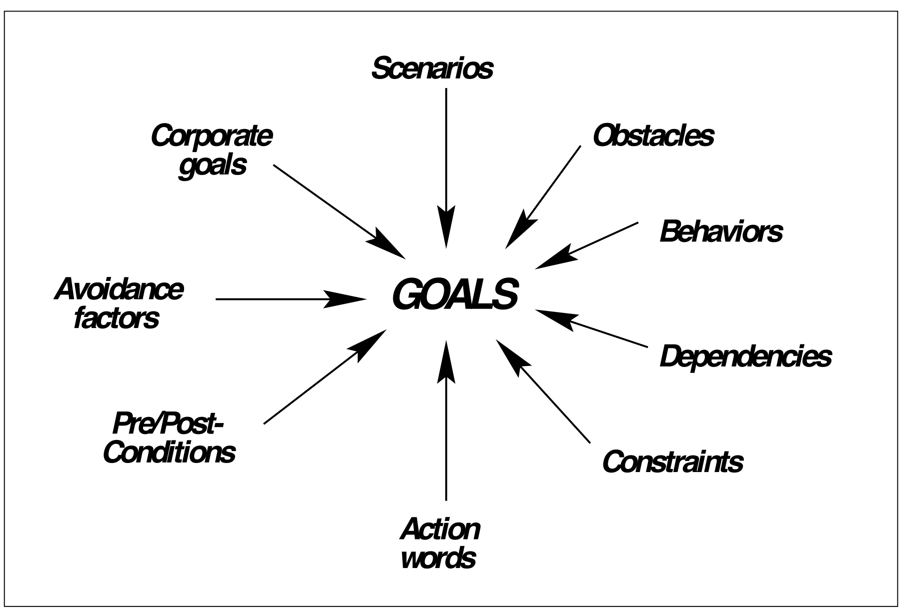

Figure 5.2. Origin of Goals

Goal identification is not limited to the initial activities of the Goal-Based Requirements Analysis Method; goals may be identified throughout the entire analysis process (e.g. during the identification of goal dependencies, obstacle analysis, and scenario analysis). Figure 4.1 on page 69 illustrates how goals may be identified during goal refinement and elaboration. The goal identification heuristics address this 'delayed' uncovering of new goals. For example, goals may be identified by considering constraints, as shown in Example 5.13 on page 150, or scenarios, as shown in Example 4.19 on page 103.

The general heuristics for identifying goals and objectives, presented below, should be considered by analysts when exploring existing documentation. Those heuristics which are not explicit are accompanied by a series of questions to guide the analyst. Figure 5.3 illustrates a possible series of inquiry points for an analyst applying the goal identification heuristics to extract goals from available documentation.

HIG 1. Goals are named in a standardized subset of natural language in which the first word is a verb that describes the kind of goal being named. For example, `AVOID` denotes one kind of goal. Goals of this kind are satisfied as long as their target conditions remain false.

> Example 5.1 In the electronic commerce WWW server discussed in Chapter 6.2, stakeholders expressed the need to prevent duplication of purchases. The analyst must specify a goal name such as `AVOID duplicate purchase`.

HIG 2. Abstraction mechanisms may be employed to extract goals from available documentation by asking:

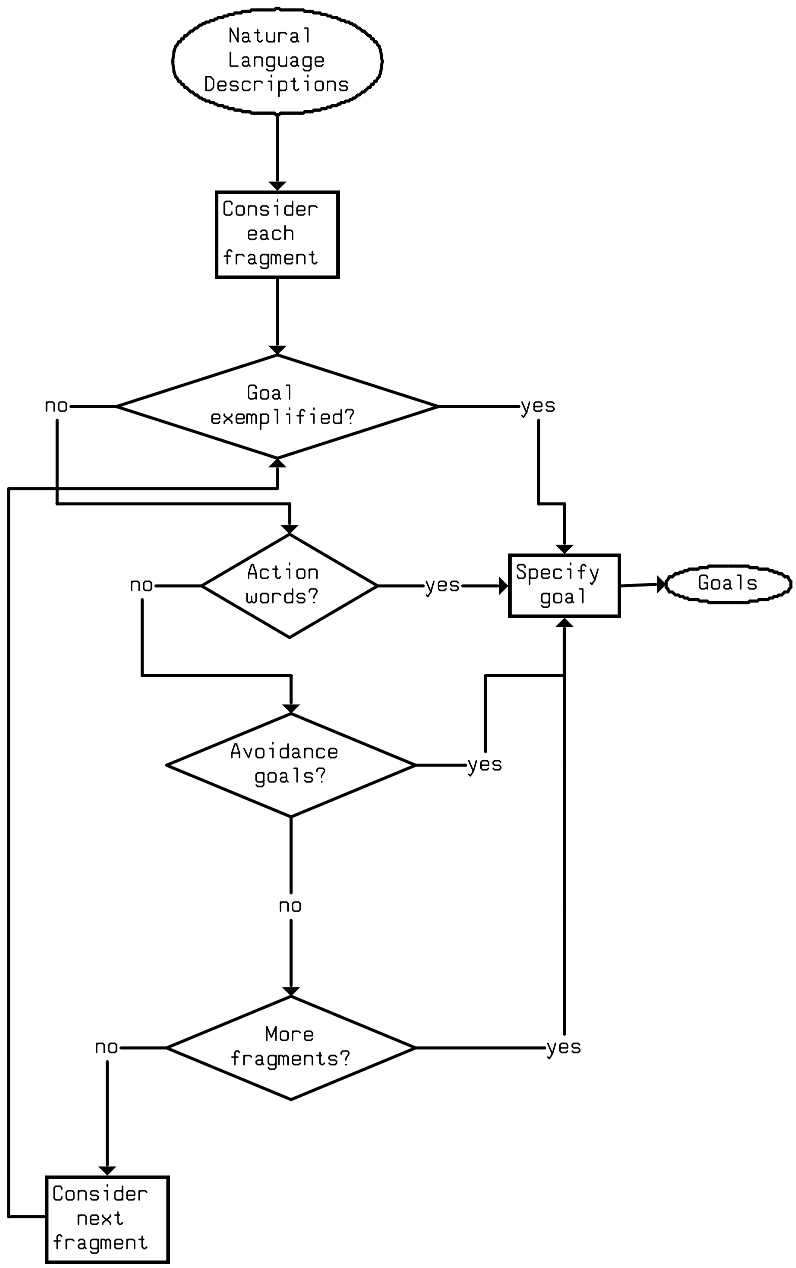

Figure 5.3. Control Flow Chart for Goal Identification

(a) What goal(s) does this statement exemplify?

(b) What goal(s) does this statement block or obstruct?

If the answer to either of these questions is yes, then express the statement as a goal which represents a state that is desired or achieved within the system. The inquiry points are shown in Figure 5.3.

Example 5.2 (Using abstraction mechanisms to identify goals) In the bug evolution example (See Appendix B; page 241), the goal `Population stabilized` is identified by using abstraction mechanisms to identify goals.

HIG 3. Action words (see Figure 5.3) that point to some state that is or can be achieved once the action is completed are candidates for goals in the system. They are identified by considering each statement in the available documentation by asking:

(a) Does this behavior or action denote a state that has been achieved, or a desired state to be achieved?

If the answer is yes, then express the answer to these questions as goals which represent a state that is desired or achieved within the system.

Example 5.3 (Using action words to identify goals) Consider the bug evolution description in Appendix B. *Move*, *eat*, *die*, and *reproduce* are all action words. Recall that if a statement denotes an action or behavior, it must be restated so that it denotes a state that has been achieved or is desired (a goal). For example, `bug moves` should be restated as `bug moved`.

HIG 4. An effective way to uncover hidden goals is to consider each action word and every description of behavior and persistently ask *"Why?"* until all the goal have been 'treated' and the analyst is confident that the rationale for each action is understood and expressed as a goal. The action words should be restated so that they denote a state that has been achieved or a desired state.

