10.6 Task Description

Descriptions of business tasks have been used within software development for many years. During the 1970s and 1980s, business scenarios were commonly used as the basis for acceptance testing, i.e. the last testing stage before the customer paid the final fee installment and accepted the system. In more recent years, due to the emphasis on involving users earlier in the development lifecycle and the large number of new interactive products now being developed, task descriptions are used throughout development, from early requirements activities through prototyping, evaluation, and testing. Consequently, more time and effort has been put into understanding how best to structure and use them.

As shown by Alexander and Maiden's (2004) collection of scenarios, stories, and use cases, there are many different flavors of task description, and they can be used for different purposes, emphasizing different elements of the product being developed. For example, Alexander and Maiden use a structuring framework that distinguishes task descriptions according to four views which are made up of nine facets including method of description (e.g. text, graphics, image or prototype, and formal, informal, or semi-formal notation), context (e.g. organizational environment and system interaction), and role (descriptive, exploratory, or explanatory).

We shall introduce three of the more common description types here: scenarios, use cases, and essential use cases (sometimes referred to as task cases). Each of these may be used to describe either existing tasks or envisioned tasks with a new product. They are not mutually exclusive and are often used in combination to capture different perspectives or to document different stages during the development lifecycle.

In this section and the next, we use two main examples to illustrate the application of techniques. These are a movie rental club and a shared travel organizer. The movie rental club allows members to rent movies of their choice; the shared travel organizer supports a group of people who are exploring vacation possibilities.

10.6.1 Scenarios

A *scenario* is an 'informal narrative description' (Carroll, 2000). It describes human activities or tasks in a story that allows exploration and discussion of contexts, needs, and requirements. It does not explicitly describe the use of software or other technological support to achieve a task. Using the vocabulary and phrasing of users means that the scenarios can be understood by the stakeholders, and they are able to participate fully in the development process. In fact, the construction of scenarios by stakeholders is often the first step in establishing requirements.

Imagine that you have just been invited along to talk to a group of users who perform data entry for a university admissions office. You walk in, and are greeted by Sandy, the supervisor, who starts by saying something like:

> ***Well, this is where the admissions forms arrive. We receive about 50 a day during the peak application period. Brian here opens the forms and checks that they are complete, that is, that all the documentation has been included. You see, we require copies of relevant school exam results or evidence of work experience before we can process the application. Depending on the result of this initial inspection, the forms get passed to …***

Telling stories is a natural way for people to explain what they are doing or how to achieve something. It is therefore something that stakeholders can easily relate to. The focus of such stories is also naturally likely to be about what the users are trying to achieve, i.e. their goals. Understanding why people do things as they do and what they are trying to achieve in the process allows us to concentrate on the human activity rather than interaction with technology.

This is not to say that the human activity should be preserved and reflected in any new product we are trying to develop, but understanding what people do now is a good starting point for exploring the constraints, contexts, irritations, facilitators, and so on under which the humans operate. It also allows us to identify the stakeholders and the products involved in the activity. Repeated reference to a particular form, book, behavior, or location indicates that this is somehow central to the activity being performed and that we should take care to understand what it is and the role it plays.

A scenario that might be generated by potential users of a movie rental club is given below:

> ***Say I want to find a movie directed by Martin Scorsese. I don't remember the title but I know it came out in the cinemas around 2006 or 2007. I go to the club website and choose the director option. A huge list of directors is displayed – I had no idea there were so many directors with surnames beginning with S! After scrolling through the list I find Martin Scorsese and choose to see further details about him. Another long list of movies eventually leads me to the movie I was looking for – The Departed. As an existing club member, I need to enter my username and password to be able to rent the movie. Once my password has been confirmed, I am given a choice of rental period and payment method. I have my preferences already registered in the system, so I just choose the defaults and download my movie.***

In this limited scenario of existing system use, there are some things of note: the long lists of names and movies that the user has to scroll through, the lack of detailed search possibilities, the importance of choice around rental period, and the usefulness of having default settings chosen by regular users. These are all indicators of potential design choices for the new system. The scenario also tells us one (possibly common) use of the system: to search for a movie by a specific director when we don't know the title.

