Chapter 5

Chapter 5 Requirements Analysis

Learning Objective

… Understanding what the *customer* requires from a software system.

Frederick T Sheldon
Assistant Professor of Computer Science
Washington State University

Requirements Analysis

Understanding the customer's requirements for a software system

CS 422 Software Engineering Principles
From *Software Engineering* by I. Sommerville, 1996.

Chapter 5
Slide 2

Objectives

To describe different approaches to requirements discovery

To explain the need for multi-perspective analysis

To illustrate a structured approach to requirements analysis

To explain why social and organizational factors influence system requirements

CS 422 Software Engineering Principles
From *Software Engineering* by I. Sommerville, 1996.

Chapter 5
Slide 3

Topics covered

Viewpoint-oriented analysis

Method-based analysis

System contexts

Social and organizational factors

CS 422 Software Engineering Principles
From *Software Engineering* by I. Sommerville, 1996.

Chapter 5
Slide 4

Requirements analysis

Sometimes called requirements elicitation or requirements discovery

Involves technical staff working with customers to find out about the application domain, the services that the system should provide and the system's operational constraints

May involve end-users, managers, engineers involved in maintenance, domain experts, trade unions, etc. These are called *stakeholders*

CS 422 Software Engineering Principles
From *Software Engineering* by I. Sommerville, 1996.
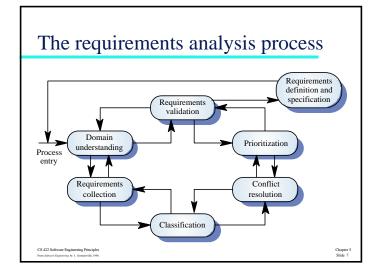
Chapter 5
Slide 5

Problems of requirements analysis

Stakeholders don't know what they really want

Stakeholders express requirements in their own terms

Different stakeholders may have conflicting requirements

Organizational and political factors may influence the system requirements

The requirements change during the analysis process. New stakeholders may emerge

CS 422 Software Engineering Principles
From *Software Engineering* by I. Sommerville, 1996.

Chapter 5
Slide 6

The requirements analysis process

Process entry

Process activities

Domain understanding

Requirements collection

Classification

Conflict resolution

Prioritization

Requirements validation

System models

Different models may be produced during the requirements analysis activity

Requirements analysis may involve three structuring activities which result in these different models

Partitioning. Identifies the structural (part-of) relationships between entities

Abstraction. Identifies generalities among entities

Projection. Identifies different ways of looking at a problem

System models covered in Chapter 6

Viewpoint-oriented analysis

Stakeholders represent different ways of looking at a problem or problem viewpoints

This multi-perspective analysis is important as there is no single correct way to analyze system requirements

Autoteller system

The example used here is an auto-teller system which provides some automated banking services

I use a very simplified system which offers some services to customers of the bank who own the system and a narrower range of services to other customers

Services include cash withdrawal, message passing (send a message to request a service), ordering a statement and transferring funds

Autoteller viewpoints

Bank customers

Representatives of other banks
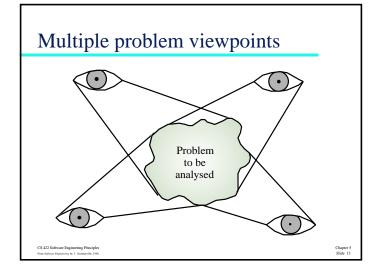
Hardware and software maintenance engineers

Marketing department

Bank managers and counter staff

Database administrators and security staff

Communications engineers

Personnel department

Multiple problem viewpoints

Problem
to be
analysed

Types of viewpoint

Data sources or sinks

Viewpoints are responsible for producing or consuming data.
Analysis involves checking that data is produced and consumed and
that assumptions about the source and sink of data are valid

Representation frameworks

Viewpoints represent particular types of system model. These may be
compared to discover requirements that would be missed using a
single representation. Particularly suitable for real-time systems

Receivers of services

Viewpoints are external to the system and receive services from it.
Most suited to interactive systems

External viewpoints

Natural to think of end-users as receivers of system
services

Viewpoints are a natural way to structure
requirements elicitation

It is relatively easy to decide if a viewpoint is valid

Viewpoints and services may be sued to structure
non-functional requirements
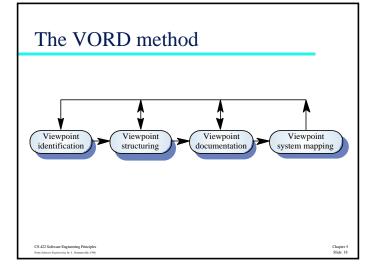
Method-based analysis

