

# Requirements Analysis of the VoD Application Using the Tools in TRADE\*

R.J. Wieringa<sup>†</sup>, Sander Huyts<sup>† ‡</sup>

October 10, 1996

## Abstract

This report contains a specification of requirements for a video-on-demand (VoD) application developed at Belgacom, used as a trial application in the 2RARE project. The specification contains three parts: an informal specification in natural language; a semiformal specification consisting of a number of diagrams intended to illustrate the informal specification; and a formal specification that makes the requirements on the desired software system precise. The informal specification is structured in such a way that it resembles official specification documents conforming to standards such as that of IEEE or ESA. The semiformal specification uses some of the tools in from a requirements engineering toolkit called TRADE (Toolkit for Requirements And Design Engineering). The purpose of TRADE is to combine the best ideas in current structured and object-oriented analysis and design methods within a traditional systems engineering framework. In the case of the VoD system, the systems engineering framework is useful because it provides techniques for allocation and flowdown of system functions to components. TRADE consists of semiformal techniques taken from structured and object-oriented analysis as well as a formal specification language, which provides constructs that correspond to the semiformal constructs. The formal specification used in TRADE is LCM (Language for Conceptual Modeling), which is a syntactically sugared version of order-sorted dynamic logic with equality. The purpose of this report is to illustrate and validate the TRADE/LCM approach in the specification of distributed, communication-intensive systems.

---

\*Research partly supported by Esprit Project 2RARE, contractnr. 20424.

†Faculty of Mathematics and Computer Science, Free University, De Boelelaan 1081a, 1081 HV Amsterdam, roelw@cs.vu.nl, <http://www.cs.vu.nl/~roelw>. Sander Huyts now at Cambridge Technology Partners, Apollohouse, Apollolaan 15, 1077 AB Amsterdam, the Netherlands.

‡Research performed at Belgacom Research, Development and Engineering Bd. E.Jacqmain 177 - 12T83 B1030 Brussels, Belgium.

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Requirements according to ESA PS-05-0</b>	<b>4</b>
<b>3</b>	<b>TRADE Analysis</b>	<b>15</b>
3.1	Analysis of the problem environment . . . . .	15
3.2	Decomposition of the operating environment . . . . .	16
3.3	Specification of product functions . . . . .	19
3.4	Allocation and flowdown of functions . . . . .	19
3.5	Component Behavior Specification . . . . .	20
3.6	Decomposition and flowdown of functions for the video server . . . . .	20
3.7	Decomposition of the subject domain . . . . .	20
<b>4</b>	<b>Discussion and conclusions</b>	<b>35</b>
4.1	The UoD of a requirements specification . . . . .	35
4.2	The UoD of a computer-based system . . . . .	35
4.3	Simulation . . . . .	36
4.4	Formal specification . . . . .	36
4.5	Semiformal specifications . . . . .	37

# Chapter 1

## Introduction

TRADE is a Toolkit for Requirements and Design Engineering for software products. The tools in TRADE are *conceptual*: they are concepts, techniques and heuristics that can be used to specify requirements of a software product and to decompose the product into high-level components. Definition of TRADE is an ongoing project performed at the *Vrije Universiteit*, Amsterdam. The TRADE project has two goals:

- Integration of the best ideas from structured and object-oriented analysis;
- Integration of formal and semi-formal specification techniques.

The tools in TRADE are independent from the development strategy followed, e.g. linear, incremental or evolutionary development.

To support usage of the tools in TRADE, a manual is under development that explains the tools and provides heuristics for their use [35]. In addition, a freely available software tool called TCM (Toolkit for Conceptual Modeling) is under construction that can be used to draw the semiformal diagrams of TRADE [6]. The current version of TCM is available from <http://www.cs.vu.nl/~tcm>. Finally, the formal specification tool used in TRADE is LCM (Language for Conceptual Modeling), a language based upon dynamic logic [9]. This language is currently being extended with constructs for the specification of temporal and real-time constraints.

To reach the goals of the TRADE project, structured and object-oriented methods have been and still are analyzed in order to identify concepts and heuristics that can be used in TRADE [20, 30, 32, 38, 34, 33]. In addition, a semantics for LCM is being defined that agrees with the use of the semiformal techniques in TRADE [2, 27, 26, 28, 29, 37]. Finally, some preliminary case studies with TRADE/LCM have been done [31, 36].

As usual, the future is bigger than the past: Extensions planned for TCM include hypertext links with requirements stated in natural language, code generation and executable specifications. Extensions planned for LCM are the capability to specify temporal and real-time constraints on the software product. Finally, case studies are actively sought in which we can validate the ideas in TRADE, the support by TCM, and the formalization of specifications in LCM. The current report presents one such case study.

This report applies the tools in TRADE to the specification of requirements of a video-on-demand (VoD) system developed by a consortium of companies called Tectris (**T**echnical **T**rial for **I**nteractive **S**ervices) for Belgacom, Brussels.

A VoD system is a system in which a television user has a set top box (STB) with the functionality of roughly that of a video player. The STB is connected to the telephone network and the user can select a video located remotely at a video server. When a video is selected, it can be viewed on the television of the user. More details of the VoD system are introduced as we go along. The purpose of the VoD system was to become acquainted with these services and develop the technical know-how for delivering broadband multimedia services. It was therefore a trial system, not intended to be taken into commercial use. The VoD system was tested in a trial

that lasted 6 weeks, starting from February 6, 1996. Information on the Video-on-demand trial and the Tectris consortium can be found at URL <http://www.belgacom.be/rde/tectris/index.htm>.

After the trial VoD system was delivered to Belgacom by Tectris, a “shadow project” was performed at Belgacom Research in the context of the Esprit project 2RARE\*, in which formal requirements specification techniques are applied to two trial applications. One of these trial applications was the VoD system at Belgacom; the other was the Icarus system at Alcatel, Toulouse. In both trials, the requirements were redone, using the Albert [1] and Oblog [22] specification languages and the requirements management tool DOORS [25]. At Belgacom research, the following documents were used as a starting point:

- Belgacom’s request for proposal, calling companies to develop the VoD system;
- A revised request for proposal;
- A preliminary version of the contract between Belgacom and the Tectris consortium.

In addition, developers that worked on the VoD system were interviewed.

At Belgacom research, some of the tools in TRADE were used to facilitate the transition from the above documents to the formal specification languages. This report is written to summarize what we have learned from this use of TRADE. It is intended to be self-contained. We explain the concepts in TRADE as we use them. All diagrams in the report have been drawn by means of TCM. In chapter 2, we present the informal requirements on the system and in chapter 3 we use some TRADE tools to analyze these semiformaly. Chapter 4 discusses the results and contains some conclusions.

---

\*2 Real Applications of Requirements Engineering, contractnr. 20424. URL <http://www.info.fundp.ac.be/phe/2rare.html>.

# Chapter 2

## Requirements according to ESA PS-05-0

In order to obtain a software requirements specification (SRS) for the VoD system, the request for proposal, final proposal, and the preliminary and final versions of the contract were analyzed. These were written in narrative natural language. Different documents used different terminology and there was no glossary of terms and abbreviations. For example “end-user”, “customer” and “subscriber” were all used to indicate the same entity. These were organized into an informal SRS structured according to the ESA standard PS-05-0 [8].

To analyze the documents, we followed heuristics provided by Gause and Weinberg [10]. Each requirement was assigned to a *system function* and for each system function, *attributes* were specified. For each attribute, *constraints* and *preferences* were specified, where a constraint is a statement about an attribute that must be satisfied in the delivered product, while a preference is something that would be nice to have. Finally, all requirements were allocated to components in the context diagram. During the process, differences and inconsistencies in terms were resolved and a dictionary of terms was built. The result was a structured collection of requirements, stated in natural language. We used the DOORS system to store and manage the requirements [25]. We give an outline of the resulting document, with the end user requirements filled in.