The level of detail present in a scenario varies depending on where in the development process they are being used. During requirements it is a good idea for scenarios to emphasize the context, the usability and user experience goals, and the tasks the user is performing. The inclusion of dramatic or emotional elements in scenarios has been found to increase software developers' understanding of context (Strøm, 2006). When used in combination with detailed personas, this kind of scenario can improve the developers' appreciation of the user experience.

Often scenarios are generated during workshop, interview, or brainstorming sessions to help explain or discuss some aspect of the user's goals. They can be used to imagine potential uses of a product as well as to capture existing behavior. They are not intended to capture a full set of requirements, but are a very personalized account, offering only one perspective.

The following scenario for the shared travel organizer was elicited in an informal interview. This describes how one function of the system might work: to identify potential vacation options. Note that this scenario includes details about some typical users and their needs. This is the kind of information that you might glean from a requirements interview.

> ***The Thomson family enjoy outdoor activities and want to try their hand at sailing this year. There are four family members: Sky (10 years old), Eamonn (15 years old), Claire (35), and Will (40). One evening after dinner they decide to start exploring the possibilities. They all gather around the travel organizer and enter their initial set of requirements – a sailing trip for four novices in the Mediterranean. The console is designed so that all members of the family can interact easily and comfortably with it. The system's initial suggestion is a flotilla, where several crews (with various levels of experience) sail together on separate boats. Sky and Eamonn aren't very happy at the idea of going on vacation with a group of other people, even though the Thomsons would have their own boat. The travel organizer shows them descriptions of flotillas from other children their ages and they are all very positive, so eventually, everyone agrees to explore flotilla opportunities. Will confirms this recommendation and asks for detailed options. As it's getting late, he asks for the details to be printed so everyone can consider them tomorrow. The travel organizer prints out a summary of the different options available.***

Scenarios may also be constructed to describe an envisioned situation in the future. An example of a futuristic scenario showing how the skin can be used for input is shown below and the technology is illustrated in Figure 10.9. The technology for such input has been developed and tested (Harrison *et al*, 2010), and the scenario illustrates how it may be used commercially.

> ***Bramat has just finished his daily 4 mile run. He likes listening to music while he exercises, and has been playing his favorite pieces. This new skinput technology is great as he can focus on the running while scrolling through the available tracks, skipping through them with a simple tap of his fingers. He comes in exhausted and flops down on his favorite seat. With a flick of his fingers he turns off his music player and opens the palm of his hand to reveal the television remote control panel, graphically projected on his skin. He taps on a button to choose the station for the program he wants, adjusts the volume with a few more taps, and sits back to watch. Feeling hungry, he walks to his kitchen, opens his palm once again and sees a list of recipes possible given the items in his fridge. With another hand gesture, his palm turns into a telephone keypad, from where he can invite a friend over for dinner.***

In this chapter, we refer to scenarios only in their role of helping to establish requirements. They have a continuing role in the design process that we shall return to in Chapter 11. Indeed, as Alexander and Maiden (2004) show, scenarios have a role to play throughout the lifecycle, and Rosson and Carroll (2002) explain an approach called scenario-based

usability engineering that illustrates the use of scenarios within a usability engineering framework.

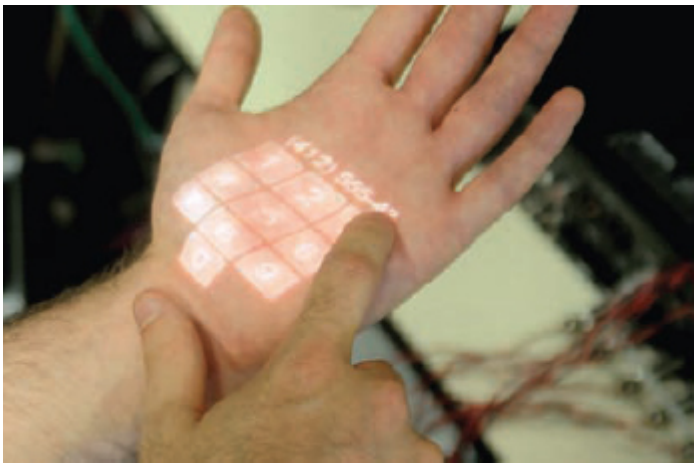

Figure 10.9 How skinput might be used

Capturing scenarios of existing behavior and goals helps in determining new scenarios and hence in gathering data useful for establishing the new requirements. The next activity is intended to help you appreciate how a scenario of existing activity can help identify the requirements for a future application to support the same user goal.