Example 5.4 (Questions for action words) Given the statement *weight gained*, the analyst would inquire as to why the bug gained weight. Given the description in Appendix B, one can determine that the bug gained weight because the bug ate bacteria (*bacteria eaten*). While this technique may seem obvious, it has been applied successfully to several system analysis efforts (Refer to Chapter 3; page 35).

HIG 5. Key action words such as: *track, monitor, provide, supply, find out, know, avoid, ensure, keep, satisfy, complete, allocate, increase, speedup, improve, make,* and *achieve* are useful for pointing to candidate goals.

Example 5.5 Example 4.2 on page 76 illustrates how the identification of the action words *coordinate*, *track*, and *improve* led to the identification of the goals `Training coordinated`, `Progress tracked`, and `Qualifications improved`.

HIG 6. If a statement seems to guide design decisions at various levels within the system or organization, express it as a goal.

Example 5.6 Example 4.1 on page 75 illustrates how the statement "DoD must ... spend tax payers' money ... more effectively and efficiently" affects decisions throughout the organization, and can be expressed as the goal `Tax payers money spent efficiently`.

Analysts may find this mode of express awkward for this particular goal. This is mainly a consequence of GBRAM stating goals as state expressions. If this creates confusion for the user, the goal may be expressed in a manner with which they feel comfortable.

HIG 7. Goals may be uncovered by examining available information fragments to identify avoidance goals*. Avoidance goals are found by identifying bad states or states that should be avoided within the system.

Example 5.7 (Identifying goals by considering avoidance factors) In a budget system, `Avoid overdrawn account` is an avoidance goal.

HIG 8. Goals can be uncovered or discovered by considering the goal dependencies for the previously specified goals by asking:

(a) What are the pre-conditions† of this goal?

(b) What are the post-conditions‡ of this goal?

Since preconditions and postconditions are expressed as goals in GBRAM, it is possible to identify new goals that had not been previously considered or identified by considering each goal's dependencies.

---

* *Avoidance goals* are satisfied as long as their target conditions remain false.
† The *precondition* characterizes the conditions under which the goal may be achieved.
‡ The *postcondition* characterizes the state of the system once the goal is completed.

Example 5.8 Example 4.22 on page 107 illustrates how goals may be discovered by considering the potential postconditions of a goal.

HIG 9. Stakeholders tend to express their requirements in terms of operations and actions rather than goals [7, 25]. Thus, when given an interview transcript, it is beneficial to apply the action word strategy to extract goals from stakeholders' descriptions.

Example 5.9 (Extracting goals from stakeholder descriptions) In a meeting scheduler system, stakeholders may use action words such as 'schedule' and 'reserve,' which give rise to goals such as `Meeting scheduled` and `Room reserved`.

HIG 10. Analysts should first seek to understand the stakeholder's application domain and goals before concentrating on the actual or current system so that the system requirements may be adequately specified. Previous research indicates that customers tend to express their goals within the context of their application domain, not in terms of an existing or desired system [7].

Example 5.10 The goal of a college financial services system is not to maintain a financial ledger/database (system goal) as typically described by management level stakeholders, but to ensure that, among other requirements, the budget remains balanced; sponsors are charged according to their contracts; and faculty are paid according to state research contracts, as typically described by customers using their application domain vocabulary.

HIG 11. Goals are also identified by considering the possible goal obstacles for previously specified goals.

Example 5.11 Example 4.20 on page 105 illustrates how the goal `Course closed` was identified by considering the possible obstacles for the goal `Available course slots announced`.

HIG 12. Goals may be identified by considering possible scenarios.

Example 5.12 Example 4.21 on page 106 illustrates how an analysis of the scenario "Employee not reminded" facilitates the identification of new goals to handle the exceptional cases represented by the scenarios.

Given each goal obstacle, the analyst should determine whether or not the occurrence of the goal obstacle would cause the system to fail. If the occurrence of the goal obstacle would initiate system failures, these obstacles are key candidates for scenario construction and analysis since the analyst must be sure to specify the goals and requirements to enable the system to handle exceptional cases. Goals may also be identified by considering the normal non-exceptional scenarios.

HIG 13. Goals may be identified by considering constraints.

Example 5.13 (Identifying a goal from a constraint) Several statements in the Career Track Training System (Chapter 3.2; page 45) describe requirements that have to be met in order for a goal to be achieved. Consider the circumstances illustrated in Example 4.25 on page 110. A review of NLD #3 on page 81 leads to the identification of the constraint `Course must qualify employee to advance to another level`; the system must 'know' which courses an employee can take. Thus, the goal

`Courses which employee qualifies for identified`) can be identified from this constraint.

Example 5.14 In the CTTS case study, several statements describe requirements that have to be met in order for a goal to be achieved. For example, consider the CTTS Process Scenario: a review of NLD #3 (shown on page 81) points to the identification of the constraint `Course must qualify employee to advance to another level`. The word 'qualify' was the key indicator that a requirement must be met. Before an employee can advance to a certification level, the course taken must officially qualify them for advancement.

HIG 14. Goals may be extracted from process diagrams by searching for actions and behaviors, as well as by consistently applying the Inquiry Cycle [64] to clarify the goals and requirements.

Example 5.15 (Extracting goals from process diagrams) In the CTTS example (Chapter 3.2; page 45), in-depth interviews were conducted with AFB personnel, as well as with professionals in the training acquisition process, to develop an understanding of the current process. These interactions resulted in the construction of an informal, but detailed flow chart model of the current process; a portion of the flow chart for a career training system is shown in Figure 5.4. This particular flow chart provides a good bit of detail so it is possible to extract goals using action word location as well as by identifying behaviors in the system. Given the level of detail provided in this flow chart, 14 goals were extracted, as detailed in Table 5.2.

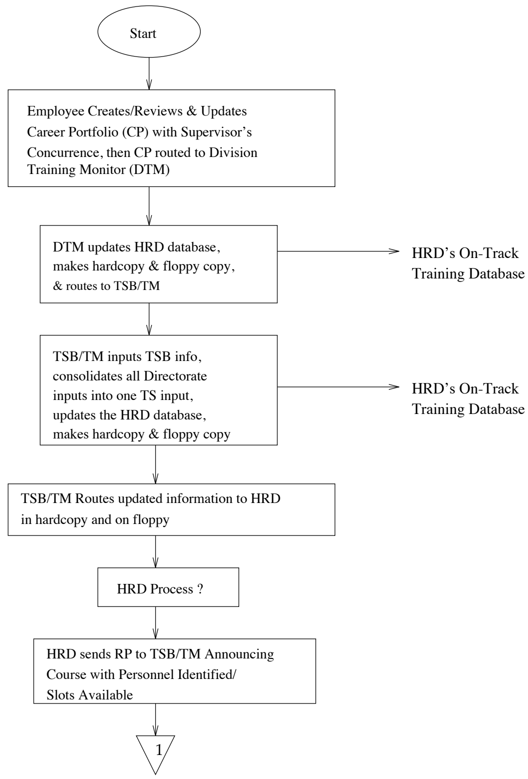

Figure 5.4. Flow Chart for a Portion of the CTTS

Table 5.2. Goals Extracted from the CTTS Flow Chart

Goals	Agent
G_1 : Career portfolio created	employee
G_2 : Career portfolio reviewed	employee
G_3 : Career portfolio updated	employee
G_4 : Career portfolio routed to DTM	employee
G_5 : Database routed to TSD	HRD
G_6 : Database routed to HRD	TSD
G_7 : IP sent to TSD	HRD
G_8 : Database updated	HRD, TSD
G_9 : Database updated with TSD info	TSD
G_{10} : Database hard copied	HRD, TSD
G_{11} : Database floppy copied	HRD, TSD
G_{12} : Directorate inputs consolidated	TSD
G_{13} : HRD process completed	HRD
G_{14} : Course & personnel matched	HRD

The goal identification heuristics offer analysts several approaches for extracting goals, based upon the resources available to the analyst. A summary of these heuristics is provided in Tables C.1 and C.2 (Appendix C; pages 244 and 245). The stakeholder and agent identification heuristics guide the process of considering the stakeholders of the proposed system and the agent responsibilities. The constraint identification heuristics allow analysts to identify conditions that must exist or be met for a goal to be realized. These heuristics are discussed in the following subsections.