### 1. Introduction

#### 1.1 Purpose of the Document

#### 1.2 Scope of the Software

#### 1.3 Definitions Acronyms, and Abbreviations

##### 1.3.1 Definitions

##### 1.3.2 Acronyms

##### 1.3.3 Abbreviations

First Screen = THE SERVICES SCREEN, presentation screen

Second Screen = MOVIE CATEGORY SCREEN, category select screen

Third Screen = MOVIE LIST SCREEN, movie select screen

Fourth Screen = PLAY MOVIE SCREEN

#### 1.4 References

##### 1.4.1 Original documents:

- VoD Contract 1.5.4
- Request for Preposal
- TECTRIS Application Specification
- Annex 1
- Annex 2

- Annex 3
- Acceptance test document

#### 1.4.2 **Standards:**

- ESA PSS-O5-0 User Requirements Document Standard [8]
- Davic 1.0 Specification Part 00 [4]

### 1.5 **Overview of the Document**

## 2. **General Description**

### 2.1 **Relation to Current Projects**

### 2.2 **Relationship to Predecessor and Successor Projects**

### 2.3 **Functions and Purpose**

### 2.4 **Environmental Considerations**

### 2.5 **Relationship to other Systems**

### 2.6 **General Constraints**

### 2.7 **Model Description**

## 3. **Specific Requirements: End User View**

### 3.1 **Power on STB.**

Creating connection between STB and Video Server

#### 3.1.1 **Interface Requirements.**

The STB sends a Boot-Request message on the V.24 channel. This message is relayed by the ATM-LAN router as a broadcast message on the Ethernet. The dedicated BOOTP server replies to the STB with a Boot\_Reply message indicating the boot parameters to the STB. The STB is then able to request opening the session with the server. This session is granted by the assigned Server Element. As first request in the session, the STB asks for the VoD application to be downloaded. The downloaded application is then executed.

#### 3.1.2 **Operational Requirements.**

The language of the screens is English. At power-up/reset of the Set-Top Box, the user is greeted with a welcome screen, while the Set Top Box waits for the network initialisation phases to be completed (e.g. APON ranging, ...). If there is a communication problem with the server the following message is displayed: SORRY...THE SYSTEM IS DOWN PLEASE CONTACT THE HELPDESK. If the maximum number of simultaneously active users is exceeded, the VoD application will not start and a message explaining that the system is currently full will be displayed on the screen: SORRY...SERVICE NOT AVAILABLE, MAXIMUM NUMBER OF USERS REACHED. PLEASE TRY AGAIN LATER ON. After a starting procedure, the navigation and selection of programmes is operated under local control of the STB. The viewer is presented with a serie of four screens that provide navigational assistance leading to the selection and playback of the desired movie:

THE SERVICES SCREEN  
 MOVIE CATEGORY SCREEN  
 MOVIE LIST SCREEN  
 PLAY MOVIE SCREEN

### 3.2 **Select service.**

Allows choice between several multimedia services by the end user.

#### 3.2.1 **Performance Requirements.**

The delay between the activation of the "select" button on the remote control and the display of the second screen will be less than 1 second.

#### 3.2.2 **Interface Requirements.**

Interface Remote Control → STB:

- arrow up
- arrow down
- arrow left
- arrow right
- select

#### 3.2.3 **Operational Requirements.**

Four types of services will be identified. However, only category "VIDEO ON DEMAND", will be active. The types of services will be displayed as buttons on the screen. The user can select one service by pushing the arrows of the remote control (up, down, left, right). The selected button will be highlighted. The user confirms the selection by pressing the "select" button on the remote control. Once selected, the second screen will be displayed and allows the viewer to select the type of movie desired.

### 3.3 **Select category.**

Allows choice between several movie categories by the end user.

#### 3.3.1 **Performance Requirements.**

The delay between the activation of the "select" button on the remote control and the display of the third screen will be less than 1 second.

#### 3.3.2 **Interface Requirements.**

Interface Remote Control → STB:

- arrow up
- arrow down
- arrow left
- arrow right
- select
- shift
- blue

#### 3.3.3 **Operational Requirements.**

Display MOVIE CATEGORY SCREEN. The second screen presents 4 categories :

- Movies
- Information
- Entertainment
- Various

The movie categories will be displayed as buttons on the screen. On pressing the blue button one is reset to the first screen. The user can select one movie category by pushing the arrows of the remote control (up, down, left, right). The selected button will be highlighted. The user confirms the selection by pressing the "select" button on the remote control. The "shift" command highlights the different category boxes and but the state remains

the same. Upon selection of a movie category, the viewer is then presented with a list of available movies in that category displayed in the next screen.

### 3.4 **Select movie.**

Allows choice between several movies by the end user.

#### 3.4.1 **Performance Requirements.**

Timeout of the selection menu after 5 min. and return to the category menu.

#### 3.4.2 **Interface Requirements.**

Interface Remote Control → STB:

- arrow up
- arrow down
- select
- shift
- blue

#### 3.4.3 **Operational Requirements.**

Display MOVIE LIST SCREEN. All available movies that fall into the category selected on the MOVIE CATEGORY SCREEN are presented as a list. The total width of this screen is 40 characters.

- 30 will be used for the title of the program.
- Behind the title, there will be information about the language of the program (3 characters). There are three possible values: Dutch, abbreviated by by N, French abbreviated by with F, and both, abbreviated by F/N.
- Next, there is a one-letter indication of the type of movie: A (action), C (comedy), M (Mystery), D (drama), space (other).

The currently selected movie is highlighted. The user can scroll by means of the up and down arrow on the remote control, and confirms the selection by means of the “select” button on the remote control. On pressing the blue button one is reset to the second screen. The “shift” command highlights the different titles and but the state remains the same. Once the desired movie has been selected the movie begins to play in full screen mode on the PLAY MOVIE SCREEN.

### 3.5 **Start movie.**

Start movie with VCR interaction in fourth screen mode.

#### 3.5.1 **Performance Requirements.**

Reponse by displaying MPEG frame (playing movie) < 10 sec. The maximum delay between the selection and the display of the first frame of the movie is 3 seconds, if the new stream can be accomodated by an existing disk cache, and is 10 seconds if a new on-disk replica of the movie has to be created first.

#### 3.5.2 **Interface Requirements STB ↔ SERVER.** When a programme is selected, a Program Request is sent to the server, and the programme is played on the 2 Mbps channel.

#### 3.5.3 **Operational Requirements.**

Display PLAY MOVIE SCREEN. The screen for the beginning of the movie is the first frame of the MPEG stream. If the screen for the beginning is of very poor visual quality (due to compression artefacts for example) during the A1 tests, the frame should be overlaid by a black screen from the application. After the movie starts playing, the viewer may activate motion control commands by pushing VCR buttons on the Remote Control device.

### 3.6 **Execute command.**

Execute actions related to VCR button, pushed on the remote control by the user.

#### 3.6.1 **Interface Requirements.**

##### 3.6.1.1 Remote Control → STB:

- pause / play
- stop
- FF
- Rew
- JumpB
- JumpF
- Red
- Green
- Yellow
- Blue

Only authorized commands are taken into consideration by the STB.

##### 3.6.1.2 RC → STB ↔ Server:

Several Motion Control requests may be initiated by the customer on his Remote Control, causing appropriate responses by the server. Only authorized commands are sent to the server.

#### 3.6.2 **Operational Requirements.**

Unauthorized commands are responded to by a “beep”. All subsequent motion control commands will execute continuous from the current position.

### 3.7 **End movie**

Automatic end of a movie when last MPEG packet is received.

#### 3.7.1 **Performance Requirements.**

Timeout text box after 5 sec and return to category screen.

#### 3.7.2 **Interface Requirements.**

The 2 Mbps downstream channel is deactivated, and the session is closed. Possible functions:

- \* Rew,
- \* JumpB,
- \* Stop.

#### 3.7.3 **Operational Requirements.**

Display after Last frame a Text box : “End of program”. The screen for the end of the movie is the last frame of the MPEG stream. If the screen for the end is of very poor visual quality (due to compression artefacts for example) during the A1 tests, the frame should be overlaid by a black screen from the application

### 3.8 **Pause/Play (P/P).**

Toggles between Pause and Play.

#### 3.8.1 **Play.**

Movie playback.

### 3.8.1.1 Performance Requirements.

The " Play " should produce a delay to be limited to a few frames, meaning less than 1 second.

