

---

# Requirements Engineering

## Surf 2022

# Requirements Engineering Processes

---

- Processes used to discover, analyse and validate system requirements

# Objectives

---

- To describe the principal requirements engineering activities
- To introduce techniques for requirements elicitation and analysis
- To describe requirements validation
- To discuss the role of requirements management in support of other requirements engineering processes

# Topics covered

---

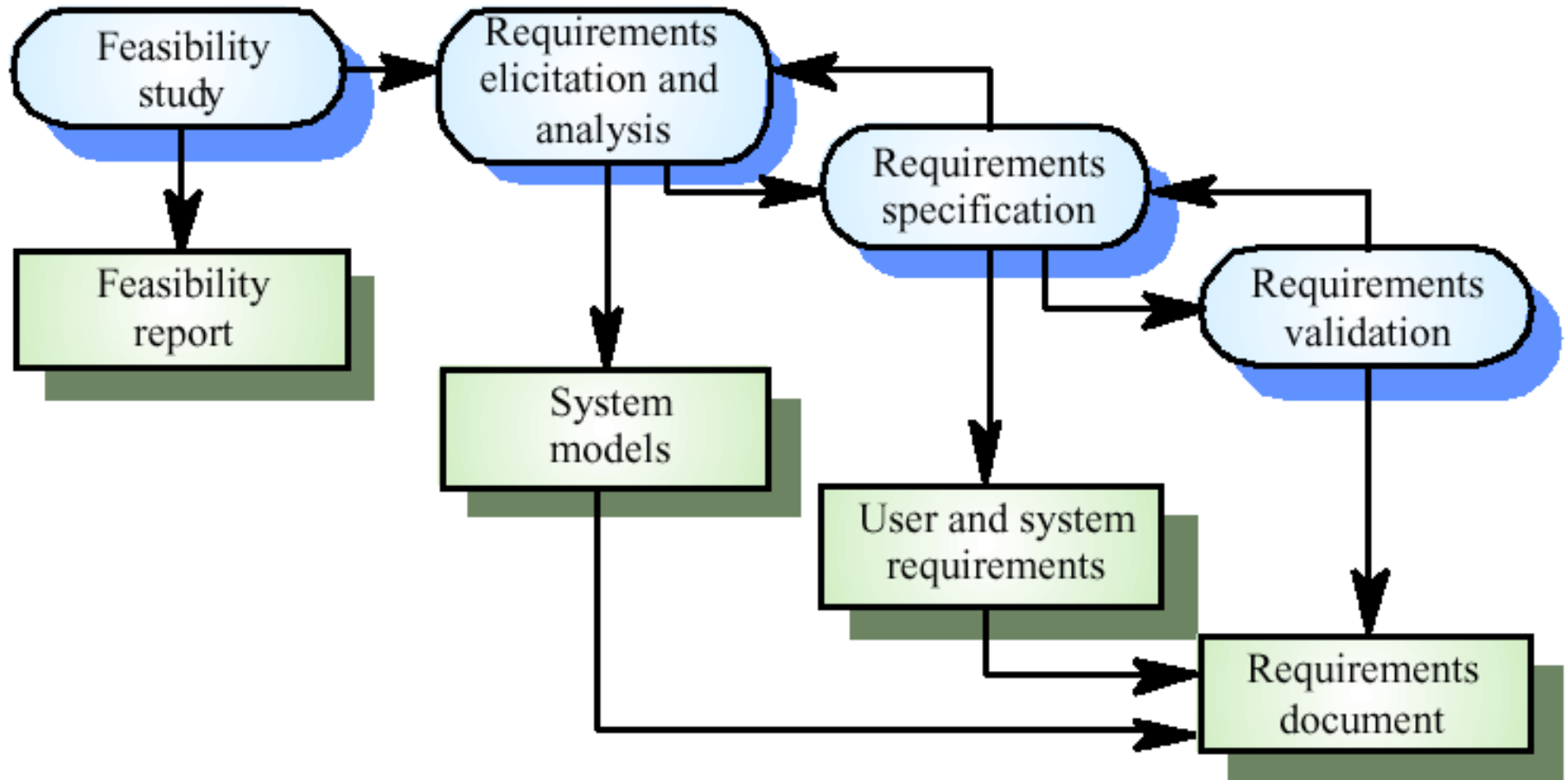
- Feasibility studies
- Requirements elicitation and analysis
- Requirements validation
- Requirements management

# Requirements engineering processes

---

- The processes used for RE vary widely depending on the application domain, the people involved and the organisation developing the requirements
- However, there are a number of generic activities common to all processes
  - Requirements elicitation
  - Requirements analysis
  - Requirements validation
  - Requirements management

# The requirements engineering process



# Feasibility studies

---

- A feasibility study decides whether or not the proposed system is worthwhile
- A short focused study that checks
  - If the system contributes to organisational objectives
  - If the system can be engineered using current technology and within budget
  - If the system can be integrated with other systems that are used

# Feasibility study implementation

---

- Based on information assessment (what is required), information collection and report writing
- Questions for people in the organisation
  - What if the system wasn't implemented?
  - What are current process problems?
  - How will the proposed system help?
  - What will be the integration problems?
  - Is new technology needed? What skills?
  - What facilities must be supported by the proposed system?



