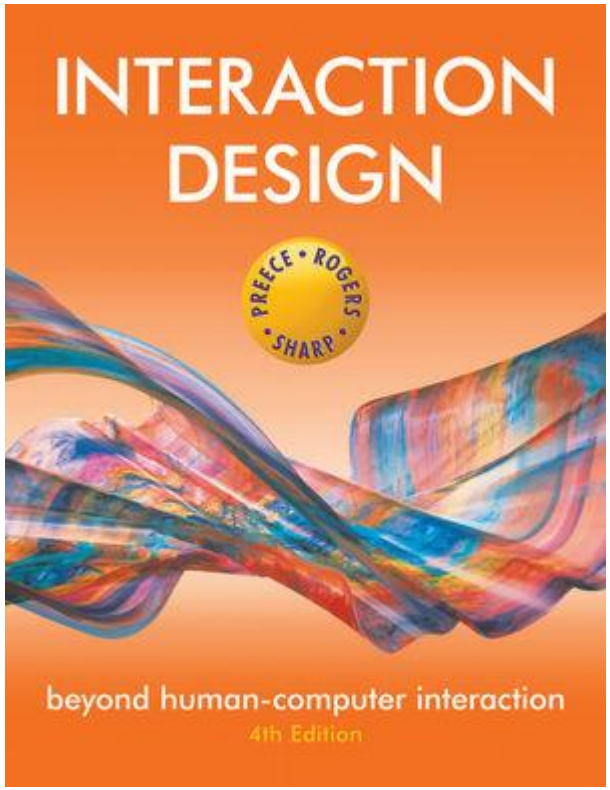


Chapter 10

ESTABLISHING REQUIREMENTS

Overview

- The importance of requirements
- Different types of requirements
- Data gathering for requirements
- Data analysis and presentation
- Task description: Scenarios

  Use Cases

  Essential use cases
- Task analysis: HTA

What, how and why?

What needs to be achieved?

1. Understand as much as possible about users, task, context
2. Produce a stable set of requirements

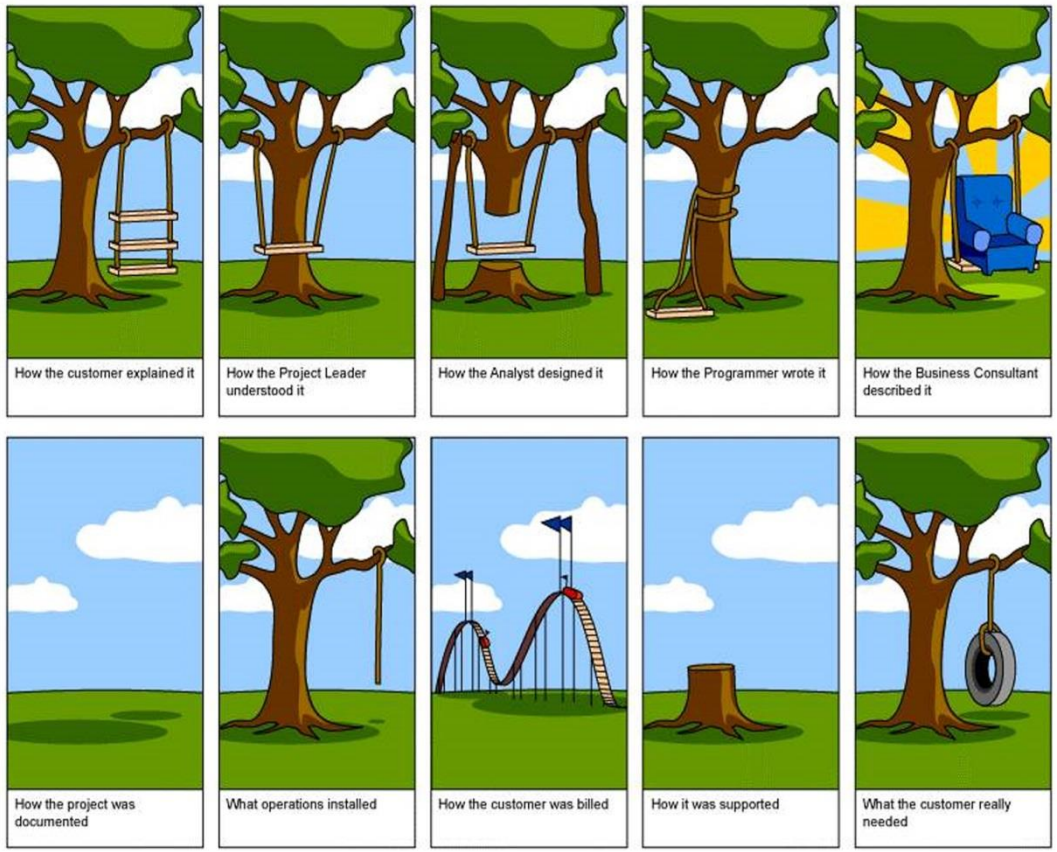
How can this be done?