### 3.8.1.2 Interface Requirements.

Possible Remote Control functions:

- Play/Pause,
- JumpF,
- JumpB,
- FF,
- Rew,
- Stop.

### 3.8.1.3 Operational Requirements.

Display Green "Play" logo in the corner.

## 3.8.2 Pause.

Pause movie playback.

### 3.8.2.1 Performance Requirements.

Time out Pause state after 20 min and return to category screen.

### 3.8.2.2 Interface Requirements.

The command to stop the transmission of the MPEG stream is sent to the server. Possible Remote Control functions:

- Play/Pause,
- JumpF,
- JumpB,
- FF,
- Rew,
- Stop.

### 3.8.2.3 Operational Requirements.

When the " Pause" button is pressed, the current frame stored in the STB is displayed and frozen. Display Green "Pause" logo in the corner.

## 3.9 Forward.

### 3.9.1 Forward (FF).

The movie is speeded up.

#### 3.9.1.1 Interface Requirements

The FAST FORWARD speed is established when the actual title is processed through the New Content Gateway. Possible Remote Control functions:

- Play/Pause,
- JumpF,
- JumpB,
- Rew.

#### 3.9.1.2 Operational Requirements.

Display Green "FF" logo in the corner. In case there are no fast forward file available, display FUNCTIONALITY NOT AVAILABLE. This message is accompanied by a beep.

### 3.9.2 Forward at end movie.

Forward (still) Activated at end of movie.

#### 3.9.2.1 Performance Requirements.

Time out Text Box after 2 min and return to category screen.

#### 3.9.2.2 Interface Requirements.

Possible Remote Control functions:

- JumpB,
- Rew,
- Stop.

#### 3.9.2.3 Operational Requirements.

At Last frame display Text box : "End of program"

### 3.10 Rewind.

#### 3.10.1 Rewind (Rew).

The movie is speeded up backwards.

##### 3.10.1.1 Interface Requirements.

The REWIND speed is established when the actual title is processed through the New Content Gateway. Possible Remote Control functions:

- Play/Pause,
- JumpF,
- JumpB,
- FF.

##### 3.10.1.2 Operational Requirements.

Display Green "RW" logo in the corner. In case there are no rewind file available, display FUNCTIONALITY NOT AVAILABLE. This message is accompanied by a beep.

#### 3.10.2 Rewind at begin of movie.

Rewind (still) activated at begin of movie.

##### 3.10.2.1 Performance Requirements.

Time out Text Box after 20 min and return to category screen.

##### 3.10.2.2 Interface Requirements.

Possible Remote Control functions:

- JumpF,
- FF,
- Stop,
- Play/Pause.

##### 3.10.2.3 Operational Requirements.

At First frame display Text box : "Begin of program".

### 3.11 Jump forward.

#### 3.11.1 JumpF.

Jumps forward in a movie.

##### 3.11.1.1 Performance Requirements.

The initial speed is SLOW (about 20 x normal speed).

##### 3.11.1.2 Interface Requirements.

Possible Remote Control functions:

- Play/Pause,
- JumpB,
- FF,
- Rew,
- Yellow.

#### 3.11.1.3 **Operational Requirements.**

The movie playback is suspended. Display Black screen. Display On-going moving bar. Selecting "Play" , restarts the movie at the position indicated by the moving bar. Selecting "Pause" , stops the moving bar at its current position and clears the screen. Display Green "JumpF" logo and speed info in the corner

#### 3.11.2 **JumpF double speed.**

Change JumpF speed from SLOW to FAST when in JumpP state the Yellow Remote Control button is pressed.

##### 3.11.2.1 **Performance Requirements.**

Speed is FAST (about 40 x normal speed).

##### 3.11.2.2 **Interface Requirements.**

As JumpF.

##### 3.11.2.3 **Operational Requirements.**

When the "Yellow" button is pressed, the speed indication in the corner of the screen changes to FAST.

#### 3.11.3 **JumpF at end of movie.**

JumpF (still) activated at end of movie.

##### 3.11.3.1 **Performance Requirements.**

Time out Text Box after 2 min and return to category screen.

##### 3.11.3.2 **Interface Requirements.**

Possible Remote Control functions:

- JumpB,
- Rew,
- Stop.

##### 3.11.3.3 **Operational Requirements.**

Display Black screen. Display Full moving bar. Display Text box : "End of program"

### 3.12 **Jump back.**

#### 3.12.1 **JumpB.**

Jumps Back in a movie.

##### 3.12.1.1 **Performance Requirements.**

The initial speed is SLOW (about 20 x normal speed).

##### 3.12.1.2 **Interface Requirements.**

Possible Remote Control functions:

- Play/Pause,
- JumpF,
- FF,
- Rew,
- Yellow.

### 3.12.1.3 **Operational Requirements.**

The movie playback is suspended. Display Black screen. Display Back-going moving bar. Selecting "Play", restarts the movie at the position indicated by the moving bar. Selecting "Pause", stops the moving bar at its current position and clears the screen. Display Green "JumpB" logo and speed info in the corner.

### 3.12.2 **JumpB double speed.**

Change JumpB speed from SLOW to FAST when in JumpB state the Yellow Remote Control button is pressed.

#### 3.12.2.1 **Performance Requirements.**

Speed is FAST (about 40 x normal speed).

#### 3.12.2.2 **Interface Requirements.**

As JumpB.

#### 3.12.2.3 **Operational Requirements.**

When the "Yellow" button is pressed, the speed indication in the corner of the screen changes to FAST.

### 3.12.3 **JumpB at begin of movie.**

JumpB (still) activated at begin of movie.

#### 3.12.3.1 **Performance Requirements.**

Time out Text Box after 20 min and return to category screen.

#### 3.12.3.2 **Interface Requirements.**

Possible Remote Control functions:

- JumpF,
- FF,
- Stop,
- Play/Pause.

#### 3.12.3.3 **Operational Requirements.**

Display Black screen. Display Empty moving bar. Display Text box : "Begin of program".

### 3.13 **Stop confirmation.**

Stop movie playback with confirmation.

#### 3.13.1 **Performance Requirements.**

Time out Text Box after 5 sec and return to previous state.

#### 3.13.2 **Interface Requirements.**

##### 3.13.2.1 STB → Server.

A STOP request is generated after confirmation. This request is optional when the programme ends. The server may also close the session without receiving a Session-Release, eg. when Power-off the STB, after a time-out of 15 minutes. The 2 Mbps downstream channel is the deactivated, and the session is closed.

##### 3.13.2.2 RC → STB Possible Remote Control functions:

- Green
- Red

#### 3.13.3 **Operational requirements.**

Display Green "Stop" logo in the corner. Display Text box : "Return to menu ?" Green

button is confirmation stopping playback with returning to menu. Red button is proceed playback. There is no stop command possible in the Fast forward, rewind, jumpf and jumpb states.

### 3.14 **display VCR logo.**

Display logo on screen when authorized VCR button is pushed.

#### 3.14.1 **Operational Requirements.**

The little sign in the upper right corner, appears on the screen immediately after the user has pushed the button and remains on the screen until the action is completely finished. In the play mode the sign appears on the screen when pushing play, but disappears again after the first MPEG frame actually appears on the screen. (max. 10 sec). The sign is visible on the screen when the user pushes fast forward (rewind, jumpb or jumpf, pause), and stays there until the user leaves the fast forward (rewind, jumpb or jumpf, pause) mode. When arriving in the fast forward (rewind, jumpb or jumpf) end state, the sign disappears again. For the jump screens, there is also a word displayed under the sign. This text will either be "fast" or "slow". When pushing the yellow button during jumpf or jumpb the text on the screen changes from "slow" to "fast".

### 3.15 **Display text box.**

Display a text box on the screen.

#### 3.15.1 **Operational Requirements.**

Text box : white background and black text.

### 3.16 **Display moving bar.**

Display a moving bar which shows the current position of the movie.

#### 3.16.1 **Operational Requirements.**

Moving bar : white background and red foreground (bar).

