Chapter 3 – Identifying User Stories and Use Cases

Table of Contents

Chapter Overview

This chapter extends the concepts learned in Chapter 2 about building analysis models that capture the processing requirements of the new system. For any system there are two types of requirements that must be defined and modeled: Processing requirements and data requirements. This chapter addresses the processing requirements by teaching the students how to identify and document use cases. Use case modeling is a powerful technique to assist system analysts to identify, understand, communicate, and document the processing requirements of the new system.

The first section of the chapter teaches students how to find and identify use cases using the user goal technique. In this technique a systems analyst identifies the users of the system, as a role or type of user, and then identifies each goal or "actions to perform." These goals then are used to define use cases.

The second method of identifying use cases is the "event-decomposition" technique. This technique first identifies the business events that occur. By understanding the business events, the actions leading up to the event, and the resulting processing required to support each event, a list of use cases can be developed. This is a powerful technique that takes a broader business point of view and can be used to identify many different types of events, which then produce a comprehensive list of use cases.

The chapter concludes with instructions on how to build use case diagrams. Use case diagrams are a straightforward method to document and visualize the use cases that have been identified. Use case diagrams are effective in showing the actors, use cases, and relationships between use cases.

Note that this chapter does not get into the detailed descriptions or documentation of internal steps within a use case. That material is covered in Chapter 5.

Learning Objectives

After reading this chapter, the student should be able to:

- Explain why identifying user storied and use cases is the key to defining functional requirements
- Write user stories with acceptance criteria
- Describe the two techniques for identifying use cases
- Apply the user goal technique to identify use cases
- Apply the event decomposition technique to identify use cases
- Describe the notation and purpose for the use case diagram
- Draw use case diagrams by actor and by subsystem

Notes on Opening Case and EOC Cases

Opening Case

Waiters on Call Meal-Delivery System: This case shows how the analyst uses events—and the use cases they trigger—to summarize what the users require in their order and delivery system. As the users discuss their business, the analyst notes what events occur that require the system to respond in some way. A suggested in-class exercise is to complete a list of use cases.

Key points to note include:

- Events (and use cases) can be discovered by talking with the users.
- Talking about events is natural for business-oriented users.
- The analyst can focus on requirements independent of current or proposed technology (a logical model), which is something that students often find difficult. Even if the users want to talk about technology, it is important to define the requirements in this way.

EOC Cases

The running cases go throughout the rest of the book, e.g. for each chapter.

The State Patrol Ticket-Processing System: This case describes a traffic ticket system that must interface with other systems. One benefit of this case is that the students must identify use cases that belong only to the ticket-processing system. In other words, they should not include use cases for the other systems that receive information from, or sent information to the ticket-processing system. This is a good exercise to teach students to identify only appropriate use cases.

Community Board of Realtors (running case)**:** Community Board of Realtors is a professional

organization that supports real estate offices and agents. This chapter describes many of the processing and informational requirements of real estate agents. Students are asked to identify use cases and build a use case diagram based needs of real estate agents. Students are also asked to act as a real estate buyer and identify any other use cases, e.g. information and processes for customers that they desire.

Spring Breaks 'R' Us Travel Services (SBRU) (running case)**:** SBRU is an online travel services that books spring break trips to resorts for college students. This chapter describes student activities to find and book reservations for spring break vacation trips. Students are asked to identify appropriate use cases and draw a use case diagram. The second problem in this case asks students to consider their own desires, as student customers, on what social networking functions they might also want to include in this system. This should provide some interesting answers.

On the Spot Courier Services (running case)**:** On the Spot is a small, but growing, courier service that needs to track customers, package pickups, package deliveries, and delivery routes. This chapter describes many of the functions that the new system must provide. Since there are multiple actors that will use the system, the students are asked to first identify the actors and then the associated use cases. Finally the students are asked to find use cases specifically using the event decomposition technique.

Sandia Medical Devices (running case)**:** Sandia Medical Devices is a company that specializes in medical monitoring through remote, mobile telecommunication devices. This chapter provides a rich description of processing and requirements of both patients and medical professionals. The students are asked to identify the actors and the associated use cases. They are asked to use the event decomposition technique to find use cases.