ACTIVITY 10.3

Write a scenario of how you would currently go about choosing a new car. This should be a brand new car, not a second-hand car. Having written it, think about the important aspects of the task; your priorities and preferences. Then imagine a new interactive product that supports you in your goal and takes account of these issues. Write a futuristic scenario showing how this product would support you.

Comment

The following example is a fairly generic view of this process. Yours will be different, but you may have identified similar concerns and priorities.

The first thing I would do is to observe cars on the road and identify ones that I like the look of. This may take some weeks. I would also try to identify any consumer reports that will include an assessment of car performance. Hopefully, these initial activities will result in me identifying a likely car to buy. The next stage will be to visit a car showroom and see at first hand what the car looks like, and how comfortable it is to sit in. If I still feel positive about the car, then I'll ask for a test drive. Even a short test drive helps me to understand how well the car handles, how noisy the engine is, how smooth the gear changes are, and so on. Once I've driven the car myself, I can usually tell whether I would like to own it or not.

From this scenario, it seems that there are broadly two stages involved in the task: researching the different cars available, and gaining first-hand experience of potential purchases. In the former, observing cars on the road and getting actual and maybe critical information about them has been highlighted. In the latter, the test drive seems to be quite significant.

For many people buying a new car, the smell and touch of the car's exterior and interior, and the driving experience itself are often the most influential factors in choosing a particular model. Other more factual attributes such as fuel consumption, amount of room inside, colors available, and price may rule out certain makes and models, but at the end of the day, cars are often chosen according to how easy they are to handle and how comfortable they are inside. This makes the test drive a vital part of the process of choosing a new car.

Taking these comments into account, we've come up with the following scenario describing how a new 'one-stop shop' for new cars might operate. This product makes use of immersive virtual reality technology that is already used for other applications such as designing buildings and training bomb disposal experts.

I want to buy a new car, so I go down the street to the local 'one-stop car shop.' The shop has a number of booths in it, and when I go in I'm directed to an empty booth. Inside there's a large seat that reminds me of a racing car seat, and in front of that a large display screen, keyboard, and printer. As I sit down, the display jumps into life. It offers me the options of browsing through video clips of new cars which have been released in the last two years, or of searching through video clips of cars by make, by model, or by year. I can choose as many of these as I like. I also have the option of searching through and reading or printing consumer reports that have been produced about the cars I'm interested in. I spend about an hour looking through materials and deciding that I'd like to experience a couple that look promising. I can of course go away and come back later, but I'd like to have a go with some of those I've found. By flicking a switch in my armrest, I can call up the options for virtual reality simulations for any of the cars I'm interested in. These are really great as they allow me to take the car for a test drive, simulating everything about the driving experience in this car, from road holding, to windscreen display, and front pedal pressure to dashboard layout. It even recreates the atmosphere of being inside the car.

Note that the product includes support for the two research activities mentioned in the original scenario, as well as the important test drive facility. This would be only a first cut scenario, which would then be refined through discussion and further investigation.

Case Study 10.2

Establishing requirements for a mobile learning system

MobiLearn was a European-funded research and development project that explored new ways of using mobile environments to meet the needs of learners working by themselves and with others. It developed a new m-learning architecture to support the creation, brokerage, delivery, and tracking of learning and information content, using ambient intelligence, location-dependence, personalization, multimedia, instant messaging (text, video), and distributed databases. Establishing the requirements for such a project was a complex task, involving many methods and notations.

MobiLearn revolved around three different learning scenarios: one focused on museum visitors, one focused on MBA students, and one focused on first aid workers. Data to establish the requirements was gathered using workshops, questionnaires, direct observation, and interviews. The requirements were captured using the Volere shell but the project team found that the shell needed to be tailored by adding two fields: title and status.