- Data gathering activities
- Data analysis activities
- Expression as ‘requirements’
- All of this is iterative

What, how and why?

- Why bother?

  Requirements definition is the stage where failure occurs most commonly



Getting requirements right is crucial

Establishing requirements

- What do users want? What do users ‘need’?

  Requirements need clarification, refinement, completion, re-scoping

  Input: Requirements document (maybe)

  Output: stable requirements

- Why ‘establish’?

  Requirements arise from understanding users’ needs

  Requirements can be justified & related to data

Volere shell

Requirement #: **75** Requirement Type: **9** Event/use case #: **6**

Description: **The product shall issue an alert if a weather station fails to transmit readings.**

Rationale: **Failure to transmit readings might indicate that the weather station is faulty and needs maintenance, and that the data used to predict freezing roads may be incomplete.**

Source: **Road Engineers**
Fit Criterion: **For each weather station the product shall communicate to the user when the recorded number of each type of reading per hour is not within the manufacturer's specified range of the expected number of readings per hour.**

Customer Satisfaction: **3** Customer Dissatisfaction: **5**

Dependencies: **None** Conflicts: **None**

Supporting Materials: **Specification of Rosa Weather Station**

History: **Raised by GBS, 28 July 99**

Volere

Volere requirements template

PROJECT DRIVERS

1. The Purpose of the Product
2. The Stakeholders

PROJECT CONSTRAINTS

3. Mandated Constraints
4. Naming Conventions and Definitions
5. Relevant Facts and Assumptions

FUNCTIONAL REQUIREMENTS

6. The Scope of the Work
7. Business Data Model and Data Dictionary
8. The Scope of the Product
9. Functional and Data Requirements

NON-FUNCTIONAL REQUIREMENTS

10. Look and Feel Requirements
11. Usability and Humanity Requirements
12. Performance Requirements
13. Operational and Environmental Requirements
14. Maintainability and Support Requirements
15. Security Requirements
16. Cultural and Political Requirements
17. Legal Requirements

PROJECT ISSUES

18. Open Issues
19. Off-the-Shelf Solutions
20. New Problems
21. Tasks
22. Migration to the New Product
23. Risks
24. Costs
25. User Documentation and Training
26. Waiting Room
27. Ideas for Solutions

Different kinds of requirements

- Functional:
  - —What the system should do
- (Non-functional: security, response time...)
- Data:
  - —What kinds of data need to be stored?
  - —How will they be stored (e.g. database)?

Different kinds of requirements

Environment or context of use:

- physical: dusty? noisy? vibration? light? heat? humidity? …. (e.g. ATM)
- social: sharing of files, of displays, in paper, across great distances, synchronous, privacy for clients
- organisational: hierarchy, IT department's attitude and remit, user support, communications structure and infrastructure, availability of training

