Discovery

CHAPTER

6

Discovering Initial Requirements

You may have arrived at the discovery phase from various previous points. There may have been a prior business process engineering or reengineering effort. Or you may have completed the scoping steps outlined in Chapter 4. Regardless, the discovery phase begins when there is a defined project scope to develop a target information system, a business commitment to build the information system, a detailed plan for how to proceed, and a desire to develop the information system with a business rules approach.

Figure 6.1 reminds you of where the discovery phase fits with respect to the other phases in the methodology. Note that the discovery phase addresses the discovery of requirements in four tracks: process, data, technology, and rules. This chapter focuses

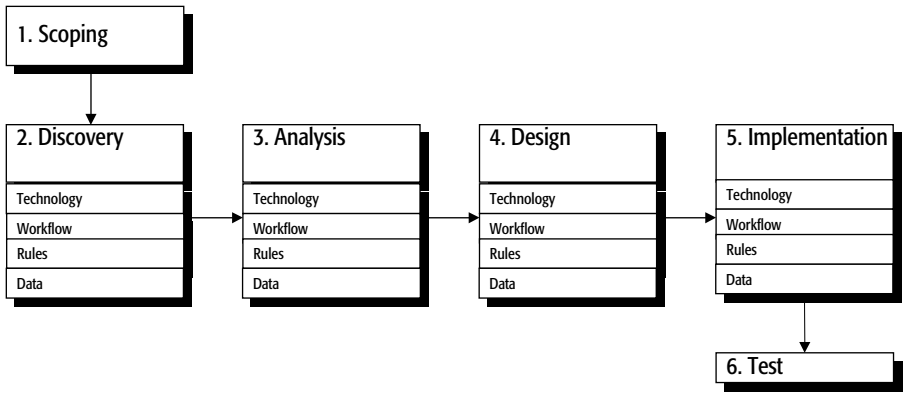

Figure 6.1 Business rule systems methodology phases.

mostly on the discovery of requirements in the process track as a starting point. Because the discovery process in this book places new emphasis on the rules track, you will want to address the concepts in Chapter 15 for managing rules.

What Is the Discovery of Initial Requirements?

In this book, the discovery of initial requirements means gaining a preliminary understanding of four aspects: potential process flow (most likely through documenting use-case descriptions), instances for testing purposes (through collecting concrete scenarios), intellectual decision-making behind the process (through capture of decisions), and the potential for sharing information (through developing a conceptual model).

For tutorial purposes, and to provide a new emphasis on rules, this book separates the discovery of process-oriented (who, when, where, and how) requirements from data (what) and rules (why). In reality, you may discover them all at once, most likely over several iterations.

How Is Discovery of Initial Requirements Different for a Business Rules Approach?

Most of you will notice that the first two steps in this chapter (create use-case descriptions and identify concrete scenarios) are similar to how you would begin discovering requirements for most object-oriented and even non-object-oriented development efforts. The third step, perhaps may seem new to you. It represents the discovery of decisions, rather than moving directly to an initial discovery of objects or of sequences of responsibilities among objects. This shift in emphasis occurs because of six subtle differences in discovering initial requirements for a business rules approach outlined in Table 3.1:

- Separating rules by decomposing business events into business decisions.
- Tracing rules by correlating decisions to business context (organizational policies, strategies, objectives, goals).
- Tracing rules by associating decisions and rules with use cases.
- Externalizing rules by identifying concrete scenarios.
- Positioning rules for change by correlating rules to information referenced and created.
- Positioning rules for change by avoiding premature commitment to execution sequence.

Let's look at each of these differences.

The first, most unique difference is the decomposition of a business event into a set of activities, but focusing first on those activities that are decision-rich. The second

difference is that you make sure the decisions occur in concert with business context, such as organizational policies, strategies, and objectives. Because you perceive a process primarily as a series of decisions, you focus early on policies (if any and if known) for each decision. After all, if there is a decision to be made about something, there probably is a policy to guide the decision (somewhere). Otherwise, why would the organization bother making the decision? Further, if you have done your scoping and discovery activities with diligence, the policy should provide a lot of useful information such as: What does it intend for the result to be? Is it a mandatory policy or a guideline only? Who is most responsible for the policy? Over what jurisdiction does it apply? And, most important of all, what is its rationale? In other words, you are rounding out the motivation and intelligence behind the decision to make sure it has maximum business value.

The third difference is that you will associate rules or decisions to those use cases that rely on them. You will learn that decisions and rules may be shared across use cases, much like information may be shared.

Next is the fourth difference which is the identification of a more complete set of concrete use cases to be sure that every rule can be tested.

This brings you to the fifth difference, which is avoidance of premature commitment to execution sequence. That is, you do not obsess over sequence of activities or tasks at this time. You want to understand the decisions (intelligence) behind the event first, and only later on work out how communications happen among objects to carry out that intelligence.

The sixth difference is a focus on the information referenced and knowledge created by the business event, or actually on the decisions employed by a business event.