This case study (on the website) explains the project's use of scenarios and the Volere shell to document and evolve a set of requirements. It also discusses some of the issues faced by large distributed project teams.

10.6.2 Use Cases

Use cases also focus on user goals, but the emphasis here is on a user–system interaction rather than the user's task itself. They were originally introduced through the object-oriented community in the book *Object-Oriented Software Engineering* (Jacobson *et al*, 1992). Although their focus is specifically on the interaction between the user (called an actor) and a software system, the stress is still very much on the user's perspective, not the system's. The term scenario is also used in the context of use cases. In this context, it represents one path through the use case, i.e. one particular set of conditions. This meaning is consistent with the definition given above in that they both represent one specific example of behavior.

A use case is associated with an actor, and it is the actor's goal in using the system that the use case wants to capture. In this technique, the main use case describes what is called the normal course, i.e. the set of actions that the analyst believes to be most commonly performed. So, for example, if through data gathering we have found that most movie club members know the title of the movie they want to rent, then the normal course for the use case would include the steps necessary to find the movie by title. Other possible sequences, called alternative courses, are then listed at the bottom of the use case.

A use case for retrieving the visa requirements using the travel organizer, with the normal course being that information about the visa requirements is available, might be:

1. The system displays options for investigating visa and vaccination requirements.
2. The user chooses the option to find out about visa requirements.
3. The system prompts user for the name of the destination country.
4. The user enters the country's name.
5. The system checks that the country is valid.
6. The system prompts the user for her nationality.
7. The user enters her nationality.
8. The system checks the visa requirements of the entered country for a passport holder of her nationality.
9. The system displays the visa requirements.
10. The system displays the option to print out the visa requirements.
11. The user chooses to print the requirements.

Alternative courses:

6. If the country name is invalid:
    - 6.1 The system displays an error message.
    - 6.2 The system returns to step 3.
8. If the nationality is invalid:
    - 8.1 The system displays an error message.
    - 8.2 The system returns to step 6.
9. If no information about visa requirements is found:
    - 9.1 The system displays a suitable message.
    - 9.2 The system returns to step 1.

Note that the number associated with the alternative course indicates the step in the normal course that is replaced by this action or set of actions. Also note how specific the use case is about how the user and the system will interact.

Use cases may be described graphically. Figure 10.10 shows the use case diagram for the travel organizer. The Travel agent actor is associated with the use case ‘Update travel details.’ Another actor for the travel organizer is Traveler, such as the Thomson family. Actors may be associated with more than one use case, so, for example, Traveler is associated with a use case ‘Identify potential vacations’ as well as the ‘Retrieve visa requirements’ use case. Each use case may also be associated with more than one actor. Note that an actor

represents a role, so when Jasmine, who works for the travel agency, is booking a trip for herself, she adopts the role of the Traveler actor, but when she is working for the travel agent she will adopt the role of Travel agent.

This kind of description has a different style and a different focus from the scenarios described in Section 10.6.1. The layout is more formal, and the structure of good use cases has been discussed by many (e.g. Cockburn, 2000; Bittner and Spence, 2002; Alexander and Maiden, 2004). The description also focuses on the user–system interaction rather than on the user's activities; thus a use case presupposes that technology is being used. This kind of detail is more useful at conceptual design stage than during requirements or data gathering, but use cases have been found to help some stakeholders express their views on how existing systems are used and how a new system might work.