### 3.17 **View data concerning the service.**

TBD

#### 4. **Specific Requirements: Service Provider view**

#### 5. **Specific Requirements: Content Provider**

#### 6. **Specific Requirements: Network Provider**

#### 7. **System Core Functions**

7.1 Bit Transport

7.2 Network Management

7.3 Session Picture Reference: Typical VoD session (VoD Contract 1.5.4)

7.4 Access Control

7.5 Navigation, Programme Selection & Choise Picture Reference: Annex 3 a Picture Reference: Annex 3 b Picture Reference: Annex 3 c

7.6 Application Launch

7.7 Media Synchronisation Links

7.8 Application Control

## 7.9 Presentatie Control

### 7.10 Usage Data

#### 7.10.1 keep track of STB user records

7.10.1.1 Interface Each user record should contain the following information elements for one VoD connection:

1. one VPI/VCI for the downstream 2 Mbps channel,
  2. one bidirectional VPI/VCI for the up/downstream 9.6kbps channel,
  3. one port number for Video Server SDH,
  4. one port number for the Router SDH,
  5. one port number for the Access SDH connection,
  6. one ASU identification number,
  7. one LIM identification number,
  8. one port number in the E1 LIM,
  9. one IP address for the Set-Top Box,
  10. one SerialID/Physical address for the Set-Top Box, bandwidth allocated.
- Additional information like Name, Address and Telephone number may be inserted as well.

#### 7.10.2 keep track of POTS user records

7.10.2.1 Interface A similar record has to be maintained for the POTS service :

1. one bidirectional VPI/VCI for the up/downstream POTS channel,
2. one port number for POTS Mirror SDH,
3. one port number for the Access SDH connection,
4. one POTS ASU identification number,
5. one POTS Mirror ASU identification number,
6. one POTS LIM identification number,
7. one port number in the POTS LIM,
8. one POTS Mirror LIM identification number,
9. one port number in the POTS Mirror LIM,
10. one telephone number, bandwidth allocated.

#### 7.10.3 keep track of OAM user records

7.10.3.1 Interface Additional records for OAM connections have to be maintained. one connection is needed per LIM (E1/V24, POTS or POTS Mirror). This connection has the following fields :

1. one bidirectional VPI/VCI for the OAM channel,
2. one port number for the OAM Station (always the same),
3. one port number for the Access SDH connection,
4. one ASU identification number,
5. one LIM identification number,
6. one connection-ID, bandwidth allocated.

# Chapter 3

## TRADE Analysis

### 3.1 Analysis of the problem environment

In order to specify the requirements on a product, we must understand the environment of the product. TRADE identifies a number of environments of a software product. Two important environments are the *problem environment*, which is the place and the time of the world in which a problem exists that must be solved by the product, but in which the product does not exist; and the *solution environment*, which is the place and time of the world in which the product exists and interacts with entities in its environment. The solution environment is also called the *operating environment* of the product, because it is the environment in which the product operates.

The intention is that a problem environment is transformed into a solution environment. This means that we can also view the problem environment as the operating environment with a number of unknowns, where the unknowns are to be resolved by the development project. To understand the problem environment is to understand the solution environment — they are the same part of the world at different times, namely before and after the product is implemented.

Furthermore, to understand the problem and solution environments, we must first understand what kind of product is being asked for:

- A custom-made product to be used by an individual client,
- A mass-produced product to be used by consumers in a market.

These two kinds of product have different kinds of stakeholders and we must do different things to understand their problem/solution environment.

The VoD system is to be developed for a single client, namely Belgacom. However, it will then be used to sell a service to a market. Thus, there are two kinds of users: those at Belgacom itself and the consumers in the market.

The problem environment is identified in TRADE by

- identifying the stakeholders in the product,
- identifying the needs they have and that must be met by the product,
- identifying the external entities with which the product will interact.

TRADE provides a list with suggested stakeholders for client-oriented products as well as for market-oriented products; see also [34, page 35].

Drawing upon the TRADE suggestions, domain knowledge and upon our common sense, we get the following list of stakeholders.

- Sponsor (person or institution paying for development of VoD system)
- Developer
- Maintenance personnel
- Instructors of all people interacting with the system
- Service provider
- Content provider
- Network provider (network operator)
- Marketing (department that makes the availability of VoD known to the consumer)
- Distribution (department that makes the VoD service available to the user)
- Customer of VoD (person or institution paying for the use of the VoD service)
- User (person using the VoD service)

Note that some of these stakeholders are identified only after we have gained some knowledge of the solution. For example the service provider and content provider are stakeholders that play a role in a particular solution. To gain knowledge of the solution, we either develop part of the solution ourselves or decide to use a standard solution, available on the market or as standardized domain knowledge. Thus, problem analysis and solution specification are two parallel processes that interact. Over time, the locus of activity shifts from problem analysis to solution specification, but there is no point in time where either process is idle. It is important though to be aware which of the two processes one is working on and to classify the outcome of one's activity accordingly. It is for example important not to mistake part of a solution as a result of problem analysis.

Each of the stakeholders has certain needs. In the case study, these have not been analyzed because the focus was on specification techniques. However, we mention the things that should be done to understand the problem environment.

- Understand the business strategy of the sponsor
- Understand the marketing strategy of the marketing department.
- Understand the desires of the customer and user.
- Understand the needs of maintenance personnel, of the distribution department, and the service and content providers.

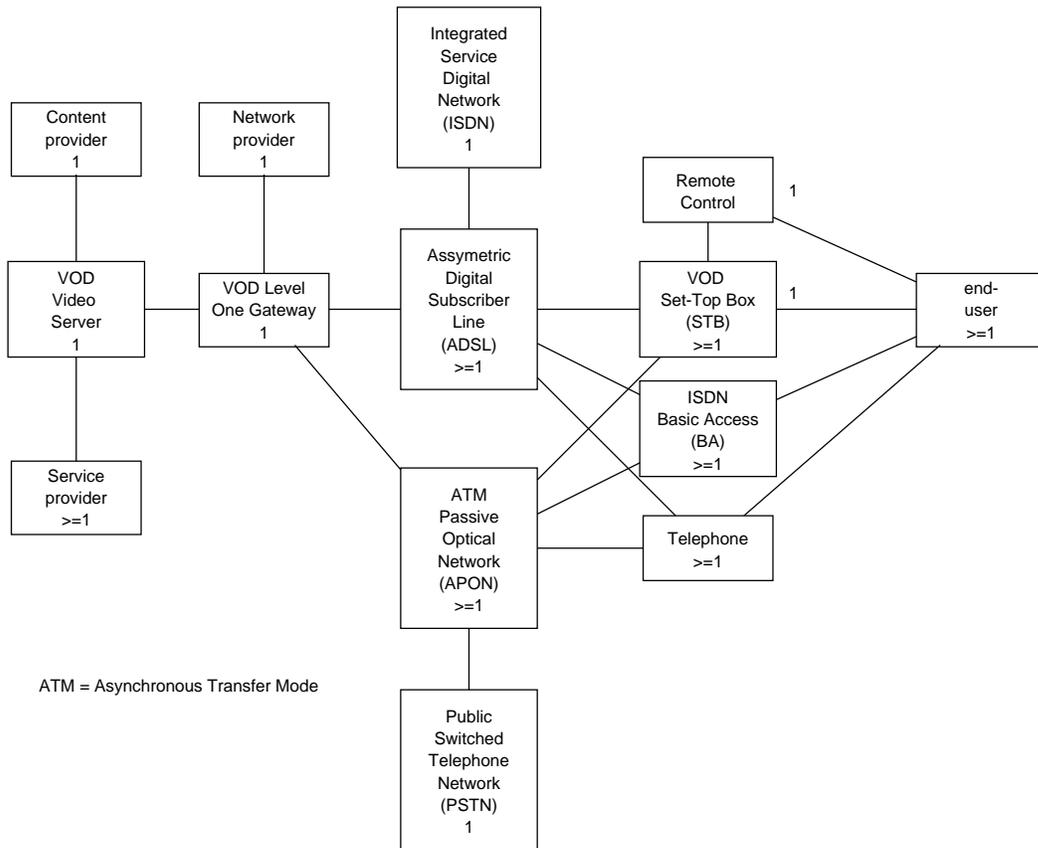
Analysis of the operating environment along these lines should lead to an understanding of the reasons for the functional requirements as well as give a hint what the desired properties of the product are. To specify these properties, techniques like that of Gilb [11] can be used. We are considering adopting these in TRADE. Another useful technique to discover desired properties is Quality Function Deployment [3, 12]

## **3.2 Decomposition of the operating environment**

To understand the requirements on the VoD system, we must decompose the environment in which it will operate.