Widely used approach to requirements analysis. Depends on the application of a structured method to understand the system
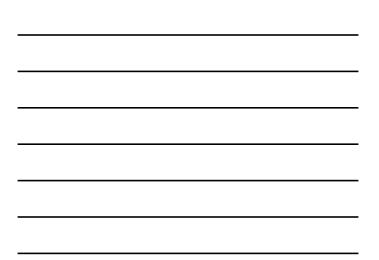
Methods have different emphases. Some are designed for requirements elicitation, others are close to design methods

A viewpoint-oriented method (VORD) is used as an example here. It also illustrates the use of viewpoints

Structured methods

Process model

System modeling notations

Rules applied to the system model

Design guidelines

Report templates

The VORD method

VORD process model

Viewpoint identification
Discover viewpoints which receive system services and identify the services provided to each viewpoint

Viewpoint structuring
Group related viewpoints into a hierarchy. Common services are provided at higher-levels in the hierarchy
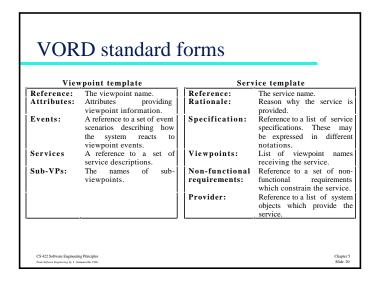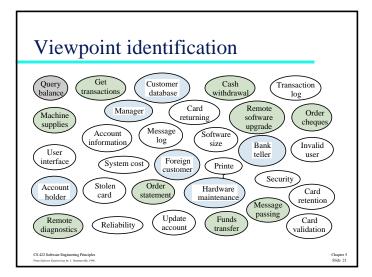
Viewpoint documentation
Refine the description of the identified viewpoints and services

Viewpoint-system mapping
Transform the analysis to an object-oriented design

VORD standard forms

Viewpoint template	
Reference:	The viewpoint name.
Attributes:	Attributes providing viewpoint information.
Events:	A reference to a set of event scenarios describing how the system reacts to viewpoint events.
Services	A reference to a set of service descriptions.
Sub-VPs:	The names of sub-viewpoints.

Service template	
Reference:	The service name.
Rationale:	Reason why the service is provided.
Specification:	Reference to a list of service specifications. These may be expressed in different notations.
Viewpoints:	List of viewpoint names receiving the service.
Non-functional requirements:	Reference to a set of non-functional requirements which constrain the service.
Provider:	Reference to a list of system objects which provide the service.

Viewpoint identification

Viewpoint service information

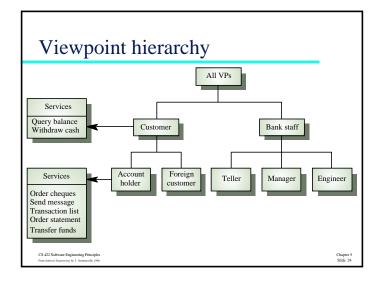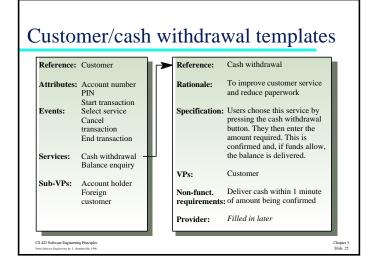| ACCOUNT HOLDER	FOREIGN CUSTOMER	BANK TELLER
Service list	**Service list**	**Service list**
Withdraw cash Query balance Order cheques Send message Transaction list Order statement Transfer funds	Withdraw cash Query balance	Run diagnostics Add cash Add paper Send message

---

Viewpoint data/control

ACCOUNT HOLDER

Control input	Data input
Start transaction Cancel transaction End transaction Select service	Card details PIN Amount required Message

---

Viewpoint hierarchy

Services
Query balance
Withdraw cash

All VPs

Customer

Bank staff

Services
Order cheques
Send message
Transaction list
Order statement
Transfer funds

Account holder

Foreign customer

Teller

Manager

Engineer

Customer/cash withdrawal templates

Reference:	Customer
Attributes:	Account number PIN
Events:	Start transaction Select service Cancel transaction End transaction
Services:	Cash withdrawal Balance enquiry
Sub-VPs:	Account holder Foreign customer

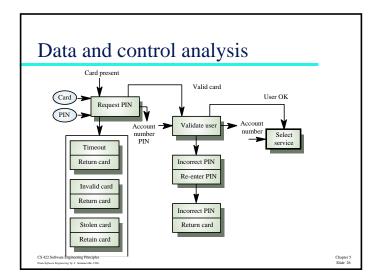|	
Reference:	Cash withdrawal
Rationale:	To improve customer service and reduce paperwork
Specification:	Users choose this service by pressing the cash withdrawal button. They then enter the amount required. This is confirmed and, if funds allow, the balance is delivered.
VPs:	Customer
Non-funct. requirements:	Deliver cash within 1 minute of amount being confirmed
Provider:	*Filled in later*