Heuristics for Stakeholder Identification

As previously discussed in Chapter 4 on page 78, anyone who claims an interest in a proposed system is a stakeholder, while anyone or anything responsible for the actual completion of goals within an organization or system is an agent. The heuristics for stakeholder identification address the problem of determining who and what parties claim an interest in the proposed system. A stakeholder need not only be a user in the classical sense; rather, any representative affected by the completion or prevention of a goal may be considered a stakeholder. A stakeholder may be a customer, actor, owner, or representative of an organization (Refer to Examples 4.3 and 4.4, found on page 79).

Although goal identification is discussed prior to stakeholder identification, the focus in this thesis and GBRAM is on the activities, not on the sequence of activities. However, there are times when stakeholders must be identified before the goals and objectives can even be established. For example, if the manager of a department in a telecommunications company requests an information system, there may be no existing documentation or description of the desired system. The starting point for the analyst thus becomes an in-depth interview with the telecommunications department manager or "customer" to determine who the stakeholders are so that the analyst may begin interviewing those persons. Requirements elicitation is not the focus of this dissertation; however, this has been addressed extensively in other literature [22, 23, 28, 33, 35].

Stakeholders are identified to determine which agents claim a stake in each goal and to develop an understanding of the different viewpoints involved in the system for conflict resolution. Stakeholder identification is important for conflict resolution and for prioritizing goals in that capturing stakeholders' viewpoints allows conflicts to be detected early. Analysts should consider the role that stakeholders play in prioritizing goals during negotiation [65]. In the event of conflicting requirements, the respective goals should be used to guide the process of resolving the conflict. While a balanced resolution is not always possible, the goals may have different priorities or 'importance' ratings that may guide the process to determine the most feasible trade-off.

As illustrated in Figure 5.5, stakeholders may be identified before and/or after goal identification, enabling conflicts to be surfaced early. When a conflict is detected, or when there are multiple stakeholders, the analyst should apply the inquiry cycle, using some of the questions in this section to clarify the goals and the reason for conflict.

HIS 1. Systems or subsystems which do not involve multiple stakeholders may not require stakeholder identification; analysts may choose to skip stakeholder identification entirely in these systems.

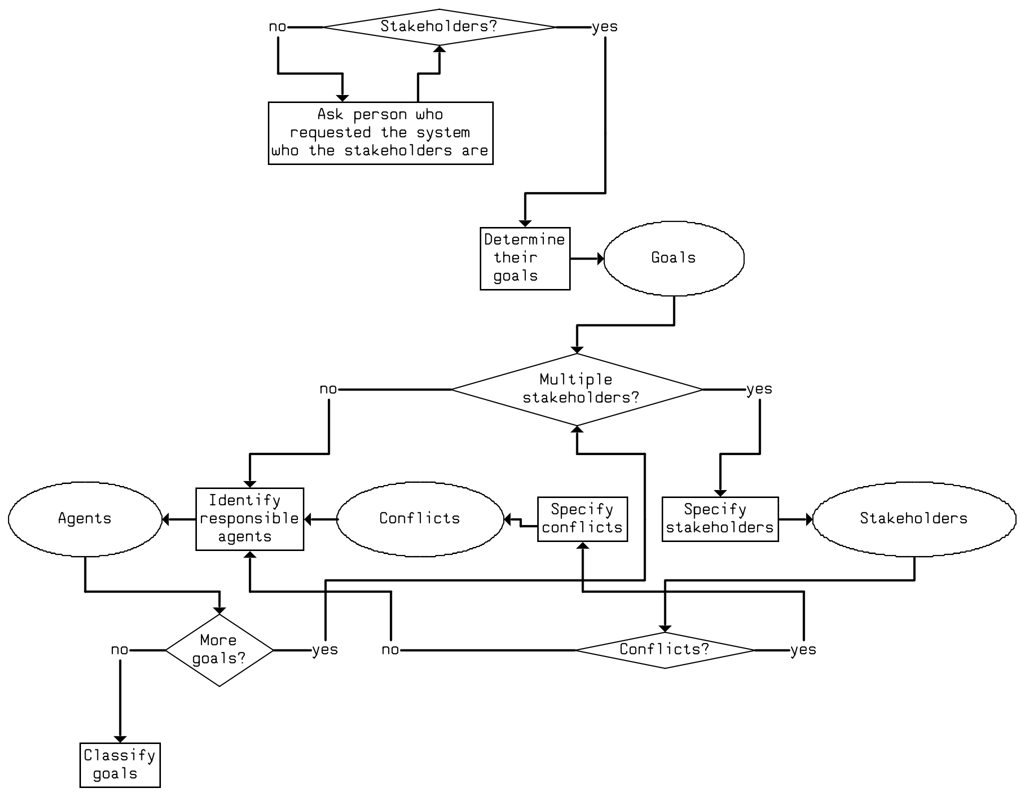

Figure 5.5. Control Flow Chart for Stakeholder and Agent Identification

Example 5.16 In GBRAT (Detailed in Chapter 6.1; page 187), there was only one stakeholder at the time of development. Thus, stakeholder identification was deemed unnecessary.

HIS 2. Multiple stakeholders may be associated with one goal.

Example 5.17 Example 4.3 on page 79 illustrates this association of multiple stakeholders with one goal. In this example, the goals `Skills improved` and `Career tracks provided` both have two stakeholders (AFB and employee).

If different stakeholders are associated with a goal but their associations occur at different times within the life of the system, the analyst should document these variances to ensure that the role of stakeholders throughout the lifetime of a goal or the system is well understood.

HIS 3. Any representative affected by the completion or prevention of a goal is a stakeholder. A customer or person representing the enterprise requesting the system or an analysis effort is a stakeholder. Users of the proposed system are stakeholders. Stakeholders are thus identified by asking:

(a) Who or what claims a stake in this goal?

(b) Who or what stands to gain or lose by the completion or prevention of this goal?

(c) Who will use the system?

Example 5.18 Consider the vacation/sick leave problem description in Appendix A. `Hour tracked` has more than one stakeholder and these stakeholders are identified by asking the questions above. The stakeholders for this goal are: academic institution, employee, financial services office, and payroll office.

The goal identification heuristics allow analysts to extract goals from various sources of information; stakeholder identification heuristics allow analysts to consider all parties who claim an interest in the system. A summary of the heuristics for stakeholder identification is provided in Table C.3 (Appendix C; page 246). Agents must be identified so that responsibility for ensuring the achievement of a goal at any given time may be determined and assigned. The following subsection presents the agent identification heuristics.

Heuristics for Agent Identification

Every goal has at least one responsible agent, be it a person, organization, or even the system. The agent identification heuristics presented below address the problem of determining who or what the analyst should allocate responsibility to for the goals.

HIA 1. At least one agent must be responsible for the completion of each goal. If the analyst is unable to allocate responsibility for a goal to any agent, then the analyst can assume that the goal lies outside the scope of the proposed system. If the analyst believes there is a responsible agent, but doesn't know who or what, then the inquiry cycle should be applied using the Who-is question.

HIA 2. Responsible agents may be identified by considering each goal and asking:

(a) Who or what agent *is*, *could be*, or *should be* responsible for this goal?

The answer to this question will be the name of the responsible agent. The agent's name should be 'attached' to the goal for which it is responsible; Table 4.4 on page 81 illustrates how agents can be attached to goals using a tabular notation.

Example 5.19 Example 4.5 on page 81 illustrates how the responsible agent (employee) was identified for the goals `Course completed` and `Proof of course completion submitted` by asking the questions in HIA 2 for the goals

Example 5.20 Consider the bug evolution description in Appendix B. For each goal, answer the question: *Who or what agent is responsible for this goal?* For example, `bug` is responsible for the goal `Bacteria eaten`.

HIA 3. Different agents can be responsible for the completion of the same goal at different times.

HIA 4. Agents may be either the system, organization, or a human agent.