Figure 3.1 gives a decomposition of the operating environment of the VoD system. Explanations about the conventions used:

- The nodes in the graph represent systems.



**Figure 3.1:** Decomposition of the operating environment of the VoD system.

- The numbers in the nodes represent cardinality constraints, i.e. how many instances of a node can exist at the same time.
- The lines represent possible communications.
- The cardinality constraints on the lines represent how many instances of a node on one side can be linked to how many nodes on the other side. The default is “arbitrarily many, including 0”. The diagram says that one end user communicates with exactly one remote control and one set top box.

The three nodes with VOD in their label are the system to be developed. The nodes labeled ISDN and PSTN represent networks that are linked to the VoD system but with which the VoD system should *not* interact. A requirement on the VoD system is that there is no feature interaction with those networks.

Figure 3.1 is what Jackson [18] calls a *context diagram*. We will follow this terminology. An interesting property of the context diagram of the VoD system is that in a distributed system like this, we cannot separate the decomposition of the environment from the decomposition of the product. In the diagram, the VoD system is decomposed into three components, the set top box (STB), the gateway, and the video server.

Note also that some of the stakeholders of the VoD system appear in the diagram as entities that interact with the system:

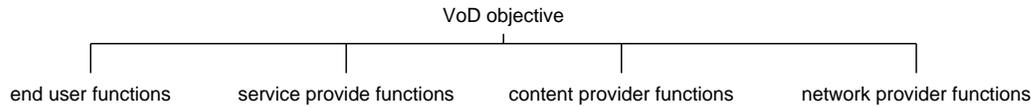
- User (called end-user in the diagram)
- Content provider
- Service provider
- Network provider

A context diagram shows the system under development (SuD) and the systems with which it interacts. As mentioned above, the operating environment consists of the systems with which the product interacts. Now, TRADE defines another kind of environment, called the *implementation environment*. This consists of the lower-level components out of which the SuD will be built. The problem is how we can distinguish the operating environment from the implementation environment. Is the ADSL part of the operating environment or of the implementation environment of the VoD system? We use the following heuristic to resolve this dilemma:

- A system is external to the SuD if and only if the developer has no discretion in specifying and realizing its behavior.

In other words, the SuD is that part of the future world over which the developer has the freedom to choose its behavior and realization. The implementation environment is that part of the future world that is chosen and implemented by the developer. The operating environment of the SuD is part of nature, previously installed products, or products developed by other groups. This is very close to the concept of environment defined by Parnas [23]. Products developed by other groups have a behavior that may or may not be negotiable. In any case, this behavior is influenced by negotiations of the developer of the SuD and other developers; it is not determined by the requirements analysis of the the SuD alone. And the developer of the SuD has nothing to say about the implementation of these other systems at all. This is related to the essential modeling heuristic of McMenamin and Palmer [21]: The developer has the freedom to choose the implementation of the SuD. In order not to make implementation decisions during specification, the developer assumes perfect implementation technology during specification.

Applying this heuristic to the VoD system, all nodes in figure 3.1 not labeled by VOD are part of the operating environment of the VoD system.



**Figure 3.2:** Top of the function refinement tree for the VoD system.

### 3.3 Specification of product functions

To specify product functions, we must know what the product objectives are. The *product objective* is the reason why the product exists. It is comparable to the mission of an organization. The product functions are all related to the product objective because the objective gives the reason why these functions should be present.

The objective of the VoD system is to offer video retrieval services to users, where the videos are stored centrally in databases and where the retrieval service is honored on an individual basis.

A *product function* is a useful interaction between the product and an external entity. The external entities are the components of the operating environment, and they are shown in the context diagram.

The VoD system has functions for the end user, the content provider, the service provider and the network provider. It was remarked already that the ISDN and PSTN are shown in the context diagram to indicate that the VoD system should *not* have an interaction with them. Finally, the interaction between the VoD system and the level one gateway, the ADSL and the ISDN Basic Access network are left out of consideration because these belong to a confidential part of the specifications.

We partition the functions of the VoD system according to the viewpoints of the external entities. Figure 3.2 then gives the top level function refinement tree and figure 3.3 to 3.6 show the refinements for each of the five viewpoints. The functions for the content provider and the accounting functions for the network provider have not been worked out, because there was no information about these in the original documents.

These trees can be used to illustrate the requirements. During requirements specification, they were useful in finding functions that logically belonged to the system but that were not mentioned in the original documents. Note that the tree does *not* show a decomposition of the VoD system into components. Despite its organization according to the external entities with which the system must interact, it does not even show the decomposition of the environment of the system. It merely shows which functions the system should have and why it should have these.

### 3.4 Allocation and flowdown of functions

We can now allocate the required functions of the system to the components of the context diagram. This means that we show for each function what the part played of each of the components in the context diagram is. To *allocate* a function is to determine which component plays a role in the function, *flowdown* is the specification of a function of that the component must have in order to play this role in the higher-level function. Allocation and flowdown is a standard practice in systems engineering [5]. The result of allocation and flowdown can be represented in a matrix. Figure 3.7 shows the flowdown matrix for the VoD context diagram.

To illustrate the sequence of events, figure 3.8 shows a message trace diagram of the interactions between the components. Message trace diagrams are standardized in ITU [17] and they play an important role in the Objectory method [19].

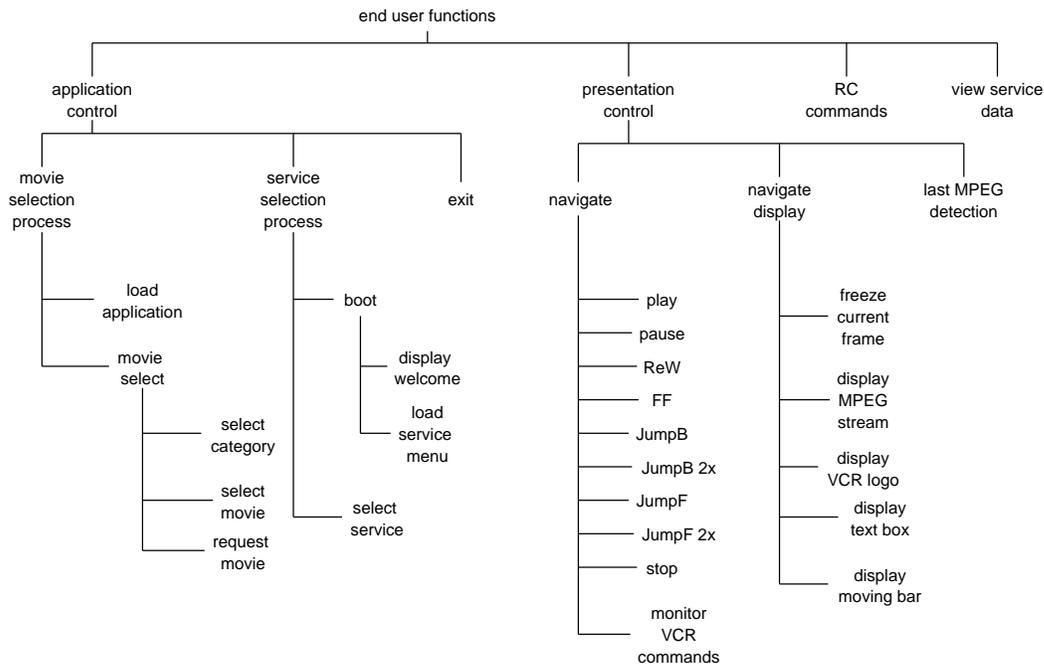


Figure 3.3: Refinement tree for the functions of the VoD system for the user.

