

# Requirements Engineering

---

## Chapter 7.4 Viewpoint-Oriented Requirements Definition

### Learning Objective

...to describe the VORD process primarily intended for specifying interactive systems. VORD is based on viewpoints that focus on the user issues and organizational concerns. This segment is intended to give a concrete example as a reference point for *Viewpoint-Oriented Requirements Methods*. Chapter 9 uses VORD to specify an interactive system.

### Frederick T Sheldon

Assistant Professor of Computer Science  
University of Colorado at Colorado Springs

## Why Viewpoints?

---

- ⊗ To understand the requirements for a system, we need to understand a number things:
  - The services the system provides . . .
  - The application domain of the system
  - Non-functional constraints on the system and its development process
  - The environment where the system is to be installed and organizational issues affecting the system's operation.
- ⊗ Why are these issues important to the analysis and specification of requirements?

## Why Viewpoints Work . . .

---

- ⊗ Consequently, the requirements engineering process involves the capture, analysis and resolution of many ideas, perspectives and relationships at varying levels of detail.
- ⊗ To address this problem a number of methods have evolved *based on the notion of viewpoints*.

## Potential advantages of viewpoint-based methods

---

- ⊗ They explicitly recognize of the diversity of the sources of requirements
- ⊗ They provide a mechanism for organization and structuring diverse information
- ⊗ They provide a means for requirements sources or stakeholders to identify and check their contribution to the requirements
- ⊗ They provide a framework for exposing conflicting requirements

## Viewpoints-oriented requirements definition (VORD)

---

- ⊗ Viewpoint-based method primarily intended to specify interactive systems
- ⊗ VORD is based on viewpoints that focus on user issues and organizational concerns
- ⊗ The model adopted for viewpoints is a service-oriented model
  - Viewpoints are analogous to clients in a client-server system
  - The system delivers services to viewpoints and viewpoints pass control information to the system

## VORD viewpoints

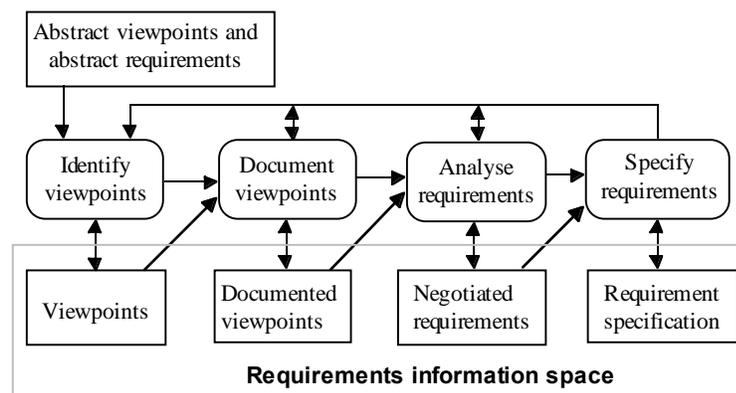
---

- ⊗ VORD viewpoints fall into two classes:
  - Direct viewpoints* These correspond directly to clients in that they receive services from the system and send control information and data to the system.
    - Are either system operators/users or other systems which are interfaced to the system being developed
  - Indirect viewpoints* Indirect viewpoints have an 'interest' in some or all the services which are delivered by the system but do not interact directly with it.
    - Indirect viewpoints may generate requirements which constrain the services delivered to direct viewpoints
    - Indirect viewpoints vary greatly