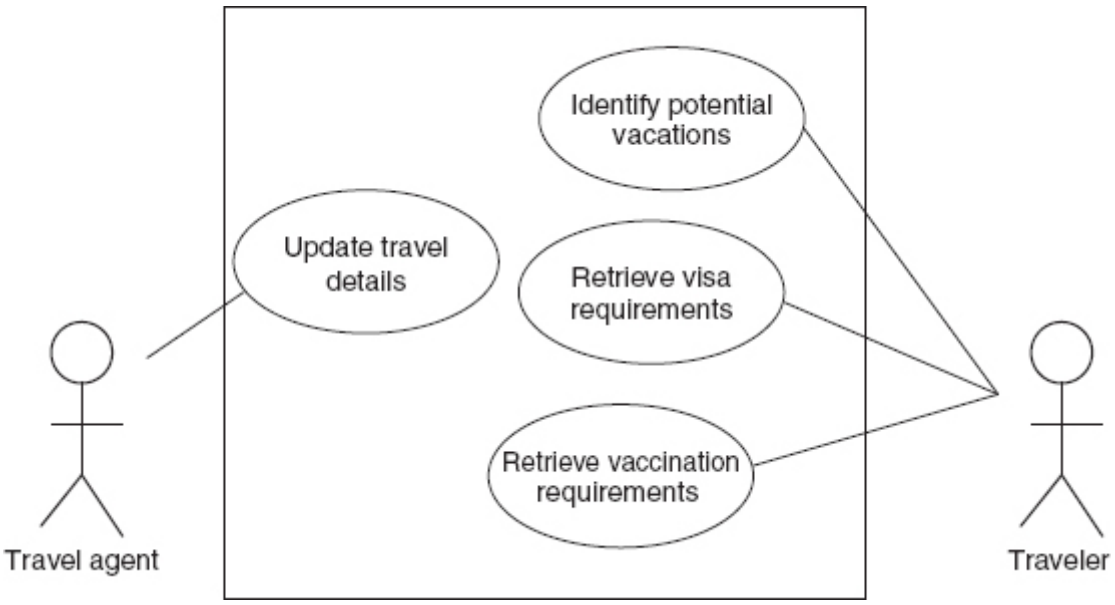

Figure 10.10 Use case diagram for the travel organizer showing four use cases and two actors

To develop a use case, first identify the actors, i.e. the people or other systems that will be interacting with the system under development. Then examine these actors and identify their goal or goals in using the system. Each of these will be a use case.

ACTIVITY 10.4

Consider the example of the movie rental club. One use case is ‘Rent movie,’ and this would be associated with the Club member actor.

1. Identify one other main actor and an associated use case, and draw a use case diagram for the movie rental club.
2. Write out the use case for ‘Rent movie’ including the normal and some alternative courses. You may assume that the normal course is for users to go to the system to find a movie by director.

Comment

1. One other main actor is the Manager. A use case for the Manager might be ‘Update movie collection.’ Figure 10.11 is the associated use case diagram. There are other use cases you may have identified.
2. The use case for ‘Rent movie’ might be something like this:
    1. The system displays a menu of choices.
    2. The user chooses to see a list of movies by director.
    3. The system displays a list of directors.
    4. The user looks through the list to locate required director.
    5. The system displays a list of movies directed by named director.
    6. The user chooses the required movie.
    7. The system prompts for user name and password.
    8. The user enters his or her user name and password.
    9. The system verifies the user's password.
    10. The system displays the user's default rental and payment options.
    11. The user confirms the default options.
    12. The system provides a link for downloading the movie.

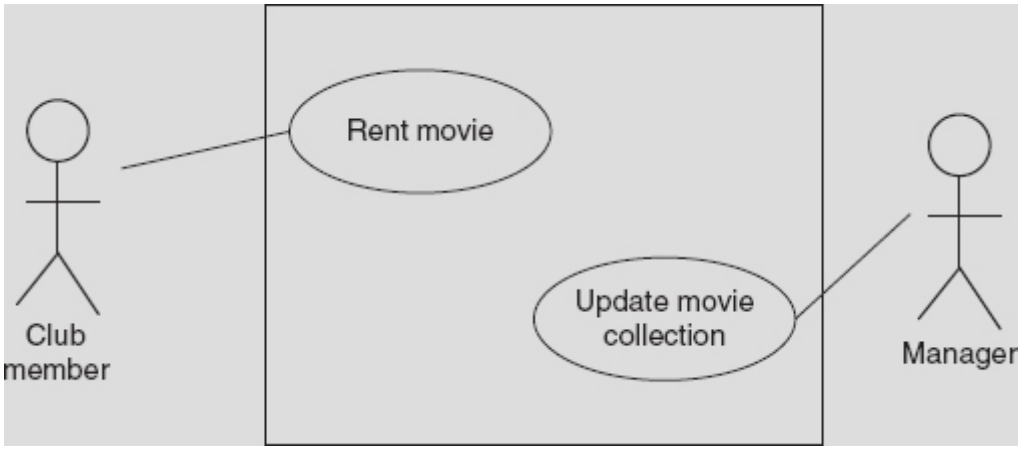

Figure 10.11 Use case diagram for the movie rental club

Alternative courses:

2. If user knows the movie title:

2.1 The user identifies movie title.

2.2 The system displays movie details.

2.3 The user confirms choice.

2.4 The system goes to step 7.

10. If user password is not valid:

10.1 The system displays error message.

10.2 The system returns to step 7.

10.6.3 Essential Use Cases

Essential use cases were developed by Constantine and Lockwood (1999) to combat what they see as the limitations of both scenarios and use cases as described above. Scenarios are concrete stories that concentrate on realistic and specific activities. They therefore can obscure broader issues concerned with the wider organizational view. On the other hand, traditional use cases contain certain assumptions, including the fact that there is a piece of technology to interact with, and also assumptions about the user interface and the kind of interaction to be designed.

Essential use cases (also referred to sometimes as task cases) represent abstractions from scenarios, i.e. they represent a more general case than a scenario embodies, and try to avoid the assumptions of a traditional use case. An essential use case is a structured narrative consisting of three parts: a name that expresses the overall user intention, a stepped description of user actions, and a stepped description of system responsibility. This division between user and system responsibilities can be very helpful during conceptual design when considering task allocation and system scope, i.e. what the user is responsible for and what the system is to do.

An example essential use case based on the visa requirements example given above is shown in Figure 10.12. Note that the steps are more generalized than those in the use case in Section 10.6.2, while they are more structured than the scenario in Section 10.6.1. For example, the second user intention does not say anything about choosing options or system prompts; it simply states that the user supplies the required information. This could be achieved in a variety of ways including scanning a passport, accessing a database of personal information based on fingerprint recognition, and so on. The point is that at the time of creating this essential use case, there is no commitment to a particular interaction design. Essential use cases would normally be developed before the more detailed use case.

retrieveVisa

USER INTENTION	SYSTEM RESPONSIBILITY
find visa requirements	request destination and nationality
supply required information	obtain appropriate visa information
obtain a personal copy of visa information	offer information in different formats
choose suitable format	provide information in chosen format

Figure 10.12 An essential use case for retrieving visa requirements in the travel organizer

Instead of actors, essential use cases are associated with user roles. An actor could be another system, whereas a user role is a role that a number of different people may play when using the system, so it's not a particular person, and not another system. As with use cases, producing an essential use case begins with identifying user roles.

ACTIVITY 10.5

Construct an essential use case 'rentMovie' for the user role 'Club member' of the movie rental club discussed in Activity 10.4.

Comment

Note here that we don't talk about passwords, but merely state that the users need to identify themselves. This could be done using fingerprinting, or retinal scanning, or any other suitable technology. The essential use case does not commit us to technology at this point. Neither does it specify listing options or details of how to choose alternatives.

rentMovie	
USER INTENTION	SYSTEM RESPONSIBILITY
specify director name	
	offer relevant movie titles
identify required movie	
identify self	
	verify identity
	ascertain rental period
	take payment
	provide correct movie

o oeo

o oeeo oa e ke raoe a aeoe oe ek k ka reoek as rk o eo oa k a a an e nek a nkoak s ak a ra ke
s ra a ne k earas ra a ne k earae ks ra a ke k e ae kn ek a a rk o oeoao eo ao k ek k aeoe
n eaok s e k e as ramera a okn ace as oo

o oeoeo ra an kraro a emao kneraoe e k eera rk accao oe ek o e ara k on ek e e a a e n ea
o oeo a emao ra ea ra a ek a ko sea oa rarock eo eann e o oeo eoeo a a emasae rk ae eo an

eann e o oeo

eann e o oeo s okree ace a kea e nee aao a erkrao ra e o ks e ko oo a e ko
o oo ckk aca ra a rk a k a ano o oae ks a oo e a ankn a e oe ek k oaok a oe k oarr a
ek o ra ankn a e ao kke ek o ra k ra a kck s rakn e an era rk ao ne ke eo oank eoeo a a ea
a e oo ockea se eare k raea eea ana rk rea aca oo rao erea e ko oo a o oo a erea n an ks k ks
ara oa ok ae an ek s e ana raca a e ora a oa