Example 5.21 In a meeting scheduler system the goal `Meeting scheduled` is the responsibility of the meeting scheduler. Depending on the desired implementation, the agent may be either the automated system or a human agent. In an email-based implementation of this system the responsible agent could be someone other than the automated meeting scheduler system such as a member of the clerical staff.

Agents must be identified so that responsibility for ensuring the achievement of a goal at any given time may be determined and assigned. A summary of the heuristics for agent identification is provided in Table C.3 (Appendix C; page 246). The following subsection presents the heuristics for constraint identification.

Heuristics for Constraint Identification

A constraint, as discussed in Chapter 4 on page 110, places a condition on the achievement of a goal. In GBRAM, statements which seem to be independent of other goals or requirements should be stated as constraints. The heuristics for identifying constraints are presented below.

HIC 1. Constraints can be identified by considering each statement and asking:

(a) Does this fragment impose some constraint on the goal(s)?

(b) Does this statement specify some requirement that must be met?

Given an answer of 'yes' to either of these two questions, restate as a constraint every statement that exemplifies or states a requirement which must be met to achieve some goal.

Example 5.22 Example 4.25 on page 110 illustrates a such a case where a statement specifies a requirement that must be met. In this example, the word *qualifies* is a key indicator of a condition that must be met.

HIC 2. Constraints can be identified by searching for temporal connectives (i.e., *during, before, after,* etc.). Restate statements that describe *when* some condition is true or *when* a goal can be completed as a constraint.

Example 5.23 Example 4.24 on page 110 illustrates a how the identification of the temporal connective *during* led to the identification of the constraint: Meeting room must be available during the meeting date/time.

HIC 3. Constraints can be identified by searching statements which place limits on the completion of a goal.

Example 5.24 Interview transcripts with stakeholder for the meeting scheduler system were analyzed to identify possible constraints. The constraint `At least 35 participants should be handled efficiently` places a limit or minimum threshold for the system. This constraint can eventually be mapped into a non-functional requirement for the system.

HIC 4. Since constraints may place a condition on the achievement of a goal, they should be restated as goal obstacles to allow for subsequent elaboration of the obstacle using scenario analysis. This enables the consideration of exception cases which the system is required to handle.

Example 5.25 A constraint places a condition on the achievement of a goal. For example, the constraint `Member must be able to ascertain if product was previously purchased` in the Electronic Commerce Web Server, discussed in Chapter 6.2 on page 193, places a condition on the achievement of the goal `AVOID duplicate purchase`.

A summary of the heuristics for constraint identification is provided in Table C.3 (Appendix C; page 246). The following section presents the heuristics for goal classification.

5.3 Goal Classification Heuristics

The goal classification heuristics in this section address the problems of determining the type of each goal. GBRAM differentiates among types of goals according to their target conditions, classifying goals as either *achievement** or *maintenance goals*†. In general, it is best to classify a goal at the moment it is initially identified, since it is convenient and more efficient to examine each goal once, upon identification, instead of revisiting each goal at a later date for classification. Thus, analysts are encouraged to perform the classification activity concurrently with the goal identification activity. When analysts experience difficulty in classifying a goal as either an achievement or a maintenance goal, it may be because the goal is either an organizational/policy level goal or a quality goal (e.g. `SPEEDUP time required to process claim`). This section presents the heuristics for classifying goals as either achievement or maintenance goals.

Classifying Achievement Goals

Achievement goals generally map to actions that occur in the system. Thus, they are helpful in identifying functional requirements in the system. The heuristics for classifying goals as achievement goals are presented below.

HCA 1. **Goals are classified as achievement goals by considering each goal and asking:**

(a) Is completion of this goal self-contained?

* *Achievement goals* are satisfied when their target conditions are attained.

† *Maintenance goals* are those goals which are satisfied while their target condition remains constant or true.

(b) Does this goal denote a state that has been achieved or a desired state?

(c) Does the completion of this goal depend on the completion of another goal?

(d) Is the ability of another goal to complete depend upon the completion of this goal?

Given an answer of 'yes' to any of the questions above, classify the goal as an achievement goal.

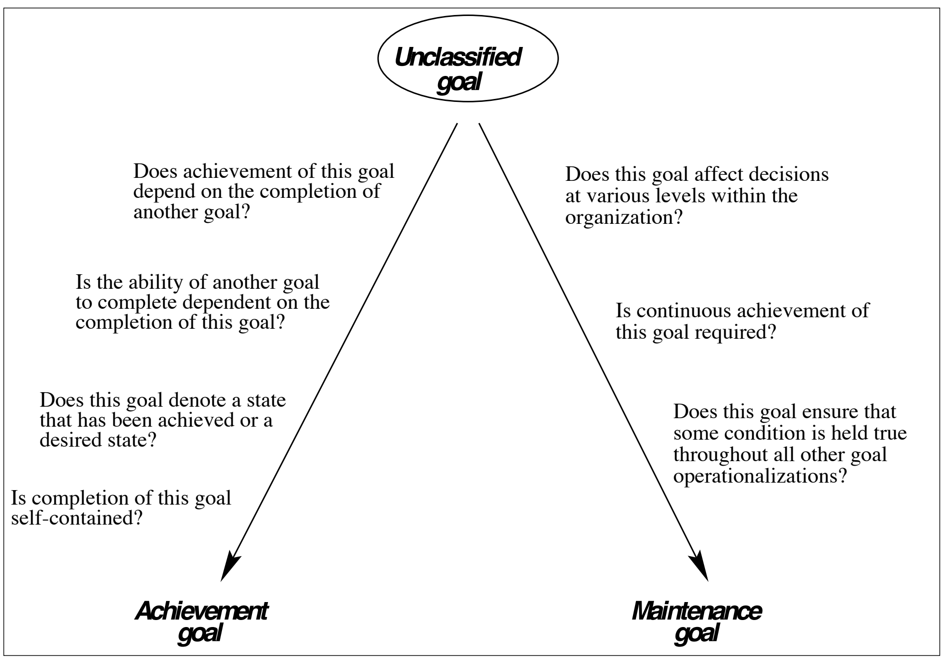

Figure 5.6. Classifying Goals

HCA 2. Achievement goals can be identified by searching for key words representing desired behaviors within the system (i.e., *make, improved, speed up, increase, satisfied, completed, allocated,* etc.)

Example 5.26 Recall that achievement goals are objectives of the system that are named by verbs such as *make*. Consider a seminar registration system that may need to satisfy the goal of enrolling organization members for a seminar before the actual seminar begins. The object of the goal is seminar registration, thus the goal would be named `MAKE member registered`.

HCA 3. **Achievement goals are relatively self-contained. While other goals may depend on the completion of the given goal, achievement goals rarely impose constraints upon an entire class of goals (e.g. a group of security and access goals). In contrast, a Maintenance goal is likely to impose a constraint upon an entire class of achievement goals.**

A summary of the heuristics for classifying achievement goals is provided in Table C.5 (Appendix C; page 248). The following subsection presents the heuristics for classifying maintenance goals.

Classifying Maintenance Goals

A maintenance goal is satisfied as long as its target condition remains true. Since maintenance goals suggest a continuous state within the system, they may generally be mapped to nonfunctional requirements*. Not all maintenance goals map to nonfunctional requirements, some generally map to safety requirements. The heuristics for classifying goals as Maintenance goals are presented below.

***Nonfunctional requirements* describe the nonbehavioral aspects of a system, capturing the properties and constraints under which a system must operate.

HCM 1. Goals are classified as maintenance goals by considering each identified goal and asking:

(a) Does this goal ensure that some condition is held true throughout all other goal operationalizations?

(b) Does this goal affect decisions at various levels within the organization?

(c) Is continuous achievement of this goal required?

Example 5.27 In the Career Track Training System (Chapter 3.2, page 45), the goal G_2 (`Tax payers money spent efficiently`) must be achieved on a 'continuous' basis. The system mandates that career tracks be provided in order to ensure that tax payers' money is spent efficiently. This goal characterizes a condition which must be held true.

HCM 2. Maintenance goals can be identified by searching for key words that suggest a continuous state within the system (i.e., *keep, ensure, avoid, know, monitor, track, provide, supply,* etc.).