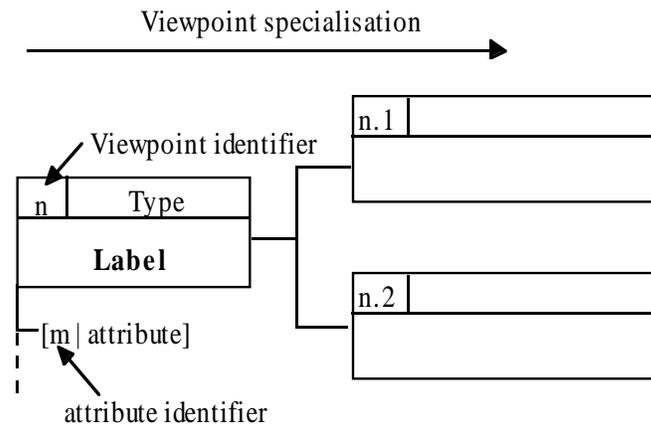
# Formulating requirements with VORD

- ⊗ VORD is based on three main iterative steps, namely:
  1. Viewpoint identification and structuring
    - Concerned with identifying and structuring relevant problem domain viewpoints
  2. Viewpoint documentation
    - Concerned with documenting the documenting the viewpoints
  3. Viewpoint requirements analysis and specification
    - Identifying errors and inconsistencies in the viewpoint documentation and resolving them

# Viewpoint process model



# Viewpoint notation



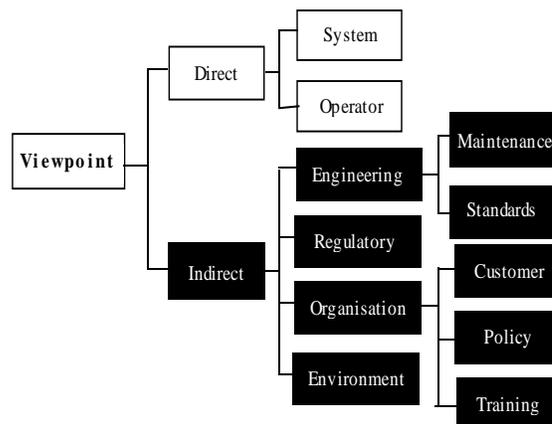
# Video on demand service example

- ⊗ Consider the requirements of a system intended to provide its users with a video-on demand service (pay-per-view).
  - The system users are to be provided with decoders to access the system and request for their favorite films to be played.
  - The system is intended to support two types of customers; standard and gold.
    - » *Standard customers* will allowed to view any 4 films no more than once in a day
    - » *Gold customers* will be allowed continuous viewing of any 5 films in a day. In addition gold customers will be allowed to 'pause', 'rewind' and 'forward' their films

## Video on demand service example (Cont.)

- ⊗ More requirements for the video-on demand service (pay-per-view).
  - The system is intended to support multiple requests for a film by up to 5 users.
  - The system is to be interfaced to a digitized video archive for film access
  - The quality manager requires that the system delivers a minimum film quality of 22 frames/sec (at any one time) with good sound quality
  - The users of system would like a service availability of no less than 98%
  - The system administrator requires that system be taken down for maintenance twice every month month