# Elicitation and 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*

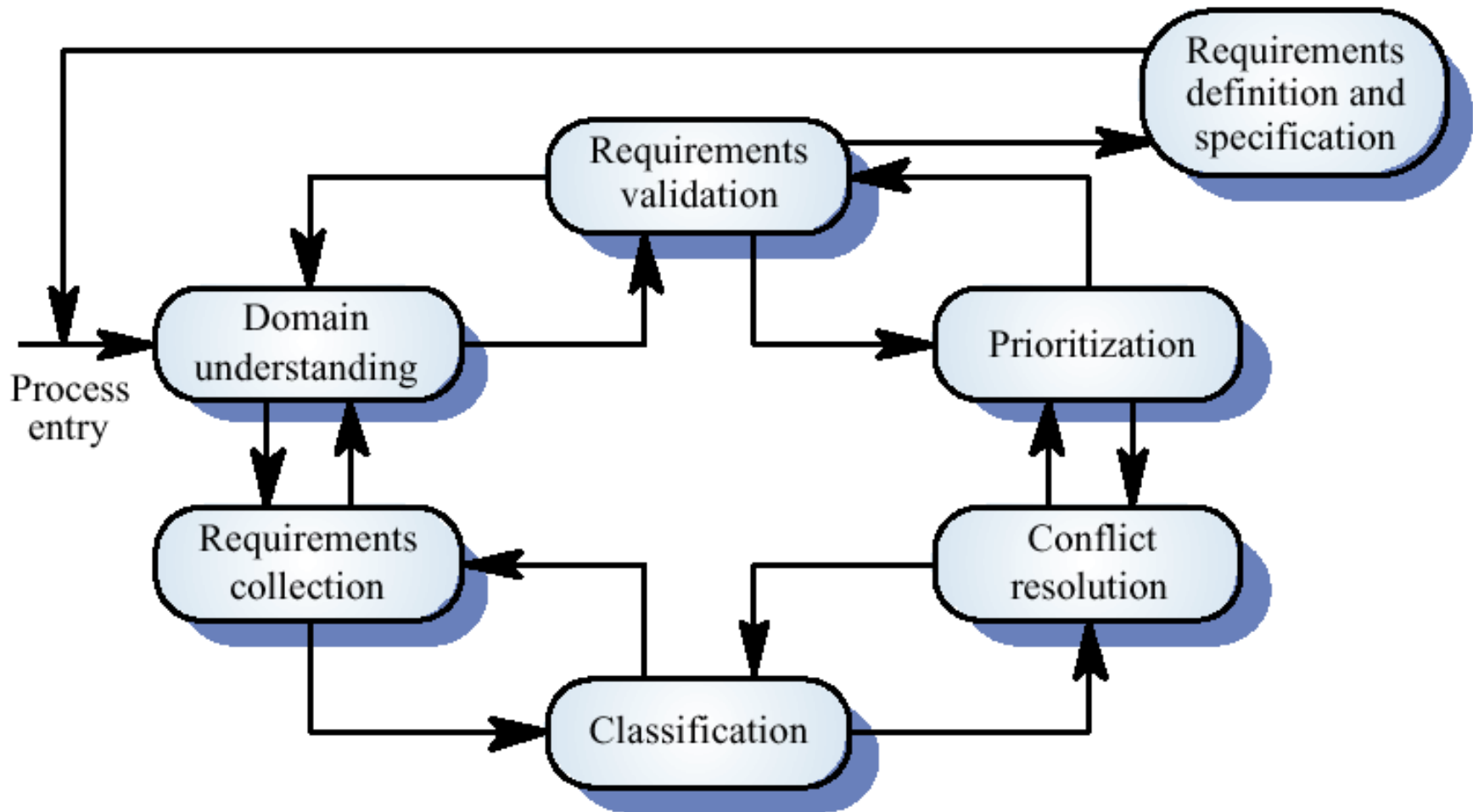
# Problems of requirements analysis

---

- Stakeholders don't know what they really want
- Stakeholders express requirements in their own terms
- Different stakeholders may have conflicting requirements
- Organisational and political factors may influence the system requirements
- The requirements change during the analysis process. New stakeholders may emerge and the business environment change

# The requirements analysis process

---



# Process activities

---

- Domain understanding
- Requirements collection
- Classification
- Conflict resolution
- Prioritisation
- Requirements checking

# 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 7

# Viewpoint-oriented elicitation

---

- 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 analyse system requirements

# Banking ATM 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



# 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 used to structure non-functional requirements

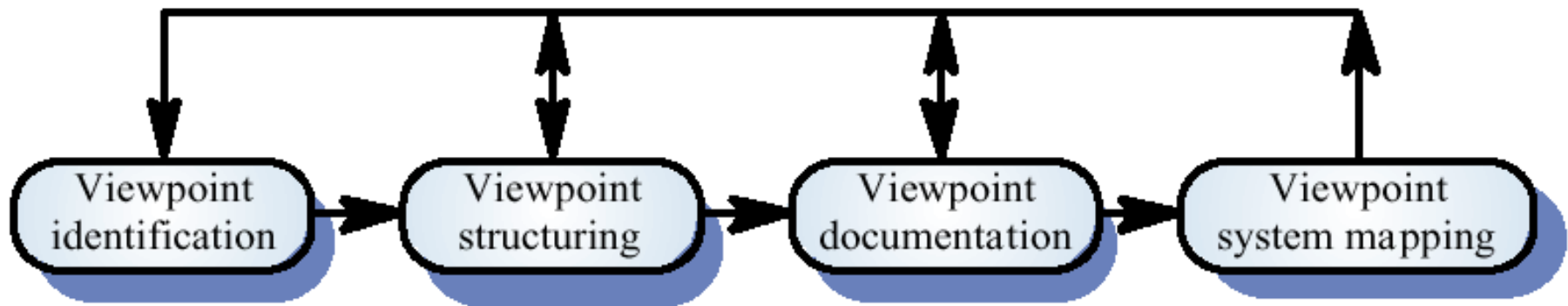
# Method-based analysis

---

- 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

# 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

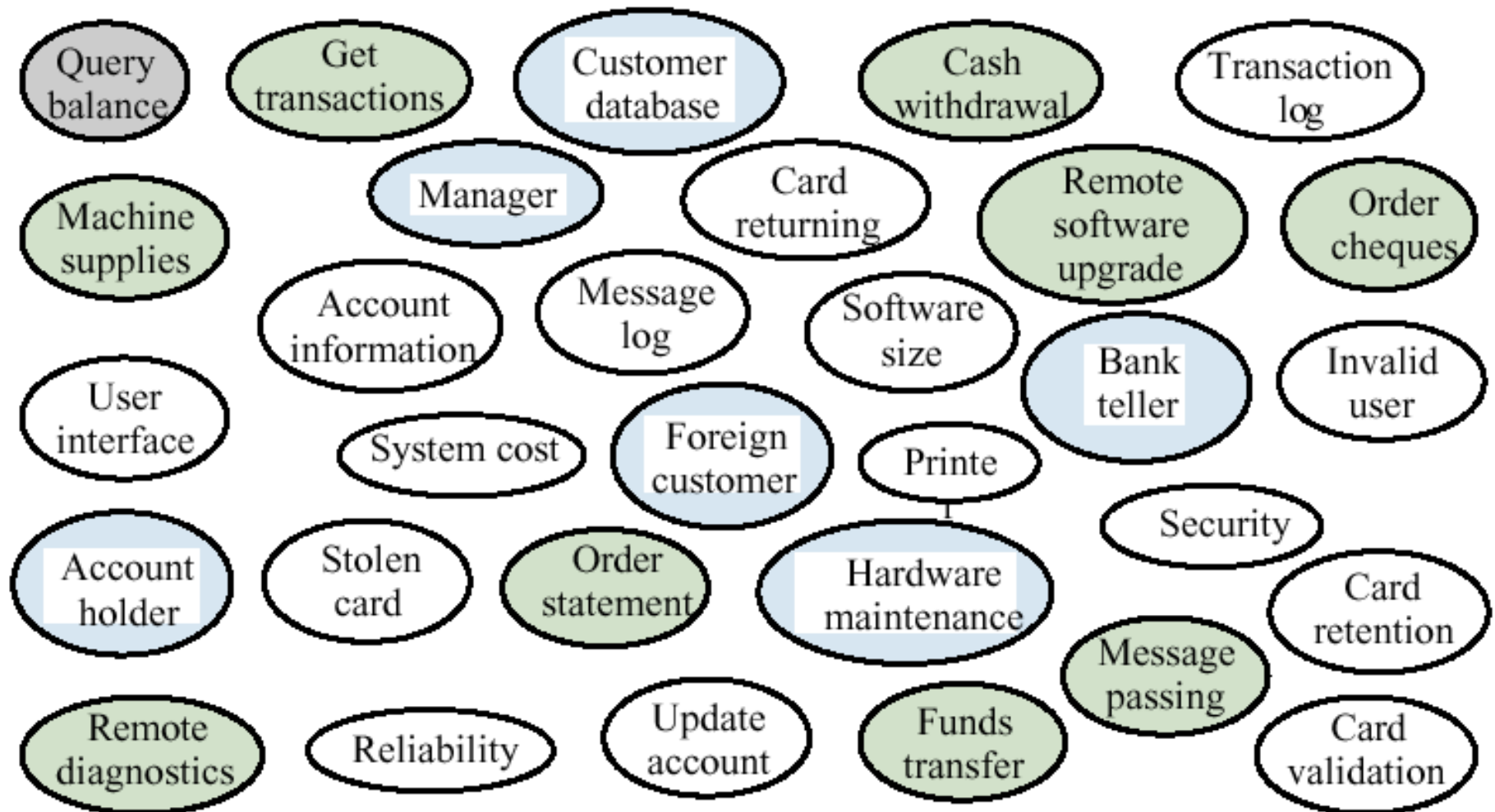
<b>Reference:</b>	The viewpoint name.
<b>Attributes:</b>	Attributes providing viewpoint information.
<b>Events:</b>	A reference to a set of event scenarios describing how the system reacts to viewpoint events.
<b>Services</b>	A reference to a set of service descriptions.
<b>Sub-VPs:</b>	The names of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