Example 5.28 Example 4.1 on page 75 illustrates how the word *provided* allowed for the identification of two maintenance goals, one pertaining to the provision of training courses and another pertaining to the provision of career tracks.

Example 5.29 Example 4.2 on page 76 illustrates how the words *tracked* served as an indicator for the maintenance goal `Progress tracked`.

HCM 3. Maintenance goals tend to be operationalized as actions that prevent certain states from being reached within the system. Since maintenance goals are those goals which are satisfied while their target condition remains true, they are named using the verbs MAINTAIN, KEEP, AVOID and ENSURE.

Example 5.30 The goals in Table 4.6 (page 87) are examples of maintenance goals.

A summary of the heuristics for classifying maintenance goals is provided in Table C.5 (Appendix C; page 248). The following subsection presents the GBRAM goal refinement heuristics.

5.4 Goal Refinement Heuristics

GBRAM refinement heuristics allow analysts to prune or refine the size of the goal set and to clarify stakeholders' goals. GBRAM provides three sets of refinement heuristics:

- heuristics for the elimination of redundancies;
- heuristics for the reconciliation of synonymous goals; and
- heuristics for the refinement of system-specific goals.

Figure 5.7 illustrates a likely sequence of goal refinement activities. In this control flow chart, the goals are refined after the goal set has been ordered according to precedence relations. When analysts have a small goal set to work with it may be convenient to apply the refinement heuristics before applying the dependency heuristics. However, for large goal sets it may be more convenient to apply the refinement heuristics after ordering the goals, so that the investment of time can be greatly reduced due to the clustering effect exhibited by ordered goals, as detailed in Chapter 4. This section presents the three sets of heuristics for refining the goal set.

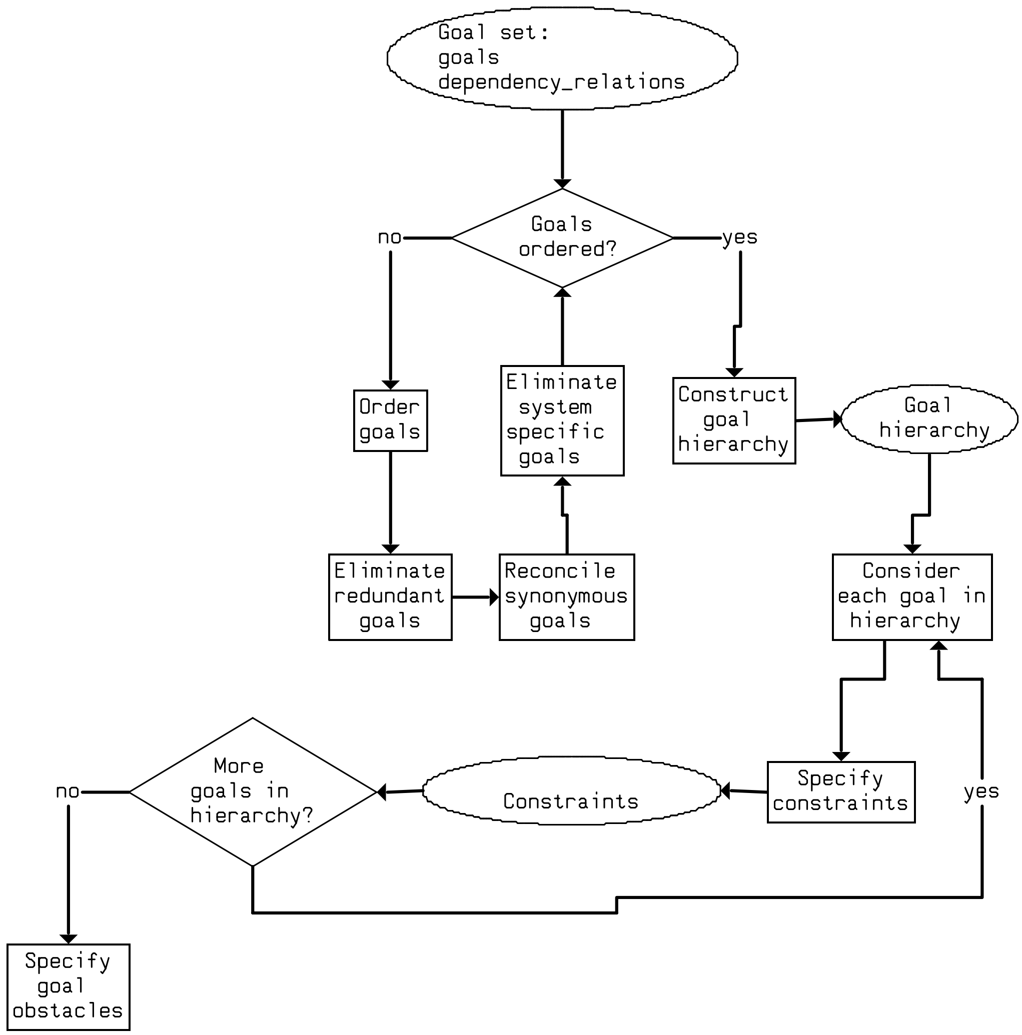

Figure 5.7. Control Flow Chart for Refining Goal Set

Heuristics for the Elimination of Redundancies

Analysts should examine each occurrence of redundant goals according to the following heuristics and resolution strategies.

HRR 1. If the same goal appears more than once AND the same agent is responsible for the goal on each occurrence, then all but one of the goals may be eliminated.

HRR 2. If the same goal appears more than once but two or more different agents are responsible for the same goal at different times, then all but one occurrence of the goal should be eliminated. However, to prevent the loss of information the analyst must keep track of all current and future agents who assume responsibility for the goal.

A summary of the heuristics for refining goals by eliminating redundancies is provided in Table C.6 (Appendix C; page 249). The following subsection presents the GBRAM heuristics for reconciling synonymous goals.

Heuristics for the Reconciliation of Synonymous Goals

The objective of the heuristics for the reconciliation of synonymous goals is to ensure that a common terminology is agreed upon by the stakeholders and to clarify, whether or not there is a shared understanding for goals which stakeholders may refer to by different names. Analysts can apply the inquiry cycle and the questions discussed in Section 5.1 to determine which goals are candidates for subsuming other synonymous goals. The heuristics for reconciling synonymous goals are presented below.

HRS 1. If two goals are synonymous, eliminate one of them.

Example 5.31 Example 4.6 on page 83 illustrates how the goal `Skills improved` can be semantically subsumed by the synonymous goal `Qualifications improved`.

HRS 2. If two synonymous goals are heterogeneous (e.g. one is an achievement goal and the other is a maintenance goal), then it is likely that the maintenance goal was classified incorrectly. If a maintenance goal is synonymous with an achievement goal, then the maintenance goal should be decomposed into an achievement goal. If the maintenance goal is decomposed into more than one achievement goal, then at least one of the goals should be synonymous with the originial achievement goals, and thus synonymous with the maintenance goal.

HRS 3. Consolidate and refine goals by merging synonymous goals.

Example 5.32 Example 4.7 on page 84 illustrates how the goals `Training coordinated` and `Training provided` in the Career Track Training System are synonymous and can be consolidated into one goal.

HRS 4. Since synonymous goals tend to share precedence relations, they appear clustered together when ordered. Ordering goals according to their precedence relations thus facilitates the identification of synonymous goals.

A summary of the heuristics for reconciling synonymous goals is provided in Table C.6 (Appendix C; page 249). The following subsection presents the GBRAM heuristics for refining system-specific goals.

Heuristics for the Refinement of System-Specific and Information Dissemination Goals

A goal-based model is an abstraction of *what* the desired system must do, not *how* it will be done. The goals in the model should be application-domain concepts and not computer implementation concepts, such as data structures. A good model may be readily understood by both programmers and application experts who are not programmers. The analysis model should not contain implementation decisions. Recall that GBRAM is a requirements analysis method, not an information modeling method; thus, GBRAM focuses on what the goals and requirements are and their attributes and operations rather than on specific data entities. For example, a workstation windowing system would be described in terms of attributes and operations visible to the user, not in terms of the actual implementation solution. The heuristics for refining system-specific goals are presented below.