Underwater computing

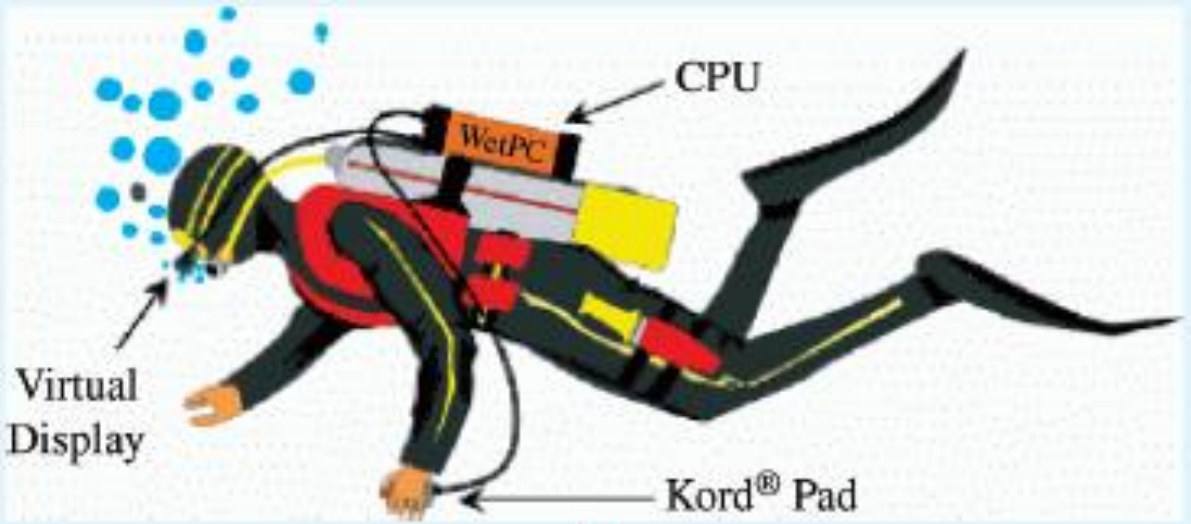
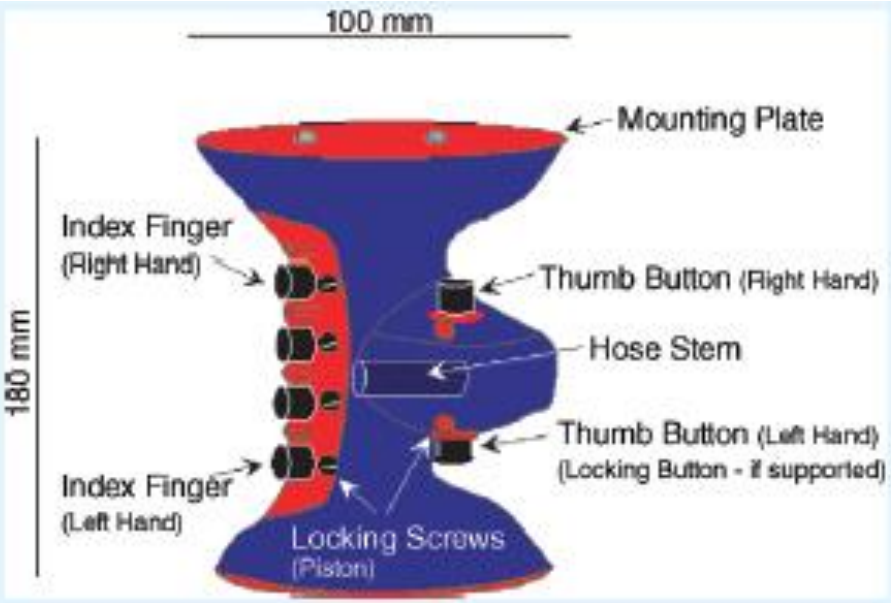


Figure 10.2 (a) The components of WetPC's underwater computer.

Source: Reproduced by permission of WetPC Pty Ltd. http://www.wetpc.com.au/WetPC.

Underwater computing



(a)

(b)

Figure 10.3 (a) The KordGrip interface and (b) the KordGrip in use underwater

Source: (a) Reproduced by permission of WetPC Pty Ltd (b) Reproduced by permission of the Australian Institute of Marine Science.

Different kinds of requirements

Users: Who are they?

- — Characteristics: nationality, educational background, attitude to computers
- — System use: novice, expert, casual, frequent
  - — Novice: prompted, constrained, clear
  - — Expert: flexibility, access/power
  - — Frequent: short cuts
  - — Casual/infrequent: clear menu paths

What are the users' capabilities?