## Service template

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

# Viewpoint identification

---



# Viewpoint service information

---

## ACCOUNT HOLDER

### Service list

Withdraw cash  
Query balance  
Order cheques  
Send message  
Transaction list  
Order statement  
Transfer funds

## FOREIGN CUSTOMER

### Service list

Withdraw cash  
Query balance

## BANK TELLER

### Service list

Run diagnostics  
Add cash  
Add paper  
Send message



# Viewpoint data/control

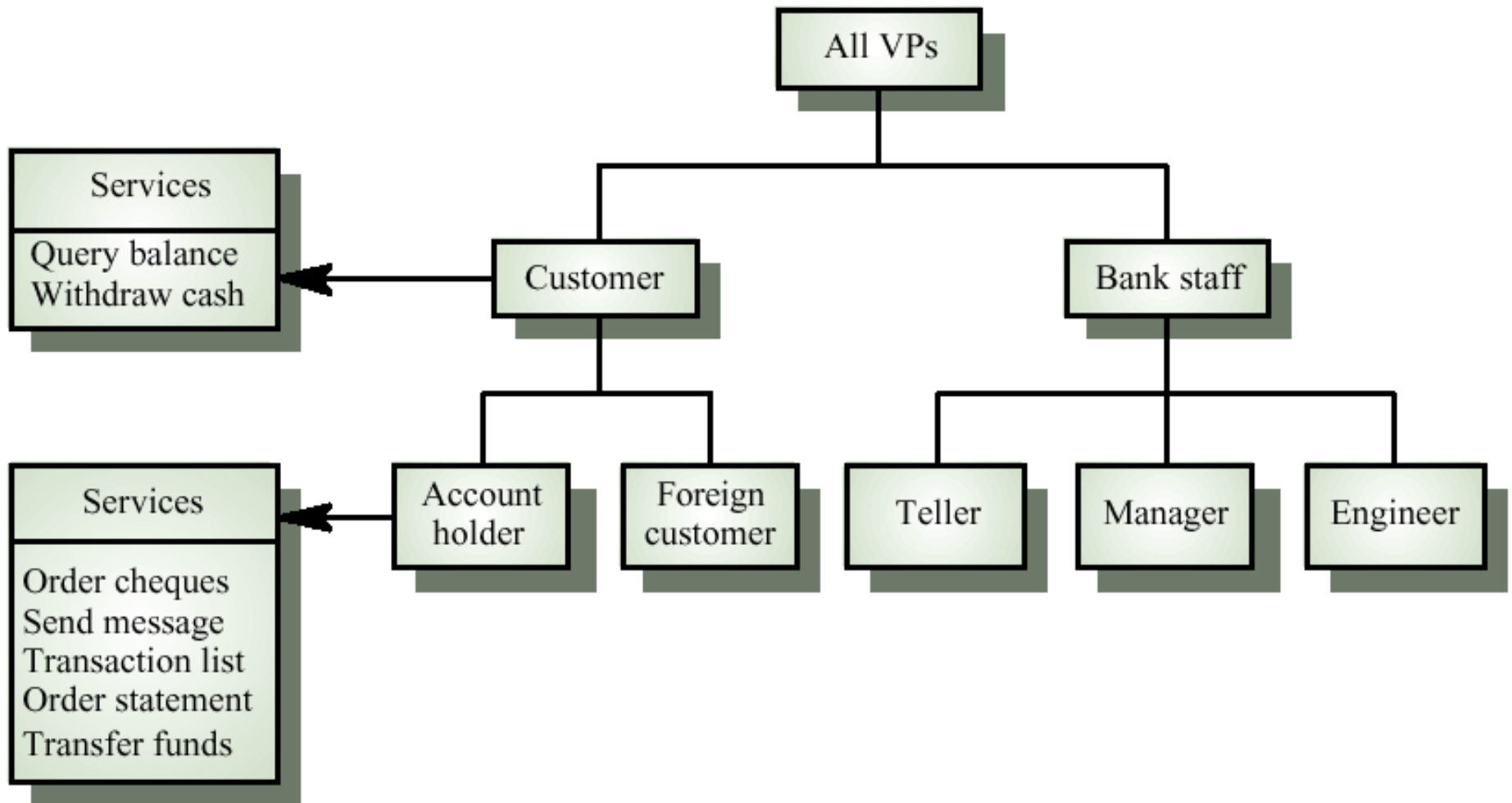
---

ACCOUNT  
HOLDER

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

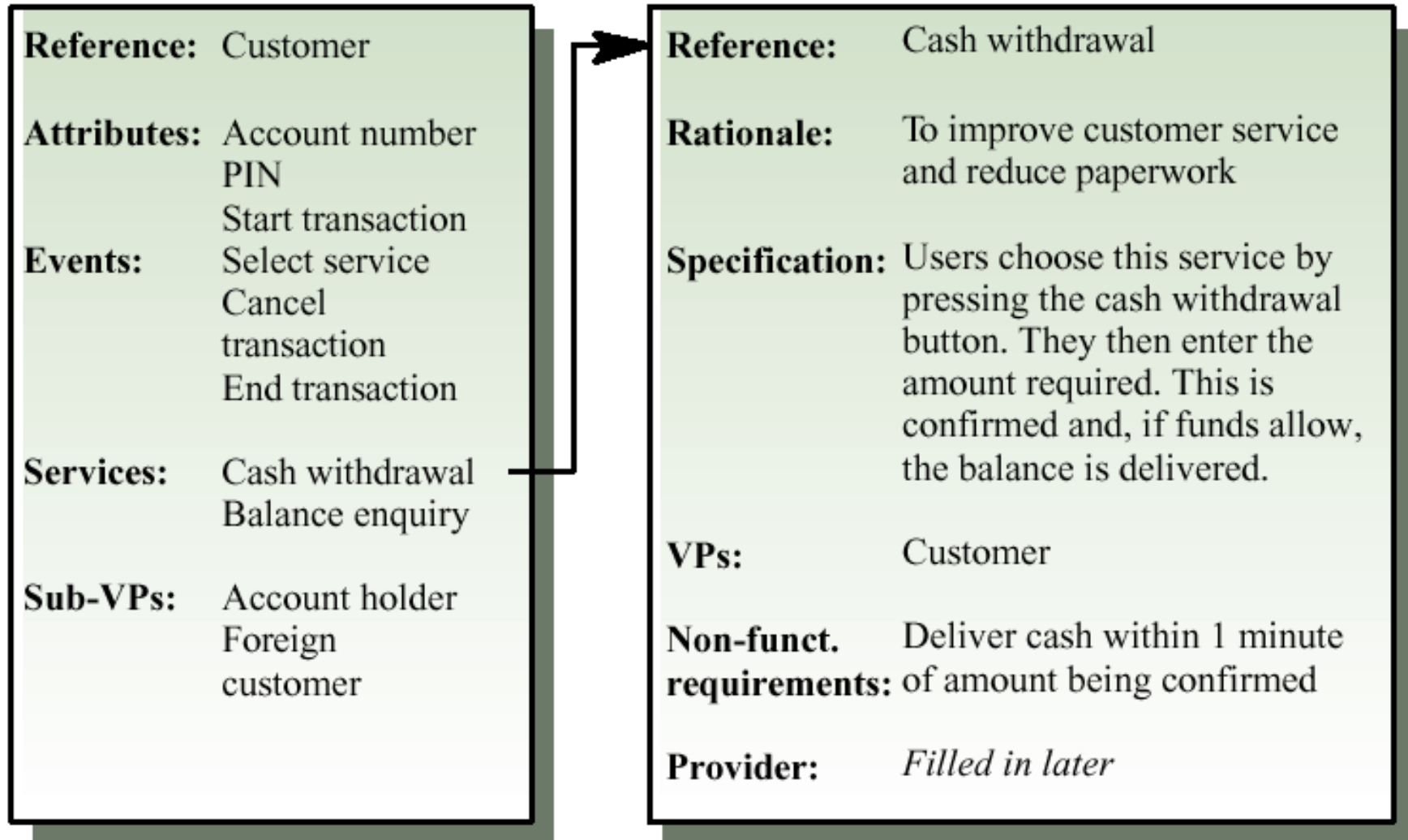
# Viewpoint hierarchy

---