HRSS 1. Individual information dissemination goals may be refined by asking:

(a) What is 'information' and why is it significant or important?

(b) Do any goals depend on the availability of this information for goal achievement?

Restate goals based on system-specific entities to capture the essence of the goal without including any system-specific information. If a goal is based on system-specific entities, it should be eliminated (e.g. `database updated` is a system-specific goal).

Example 5.33 Goals based on system entities should be refined. We initially identified the goal `HRD process completed` which clearly called for refinement due to the need to develop an understanding of what the 'HRD process' entails. By asking the stakeholder *"What is 'HRD process,' and why is it significant?"* it was possible to define `HRD process` and determine that the goal (`HRD process completed`) can be decomposed into two goals: `Available course slots announced` and `Qualified personnel identified`. Thus, in this case, one goal was eliminated and replaced with two new goals.

When an implementation bias exists (i.e., if the customer has requested a certain implementation platform) then it may not be possible to ignore system-specific information.

Example 5.34 Due to the nature of the meeting scheduler, many of the goals in this system pertain to dissemination. For example, if the meeting scheduler were implemented as a messaging system in which meetings are scheduled via email messages, then the use of system-specific or information dissemination goals cannot be avoided.

HRSS 2. Restate routing goals to avoid emphasizing the receiving party and so that the underlying process and activity is represented.

Example 5.35 In the Career Track Training System (Section 3.2; page 45), the goal `Career portfolio routed to DTM` concerns the dissemination of some information. This goal is restated so that it reflects the underlying process: `Employee prefs made available`. In order to determine what the contents of a Career portfolio are to define DTM, the analyst must apply a few of the questions discussed in Section 5.1. Notice that all references to information disemination have been eliminated from the goals.

Example 5.36 Similarly, the goal `IP sent to TSD` is restated so that the goal reflects the underlying process: `IPs made available`. As illustrated in the previous example, the questions discussed in Section 5.1 must be applied in order to clarify what IP and TSD are. By asking the stakeholder *"What is 'TSD info' and why is it significant?"* it is learned that `TSD info` and `directorate inputs` are the collection of IPs that have been submitted by employees. Each IP contains two lists: a list of all courses taken by an employee and a list of the additional training courses the employee wishes to take.

A summary of the heuristics for refining system-specific goals is provided in Table C.6 (Appendix C; page 249). The strength of GBRAM lies in its focus on goals and objectives and the derivation of operational requirements from those goals. The next section focuses on the heuristics which aid in the elaboration of goals.

5.5 Goal Elaboration Heuristics

Goal elaboration is the process of adding practical detail to the goals identifed from intially simplified descriptions. Analysts begin the goal elaboration process by considering the dependency relationships that exist between goals; this process facilitates the consideration of possible ways in which goals may be blocked or failed so that exceptional cases may be anticipated. There are three sets of GBRAM elaboration heuristics employed by analysts:

- heuristics for considering goal dependencies;
- heuristics to guide obstacle analysis; and
- heuristics to guide scenario analysis.

The objective of the goal dependency heuristics is to guide the analyst in considering goal relationships so that the goals may be ordered in the goal hierarchy. The goal obstacle heuristics allow analysts to consider the possible ways for goals to fail, facilitating the anticipation of exception cases. Scenario analysis heuristics allow analysts to evaluate changing goal priorities. Goals are further elaborated by considering the possible ways in which they may be blocked and by identifying scenarios to develop an understanding of how the goals may be operationalized. These three sets of goal elaboration heuristics are presented in the following subsections.

Heuristics for Considering Goal Dependencies

To gain a better understanding of the 'big picture,' it is helpful to model the relationships between goals. In Chapter 4 the identification of goal obstacles and scenarios to enable the consideration of such relationships was discussed. Recall that *refinement* refers to the process of pruning the goal set, while *elaboration* refers to the process of acquiring more detailed information about the goals and identifying or uncovering new goals for the goal set.

GBRAM recognizes three kinds of dependency relations among goals: *precedence dependency**, *contract dependency*†, and *agent dependency*‡. Identification of these dependency relationships facilitates ordering the goals en route to constructing a goal hierarchy. Goal dependencies also assist analysts in the essential task of identifying pre- and post-

---

*A *precedence dependency* exists between goals G_1 and G_2 if goal G_1 must be completed before goal G_2.

†A *contract dependency* exists between goals G_1 and G_2 when goal G_2 must be achieved if goal G_1 occurs.

‡An *agent dependency* exists between goals G_1 and G_2 when the agent responsible for goal G_1 must complete the goal in order for the agent responsible for goal G_2 to complete G_2.

conditions during goal elaboration. As previously mentioned, the GBRAM goal dependency heuristics are considered to be both refinement and elaboration heuristics. Figure 5.7 on page 166 illustrates how goal refinement relies on the goal dependency heuristics depending on the analysts' chosen order of activities when applying GBRAM. A set of questions guides the identification of goal dependencies through this process. Figure 5.8 illustrates, at a high level, a possible sequence of application of these questions. Again, this flow chart is an approximation of the method. Each of the diamonds in this figure represents an inquiry point during which the analyst can apply the questions discussed in below in association with the heuristics as well as follow-up questions as suggested in Section 5.1. Goals must be organized according to their precedence relations for construction of the goal hierarchy. Specifying goal dependencies is thus a prerequisite activity for ordering the goals. The goal dependency heuristics are presented below.

HED 1. An effective way to discover precedence dependencies between goals is to consider each goal and ask:

(a) What goals are the prerequisites for this goal?

(b) What goal(s) must follow this goal?

The answers to these questions indicate the given goal's precedence relations, and should be documented by the analyst so that the goal may be subsequently ordered in accordance with these relationships. Table 5.3, found on page 175 and Chapter 4, 88 detail methods which analysts may employ for these annotations.

Example 5.37 Example 4.11 on page 89 illustrates how consideration of the precedence dependencies for the goal `Course completed` in the Career Track Training System