Humans vary in many dimensions:

— size of hands may affect the size and positioning of input buttons

— motor abilities may affect the suitability of certain input and output devices

— height if designing a physical kiosk

— strength - a child's toy requires little strength to operate, but greater strength to change batteries

— disabilities (e.g. sight, hearing, dexterity)

Personas

- Capture a set of user characteristics (user profile)
- Not real people, but synthesised from real users
- Should not be idealised
- Bring them to life with a name, characteristics, goals, personal background
- Develop a small set of personas with one primary

Example Persona

CAPLIN

Ginnie

BACKGROUND

- 15, Female
- Ongoing Private Education
- Ambitious
- Comfortable using technology to communicate

MOTIVATIONS

- Keeping in touch with her network
- Fashion/street cred
- Keeping up with peers.

FRUSTRATIONS

- Sad people trying to be 'friends' on Facebook
- Having to be in bed @ 11pm
- Being swamped in friends updates
- Missing important status updates

Receives private tutoring in Maths and English as these are not her strong subjects. Enjoys playing for the school's 2nd teams for netball and Lacrosse and is good at art.

She loves recording her favourite shows: ER and Sun Valley High on Sky+ and spends some of her time on her Laptop that Daddy bought her watching videos on YouTube, downloading music, keeping up to date with her friends on Facebook and chatting via MS IM to her cousin who is at University in Leeds.

She loves Ugg boots and Abercrombie & Fitch and uses the Internet to shop and find the cheapest prices.

"I want to easily hook up with my friends whilst watching TV"

Data gathering for requirements

- Interviews:
  - Props, e.g. sample scenarios of use, prototypes, can be used in interviews
  - Good for exploring issues
  - Development team members can connect with stakeholders
- Focus groups:
  - Group interviews
  - Good at gaining a consensus view and/or highlighting areas of conflict
  - But can be dominated by individuals

Data gathering for requirements

- Questionnaires:
  - — Often used in conjunction with other techniques
  - — Can give quantitative or qualitative data
  - — Good for answering specific questions from a large, dispersed group of people
- Researching similar products:
  - — Good for prompting requirements

Data gathering for requirements

- Direct observation:
  - — Gain insights into stakeholders' tasks
  - — Good for understanding the nature and context of the tasks
  - — But, it requires time and commitment from a member of the design team, and it can result in a huge amount of data
- Indirect observation:
  - — Not often used in requirements activity
  - — Good for logging current tasks

Data gathering for requirements

Studying documentation:

- Procedures and rules are often written down in manuals
- Good source of data about the steps involved in an activity, and any regulations governing a task
- Not to be used in isolation
- Good for understanding legislation, and getting background information
- No stakeholder time, which is a limiting factor on the other techniques

Some examples

Figure 10.5 A cultural probe package

Source: B. Gaver, T. Dunne and E. Pacenti (1999): "Cultural Probes" from *Interactions* 6(1) pp.21–29.

Cultural probes

Some examples

Ethnographic study, interviews, usability tests, and user participation

(a) (b)

Figure 10.6 (a) Exploring mouse gene expression using G-nome Surfer 2.0 (b) G-nome Surfer Pro displaying the chromosome visualizations, an aligned sequence, and publications

Source: Shaer *et al* (2012) The design, development, and deployment of a tabletop interface for collaborative exploration of genomic data, *International Journal of Human–Computer Interaction* 70, 746–764. ©2012 Association for Computing Machinery, Inc. Reprinted by permission.

Contextual Inquiry

- An approach to ethnographic study where user is expert, designer is apprentice
- A form of interview, but
  - — at users' workplace (workstation)
  - — 2 to 3 hours long
- Four main principles:
  - — Context: see workplace & what happens
  - — Partnership: user and developer collaborate
  - — Interpretation: observations interpreted by user and developer together
  - — Focus: project focus to understand what to look for

Considerations for data gathering (1)

- Identifying and involving stakeholders:
  users, managers, developers, customer reps?, union reps?, shareholders?
- Involving stakeholders: workshops, interviews, workplace studies, co-opt stakeholders onto the development team
- ‘Real’ users, not managers
- Political problems within the organisation
- Dominance of certain stakeholders
- Economic and business environment changes
- Balancing functional and usability demands

Considerations for data gathering (2)

- Requirements management: version control, ownership
- Communication between parties:
  - —within development team
  - —with customer/user
  - —between users… different parts of an organisation use different terminology
- Domain knowledge distributed and implicit:
  - —difficult to dig up and understand
  - —knowledge articulation: how do you walk?
- Availability of key people