### 3.5 Component Behavior Specification

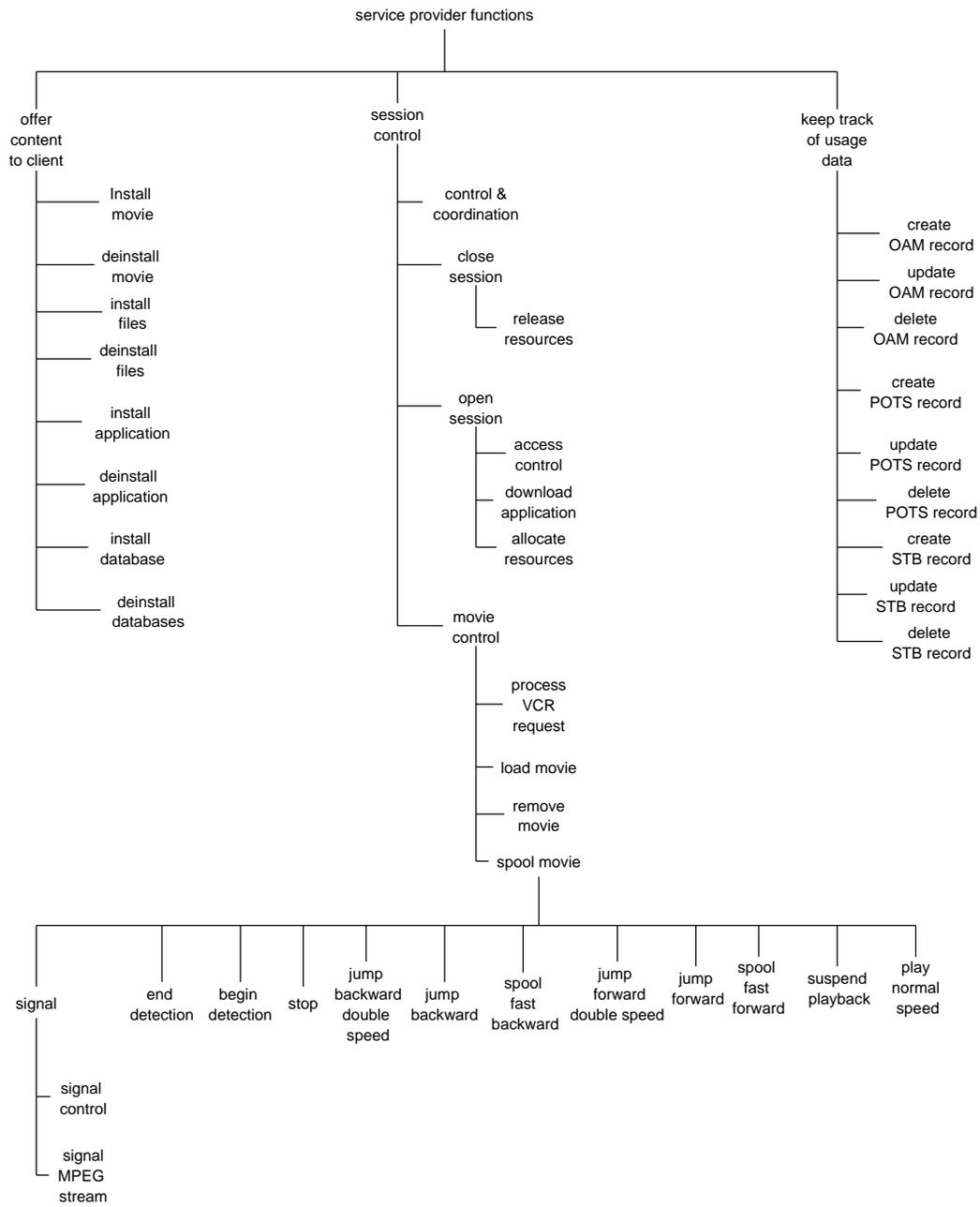
Having allocated functions to the components of the context diagram, we can specify constraints on this behavior. In the original documents, only the behavior of the STB was described. This was represented by means of a state transition diagram, but due to the large number of states and transitions, this did not help to understand the requirements. Figures 3.9 to 3.13 show a state transition table for the STB. The top row in this table shows the events to which the STB must respond and the leftmost column lists all possible states of the STB. Each entry shows the actions taken by the STB upon reception of an event and the state entered when these actions are done. Information about the behavior required of the level one gateway or of the video server is confidential and not presented here.

### 3.6 Decomposition and flowdown of functions for the video server

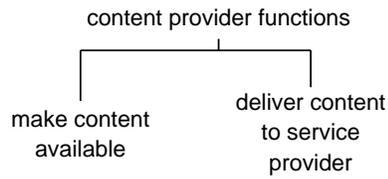
The video server can be further decomposed as shown in figure 3.14. Since from the flowdown we can derive the functions of the video server, we can repeat the allocation and flowdown one level lower, i.e. the video server functions can be allocated to the video server components. Figure 3.15 shows a flowdown of functions to these components. The same could be done for the level one gateway, but this is part of confidential information, so we do not present this here.

### 3.7 Decomposition of the subject domain

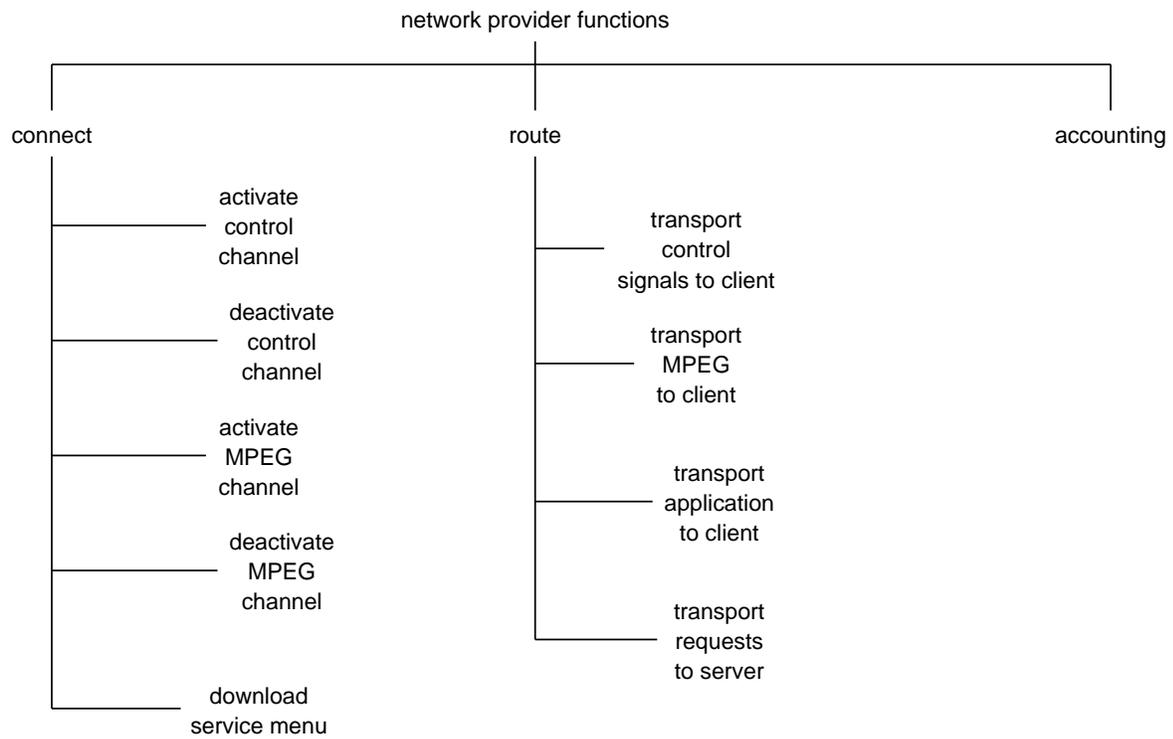
Any information system has a *subject domain*, which is the part of the world about which it stores information. The subject domain is an environment of the system, just as the problem and operating environments are. The decomposition of the subject domain is traditionally represented by an entity-relationship diagram or one of its