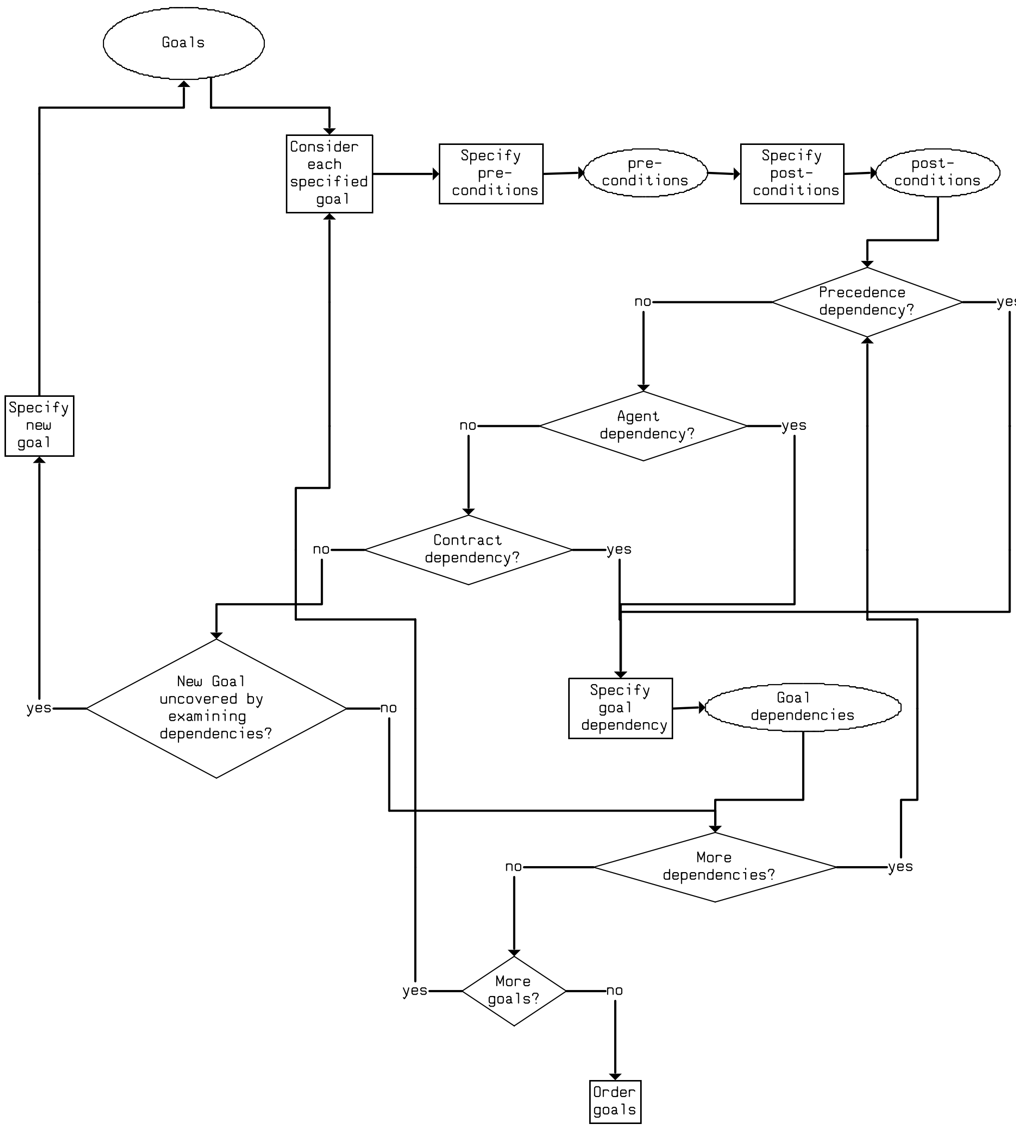

Figure 5.8. Control Flow Chart for Specifying Goal Dependencies

leads to the clarification of the relationship between this goal and the goals `Skills improved` and `Certification granted`.

Example 5.38 In the CTTS the prerequisites for each goal were considered. For example, before a career portfolio can be reviewed, or updated, it must be created. Table 5.3 shows the ordering that resulted from the analysis of the dependency relations for G_2 (`Career portfolio reviewed`). By asking *"What are the prerequisites for this goal?"* it becomes evident that $G_1 < G_2$.

HED 2. **An effective way to identify contract dependencies is to consider each goal and ask:**

(a) What goal(s) *must* be completed if this goal is achieved?

Example 5.39 Example 4.12 on page 89 illustrates how a contract relation may be observed between the goals `Course completed` and `Skills improved`. If the course is not completed, the employees skills are not improved and both goals fail; thus there exists a contract dependency between the two goals.

Table 5.3. Ordered Achievement Goals

Achievement Goals	Agent	Stakeholders
G_2 : Career portfolio reviewed	employee	employee, supervisor
G_3 : Career portfolio updated	employee	employee, supervisor
G_4 : Career portfolio made available	employee	employee, DTM
G_5 : Employee course prefs ready	employee	employee
G_6 : IPs made available	HRD	HRD, employee

Example 5.40 Example 4.12 on page 89 illustrates how contract dependencies exist between the goals `Course completed` and `Skills improved` in the Career Track Training System (Chapter 3.2; page 45).

HED 3. An effective way to identify agent dependencies between goals is to consider each goal and ask:

(a) What agent must complete the goal(s) they are responsible for before the agent responsible for this goal can achieve this goal?

Example 5.41 Consider a payroll system; before an employee can be paid, the employee's supervisor depends on the the employee to provide him with a record of the number of hours the employee worked (e.g. a time sheet).

Example 5.42 Consider goal G_2 in Table 5.3. From this goal, it is clear that an agent dependency between the supervisor and the employee exists. The supervisor depends on the employee to provide him the necessary information which enables the supervisor to decide whether or not to approve the employee's career portfolio course preferences. IP availability is similarly contingent upon this flow of information. Thus in Table 5.3, $G_3 < G_4 < G_5$.

The inquiry points for HED 2, HED 3, and HED 4 are represented by the diamonds in Figure 5.8; the questions discussed in Section 5.1 also assist the analyst in determining the goal relevant information pertaining to goal dependencies. A summary of the heuristics for considering goal dependencies is provided in Table C.7 (Appendix C; page 250). The goal dependency heuristics presented in this section enable analysts to elaborate the goal set by determining the relationships between goals and agents so that goals may be ordered and refined. Additionally, by considering the goal dependencies in a methodical manner, it is likely that analysts will recognize the need for additional goals that had not been previously identified and which may be necessary for the given goal to be realized.

Once the goal dependencies are established, the process of identifying obstacles is facilitated because each goal that fails directly affects the goals which depend upon it. The obstacle and scenario analysis heuristics serve to guide analysts as they elaborate the goal set and uncover new goals and requirements. The following section focuses on the heuristics which aid in the elaboration of goals via the analysis of goal obstacles.

Heuristics to Guide Obstacle Analysis

The objective of the obstacle analysis heuristics is to uncover hidden goals and requirements by considering the possible ways in which goals can fail. Figure 5.9 illustrates, at a high level, the relationship between obstacle and scenario analysis. The possible ways that goals may be blocked are considered by assigning a trivial obstacle to each goal. The obstacle analysis process forces analysts to consider specific cases that must be handled due to activities which prevent goal completion.

HEO 1. There is at least one goal obstacle for every goal. This is informally referred to as the trivial obstacle and formally referred to as the normal first case goal obstacles. These obstacles are worded by negating the verb in the goal name.