# Customer/cash withdrawal templates

---



# Scenarios

---

- Scenarios are descriptions of how a system is used in practice
- They are helpful in requirements elicitation as people can relate to these more readily than abstract statement of what they require from a system
- Scenarios are particularly useful for adding detail to an outline requirements description

# Scenario descriptions

---

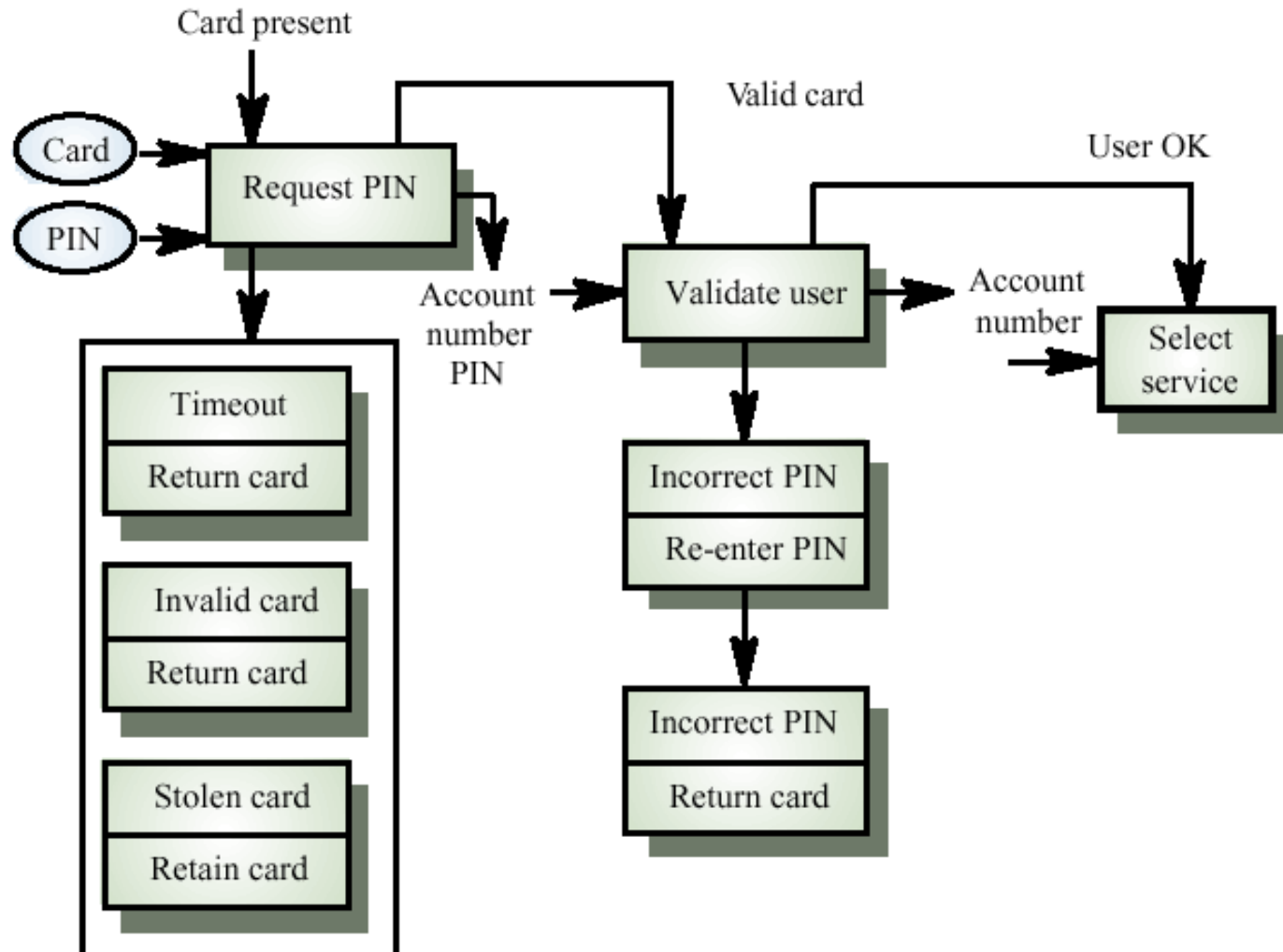
- System state at the beginning of the scenario
- Normal flow of events in the scenario
- What can go wrong and how this is handled
- Other concurrent activities
- System state on completion of the scenario