Later, during the analysis phase, you will determine which of those decisions (and underlying rules) are to be shared across organizational and application boundaries. That's because the rules are the thinking behind the business event. When a rule executes, it references pieces of information. It may create a new piece of information called knowledge. Decisions, rules, information, and rule-created knowledge are intellectual assets. All can be shared across organizational boundaries, with proper analysis, when appropriate for the business. All of these intellectual assets are important focal points in a business rules approach.

What Is the Purpose of Discovering Initial Requirements?

The purpose of discovering initial requirements is twofold. The first point is to gain an initial understanding for how the business community would like the system's process to flow. The second is to get started early in understanding the decisions, policies, and rules behind the process. The goal is to do so without considering target technology, although you will soon see that commercial rules technology can shorten this discovery time. The second goal is to develop as few deliverables as possible without sacrificing your understanding of the target system.

It is usually natural to begin by understanding in more detail how the business community perceives the system's process flow. An investigation into a preliminary system

process flow leads you to the rules. The question now is, How much investigation and discovery about the process flow is needed in a business rules approach? The answer is probably less than is needed for traditional systems development, depending on target technology.

This chapter proposes that you uncover three essential aspects related to the process behind the business event: the tasks or activities behind each business event if doing so is helpful; a sample of the real or imaginary event scenarios for unraveling and testing completeness of the process; and the decisions made on behalf of those tasks or activities to lead you to rules. The premise is that you need to understand the decisions and rules before you determine whether they will execute through object-oriented code or through other rule-oriented approaches.

This chapter does, however, encourage you to begin or refine a term-fact model in the discovery process so that you have a semantic foundation for the decisions and rules. This can be a business object model or a conceptual data model. *Business objects*, according to Paul Harmon and Mark Watson (1997), are "the kinds of things that end users talk about. . . . Business objects include things like employees, sales orders, accounts, machines, rejection slips, and company sites." They are distinguished from *infrastructure objects*, which they define as "classes that the developer creates to assure that the software works. Therefore, a business object model contains terms and facts familiar to the business person and about which decisions are made and rules are executed. You will develop class models (and other object-oriented deliverables) during process analysis in Chapter 11.

Even if you develop a business object model, this chapter also encourages you to develop a conceptual data model because a stable information architecture is critical to future business rule changes. The earlier you begin understanding it, the better. Figure 6.2

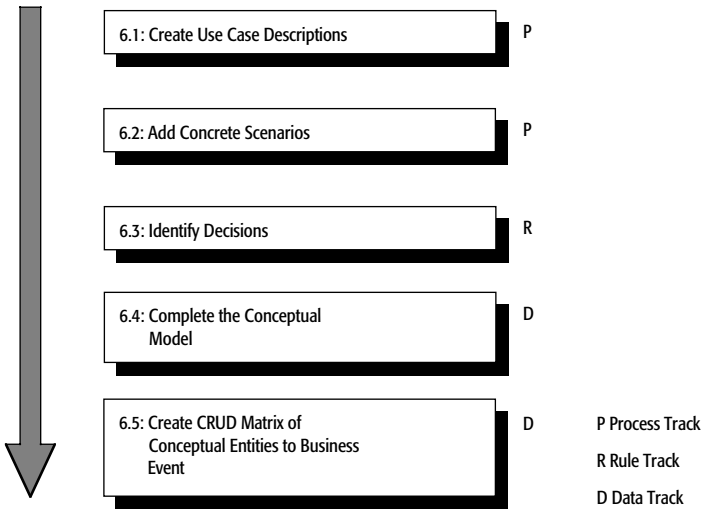

Figure 6.2 The steps of the discovering initial requirements phase.

provides the details behind the discovery of initial requirements. Note that steps 6.1 and 6.2 address deliverables in the process track because these steps deal with event-response process details and corresponding scenarios. Step 6.3, because it involves identifying decisions, takes you into the rule track. Step 6.4, because it starts or evolves a conceptual data model, takes you into the data track. Step 6.5, by creating or evolving a CRUD matrix between conceptual entities and business events, ties together the data track and the process track. Because you will use this CRUD matrix as input to designing stable, shared databases, Figure 6.2 denotes step 6.5 as a deliverable that belongs more in the data track rather than in the process track.

What Are the Deliverables of Discovering Initial Requirements?

In this book, we separate the discovery of process (this chapter) from that of rules and data (Chapter 7). You can, of course, combine the steps in these two chapters into one discovery effort. This chapter divides them here merely for ease of explanation. Should you decide not to follow the steps in this chapter, but to follow your own approach for discovering initial requirements, the separation of rule discovery into its own chapter provides an easy reference by which you can incorporate Chapter 7's concepts into your own requirements-gathering approach.

Below are the possible deliverables for the discovery of process in the process track.

- Use-case descriptions for the human interactions with each business event
- Concrete scenarios for each business event
- Decisions behind each business event
- Terms and facts
  - Business object model
  - Refined conceptual model
- CRUD matrix of conceptual data entities or business objects to business events.

At this time, you may also want to consider that a deliverable from the technology track is a technology architecture vision paper or diagram.

What Are the Steps in Discovering Initial Requirements?