## Abstract viewpoint classes

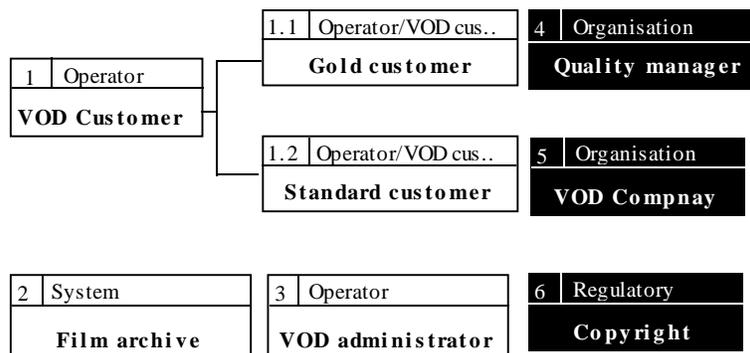


# Identifying viewpoints

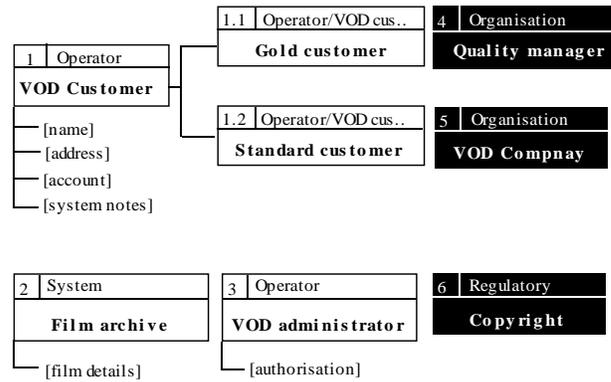
The method for identifying viewpoints has a number of stages:

1. Prune the abstract viewpoint class hierarchy, to eliminate viewpoints which are not relevant for the specific system being specified
2. Consider system stakeholders, i.e people are affected by the introduction of the system.
3. Using a model of the system architecture, identify system viewpoints, i.e. viewpoints representing other systems.
4. Identify system operators
5. For each indirect viewpoint class which is identified, consider the roles of the principal individual who might be associated with that class. For example, under the viewpoint class 'customer', we might be interested in the roles of the 'maintenance manager'

# Video on demand viewpoints



# Documenting viewpoint attributes



# Prioritizing Requirements

⊗ VORD uses a requirements priority scheme that takes into account:

- The level of importance of requirement
- The resources needed to deliver the requirement
- The risk associated with the requirement

| Weighting Factor   | High(H) | Medium(M) | Low(L) |
|--------------------|---------|-----------|--------|
| Importance         | 3       | 2         | 1      |
| Resources required | 1       | 2         | 3      |
| Risk involved      | 1       | 2         | 3      |

# Requirements and constraints

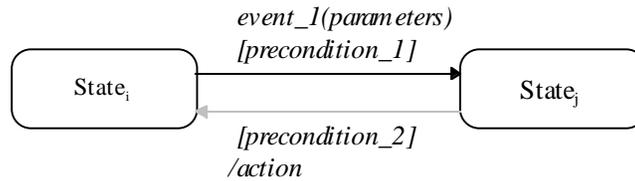
## Viewpoint service

| Identifier | Description   | Constraint                      |
|------------|---------------|---------------------------------|
| 1.1        | System access | 1.3 98% availability            |
|            |               | 3.2 Maintenance twice per month |
|            |               | 5.1 Delivery time is 8 months   |
| 1.2        | Film playback | 1.3 98% availability            |
|            |               | 3.2 Maintenance twice per month |
|            |               | 4.1 Quality should be 22 f/sec  |
|            |               | 6.1 Conform to copyright law    |
|            |               | 5.1 Delivery time is 8 months   |
| 1.2.1      | Film control  | 1.3 98% availability            |
|            |               | 3.2 Maintenance twice per month |
|            |               | 5.1 Delivery time is 8 months   |
| 3.1        | Management    | 3.2 Maintenance twice per month |
|            |               | 5.1 Delivery time is 8 months   |
|            |               | 6.1 Conform to copyright law    |

# Modeling system behavior

- ⊗ VORD uses event scenarios to model the dynamic behavior of the system.
- ⊗ An event scenario is defined as a sequence of events together with exceptions which may arise during the interchange of information between the viewpoint and the intended system
- ⊗ Viewpoint events are a reflection of control requirements as perceived by the user.
- ⊗ VORD uses an extended state transition notation to model event scenarios
- ⊗ All interactions between direct viewpoints and the system should be described using event scenarios
- ⊗ Individual scenarios combine to model the complete system behavior