Instructor's Notes

User Stories and Use Cases

Key Terms

- **user story** – one short sentence in everyday language of the end user that states what a user does as part of his or her work
- **acceptance criteria** – features that must be present in the final system for the user to be satisfied
- **use case** – an activity that the system performs, usually in response to a request by a user

Lecture Notes

User stories and use cases are very closely related concepts. User stories are simple, one sentence descriptions of a user task that utilizes the system. User stories are normally created by the users themselves. User stories are used in highly Agile development projects where there is ongoing user involvement. The template for a user story is in this form:

As a <*role played*>, I want to <*goal or desire*> so that <*reason or benefit*>

Use cases also define the tasks that a user does that utilizes the system. However, use cases are usually

documented more thoroughly and with more detail. Often a use case will also be modeled using a workflow diagram or detailed description that identifies the various steps included in the use case. The detail definition of a use case normally requires the expertise of a systems analyst.

Inasmuch as the description of a user story is rather brief, an acceptance criteria is also described with the user story. The acceptance criteria identifies the conditions that must be present in the final outcome in order for the user to be satisfied.

Quick Quiz

Q: Who takes responsibility to document a user story?

A: The users themselves

Q: What are the differences and similarities between a user story and a use case?

A: A user story is a brief one sentence statement without detailed documentation, but with acceptance criteria defined. A use case is usually more detailed and can be documented with detailed steps. Both describe the user's tasks and processes.

Use Cases and the User Goal Technique

Key Terms

- **user goal technique** – a technique to identify use cases by determining what specific goals or objectives must be completed by a user

Lecture Notes

One approach to identifying **use case**s, called the **user goal technique**, is to ask users to describe their goals for using the new or updated system. The analyst first identifies all the users and then conducts a structured interview with each user. A user goal can be thought of as a piece of work, or a task, that the user must complete. For example, for an RMO a shipping clerk might have a goal such as *ship items*, or *track shipments*.

The user goal technique for identifying use cases includes these steps:

1. Identify all the potential users for the new system.
2. Classify the potential users in terms of their functional role (e.g., shipping, marketing, sales).
3. Further classify potential users by organizational level (e.g., operational, management, executive).
4. For each type of user, interview them to find a list of specific goals they will have when using the new system. Start with goals they currently have and then get them to imagine innovative functions they think would add value. Encourage them to state each goal in the imperative verb-noun form, such as Add customer, Update order, and Produce month end report.. Create a list of preliminary use cases organized by type of user.

5. Create a list of preliminary use cases and organize it by type of user.
6. Look for duplicates with similar use case names and resolve inconsistencies.
7. Identify where different types of users need the same use cases.
8. Review the completed list with each type of user and then with interested stakeholders.

Quick Quiz

Q: What is the primary source of use cases in the user goal technique?

A: An interview with the user.

Use Cases and Event Decomposition

Key Terms

- **event decomposition technique –** a technique to identify use cases by determining the external business events to which the system must respond
- **elementary business processes (EBPs) –** the most fundamental tasks in a business process, which leaves the system and data in a quiescent state; usually performed by one person in response to a business event
- **event –** something that occurs at a specific time and place, can be precisely identified, and must be remembered by the system
- **external event –** an event that occurs outside the system, usually initiated by an external agent
- **actor –** an external agent; a person or group that interacts with the system by supplying or receiving data
- **temporal event –** an event that occurs as a result of reaching a point in time
- **state event –** an event that occurs when something happens inside the system that triggers some process
- **system controls –** checks or safety procedures to protect the integrity of the system and the data
- **perfect technology assumption –** the assumption that a system runs under perfect operating and technological conditions

Lecture Notes

The most comprehensive technique for identifying use cases is the **event decomposition technique**. The event decomposition technique begins by identifying all the business events that will cause the information system to respond, and each event leads to a use case. Starting with business events helps the analyst define each use case at the right level of detail. The appropriate level of detail for identifying use cases is one that focuses on **elementary business processes (EBP**s). An EBP is a task that is performed by one person in one place in response to a business **event**, adds measurable business

value, and leaves the system and its data in a stable and consistent state.

Event Decomposition Technique

The **event decomposition technique** focuses on identifying the events to which a system must respond and then determining how a system must respond (i.e., the system's use cases). When defining the requirements for a system, it is useful to begin by asking, "What business events occur that will require the system to respond?"