The intent of this chapter is to present one set of steps and deliverables that should suffice in discovering initial requirements so that you can move quickly into rules. This chapter contains requirements techniques that are common and span systems development paradigms, such as use cases and scenarios. You should add steps and deliverables that are familiar to and successful within your organization for gathering requirements.

Step 6.1: Describe the Event Response Process Details by Creating Use-Case Descriptions

As part of the scoping phase, you created an event-response process table. You begin the discovery phase by reviewing this table for each business event within the initial scope so that you can proceed to understand it in more detail.

Specifically, in step 6.1 you aim to understand the details behind the event-response process sufficiently enough to get started uncovering rules. There are many ways of doing this, although the most popular way today is the creation of use-case descriptions. Some people argue that older techniques, such as functional decomposition diagrams, are desirable because they depict the system functionality from a top-down perspective, allowing you to see the whole picture of functionality. Other people argue that such techniques do not lend themselves well to the eventual creation of distributed component-based systems. The truth is that there are many approaches for understanding the event-response process and all have merits. It is not the purpose of this book to contribute to the debates as to whether one is preferable over another or whether, in fact, you should use more than one technique. Instead, this chapter focuses mostly on use-case descriptions because they are common and gaining in popularity. However, the chapter also illustrates possible solutions using other alternatives.

A use-case description represents a typical sequence of interactions that may carry out a business event. According to Paul Harmon and Mike Watson (1997), a *use-case description* provides a generic, step-by-step description of the interaction between an actor and a use case. If you review Figure 4.5, it actually depicts eight use cases, one for each interaction between the VCI system and human or electronic actors.

A use-case description is generic in that it does not name a specific person but an actor or role. A use-case description usually contains a normal sequence of interactions and alternate sequences to handle error conditions, for example.

Table 6.1 illustrates a sample use-case template (Phillips 2000). Note that it contains a place for adding business rules to the use case. Because a rule may be relevant to more than one use case, it would be ideal if you can connect a use-case description to each relevant rule and have the rule repository print out the rules onto the use-case description.

Table 6.1 Sample Use-Case Template from Versata

USE CASE NAME:
Version: 0.1
Status: Draft
Project/Problem Domain: Agency Administration System
Author: Jacobson
Owner: Ms. Noma

Table 6.1 (*Continued*)

USE CASE DETAILS:	
Purpose To create a new Agency and associate an Administrator	
Actors	
Agency Administrator	
Agency Administration Supervisor	
Credit Check Bureau	
Trigger Events & Message Contents	
A "New Agency" request form is received by the Agency Administrator (via fax or email) from one of the Branch Offices	
(average 5 per week, 1 hour per request)	
Use Case Description	
Preconditions:	
Agency must not already exist	
Administrator must be Active	
Post conditions:	
Agency Active	
Intermediate States:	
Awaiting credit check	
Process Steps and Flow (Scenarios):	
Actor Actions (external events)	System Response
Select "Agency Request" input	Display "Agency Request" form
Input Agency Details	Validate Agency Details
	Generate Agency ID
	Request Agency Credit Check from credit bureau
	Set Status to "Pending Credit Check"
	Save details and notify Actor
Receive credit details	Set Agency Status to "Pending Authorization" or "Failed Credit Check"

(*continues*)

Table 6.1 Sample Use-Case Template from Versata (*Continued*)

Input Administrator ID	Validate Administrator exists and is "Active" Associate Administrator with Agency Set Agency Status to "Pending Authorization"
Business Rules (for each process and data object)	
(see data objects for attribute data types)	
If Agency must have a Credit Score over 6.9 to be approved	
If the Agency Rating is Gold, the Administrator must be a Grade 5 or above	
Exceptions or Special Conditions (maybe more business rules)	
If the selected Administrator is not active or qualified, an alternative Administrator is selected by head office.	
Data (data objects, attributes, message flows)	
Agency	
Administrator	
Credit Bureau List	
Associated Reference Materials	
Administrator's Handbook	
Acceptance Criteria for Use Case (for testing)	
Each Agency may have 3 contact addresses but only 1 current billing address	
Part entries of Agency details should be retrievable for later completion	
Example	
Implementation Issues	
Remarks	
Open Items & Risks	

Keep in mind as you write use-case descriptions that you are not overly concerned with absolute sequence at this point. You are simply gaining an understanding of a typical sequence as suggested by members of the business community. You will use your use-case descriptions first as a fast path to discovering the decisions and rules behind the event. When you later analyze rules (Chapter 10), you will methodologically uncover the essential sequence in which the system must uncover related knowledge when servicing an event. The essential sequence becomes the sequence in which decisions and

rules need to execute. You will, then, combine the essential rule sequence with overall core process flow in Chapter 11.

GUIDELINE 6.1.1

Simply use whatever technique assists you in soliciting from a business person a reasonable sequence in which the business event may be serviced.

Although this chapter favors use-case descriptions, use whatever approach works best for your organization. It is not the purpose of this book to critique the various approaches and their merits and shortcomings. Some organizations may prefer process decomposition diagrams. Still others may prefer system response tables. For complex, multiorganization and multisystem processes, a swim lane diagram may be helpful. However, use-case descriptions and analysis are the most popular and usually the most useful for today's component-based, distributed computing environments.