**Figure 3.4:** Refinement tree for the functions of the VoD system for the service provider.



**Figure 3.5:** Refinement tree for the functions of the VoD system for the content provider.



**Figure 3.6:** Refinement tree for the functions of the VoD system for the network provider.

VOD system	boot	select service	select category	select movie	play		pause
User	Power on STB	Select	Select	Select	play		pause
Remote Control	_	RC command	RC command	RC command	RC command		RC command
VOD Set Top Box	boot display welcome load service menu	select service load application	select category	select movie request movie display MPEG stream display VCR logo monitor VCR com. play	display MPEG stream display VCR logo monitor VCR com. play		freeze current screen display VCR logo monitor VCR com. pause
VOD Level One gateway	download service menu	activate control ch. activate MPEG ch. transport application		transport request accounting transport MPEG	transport request transport MPEG		transport request
VOD video Server		access control allocate resources download application		control & coord. movie control load movie play normal speed	control & coord. processVCR request play normal speed end detection signal MPEG		control & coord. processVCR request suspend playback

VOD system	fast forward	jump forward	rewind	jump back
User	fast forward	jump forward	rewind	jump back
Remote Control	RC commands	RC commands	RC commands	RC commands
VOD Set Top Box	display MPEG stream display VCR logo monitor VCR com. FF	display moving bar display VCR logo monitor VCR com. JumpF double speed JumpF	display MPEG stream display VCR logo monitor VCR com. ReW	display moving bar display VCR logo monitor VCR com. JumpB double speed JumpB
VOD Level One Gateway	transport request transport MPEG	transport request transport control	transport request transport MPEG	transport request transport control
VOD Video Server	control & coord. processVCR request spool forward end detection signal MPEG	control & coord. processVCR request jump forward (double speed) end detection signal control	control & coord. processVCR request spool backward begin detection signal MPEG	control & coord. processVCR request jump backward (double speed) begin detection signal control

VOD system	stop	confirm	timeout	exit
User	stop	confirm	-	STB off
Remote Control	RC commands	RC commands	-	_
VOD Set Top Box	display text box monitor VCR com. stop last MPEG detection display category menu	confirm display text box monitor VCR comm.	display text box monitor VCR comm. stop	exit
VOD Level One Gateway	transport request			transport request deactivate control deactivate MPEG
VOD Video Server	control & coord. remove movie time out			release resources

Figure 3.7: Flowdown of transactions to components of the context diagram.

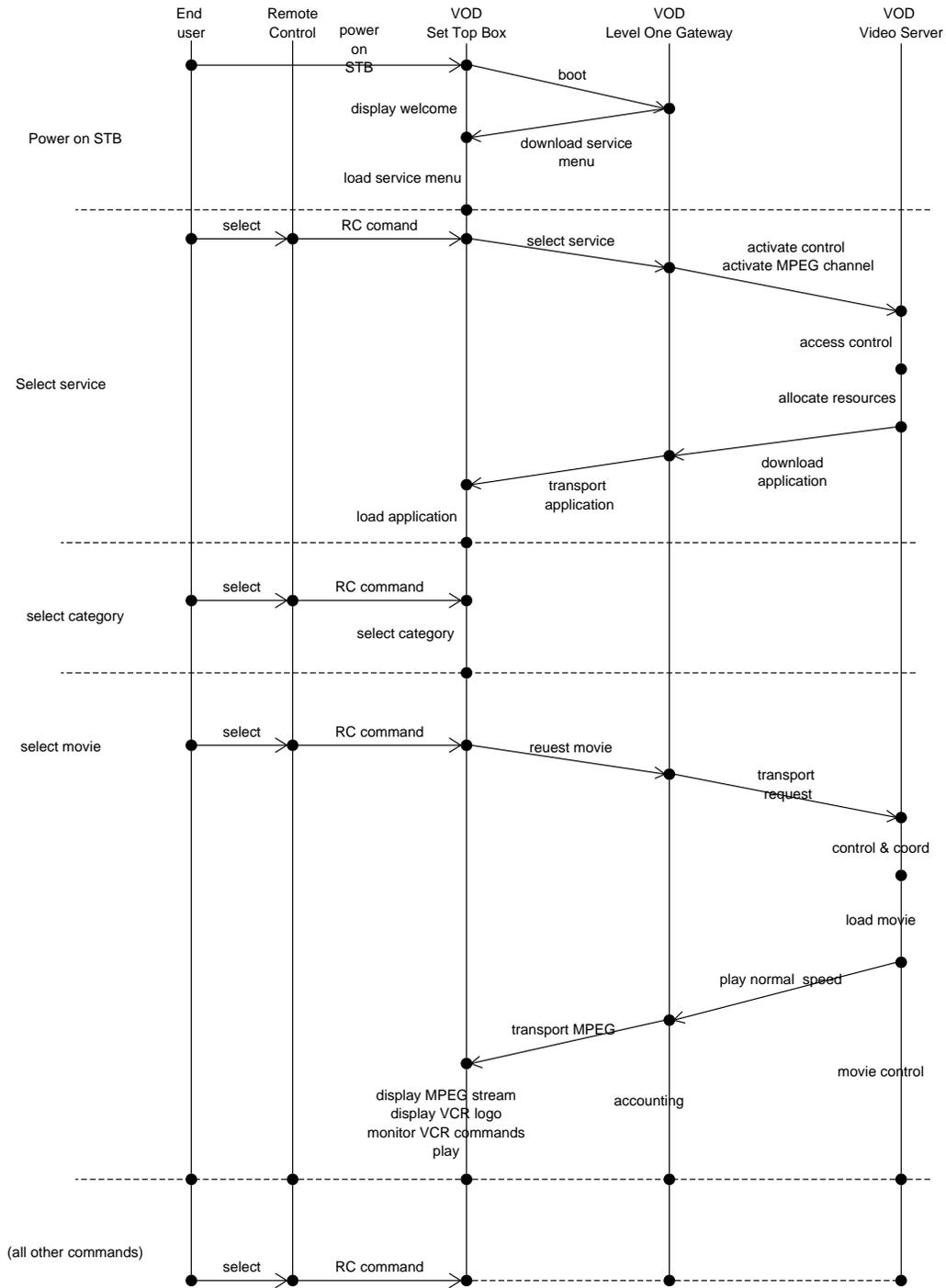


Figure 3.8: Message trace diagram.

Events show in top row, states shown in leftmost column	Power on STB	Download service menu	select service	Transport application	select category	select movie	Transport MPEG	exit
1: Power is off	boot; display welcome/ 2							
2: Booting		Load service menu / 3						exit / 1
3: Menu 1 (service select) is being displayed			select service / 4					exit / 1
4: Selecting service				load application / 5				exit / 1
5: Menu 2 (category select) is being displayed					select category / 6			exit / 1
6: Menu 3 (movie select) is being displayed						select movie; send request movie / 7		exit / 1
7: Movie requested							display MPEG stream; display VCR logo; monitor VCR commands; play / 8	exit / 1

**Figure 3.9:** State transition table for the end-user interface of the STB — part 1.

many variants. Since the VoD system manipulates data that refers to a part of the world, it has a subject domain. From the DAVIC standard for Movies on Demand [4] we get the following subjects about which the system must manipulate data:

- Data significant for the end user:
  - Movie
  - Category
  - Language
  - Additional data with regard to content playing.
  - Reservation
  - Blocked contents
  - Preferences
  - Presentation format
- Service provider data
  - Billing data
  - usage data
  - user data

Figure 3.16 gives a first version of a decomposition diagram of a domain containing these subjects. In TRADE, such a diagram is called a *class relationship diagram (CRD)*, because it shows classes of objects and their relationships, not individual objects and their links. Note that the objects whose classes are represented in the CRD are *not* physical components of the system. However, they are *conceptual components* because the system must store data about them. The system will contain a surrogate for each subject in the subject domain and will

Events show in top row, states shown in leftmost column	Pause	Play	last MPEG detection
8: Normal play	freeze current screen; display VCR logo; monitor VCR commands; pause / 9		/ 20
9: Pausing		send request movie; display MPEG stream; display VCR logo; monitor VCR commands; play / 8	
10: Moving fast forward	freeze current screen; display VCR logo; monitor VCR commands; pause / 9	send request movie; display MPEG stream; display VCR logo; monitor VCR commands; play / 8	
12: Jumping forward	freeze current screen; display VCR logo; monitor VCR commands; pause / 9	send request movie; display MPEG stream; display VCR logo; monitor VCR commands; play / 8	
15: Rewinding	freeze current screen; display VCR logo; monitor VCR commands; pause / 9	send request movie; display MPEG stream; display VCR logo; monitor VCR commands; play / 8	
16: Rewind begin reached		send request movie; display MPEG stream; display VCR logo; monitor VCR commands; play / 8	
17: Jumping backward	freeze current screen; display VCR logo; monitor VCR commands; pause / 9	send request movie; display MPEG stream; display VCR logo; monitor VCR commands; play / 8	
19: Jump backward begin reached		send request movie; display MPEG stream; display VCR logo; monitor VCR commands; play / 8	