Data gathering guidelines

- Focus on identifying the stakeholders' needs
- Involve all the stakeholder groups
- Involve more than one representative from each stakeholder group
- Use a combination of data gathering techniques
- Support the process with props such as prototypes and task descriptions

Data interpretation and analysis

- Start soon after data gathering session
- Initial interpretation before deeper analysis
- Different approaches emphasize different elements e.g. class diagrams for object-oriented systems, entity-relationship diagrams for data intensive systems

Task descriptions

- Scenarios
  - an informal narrative story, simple, ‘natural’, personal, not generalisable
- Use cases
  - assume interaction with a system
  - assume detailed understanding of the interaction
- Essential use cases
  - abstract away from the details
  - does not have the same assumptions as use cases

Scenario for travel organizer

“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 saved so everyone can consider them tomorrow. The travel organizer emails them a summary of the different options available.”

Scenarios and Personas

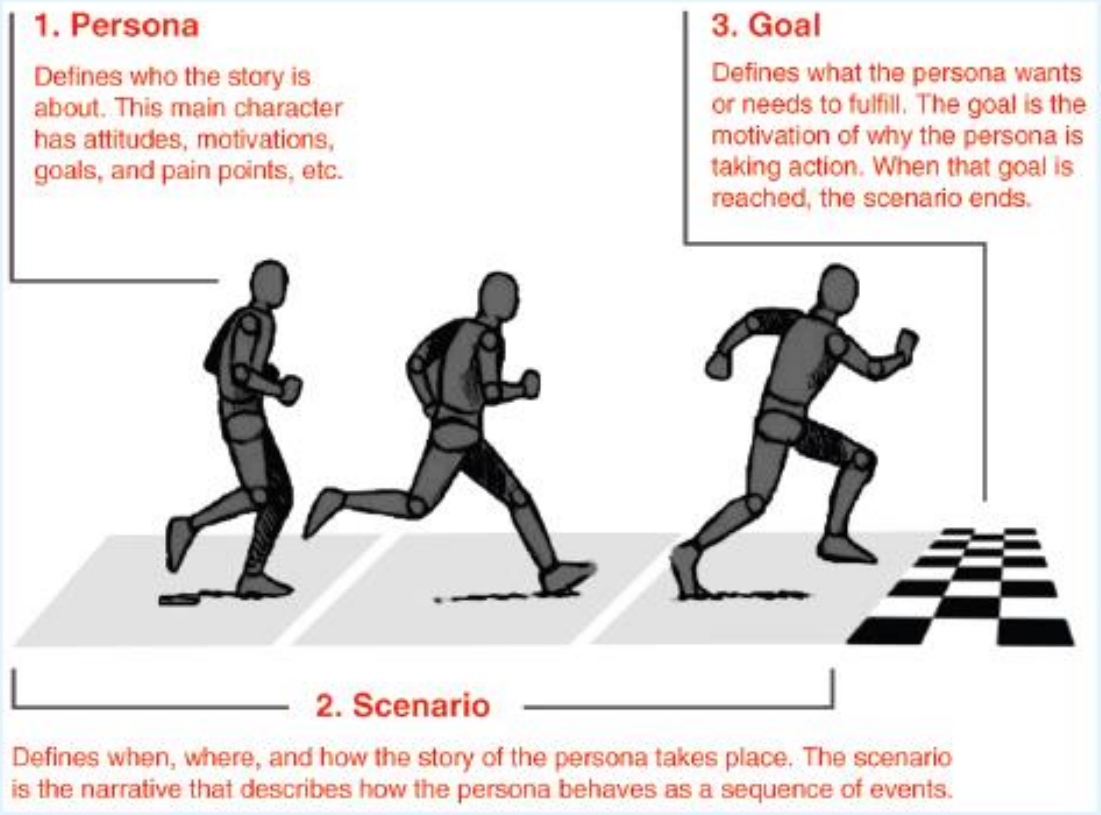


Figure 10.10 The relationship between a scenario and its associated persona

Source: http://www.smashingmagazine.com/2014/08/06/a-closer-look-at-personas-part-1/