CASE STUDY: STEP 6.1—DESCRIBE THE EVENT-RESPONSE PROCESS DETAILS BY CREATING USE-CASE DESCRIPTIONS

Case Study Instructions:

- Write a use-case description for the interaction between guardian and the system for the business event Guardian Enrolls Member.

Case Study Solution:
If you refer back to the case study description in Step 4.1, most of the documentation describes concepts behind the business event Member Requests Entrance to the Park. So, you need to find additional documentation and talk to business experts about how a guardian enrolls a member. This can be done in a facilitated session or through one-on-one interviews.

Based on what you learn, you create a preliminary use-case description shown in Figure 6.3.

STEP 6.2: ADD CONCRETE SCENARIOS

The term *concrete scenario* means an imaginary or actual instance of a business event or use case. Collecting scenarios is an effective and fun way of starting or solidifying a conversation about the business event. Later, concrete scenarios prove useful for validating that you have described all of the required processing of the system and eventually all of its rules.

Solicit scenarios from business people. Business people can have fun coming up with creative scenarios. These can be used later as test cases for a system prototype. For example, suggest that the business person bring the last 50 transactions that would have been processed by the system. Be sure to include successful and unsuccessful transactions.

Suppose your business audience for the Internet Park may propose the following scenarios for the business event Guardian Enrolls Member:

Scenario 1: Mary is a single parent with a 14-year-old daughter. She wants to enroll her daughter in all theme park functions. Her daughter always does her homework and

Normal Sequence:

1. Guardian accesses the VCI Web page.
2. Guardian accesses the enrollment Web page.
3. Guardian enters guardian information.
4. System qualifies guardian (decisions and rules!).
5. Guardian enters member information.
6. System qualifies member for enrollment in the park (decision and rules!).
7. System displays the complete enrollment screen.
8. Guardian approves the enrollment.

Alternate Sequences:

(a) Guardian is not known. The system initiates a use case for "Enter new guardian."

(b) Guardian has bad credit. The system sets the guardian's payment method to prepay status and sends a message to the guardian indicating that a payment for a prespecified amount is needed before the member will be able to enter VCI Park.

(c) Member is too young. The system sends a message that the member is not age-appropriate for this park.

(d) Member is too old. The system initiates a use case for "Recommend other parks."

(e) Guardian does not have access to the Internet.

Figure 6.3 Preliminary guardian enrolls member use-case description.

her chores, so Mary merely wants to ask the daughter, prior to park admission, if her daughter received any good grades that day. Her daughter can have 1 hour in the game park every day but gets an extra 30 minutes for every good grade. Mary is a new customer. We don't know if she is a good credit risk or not.

Scenario 2: John and Barb want to enroll a 10-year-old boy. He is to be tutored in reading for 30 minutes a day using the tutoring function of the park before he can enter the game park. His reading skill level is age 8. He can have 1 hour of access to the game park per day. John and Barb are existing customers with another child enrolled. Past history indicates that they do not have good credit.

Case Study: Step 6.2—Add Concrete Scenarios

Case Study Instructions:

- Create simple concrete scenarios for the event Member Requests Entrance into the Park. Be creative so as to include all different possibilities.

Case Study Solution:
Below are very simple concrete scenarios for the business event, Member Requests Entrance into the Park. Later, in the next chapter, when you uncover rules, you will proceed with more complicated concrete scenarios.

1. Bob G. has enrolled his daughter Kylie, who is 10 years old and is a VCI member. Bob has signed up his daughter for the theme park and the Spanish tutorial. Bob has entered the following questions to be answered by Kylie when she logs in to VCI Park:
   - Have you completed your homework?
   - Have you helped your mother with your chores?
2. Ted H. has signed up his two children as members. It seems that he is quite concerned that his children complete all their homework and all their chores before entering the VCI park system. Some of the questions that he has asked them to answer are:

   Questions to Peter:
   - Have you completed your homework today?
   - Have you cleaned up your room today?
   - Have you done two of your assigned chores (identified on the refrigerator) today?
   - You had a Math Test yesterday. If you received your test results today, what grade did you receive?

   Questions to Tricia:
   - Have you completed your homework today?
   - Have you cleaned up your room today?
   - Have you done two of your assigned chores (identified on the refrigerator) today?
   - You had a spelling test two days ago. If you received your test results today, what grade did you receive?
3. George S. has two nephews, Brian and Al, who he has signed up as members. George has entered questions to ensure that his nephews complete their homework and do well in school. The questions he has entered for his nephews are: (Note: they are the same for both of his members.)
   - Have you completed your homework?
   - What grade did you get on your Geography test today?

4. Janet W is a guardian with VCI. She has enrolled her grandchild, Nancy (age 8) as a member to VCI. Janet has prepaid for 8 hours (480 minutes) for her granddaughter's time in the VCI park system. Janet has asked her granddaughter the following questions:
   - Have you read a book today for 15 minutes?
   - Have you completed your homework for today?