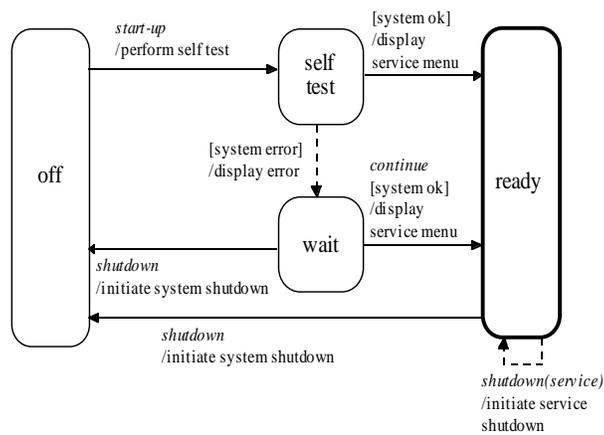
# Notation for event scenarios



## Note

- Normal sequence
- - - -> Exception sequence

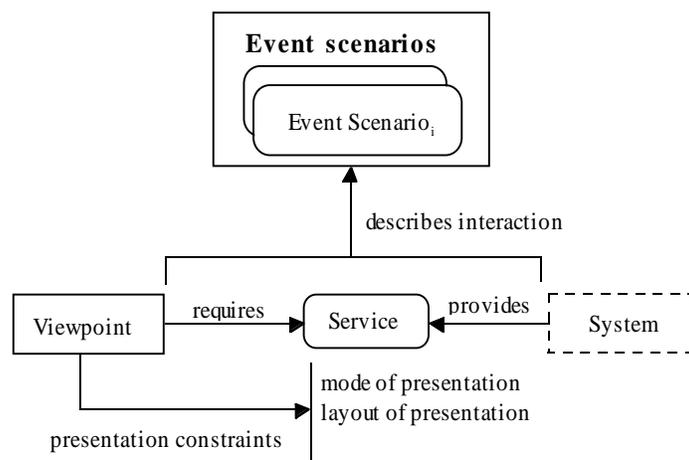
# Example Startup service



# User interface requirements

- ⊗ In VORD user interface requirements are represented as constraints on viewpoint services
- ⊗ They are associated with constraints that describe the mode and presentation of viewpoint services
- ⊗ The process of constructing the user interface is informed by viewpoint event scenarios which describe the interaction between the viewpoint and the system

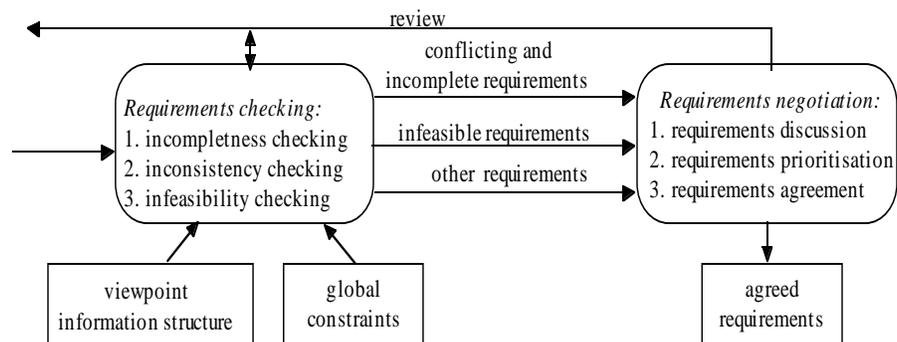
# User interface requirements & event scenarios



# Requirements analysis

- ⊗ The objective of viewpoint requirements analysis is to establish that viewpoint requirements are correct and 'complete'. There are two main stages to this:
  - *Requirements checking* is concerned with ensuring the viewpoint documentation is consistent and correct
  - *Conflict analysis and requirements negotiation* is concerned with ensuring that conflicting requirements are exposed and conflicts resolved

# VP Requirements analysis process



## Viewpoint information structure

---

- ⊗ To be described later

## Service specification

---

- ⊗ VORD supports the specification of viewpoint services in a variety of notations. This is important for two reasons:
  - Enhances communication between the engineer and potential system users
  - No one requirements notation can adequately articulate all the needs of a system. More than one specification language may be needed to represent the requirements adequately.

# Requirements specification template

---

⊗ To be described later