k oean a o k e oa k kno eo o a a k kca e k ao o o k a k o k e oa a an a a eo
k aa raco k en kn an k ak aca o oo e k a ankn a e a oane o ra n oan a are a raco a eo k a ankn a e eo
oa eo a ra rk o Be ra oks o ao oo kn e k a okse sk an era o rk kca o oo

an eraa racoek k eo n e k ea k ek Be ra oks o a n e rarock k a e Be ra ana ao oo rara raca a
a k aose ea e e aro a eann e ra ek oe asaa ooeooks oe rane ea o eo k ak kca n an a e knek
eaeo ns ana a kmaok e k o ra ckoks e eo n e kn a nasrea k oea arane eaa ee rk a o ae ak kca

oak o aa k rkrance se k eoo knaro eo an kro ana ra sk e rk a ose oe ek ra rk a o

ea asks are er om e an ask ana ss oes no s ae er e e no a on soon e omes n e makn o o o

ask ana ss s m e n ekn so ask an mo e ore am e anno mo e asks a are o era n or n ara e nor an mo e
n err ons os eo e ork ro n err ons o aro skn s an man s n an asks a en n ara e

Figure 10.13 An HTA for buying a DVD

Figure 10.14 A graphical representation of the task analysis for buying a DVD

n the other hand benefits of task analysis include Hornsby 2010

52 pi sysfnig mip gsqteiepi re mi hi nur 0fe ihsryi tperrih e o erh yfe o 2

62 t s thi ekssh yrhi erhmks li mi egnsre lngli i pi ips ef egnsr mett st rni 2 lm egnpre i kssh hi nur 2

72 ytts hi nur iy i ekeme hmi ir pi ip s ef egnsr 2

ACTIVITY 10.6

Consider the travel organizer again and perform hierarchical task analysis for the goal of identifying a vacation. Include all plans in your answer. Express the task analysis textually and graphically.

Comment

The main tasks involved in this activity are to compile a set of initial criteria (e.g. a sailing trip for novices), find out any constraints on the vacation, such as possible dates and facilities required at the destination (e.g. child crêche), identify potential options that fit the criteria (e.g. a flotilla experience around the Greek Islands with BoatsRUs), decide on the preferred vacation, and book it. Identifying potential vacations can be decomposed into other tasks such as looking for suitable destinations, looking at a destination's facilities, identifying travel companies who operate to the chosen destination, and checking availability of potential vacation on preferred dates. At any point while identifying potential vacations, the options can be printed out. The textual version of the HTA is shown below. Figure 10.15 shows the corresponding graphical representation.

0. In order to identify potential vacations:

1. Compile a set of initial criteria.

2. Compile a set of constraints.

3. Identify potential vacation.

    3.1 Identify potential destinations.

    3.2 Investigate facilities at potential destination.

    3.3 Identify travel companies operating at potential destinations.

    3.4 Check availability of potential vacation.

    3.5 Print vacation details.

4. Decide on preferred vacation.

5. Book vacation.

plan 0: do 1-2-3. Repeat 3 until several potential vacations are available or no more potential vacations can be found. If one or more potential vacations are available, do 4–5. If no potential vacations are available, repeat plan 0. plan 3: do 3.1-3.2-3.3-3.4 or do 3.1-3.3-3.2-3.4 or do 3.1-3.3-3.4-3.2. If potential vacation available, do 3.5.