Step 6.3: Identify Decisions

In this book, a *decision* is a judgment to be made. For example, one decision may be determining whether a customer is of preferred status. This decision relates to customer. Think of other decisions, such as, is a product in stock? Can an order be shipped to a desired location within the requested timeframe? These are all decisions that a system (or human) may need to make when servicing a customer request for an order (a business event).

Sometimes a decision is simply the execution of one rule. For example, if the decision is that a product is always in stock simply if there is one or more of the product on the shelf, the decision is made by executing one rule. That rule is: If the quantity on the shelf of a product is greater than 0, then the product is in stock.

Often, however, a decision results from the execution of many rules. As an example, suppose the business adopts a policy that the last 10 products on the shelf are to be sent to preferred customers only. The policy can be enforced by two different rules, each compliant with the policy. One rule states that if you are a preferred customer, the product is in stock if there is one or more on the shelf. The other rule states that if you are not a preferred customer, the product is considered in stock only if there are eleven or more on the shelf, because the business needs to reserve ten for the preferred customers.

Now, you search for evidence of decisions or rules. You can do this in at least two ways: studying policies or studying event details, or both.

Guideline 6.3.1

Review policies from scoping in search of decisions. Remember to review policies related to objectives as well as those meant to mitigate risks.

You can begin by reviewing the policies uncovered as part of the scoping phase. Start with the policies behind the business event that aim for the objectives. Do any of those policies apply to the business event? If so, do those policies imply that a decision needs to be made so as to be compliant with the policy?

As an example, Chapter 4 introduced the following possible policy: Orders received before 4 P.M. must be shipped for next day arrival at the customer's location. Recall that, in this example, related considerations are: What constitutes an order? What does it mean for an order to be received? What does it mean for an order to be received before 4 P.M.? And what does it mean to ship an order for next day arrival at a customer location? Therefore, if you were performing step 6.3, Identify Decisions, for the business event Order Is Received and the event-response process Fulfill Order, you could start by discussing the following considerations that arise from studying policies.

- What constitutes an order?
  - This will lead to the rules verifying that mandatory information for order processing is present.
- What does it mean for an order to be received?
  - This may lead to rules that determine that the source of the order is valid for meeting this next day shipment policy: fax? phone? email? Web?
- What does it mean for an order to be received before 4 P.M.?
  - This will lead to rules that test the order's timestamp.
  - This will lead to rules that test the receive time against the time in a standard time zone.
- What does it mean to ship an order for next day arrival at a customer location?
  - You can also proceed to reviewing the policies whose aim is to mitigate risks. In Chapter 4, you uncovered the following risk: the unintended release of the identity of member children. An associated policy is: The identity of member children must not be released to any external person or organization. Therefore, if you were performing step 6.3, Identify Decisions, for a business event involving the request of the identity a member, you could investigate the following:
- What information constitutes the identity of a member?
  - This will lead to rules that validate the unique identity of a member.
- What is an external person?
  - This will lead to rules that determine whether an actor is an external versus internal person.
- What constitutes an external organization?
  - This will lead to rules that determine whether an actor represents an organization that is external versus internal.

GUIDELINE 6.3.2

Study event details in search of decisions.

The second way to search for decisions is to look more closely at event details, such as response interactions, a use-case description, low-level processes in a process decomposition diagram, or even concrete scenarios in search of processes driven by decision-making activity.

GUIDELINE 6.3.3

Start with discovering decisions rather than proceeding directly to discovering rules under certain circumstances.

If the business event or a use case seems riddled with decision-making activity (that is, decisions are made about many business nouns) or if the business event or use case happens over a long timeframe, you may want to start by first understanding the kinds of decisions made rather than jumping right into the discovery of detailed rules. Also, some-

times the initial business audience is aware of high-level decisions (for example, about customer or customer credit checking), but does not know the specific details as to how such decisions are made. In these cases, it is useful to identify decisions first and rules later.

GUIDELINE 6.3.4

For each step in the use-case description, determine if there is any mental processing or "thinking" involved.

Look for the following thinking-oriented words, such as:

Check
Qualify
Compute
Calculate
Estimate
Evaluate
Determine
Assess
Compare
Verify
Validate
Confirm
Decide
Diagnose
Process.

These words suggest that there are rules or decisions behind them. If, however, the step is one of "provide information" or "carry out action," there may be no rules behind it. Look back to Figure 6.3. Notice that steps 4 and 6 contain the word "qualifies." Therefore, these are marked as representing steps in which the system makes decisions (see the parentheses). There may also be hints at hidden decisions in the alternate sequences.

GUIDELINE 6.3.5

Identify the primary business noun (term) about which each decision is made.

As examples, review the decisions listed below:

- Is *customer* known?
- Does *customer* have a good credit rating?

- Is *product* known?
- Is *product* available for an order?
- Is *product* shippable as requested for an order?

Notice that each decision is about a business noun (such as customer or product). These are shown above in italics. Also note that the decision itself is about a state that the noun may or may not be in (such as *known*, *have good credit rating*). These are shown above with an underscore.

GUIDELINE 6.3.6

Be sure that each primary business noun (term) has a place in the conceptual data model or business object model.