Use case for travel organizer

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 for travel organizer

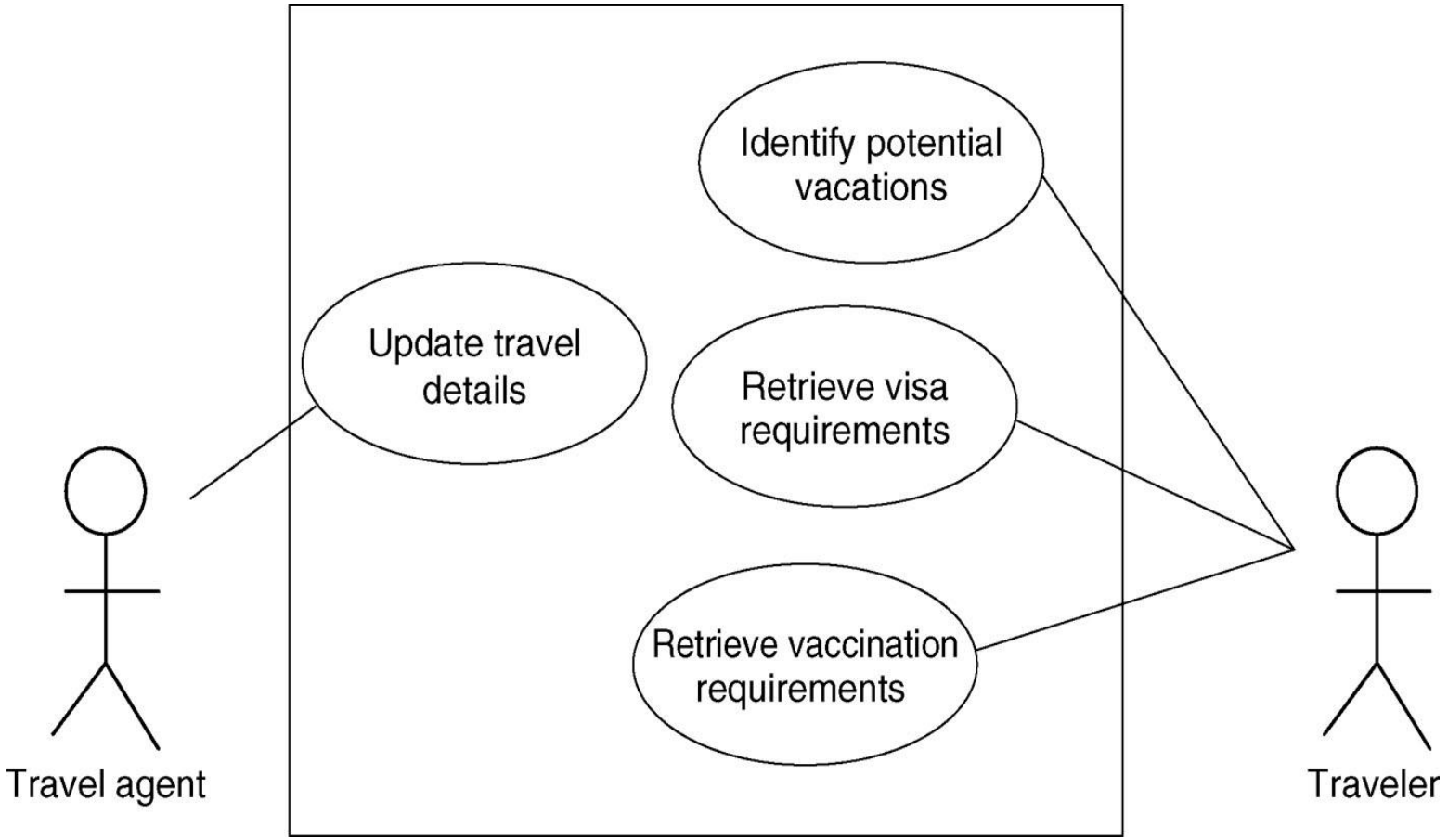
Some alternative courses:

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.