Types of Events

External Events: An **external event** is an event that occurs outside the system—usually initiated by an external agent or actor. An external agent (or actor) is a person or organizational unit that supplies or receives data from the system. To identify the key external events, the analyst first tries to identify all the external agents that might want something from the system. A classic example of an external agent is a customer. When describing external events, it is important to name the event so the external agent is clearly defined. The description should also include the action that the external agent wants to pursue.

Temporal Events: A **temporal event** is an event that occurs as a result of reaching a point in time. For example, on month-end certain automatic processing may need to be initiated. The analyst begins identifying temporal events by asking about the specific deadlines that the system must accommodate. What outputs are produced at that deadline? What other processing might be required at that deadline?

State Events: A **state event** is an event that occurs when something happens inside the system that triggers the need for processing. For example a state event can occur when an inventory level reaches a pre-order point.

Identifying Events

Events versus Prior Conditions and Responses: It is sometimes difficult to distinguish between an event and part of a sequence of prior conditions that leads up to the event. The way to determine whether an occurrence is an event or part of the interaction following the event is by asking whether any long pauses or intervals occur (i.e., can the system transaction be completed without interruption?). Or is the system at rest again, waiting for the next transaction? After the customer wants to buy the shirt, the process continues until the transaction is complete. There are no significant stops after the transaction begins. After the transaction is complete, the system is at rest, waiting for the next transaction to begin. The EBP concept defined earlier describes this as leaving the system and its data in a consistent state.

The Sequence of Events: Tracing a Transaction's Life Cycle: It is often useful in identifying events to trace the sequence of events that might occur for a specific external agent or actor. This is similar to the user goal technique, in that it focuses on a single actor. But it focuses on a narrower set of tasks. For example, what are all the things that a customer might want to do when he or she buys a shirt? Such things as buy the shirt, return the shirt, exchange the shirt, and so forth.

Technology-Dependent Events and System Controls: Sometimes, the analyst is concerned about events that are important to the system but do not directly concern users or transactions. Such events

typically involve design choices or system controls. These types of events are not part of the problem domain, e.g. not part of the users requirements. Typically those events are not addressed during analysis but are addressed during the design activities. One technique used to help decide which events apply to controls is to assume that technology is perfect. The perfect technology assumption states that events should be included during analysis only if the system would be required to respond under perfect conditions.

Using the Event Decomposition Technique

To summarize, the event decomposition technique for identifying use cases includes these steps:

1. Consider the external events in the system environment that require a response from the system by using the checklist shown in Figure 3-3.
2. For each external event, identify and name the use case that the system requires.
3. Consider the temporal events that require a response from the system by using the checklist shown in Figure 3-4.
4. For each temporal event, identify and name the use case that the system requires and then establish the point of time that will trigger the use case.
5. Consider the state events that the system might respond to, particularly if it is a real-time system in which devices or internal state changes trigger use cases.
6. For each state event, identify and name the use case that the system requires and then define the state change.
7. When events and use cases are defined, check to see if they are required by using the perfect technology assumption. Do not include events that involve such system controls as login, logout, change password, and backup or restore the database, as these are put in as system controls.

Quick Quiz

Q: What is the difference between a state event and a temporal event?

> A: Both are internal events, but a state event is triggered by a change in the “state” of the system (or data in the system, and a temporal event is trigger purely by the passage of time.

Q: How do you identify the scope of an event, i. e. what is part of the event and what is not?

> A: The primary identifier is if the system can go into a quiescent state after the end of the event. When a transaction (an event) is completed, all the data has been updated and the system can wait for another transaction to occur.

Q: Why don't we include technology dependent events such as logging onto a system? What is the assumption that we make?

> A: The assumption is the perfect technology assumption. We don't include technology dependent events during analysis because they tend to confuse user requirements, e.g. problem domain issues, with technology issues, which are system design issues.

Use Cases in the Ridgeline Mountain Outfitters Case

Key Terms

- **brief use case description** – an often one-sentence description that provides a quick overview of a use case
- **use case diagram –** the UML model used to illustrate use cases and their relationships to actors
- **automation boundary –** the boundary between the computerized portion of the application and the users who operate the application but are part of the total system
- **«includes» relationship** – a relationship between use cases in which one use case is stereotypically included within the other use case

Lecture Notes

The initial system vision (discussed in Chapter 2) identified four subsystems: the Sales subsystem, the Order Fulfillment subsystem, the Customer Account subsystem, and the Marketing subsystem. As work progressed, the analysts combined reports required by each subsystem into a fifth subsystem called the Reporting subsystem. In a system this size, the analyst should organize the use cases by subsystem to help track which subsystem is responsible for each use case.