Most of the time, you will want to make sure there is an entity in the conceptual data model or an object in the business object model for each primary business noun. In some cases, the primary business noun may best be a role played by an entity in the conceptual data model or a subclass in a business object model. This might be the case for a primary business noun of Customer, which is represented as a role of a Business Party entity in the conceptual data model.

From a list of decisions, you can proceed to the next chapter where you will uncover the rules behind each decision.

GUIDELINE 6.3.7

Collect the following meta data for each decision or computation: Decision made; information referenced for decision; knowledge created by the decision; action taken, if any.

You will discover later that most decisions (and rules) do not belong only to one business event or activity, but actually execute over and over again throughout many processes, events, and use cases. A value of the business rules approach is to identify and manage this reuse of decisions and rules to minimize coding, aim for consistency, and facilitate business changes in decisions or rules.

Specifically, decisions and rules have an existence independent of business event, use case, process, transaction, and so on, in much the same way that data have an existence that transcends these considerations.

A useful way to record decisions is with a *decision matrix*, which records each decision, information needed, knowledge created, and whether the materialization of that decision is within the target scope of the current release.

GUIDELINE 6.3.8

Document decisions that occur outside the boundaries of the target system.

Many times there will be decisions that are made by other systems or by humans.

GUIDELINE 6.3.9

Start a list of issues surrounding each decision.

Hopefully, you will address these issues when you gather rules behind the decisions, discussed in the next chapter.

CASE STUDY: STEP 6.3—IDENTIFY DECISIONS

Case Study Instructions:

- Identify the decisions made for each step in a use-case description for Member Requests Entrance into the Park.
- Identify primary business nouns.
- Walk through scenarios with a business person to learn more details about those decisions.
- Fill out a decision matrix.

Case Study Solution:
Figure 6.4 shows a simple, preliminary use-case description for Member Requests Entrance to Park.

You identify a preliminary set of decisions for each step of the use case, as shown in Table 6.2.

To identify primary business nouns, in studying these decisions, two primary business nouns emerge: Guardian and Member.

Let's now walk through the scenarios with a business person. What kinds of decisions are made; which decisions admit the member; which deny access to member?

1. Bob—Kylie
   A. Is login accepted (yes or no)—Yes
   B. Is billing status Credit or Prepay—Credit
   C. What is default time allowed—20 minutes in theme park; 20 minutes in tutorial

Normal Sequence:

1. Member accesses the VCI Web page.
2. Member presents entrance pass (identification).
3. System qualifies member.
4. System presents member-specific questions to member.
5. Member answers questions.
6. System qualifies member answers to questions.
7. System qualifies guardian billing.
8. System enables member to enter park.

Figure 6.4 Preliminary member requests entrance to park use-case description.

Table 6.2 Task Decision Table

USE CASE STEP	DECISION	RULE
1. Member accesses the VCI Web page.	None	
2 Member presents entrance pass (identification)	None	
3. System qualifies member	Is member login *accepted*?	
4. System presents member-specific questions to member	None	
5. Member answers questions	None	
6. System qualifies member answers to questions	Is homework *done*? Is chore *done*? Is activity *done*? Is subject grade *acceptable*?	
7. System sends member answers to guardian	None	
8. System qualifies guardian billing	Does guardian have money *sufficient to pay for member entrance*?	
9. System enables entrance to park	None	

D. What are the member responses to questions

Homework—Yes → Add 10 minutes to default time in theme park

Chores—No → Deduct 5 minutes from theme park allowance

2. Ted—Peter
   A. Login accepted—Yes
   B. Credit/Prepay—Credit
   C. Default time allowed—0 minutes
   D. Responses
   - Homework—Yes → 30 minutes in theme park
   - Cleaned Room—Yes → 15 minutes in theme park
   - Chores—Yes → 15 minutes in theme park
   - Math Test—C → 15 minutes in math tutorial before entering theme park
3. Ted—Tricia
   A. Login accepted—Yes
   B. Credit/Prepay—Credit

C. Default time allowed—0 minutes

D. Responses

- Homework—Yes → 30 minutes in theme park
- Cleaned room—Yes → 15 minutes in theme park
- Chores—No → Deduct 15 minutes from theme park allowance
- Spelling Test—B → 5 minutes in theme park

4. George—Brian

A. Login accepted—Yes

B. Credit/Prepay—Credit

C. Default time allowed—30 minutes in theme park; 20 minutes in library park

D. Responses

- Homework—Yes → allow default time in park
- Geography Test—B → allow default time in park

5. George—Al

A. Login accepted—Yes

B. Credit/Prepay—Credit

C. Default time allowed—30 minutes in theme park; 20 minutes in library park

D. Responses

- Homework—No → No time in theme park; allow default time in library park
- Geography Test—C → No theme park allowance all week

6. Janet—Nancy

A. Login accepted—Yes

B. Credit/Prepay—Prepay

C. Default time allowed—20 minutes in theme park, 30 minutes in math tutorial

D. Prepaid time remaining—110 minutes

E. Responses

- Read Book—Yes → allow default time in both parks
- Homework—Yes → allow default time in both parks
- Practiced piano—No → must spend 15 minutes in tutorial before entering theme park

Note: Nancy is well within her prepaid allotment. At most she will use 50 minutes and will have at least 60 minutes remaining in the prepaid account.