Example use case diagram for travel organizer



Example essential use case for travel organizer

retrieve Visa

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

Task analysis

- Task descriptions are often used to envision new systems or devices
- Task analysis is used mainly to investigate an existing situation
- It is important not to focus on superficial activities
  - What are people trying to achieve?
  - Why are they trying to achieve it?
  - How are they going about it?
- Many techniques, the most popular is Hierarchical Task Analysis (HTA)

Hierarchical Task Analysis

- Involves breaking a task down into subtasks, then sub-sub-tasks and so on. These are grouped as plans which specify how the tasks might be performed in practice
- HTA focuses on physical and observable actions, and includes looking at actions not related to software or an interaction device
- Start with a user goal which is examined and the main tasks for achieving it are identified
- Tasks are sub-divided into sub-tasks

Example Hierarchical Task Analysis

0. In order to buy a DVD
1. locate DVD
2. add DVD to shopping basket
3. enter payment details
4. complete address
5. confirm order

plan 0: If regular user do 1-2-5.
If new user do 1-2-3-4-5.

Example Hierarchical Task Analysis (graphical)

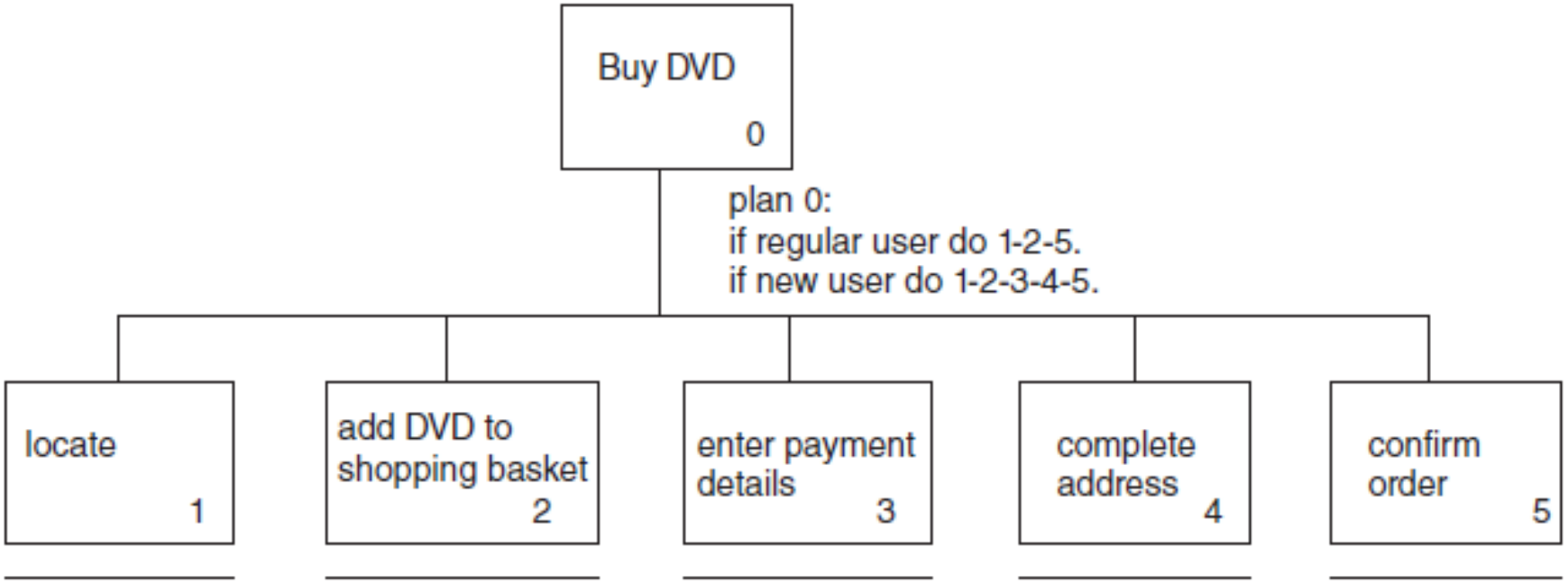


Figure 10.15 A graphical representation of the task analysis for buying a DVD

Summary

- Getting requirements right is crucial
- There are different kinds of requirement, each is significant for interaction design
- The most commonly-used techniques for data gathering are: questionnaires, interviews, focus groups, direct observation, studying documentation and researching similar products
- Scenarios, use cases and essential use cases can be used to articulate existing and envisioned work practices.
- Task analysis techniques such as HTA help to investigate existing systems and practices