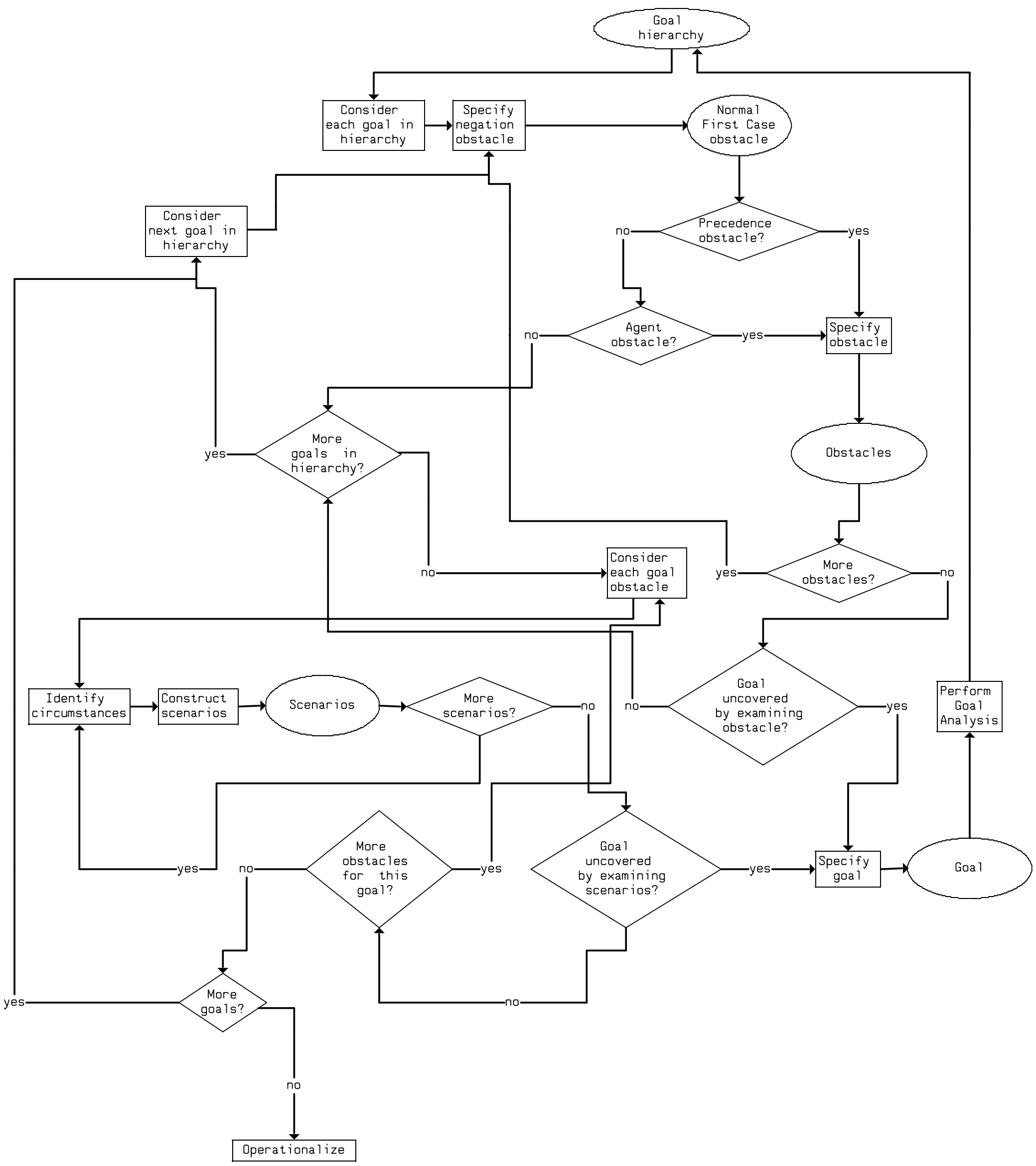

Figure 5.9. Control Flow Chart for Obstacle and Scenario Analysis

Example 5.43 Example 4.18 on page 101 illustrates how trivial obstacles enables analysts and stakeholders to consider specific cases that must be handled due to activities which prevent goal completion.

HEO 2. A statement that illustrates a condition which prevents the completion of a goal or which illustrates an example of a goal being blocked by another goal is indicative of an obstacle and should be expressed as an obstacle.

Example 5.44 Example 4.13 on page 97 illustrates how a statement which imposes a condition on an identified goal provides insights into potential goal obstacles.

HEO 3. An effective way to identify goal obstacles is to consider each goal and ask:

(a) What other goal(s) or condition(s) does this goal depend on?

(b) What other goal(s) must be completed or achieved in order for this goal to be achieved? (precondition)

(c) What goal(s) depend on this goal? (postcondition)

(d) What goal(s) must follow from this goal? (postcondition)

(e) Can the failure of another goal to complete cause this goal to be blocked?

(f) If this goal is blocked, what are the consequences?

The answer to the questions above should be worded to emphasize the state that is true, thereby denoting a goal obstacle.

Example 5.45 By asking *"What other goal(s) or condition(s) does this goal depend on?"* it was possible to determine that the goal `Career portfolio made available` in the Career Track Training System is dependent upon whether or not the supervisor's concurrence is obtained. Thus, using the "state as true" naming convention, `Supervisor's concurrence not obtained` gives rise to the obstacle `Career portfolio not made available`.

HEO 4. A prerequisite failure obstacle occurs when a goal having a precedence relation is obstructed because the precedence goal fails. Prerequisite failures are identified by considering each goal and asking:

(a) What other goal(s) does this goal depend on?

Example 5.46 Example 4.15 on page 99 illustrates how, prerequisite obstacles for the goal Career Track Training System `Qualified personnel identified` may be determined by examining the goal's precedence relations (e.g. `Preferences made available`).

HEO 5. An agent failure obstacle occurs when a goal fails because the responsible agent fails to achieve the goal. Agent failures are identified by considering each goal and asking:

(a) Can the failure of an agent to fulfill their responsibilities cause this goal to fail?

Example 5.47 Example 4.16 on page 100 illustrates how an Air Force Base (AFB) employee failing a course is an agent failure which prevents the goal `Certification granted` from being completed in the Career Track Training System.

Example 5.48 Example 4.14 on page 98 illustrates how the failure of an agent to perform a goal he/she is responsible for can prevent another goal from being completed.

HEO 6. An contract failure obstacle occurs when a goal which holds a contract with another goal fails. Contract failure obstacles are identified by considering each goal and asking:

(a) Does this goal share a contractual relation with another goal?

Example 5.49 In example 4.17 on page 101, the goals `Course completed` and `Skills improved` in the Career Track Training System share a contractual relationship. The example illustrates how a contract failure may cause a contractual goal to fail (i.e., if `Course completed` is not achieved, the goal `Skills improved` is not achieved).

A summary of the heuristics for goal obstacle analysis is provided in Table C.7 (Appendix C; page 250). While each of these activities (obstacle and scenario analysis) were treated separately in Chapter 4, the activities are complementary. Obstacle identification begins to identify ways in which goals can fail; this information is elaborated further via scenarios. The following section focuses on the heuristics which aid the analyst in constructing and analyzing scenarios.

Heuristics to Guide Scenario Analysis

Scenario analysis heuristics assist analysts in uncovering hidden goals by considering both non-special cases in the system and the circumstances under which goals may fail within the system. Obstacles denote the *reason* why a goal failed. Scenarios denote concrete *circumstances*, for example, those under which a goal may fail. Some scenarios thus instantiate goal obstacles. This section presents the heuristics to guide scenario analysis.

HES 1. An effective way to identify candidate scenarios for construction is to consider each goal and ask:

(a) What happens if this goal is not achieved?

(b) What are the circumstances under which this obstacle can occur?

The identified scenarios are elaborated by listing the activities that must occur should the scenario actually take place. The scenarios may be represented either as a simple list of actions or as the list of actions accompanied by the agent responsible for each action.

Example 5.50 Consider the Bugs problem description in Appendix B. One of the possible goals to elaborate is: `Weight gained`. The goal is elaborated by constructing a scenario which lists the activities that must occur in order to achieve the goal:

```
1. Bug moves to new grid location
2. Bacteria available
3. Bug eats bacteria
4. Bug gains weight
```

Example 5.51 In the CommerceNet Web Server (Chapter 6; page 193), the scenarios were represented as a list of agents and actions as shown below in the Scenario entitled "Member Navigates Web Pages":

```
1. Member:    Logs in members-only web page
2. Member:    Visits personal "What's New" and the
              "search" Web page
3. Member:    Submits query about payment negotiation
              protocol
4. CN Server: Gets member's preference data
5. CN Server: Searches for related web pages based on
              the preference
```

```
 6. CN Server: Generates search results web page
 7. CN Server: Responds with search results web page
 8. Member:    Visits web page X from search results web
               page
 9. Member:    Logs out
10. CN Server: Records that web page X has been visited
               by member in preference data
```

HES 2. Another effective way to identify candidate scenarios is to consider each obstacle and ask:

(a) Why did this obstacle occur?

(b) Why was this goal not achieved?

(c) Under what circumstances would this obstacle occur?

Example 5.52 Example 4.20 on page 105 illustrates how asking the questions above, with respect to the obstacle `No slots available`, led to the identification of two scenarios: `2.a All courses closed (max capacity reached)` and `2.b Course cancelled (no slots available)`.

HES 3. The scenarios which analysts should provide particular or special attention to are those which violate goals or obstacles. Scenarios should be analyzed by considering the possible ways in which goal obstacles could be prevented. This process leads to the identification of new goals and requirements for the system.

5.6 Summary

This chapter presented four sets of GBRAM heuristics and a set of recurring question types which aid analysts in applying an inquiry-driven approach. The GBRAM heuristics detailed in this chapter were:

- identification heuristics;
- goal classification heuristics;
- refinement heuristics; and
- elaboration heuristics.

The identification heuristics assist analysts in identifying goals, stakeholders, responsible agents, and constraints. Goal classification heuristics aid analysts in determining whether a goal is an achievement or maintenance goal. Refinement heuristics assist analysts in pruning the size of the goal set by eliminating redundant goals and reconciling synonymous goals. GBRAM elaboration heuristics address the need to consider goal dependencies, suggesting the goal obstacles for which scenarios should be constructed and which scenarios to elaborate. A summary of four heuristics sets is provided in Appendix C, beginning on page 243.

The following chapter discusses validation of the Goal-Based Requirements Analysis Method via its application to a large industrial case involving the reengineering of an electronic commerce Web server and through utilization of the method in an empirical investigation.