These use cases are shown in Figures 3-9 and 3-10. It is important to recognize that the list of use cases will continue to evolve as the project progresses. This **brief use case description** is usually expanded to record more of the details when the developers are designing and implementing the use case.

The use case diagram is the UML model used to graphically show the use cases and their relationship to users. In UML, a person that uses the systems is called an actor. An actor is always outside the automation boundary of the system but may be part of the manual portion of the system. Sometimes, the actor for a use case is not a person; instead, it can be another system or device that receives services from the system.

A simple stick figure is used to represent an actor. The stick figure is given a name that characterizes the role the actor is playing. The use case itself is represented by an oval with the name of the use case inside. The connecting line between the actor and the use case indicates that the actor is involved with that use case. Finally, the automation boundary, which defines the border between the computerized portion of the application and the people operating the application, is shown as a rectangle containing the use case. Figures 3-12 through 3-16 give several examples of use case diagrams.

There are many ways to organize use case diagrams for communicating with users, stakeholders, and project team members. One way is to show all use cases that are invoked by a particular actor or functional area such as a department.

Frequently during the development of a use case diagram, it becomes apparent that one use case might use the services of another use case. Therefore, one use case uses, or "includes," another use case. Sometimes, this relationship is referred to as the «includes» relationship or the « uses» relationship. Note that the word "includes" is enclosed within guillemets in the diagram

Developing a Use Case Diagram

The steps to develop use case diagrams are:

1. Identify all the stakeholders and users who would benefit by having a use case diagram.
2. Determine what each stakeholder or user needs to review in a use case diagram. Typically, a use case diagram might be produced for each subsystem, for each type of user, for use cases with the includes relationship, and for use cases that are of interest to specific stakeholders.
3. For each potential communication need, select the use cases and actors to show and draw the use case diagram. There are many software packages that can be used to draw use case diagrams.
4. Carefully name each use case diagram and then note how and when the diagram should be used to review use cases with stakeholders and users.

Quick Quiz

Q: What is the most common way to organize use cases?

A: By user or if a larger grouping is desire, by department.

Q: What are the three component part of a use case diagram and what do they represent?

A: An oval represents the use case. A stick figure represents an actor. A straight line represents a relationship between an actor and an use case. A rectangle represents the automation boundary.

Q: What does an "includes" relationship mean? And how is it represented?

A: The "includes" relationship means that one use case will use the services of another use case, hence the second use case is "included within" the first use case. It is represented by a dashed lined with an arrow. The arrow points to the "included" use case, i. e. "use case A includes use case B."

Classroom Activities

The two primary skills that students should get from this chapter are to

1. Identify use cases from user interviews (or in the case of canned case from a narrative),
2. Be able to create a use case diagram, and

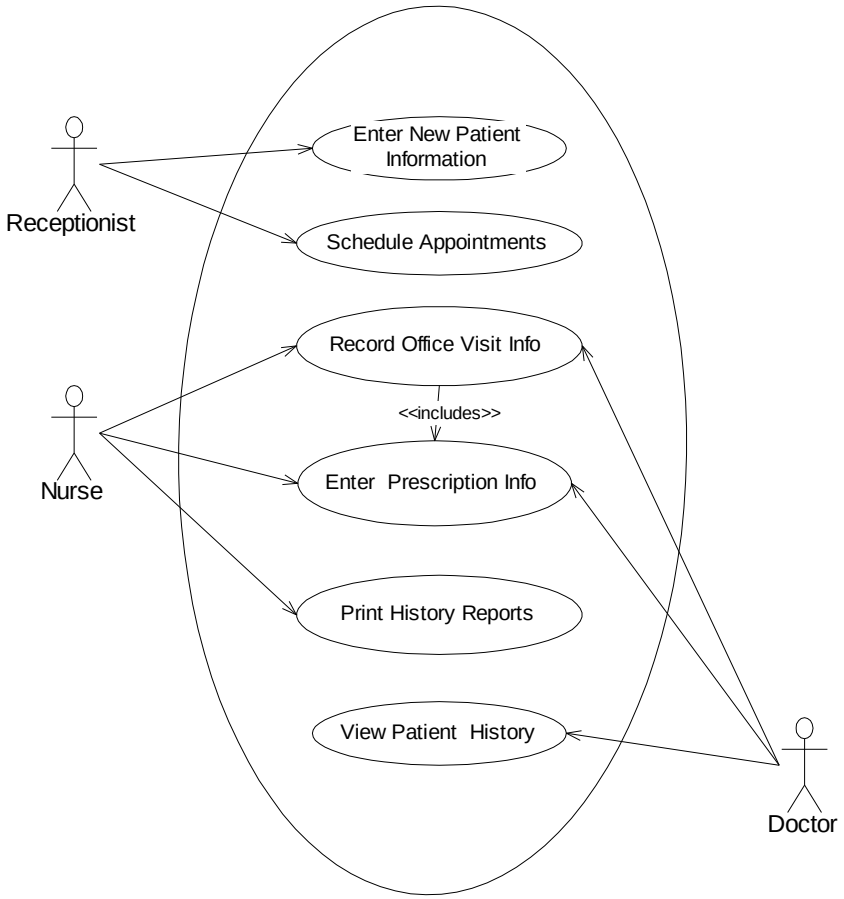
The concepts in this chapter are not complex and should not take the entire class period. The best way to teach about identifying use cases is to first do an example in class using a short case, and then have