# Event scenarios

---

- Event scenarios may be used to describe how a system responds to the occurrence of some particular event such as 'start transaction'
- VORD includes a diagrammatic convention for event scenarios.
  - Data provided and delivered
  - Control information
  - Exception processing
  - The next expected event

# Event scenario - start transaction



# 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 recognised and is returned
  - Stolen card. The card has been registered as stolen and is retained by the machine

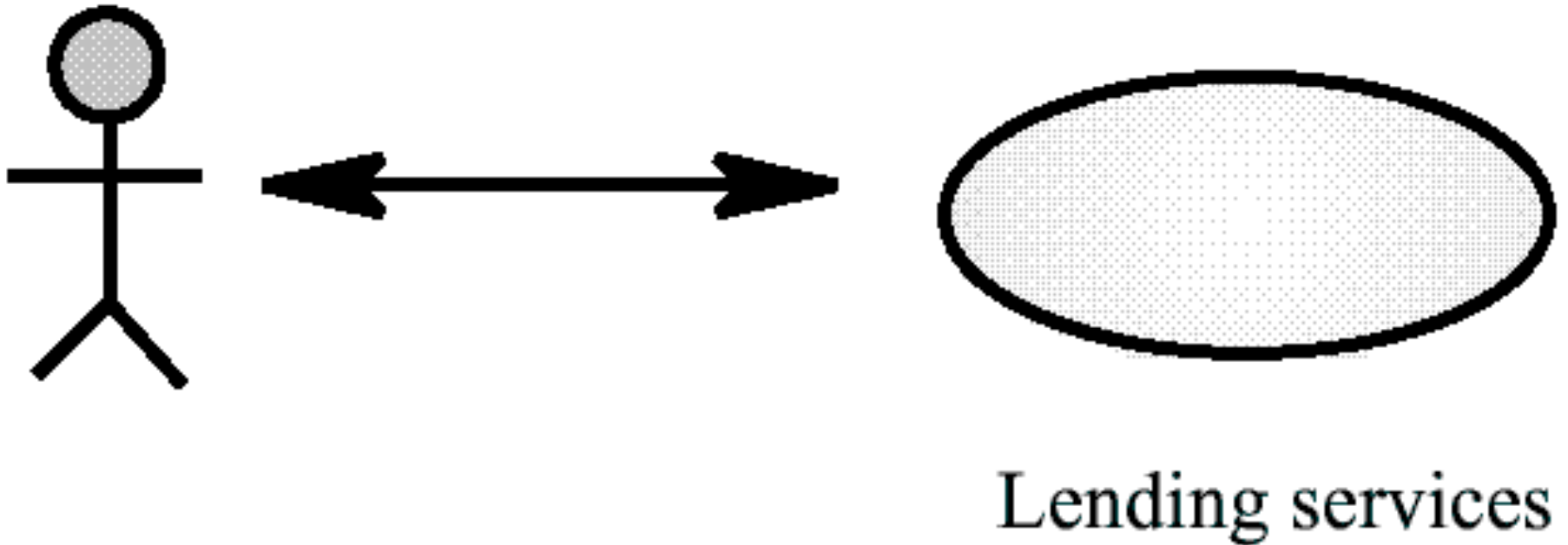
# Use cases

---

- Use-cases are a scenario based technique in the UML which identify the actors in an interaction and which describe the interaction itself
- A set of use cases should describe all possible interactions with the system
- Sequence diagrams may be used to add detail to use-cases by showing the sequence of event processing in the system