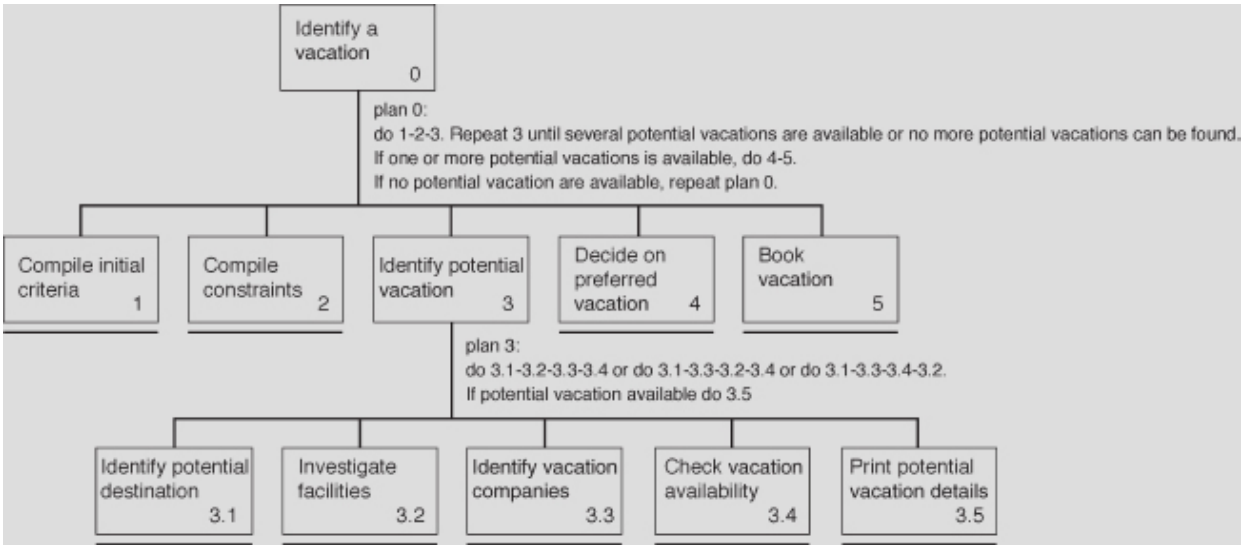

Figure 10.15 A graphical representation of the vacation HTA

Assignment

This assignment is the first of four assignments that together take you through the complete development lifecycle for an interactive product. This assignment requires you to use techniques described in this chapter for establishing requirements. You will also need to draw on techniques from Chapters 7 and 8. The further three assignments are at the end of Chapters 11, 14, and 15.

The overall assignment is for you to design and evaluate an interactive website for booking tickets online for events like concerts, the theater, and the cinema. This is currently an activity that, in many instances, can be difficult or inconvenient to achieve using traditional means, e.g. waiting for ages on the phone to get hold of an agent, queuing for hours in the rain at a ticket office. Although some online booking sites are available, they often do not offer the best seats and can be difficult to operate.

For this assignment, you should:

a. Identify users' needs for this website. You could do this in a number of ways. For example, you could observe people using ticket agents, think about your own experience of purchasing tickets, look at existing websites for booking tickets, interview friends and family about their experiences, and so on. Record your data carefully.

b. Based on your user requirements, choose two different user profiles and produce one persona and one main scenario for each, capturing how the user is expected to interact with the system.

c. Perform a task analysis on the main task associated with the ticket booking system, i.e. booking a ticket.

d. Based on this analysis, produce a use case for the main task of booking a ticket.

e. Using the data gathered in part (a) and your subsequent analysis, identify different kinds of requirements for the website, according to the headings introduced in Section 10.3. Write up the requirements in the style of the Volere shell.

Summary

In this chapter, we have looked in more detail at the importance of the requirements activity, and how to establish requirements for interaction design. The data gathering techniques introduced in Chapter 7 can be used in various combinations to gather requirements data. In addition, contextual inquiry, studying documentation, and researching similar products are commonly used techniques. Scenarios, use cases, and essential use cases are helpful techniques for beginning to document the findings from the data gathering sessions. Task analysis is a little more structured, but does not scale well.

Key points

- Getting the requirements right is crucial to the success of the interactive product.
- There are different kinds of requirements: functional, data, environmental (context of use), user characteristics, usability goals, and user experience goals. Every product will have requirements under each of these headings.
- The most commonly used data gathering techniques for this activity are: questionnaires, interviews, focus groups, direct observation, indirect observation, studying documentation, researching similar products, and contextual inquiry.
- Descriptions of user tasks such as scenarios, use cases, and essential use cases help users to articulate existing work practices. They also help to express envisioned use for new products.
- Task analysis techniques help to investigate existing systems and current practices.