Data and control analysis

Notation for data and control analysis

- Ellipses. data provided from or delivered to a viewpoint
- Control information enters and leaves at the top of each box
- Data leaves from the right of each box
- Exceptions are shown at the bottom of each box
- Name of next event is in box with thick edges

Exception description

Most methods do not include facilities for describing exceptions

In this example, exceptions are

Timeout. Customer fails to enter a PIN within the allowed time limit

Invalid card. The card is not recognized and is returned

Stolen card. The card has been registered as stolen and is retained by the machine

Method advantages/disadvantages

Methods impose structure on the requirements analysis process

May be supported by CASE tools

Can be applied systematically and can lead naturally to design

However, forces system modeling using a computational framework

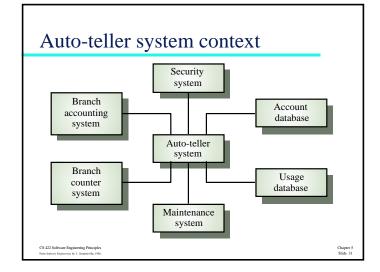Methods fail to adequately provide for the description of human activities

System contexts

The boundaries of the system must be established to determine what must be implemented

These are documented using a description of the system context. This should include a description of the other systems which are in the environment

Social and organizational factors may influence the positioning of the system boundary

Auto-teller system context

Social and organizational factors

Software systems are used in a social and organizational context. This can influence or even dominate the system requirements

Social and organizational factors are not a single viewpoint but are influences on all viewpoints

Good analysts must be sensitive to these factors but currently no systematic way to tackle their analysis

Example

Consider a system which allows senior management to access information without going through middle managers

Managerial status. Senior managers may feel that they are too important to use a keyboard. This may limit the type of system interface used

Managerial responsibilities. Managers may have no uninterrupted time where they can learn to use the system

Organizational resistance. middle managers who will be made redundant may deliberately provide misleading or incomplete information so that the system will fail

Ethnographic analysis

A social scientists spends a considerable time observing and analyzing how people actually work

People do not have to explain or articulate their work

Social and organizational factors of importance may be observed

Ethnographic studies have shown that work is usually richer and more complex than suggested by simple system models

CS 422 Software Engineering Principles
From *Software Engineering* by I. Sommerville, 1996.
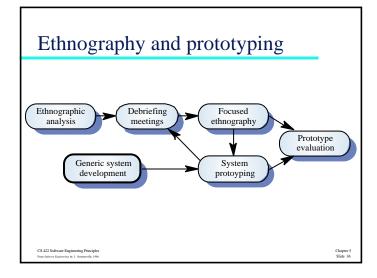
Chapter 5
Slide 34

Focused ethnography

Developed in a project studying the air traffic control process

Combines ethnography with prototyping

Prototype development results in unanswered questions which focus the ethnographic analysis

Problem with ethnography is that it studies existing practices which may have some historical basis which is no longer relevant

CS 422 Software Engineering Principles
From *Software Engineering* by I. Sommerville, 1996.

Chapter 5
Slide 35

Ethnography and prototyping

CS 422 Software Engineering Principles
From *Software Engineering* by I. Sommerville, 1996.
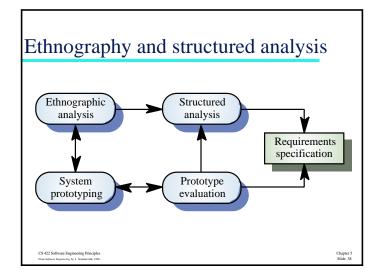
Chapter 5
Slide 36

Development of ethnography

The use of ethnography in requirements analysis needs to be developed so that it can be combined with the use of more systematic methods

As the importance of human, social and organizational factors becomes more widely recognized, these methods are likely to be developed

CS 422 Software Engineering Principles
From *Software Engineering* by I. Sommerville, 1996.

Chapter 5
Slide 37

Ethnography and structured analysis

CS 422 Software Engineering Principles
From *Software Engineering* by I. Sommerville, 1996.

Chapter 5
Slide 38

Key points

Requirements analysis requires domain understanding, requirements collection, classification, structuring, prioritization and validation

Complex systems should be analyzed from different viewpoints

Viewpoints may be based on sources and sinks of data, system models or external interaction

CS 422 Software Engineering Principles
From *Software Engineering* by I. Sommerville, 1996.

Chapter 5
Slide 39

Key points

Structured methods may be used for requirements analysis. They should include a process model, system modeling notations, rules and guidelines for system modeling and standard reports

The VORD viewpoint-oriented method relies on viewpoints which are external to the system

The boundaries between a system and its environment must be defined

Social and organizational factors have a strong influence on system requirements

CS 422 Software Engineering Principles
From *Software Engineering* by I. Sommerville, 1996.

Chapter 5
Slide 40