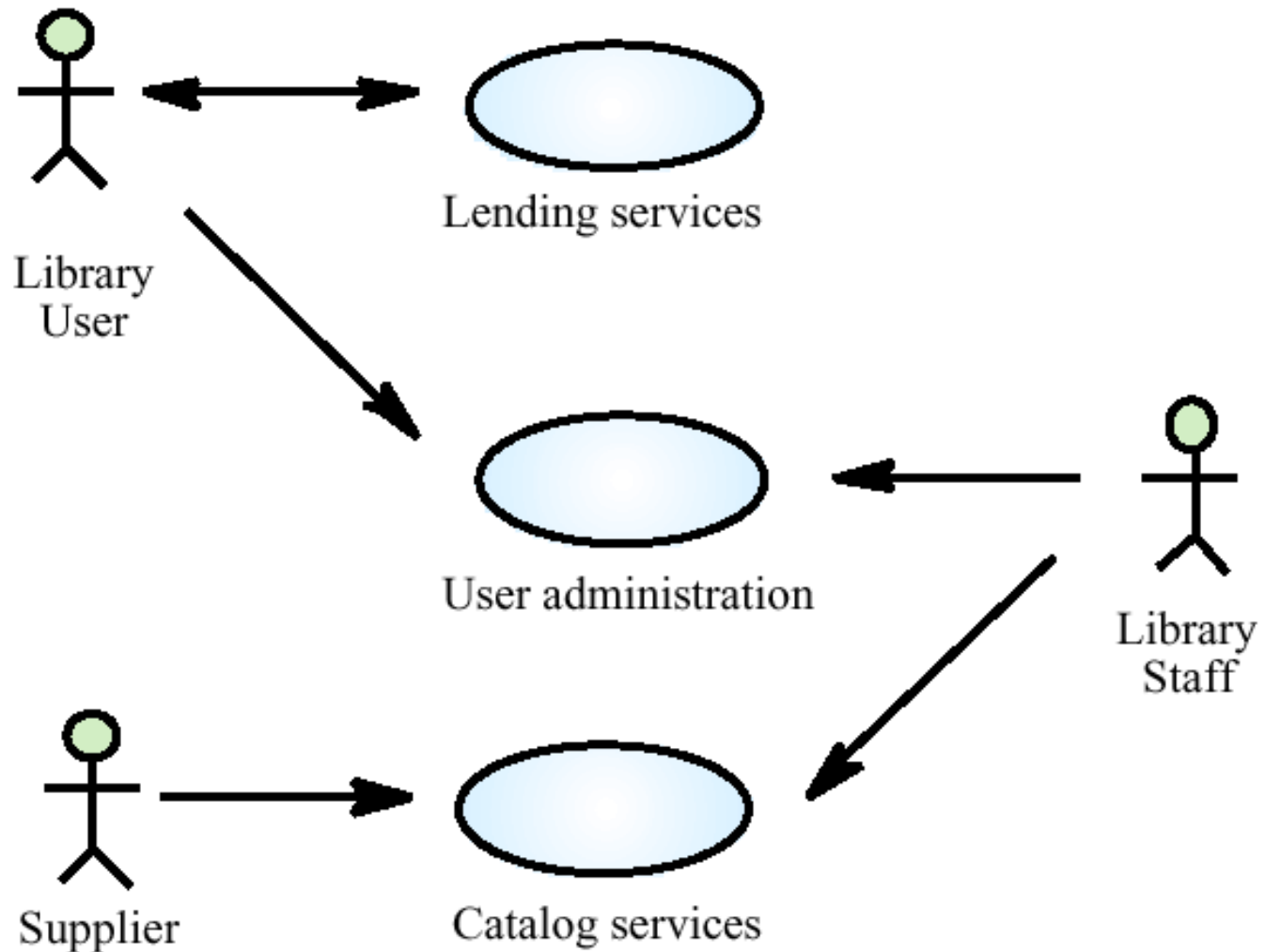
# Lending use-case

---

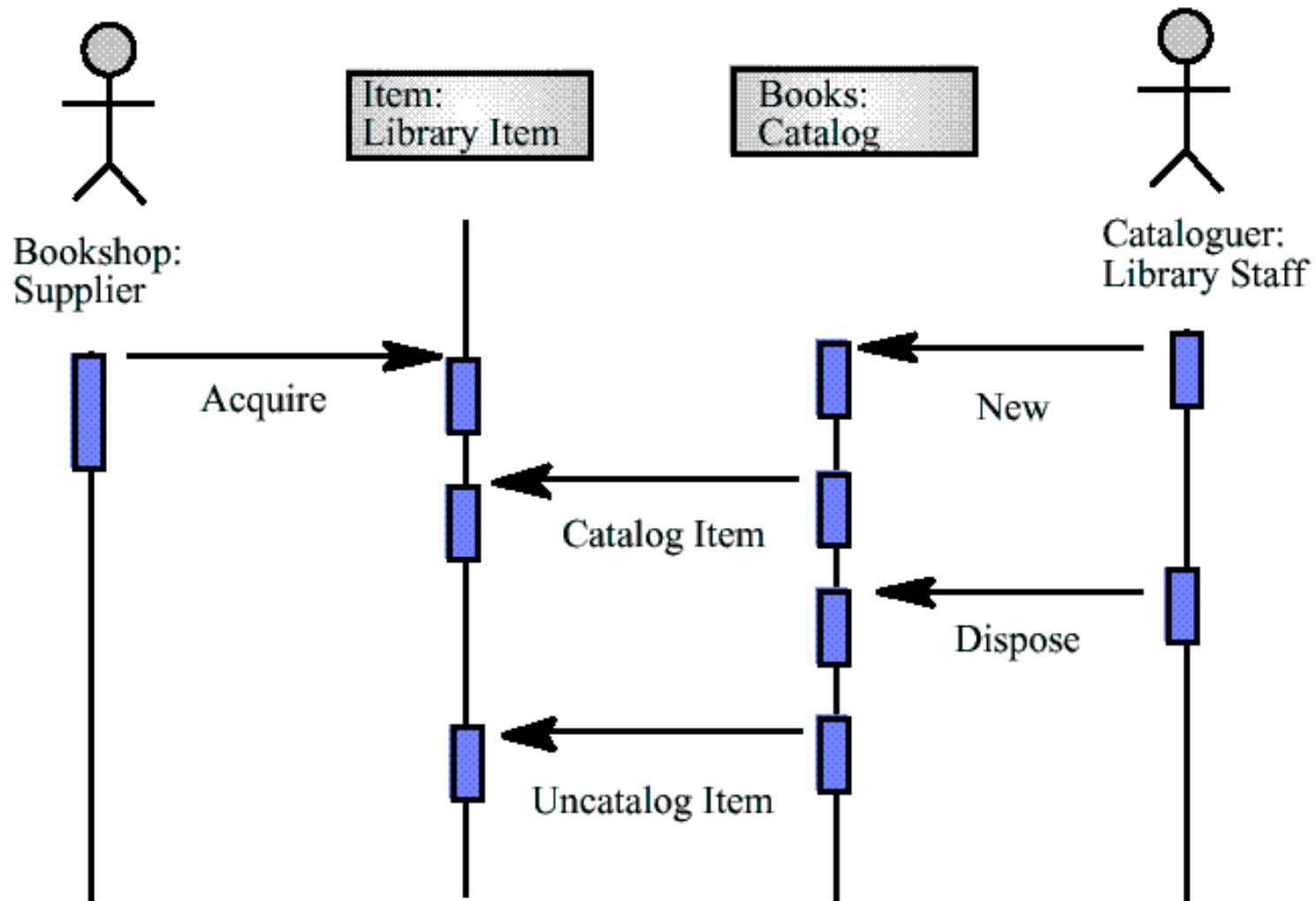


# Library use-cases

---



# Catalogue management



# Social and organisational factors

---

- Software systems are used in a social and organisational context. This can influence or even dominate the system requirements
- Social and organisational 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
  - Organisational resistance. Middle managers who will be made redundant may deliberately provide misleading or incomplete information so that the system will fail

# Ethnography

---

- A social scientists spends a considerable time observing and analysing how people actually work
- People do not have to explain or articulate their work
- Social and organisational factors of importance may be observed
- Ethnographic studies have shown that work is usually richer and more complex than suggested by simple system models