STEP 6.4: COMPLETE THE CONCEPTUAL MODEL

In step 6.4, you carry out discovery activities appropriate for the data track by refining the conceptual model you started during the scoping phase or starting one if you did not

already do so. In this book, a *conceptual data model* is a high-level view of information requirements.

The conceptual model described in this book borrows from former ideas of information engineering. Some practitioners may not be in favor of creating a conceptual model because many systems are developed today, following an object-oriented approach that gives low or no priority to information engineering concepts. We include a conceptual model in our deliverables because, without careful attention to the underlying information architecture, you are likely to proliferate poor quality databases. In a world where business rules can change easily, poor quality databases become serious barriers to business growth. To minimize the risk that the database will inhibit rule changes, the conceptual model represents the scope information requirements extended beyond the target information system scope wherever possible. Doing so positions you to deliver databases that can be shared across organizational and application boundaries with minimal disruption to the business. Doing so also positions you to develop data structures that will support a wide variety of new rules. By creating a conceptual data model, you are able to divide the information requirements into segments that can be planned, scoped, analyzed, designed, and implemented in increments over time.

Note that you can also enhance your business object model at this point to extend beyond the scope of your target system. However, your business object model, if you create one, will likely serve as a foundation for adding infrastructure objects, and iteratively refining so that it serves the processing needs of the system well. The processing needs and the data needs are not the same. You will apply different criteria to the analysis and design of objects to leverage them than you will to the analysis and design of databases to leverage them. Object models, in one way or another, aim to serve the dynamic aspect of the system and ought to be developed iteratively with the dynamic nature in mind. Attention is given to object models in Analyzing Process, Chapter 11. Data models, on the other hand, aim to serve the static informational structure of the system and are difficult to develop iteratively, as changes to data structures are expensive and time-consuming.

GUIDELINE 6.4.1

At a minimum, the conceptual model should consist of three deliverables: a conceptual data model, a conceptual process model, and a CRUD matrix showing information (entity), usage (create, update, delete, read, or usage) of the lowest levels in the conceptual process model.

The benefits of creating a conceptual model for a broad scope include:

- Enabling the integration of eventual databases (with few surprises)
- Serving as the focal point for all data requirements for all projects eventually
- Serving as insight into business-knowledgeable sources to be included in data analysis activities (that is, to scope the specific data analysis project)
- Communicating the breadth of business data to business people and to IT professionals
- Solidifying proposed data subject boundaries
- Establishing early stewardship boundaries over information.

GUIDELINE 6.4.2

For a business rules system, the conceptual data model should include major entities that are of interest to the scope under consideration.

The focus in creating a conceptual data model is on common business semantics, that is, data names, data meanings, and data relationships. It can show abstract supertypes (such as business party) with business-specific subtypes. It need include only the most prominent subtypes. Many-to-many relationships need not be resolved in a conceptual data model, unless doing so adds to business understanding. It shows cardinality of relationships but not optionality. Except for abstract supertypes, all entity and attribute names reflect business terms. Where interesting, obvious attributes are shown within entities, if known. Codes or flags are not shown as entities or attributes. The model should be complete enough to influence design but not enough to serve as a final design specification. (For more details on the items mentioned in this paragraph, refer to Chapter 9.)

If you do not create a conceptual data model during the discovery phase, you are likely to spend more time getting started in building your detailed logical data model. Moreover, the absence of a conceptual data model means that the seams among subjects are unclear, thus possibly resulting in duplicate and inconsistent analysis efforts.

GUIDELINE 6.4.3

Typically, a conceptual process model is a functional decomposition diagram with supporting text.

The functional decomposition diagram is usually decomposed down to the level necessary to indicate creation of an entity. Keep in mind that these functions are not system functions, but business functions. The reason you want a conceptual process (business function) model is because you want to understand the possibility of sharing data across business functions. This book does not favor one process or functional modeling approach over another.

GUIDELINE 6.4.4

The CRUD matrix should depict business event to conceptual entities.

A *CRUD matrix of conceptual entities to business events* is a correlation of each business event to the conceptual entities it creates, updates, reads, or deletes. This provides an early idea of the scope of the data foundation that is needed for each incremental delivery. You do not need to deliver the entire data foundation with the first release. However, you need to deliver foundational data structures against which future data pieces can be added with little negative impact on the system.

If you do not create a conceptual process model and the CRUD matrix between processes and entities, your detailed logical data model created during the analysis phase may not reflect a broad enough scope to be stable.

Case Study: Step 6.4—Begin to Create or Refine the Conceptual Model

Case Study Instructions:

- Update the conceptual data model with new information gained.

Case Study Solution:
Figure 6.5 depicts a possible preliminary conceptual data model based on what you have uncovered so far. Table 6.3 proposes a partial CRUD, showing the business event, Member Requests Entrance to Park, and its usage of conceptual data entities.