the students do an example in class using another short case. Alternatively if there is only time to do a single use case, then have the students first do their best to identify the use cases, and then work through the solution together.

The following short case makes a good classroom example without having to use the chapter cases.

Patient Record and Scheduling System

A patient record and scheduling system in a doctor's office is used by the receptionists, nurses, and doctors. The receptionists use the system to enter new patient information when first-time patients visit the doctor. They also schedule all appointments. The nurses use the system to keep track of the results of each visit including diagnosis and medications. For each visit, free form text fields are used captures information on diagnosis and treatment. Multiple medications may be prescribed during each visit. The nurses can also access the information to print out a history of patient visits. The doctors primarily use the system to view patient history. The doctors may enter some patient treatment information and prescriptions occasionally, but most frequently they let the nurses enter this information. -- Each patient is assigned to a family. The head of family is responsible for the person with the primary medical coverage. Information about doctors is maintained since a family has a primary care physician, but different doctors may be the ones seeing the patient during the visit.



Troubleshooting Tips

Probably the biggest problem that students have with the concepts in this chapter is finding and describing use cases. A big help is for students to say “the actor uses the system to .... [use case description].” For example, “the customer uses the system to 'make a purchase'” or 'fill a shopping cart.' Using this technique most frequently describes use cases at the right level of detail.

Also remember that a use case should stand alone. In other words, the system should be in a quiescent state when the use case starts and it should end in a quiescent state. For example, one type of action could be an independent use case, or that same action might exist only as one step in a use case. Take, for our example, “make a payment.” If all purchases must be paid when the purchase is made, then making a payment is always part of the “purchase an item” use case. However, if the organization allows customer lines of credit, then making a payment may also be an independent use case, e. g. the system is quiescent before and after making a payment. In this case, “purchase an item” might «include» “make a payment” use case.

The other problem that students have is use CRUD to define all of their use cases. Using CRUD as the sole source of use cases results in a lot of low level, non-business descriptions for use cases. CRUD is best used as a validating technique and not a use case definition technique.

Discussion Questions

1. Business Events

Students sometimes have a hard time distinguishing between a business event and preliminary or subsequent activity. The textbook has a good discussion about this subject. A few leading questions will help students to grasp this concept.

When there are many steps in a business activity, how can you tell which step actually identifies a use case? What happens to the other steps in relationship to the system? Does the system have to know about them? What kind of clues are available to help identify the step that is actually the use case?

You might elaborate on the above Patient Record and Scheduling system and describe a whole scenario when a patient makes an appointment, or visits the doctor. And ask the students to distill out the specific use case.

2. Perfect Technology Assumption

The perfect technology assumption concept helps the analyst identify if an event is required in the analysis phase. If the system must respond to the event even if the system is implemented with perfect technology, then the event should be included. With perfect technology, users are perfect, and the technology is always reliable. Using the perfect technology assumption, describe the perfect automobile. Which businesses and services would no longer be needed if the perfect technology assumption were a reality for the automobile? How does the perfect technology assumption help with analyzing a system?