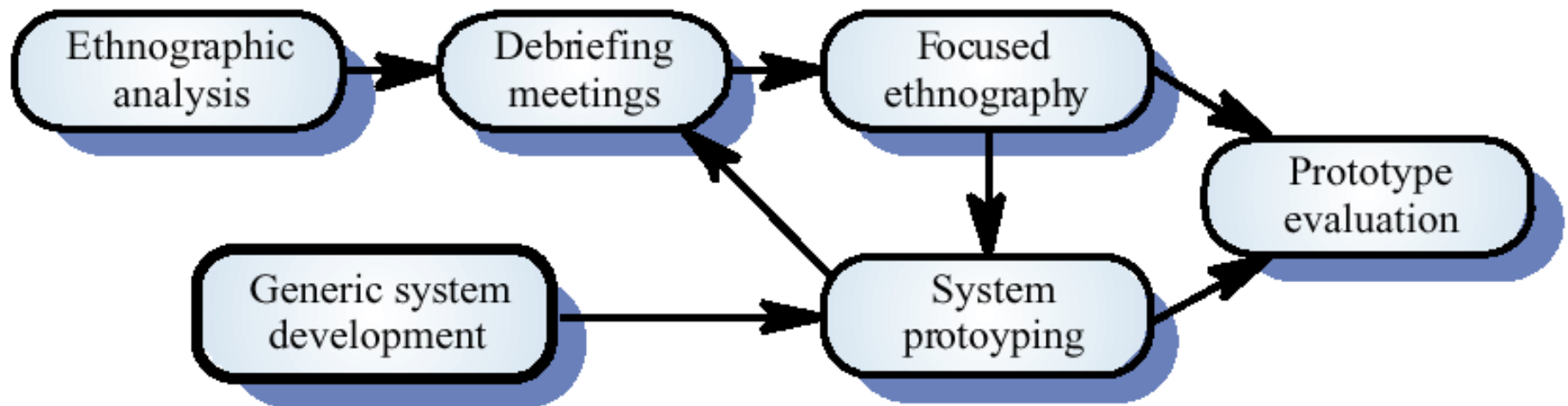
# Focused ethnography

---

- 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

# Ethnography and prototyping

---



# Scope of ethnography

---

- Requirements that are derived from the way that people actually work rather than the way in which process definitions suggest that they ought to work
- Requirements that are derived from cooperation and awareness of other people's activities

# Requirements validation

---

- Concerned with demonstrating that the requirements define the system that the customer really wants
- Requirements error costs are high so validation is very important
  - Fixing a requirements error after delivery may cost up to 100 times the cost of fixing an implementation error

# Requirements checking

---

- Validity. Does the system provide the functions which best support the customer's needs?
- Consistency. Are there any requirements conflicts?
- Completeness. Are all functions required by the customer included?
- Realism. Can the requirements be implemented given available budget and technology
- Verifiability. Can the requirements be checked?

# Requirements validation techniques

---

- Requirements reviews
  - Systematic manual analysis of the requirements
- Prototyping
  - Using an executable model of the system to check requirements. Covered in Chapter 8
- Test-case generation
  - Developing tests for requirements to check testability
- Automated consistency analysis
  - Checking the consistency of a structured requirements description

# Requirements reviews

---

- Regular reviews should be held while the requirements definition is being formulated
- Both client and contractor staff should be involved in reviews
- Reviews may be formal (with completed documents) or informal. Good communications between developers, customers and users can resolve problems at an early stage

# Review checks

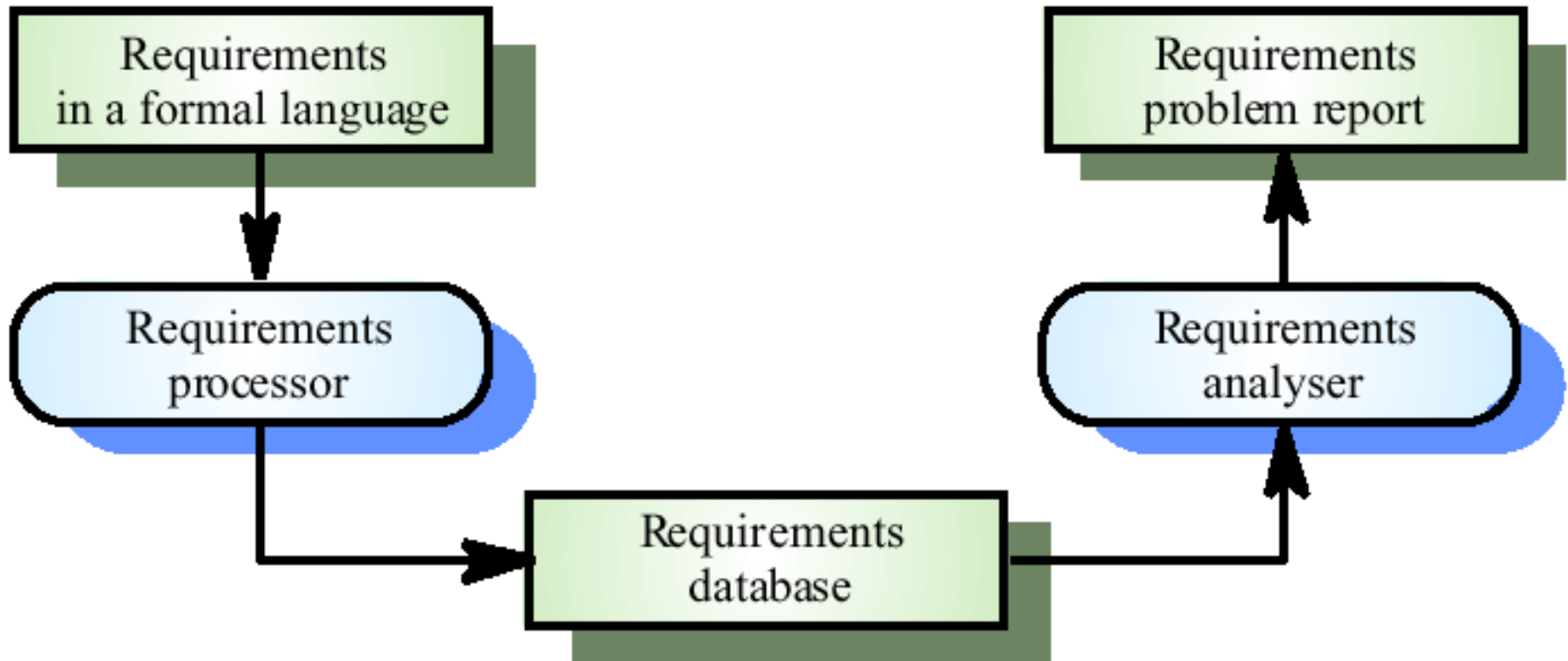
---

- Verifiability. Is the requirement realistically testable?
- Comprehensibility. Is the requirement properly understood?
- Traceability. Is the origin of the requirement clearly stated?
- Adaptability. Can the requirement be changed without a large impact on other requirements?