Considerations for Iterative and Parallel Systems Development

Some of the concepts addressed in this chapter (use cases, concrete scenarios, decisions, and the terms and facts referenced in decisions) will evolve throughout all phases of development. Specifically, you will correct and add details to use cases. You may add

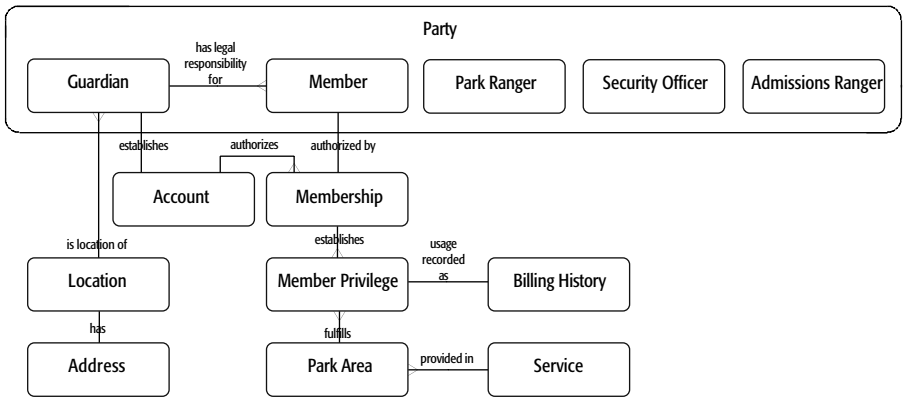

Figure 6.5 Preliminary conceptual data model.

Table 6.3 CRUD Matrix, Business Event to Business Entity

BUSINESS EVENT	BUSINESS PARTY ENTITY	GUARDIAN ENTITY	MEMBER ENTITY	MEMBERSHIP ENTITY	MEMBER PRIVILEGE ENTITY
Guardian Enrolls Member	C, R	C, R, U	C	C	C

more use cases. You will revisit concrete scenarios, adding more to ensure completeness. Decisions (and rules) will be added and changed. Especially if using commercial rules products (or building your own rules component), you will be able to change and add rules quickly for prototyping. Data can be added or changed in each phase, although not all data changes are easy to accommodate. This is why a solid information architecture assists in enabling a smooth iterative development experience, because it minimizes the costly and negative impact of data changes.

Within this part of discovery, you can carry out the steps in this chapter in parallel, if useful. For example, if you will be developing all of your own object-oriented code, you may want to develop use cases and concrete scenarios in parallel. Subsequently, you may want to work on decisions and the terms and facts within them in parallel. You might even focus on all four items in parallel:

- Use-case descriptions to use as a basis in analysis for sequence diagrams and class diagrams.
- Concrete scenarios to check for completeness.
- Decisions employed by the use case to create a starting point for rule discovery.
- Terms and facts referenced in decisions to use as a basis in analysis to refine the class and sequence diagrams and begin a logical data model.

If, instead, you will be deploying your system using certain commercial rules products, you may address these four items sequentially, develop the first two together and the last two together, or develop all four at once. You will discover that you may need later only to create class models and sequence diagrams for the core process flow that is not responsible for executing shared rules. That's because shared rule execution is not handled within your system's internal logic.

The point is that object-oriented techniques are extremely useful, but we don't need to apply them necessarily to the rules portion of the system. Not yet, anyway.

Summary

The discovery phase begins when there is a defined project scope to develop a target system, business commitment to build the system, a detailed plan for how to proceed, and a desire to develop the system with a business rules approach. The purpose of discovering initial requirements is to document aspects of the system's process that includes five items: typical details behind each business event; decisions made behind those interactions; information referenced in making those decisions; the knowledge created by those decisions; and sample real or imaginary event scenarios for testing completeness of the process.

The significant difference in a business rules approach lies in the unearthing of decisions behind business events. That is, you begin, even during the discovery phase, to focus on the decision-making (intellectual processing) behind a business event as the starting point for uncovering detailed rules (and thought processes) behind each decision. What this means is that a business rules approach places emphasis on the intelligence (computations, constraints, inferences) required to handle a business event. Later,

you will want to determine the process by which those decisions and rules are executed and which ones are to be shared across organizational and application boundaries.

While you begin by studying the sequence of interactions, you are truly not concerned with actual sequence at this point in time. You are only looking for a way that will lead you to the rules behind the event. When you later analyze the rules (Chapter 10), you will methodically uncover the essential sequence of processing behind the event. The *essential processing sequence* represents the sequence in which the system must uncover related knowledge (or decisions) when servicing an event.

Traditional systems development methodologies usually do not include a formal capturing or analysis of decisions and relating them to business context, such as goals, objectives, strategies, and policies.

During the discovery phase, you should create a conceptual model. This provides an early idea of the scope of the data foundation that is needed for each incremental delivery. You do not need to deliver the entire data foundation with the first release. However, you need to deliver foundational data structures against which future data pieces can be added with little negative impact on the system.

At this point, you have an understanding of your use cases, the decisions behind them, sample concrete scenarios, a semblance of semantics through the identification of terms. You are now ready for the next chapter. In Chapter 7, you seek the detailed rules and information pieces behind each use-case and possibly shared across them.