# Automated consistency checking

---



# Requirements management

---

- Requirements management is the process of managing changing requirements during the requirements engineering process and system development
- Requirements are inevitably incomplete and inconsistent
  - New requirements emerge during the process as business needs change and a better understanding of the system is developed
  - Different viewpoints have different requirements and these are often contradictory

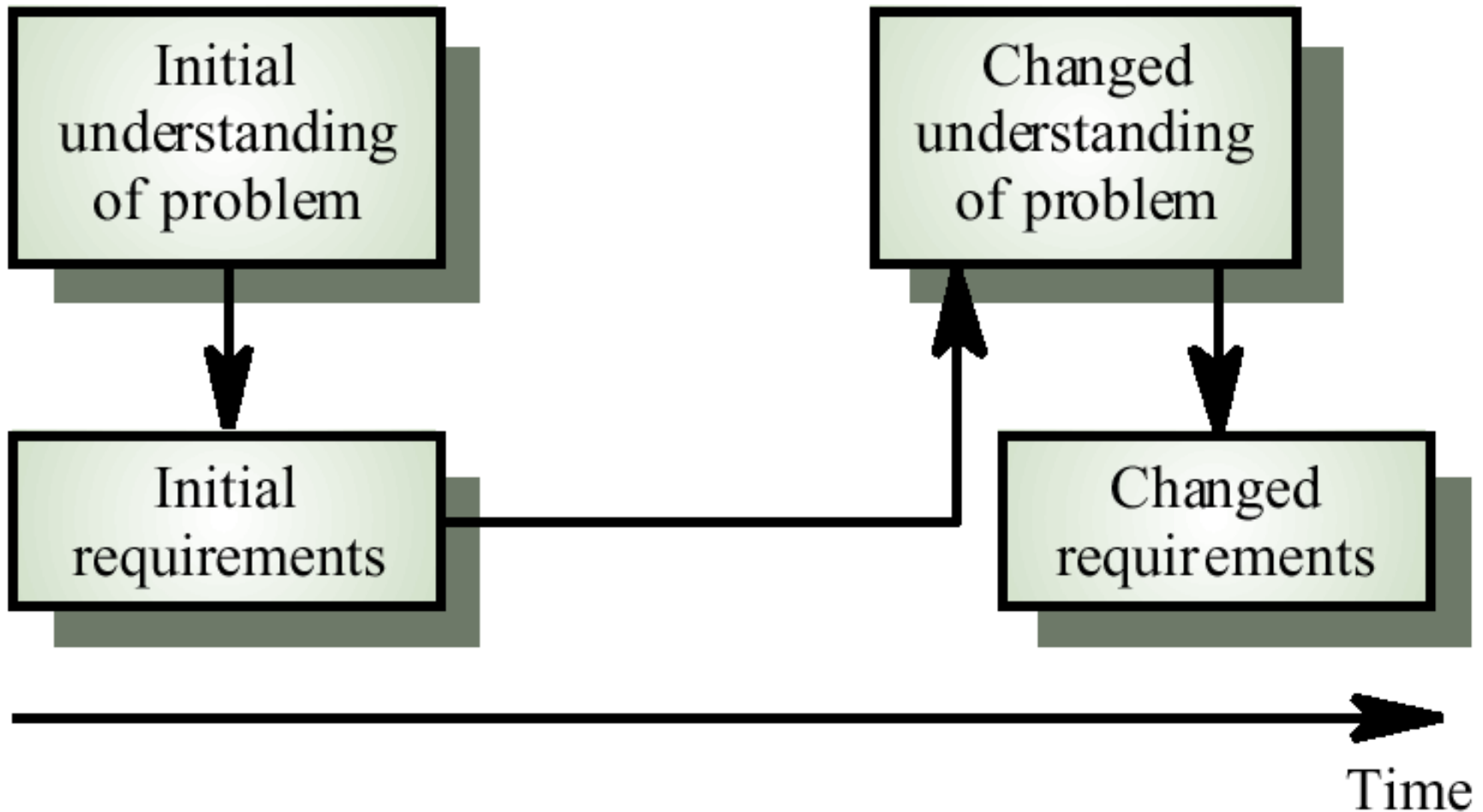
# Requirements change

---

- The priority of requirements from different viewpoints changes during the development process
- System customers may specify requirements from a business perspective that conflict with end-user requirements
- The business and technical environment of the system changes during its development

# Requirements evolution

---



# Enduring and volatile requirements

---

- Enduring requirements. Stable requirements derived from the core activity of the customer organisation. E.g. a hospital will always have doctors, nurses, etc. May be derived from domain models
- Volatile requirements. Requirements which change during development or when the system is in use. In a hospital, requirements derived from health-care policy

# Classification of requirements

---

- **Mutable requirements**
  - Requirements that change due to the system's environment
- **Emergent requirements**
  - Requirements that emerge as understanding of the system develops
- **Consequential requirements**
  - Requirements that result from the introduction of the computer system
- **Compatibility requirements**
  - Requirements that depend on other systems or organisational processes

# Requirements management planning

---

- During the requirements engineering process, you have to plan:
  - Requirements identification
    - » How requirements are individually identified
  - A change management process
    - » The process followed when analysing a requirements change
  - Traceability policies
    - » The amount of information about requirements relationships that is maintained
  - CASE tool support
    - » The tool support required to help manage requirements change

# Traceability

---

- Traceability is concerned with the relationships between requirements, their sources and the system design
- Source traceability
  - Links from requirements to stakeholders who proposed these requirements
- Requirements traceability
  - Links between dependent requirements
- Design traceability
  - Links from the requirements to the design



# A traceability matrix

---

Req. id	1.1	1.2	1.3	2.1	2.2	2.3	3.1	3.2
1.1		U	R					
1.2			U			R		U
1.3	R			R				
2.1			R		U			U
2.2								U
2.3		R		U				
3.1								R
3.2							R	

# CASE tool support

---

- Requirements storage
  - Requirements should be managed in a secure, managed data store
- Change management
  - The process of change management is a workflow process whose stages can be defined and information flow between these stages partially automated
- Traceability management
  - Automated retrieval of the links between requirements

# Requirements change management

---

- Should apply to all proposed changes to the requirements
- Principal stages
  - Problem analysis. Discuss requirements problem and propose change
  - Change analysis and costing. Assess effects of change on other requirements
  - Change implementation. Modify requirements document and other documents to reflect change

# Requirements change management

---



# Key points

---

- The requirements engineering process includes a feasibility study, requirements elicitation and analysis, requirements specification and requirements management
- Requirements analysis is iterative involving domain understanding, requirements collection, classification, structuring, prioritisation and validation
- Systems have multiple stakeholders with different requirements

# Key points

---

- Social and organisation factors influence system requirements
- Requirements validation is concerned with checks for validity, consistency, completeness, realism and verifiability
- Business changes inevitably lead to changing requirements
- Requirements management includes planning and change management