**Figure 3.10:** State transition table for the end-user interface of the STB — part 2.

Events show in top row, states shown in leftmost column	FF	JumpF	double speed	end detected
8: Normal play	display MPEG stream; display VCR logo; monitor VCR commands; FF/ 10	display moving bar; display VCR logo; monitor VCR commands; JumpF/ 12		
9: Pausing	display MPEG stream; display VCR logo; monitor VCR commands; FF/ 10	display moving bar; display VCR logo; monitor VCR commands; JumpF/ 12		
10: Moving fast forward				end detected/ 11
12: Jumping forward	display MPEG stream; display VCR logo; monitor VCR commands; FF/ 10		double speed/ 13	end detected/ 14
13: Jumping forward at double speed				end detected/ 14
15: Rewinding	display MPEG stream; display VCR logo; monitor VCR commands; FF/ 10			
16: Rewind begin reached	display MPEG stream; display VCR logo; monitor VCR commands; FF/ 10	display moving bar; display VCR logo; monitor VCR commands; JumpF/ 12	double speed/ 18	
17: Jumping backward	display MPEG stream; display VCR logo; monitor VCR commands; FF/ 10	display moving bar; display VCR logo; monitor VCR commands; JumpF/ 12		
19: Jump backward begin reached	display MPEG stream; display VCR logo; monitor VCR commands; FF/ 10	display moving bar; display VCR logo; monitor VCR commands; JumpF/ 12		

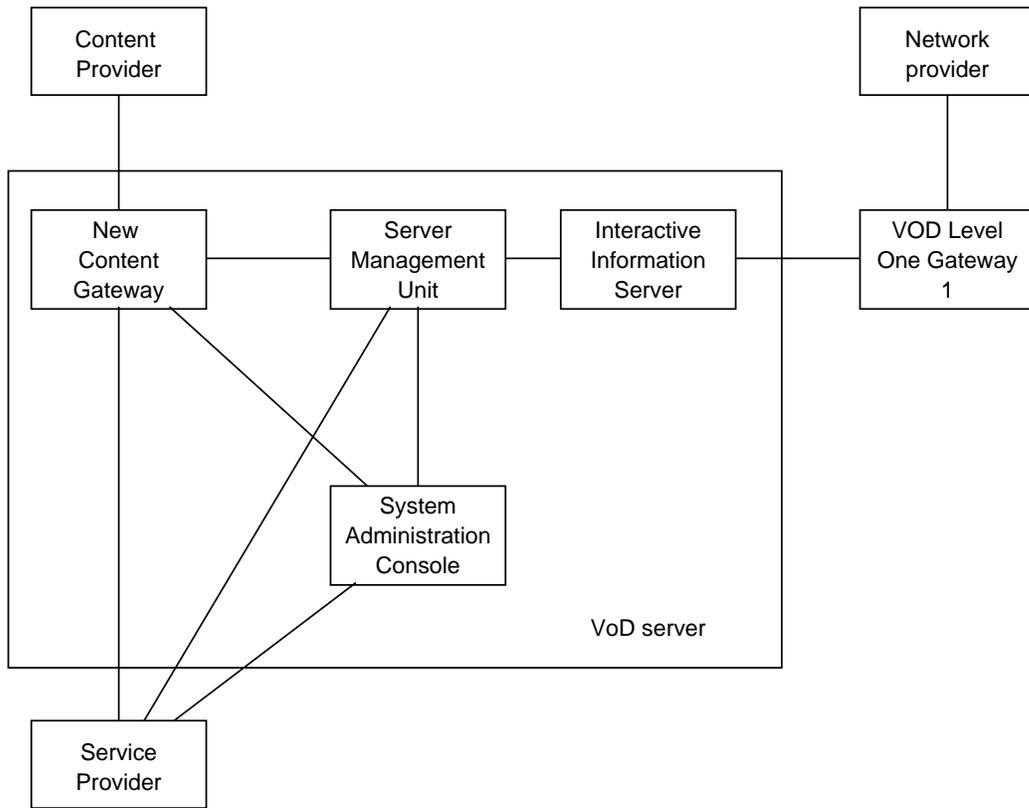
**Figure 3.11:** State transition table for the end-user interface of the STB — part 3.

Events show in top row, states shown in leftmost column	Rew	JumpB	begin detected
8: Normal play	display MPEG stream; display VCR logo; monitor VCR commands; Rew / 15	display moving bar; display VCR logo; monitor VCR commands; JumpB / 17	
9: Pausing	display MPEG stream; display VCR logo; monitor VCR commands; Rew / 15	display moving bar; display VCR logo; monitor VCR commands; JumpB / 17	
11: Fast forward end reached	display MPEG stream; display VCR logo; monitor VCR commands; Rew / 15	display moving bar; display VCR logo; monitor VCR commands; JumpB / 17	
12: Jumping forward	display MPEG stream; display VCR logo; monitor VCR commands; Rew / 15	display moving bar; display VCR logo; monitor VCR commands; JumpB / 17	
13: Jumping forward at double speed	display MPEG stream; display VCR logo; monitor VCR commands; Rew / 15		
14: Jump forward end reached	display MPEG stream; display VCR logo; monitor VCR commands; Rew / 15	display moving bar; display VCR logo; monitor VCR commands; JumpB / 17	
15: Rewinding		display moving bar; display VCR logo; monitor VCR commands; JumpB / 17	begin detected / 16
17: Jumping backward	display MPEG stream; display VCR logo; monitor VCR commands; Rew / 15		
18: Jumping backward at double speed	display MPEG stream; display VCR logo; monitor VCR commands; Rew / 15		
20: At movie end	display MPEG stream; display VCR logo; monitor VCR commands; Rew / 15	display moving bar; display VCR logo; monitor VCR commands; JumpB / 17	

**Figure 3.12:** State transition table for the end-user interface of the STB — part 4.

Events show in top row, states shown in leftmost column	Stop	confirm	timeout	Exit
8: Normal play	stop / 21			exit / 1
9: Pausing	stop / 21		display text box; monitor VCR commands; stop / 5	exit / 1
10: Moving fast forward				exit / 1
11: Fast forward end reached	stop / 21		display text box; monitor VCR commands; stop / 5	exit / 1
12: Jumping forward				exit / 1
13: Jumping forward at double speed				exit / 1
14: Jump forward end reached	stop / 21		display text box; monitor VCR commands; stop / 5	exit / 1
15: Rewinding				exit / 1
16: Rewind begin reached	stop / 21		display text box; monitor VCR commands; stop / 5	exit / 1
17: Jumping backward				exit / 1
18: Jumping backward at double speed				exit / 1
19: Jump backward begin reached	stop / 21		display text box; monitor VCR commands; stop / 5	exit / 1
20: At movie end	stop / 21		display text box; monitor VCR commands; stop / 5	exit / 1
21: Stopped		display text box; monitor VCR commands / 5	display text biox / return to previous state	exit / 1

**Figure 3.13:** State transition table for the end-user interface of the STB — part 5.



**Figure 3.14:** Decomposition of the video server.

Video server	boot	select service	select category	select movie	play	pause
Interactive Information Server(IIS)				movie control load movie	processVCR request play normal speed end detection signal MPEG	processVCR request suspend playback
Server Management Unit (SMU)		access control allocate resources download application		control & coord.	control & coord.	control & coord.
System Administration Console (SAC)						
New Content Gateway (NCG)						

VOD server	fast forward	jump forward (2x) mode	rewind mode	jump back (2x)	end movie	exit
Interactive Information Server(IIS)	processVCR request spool forward end detection signal MPEG	processVCR request jump forward (double) end detection signal control	processVCR request spool backward begin detection signal MPEG	processVCR request jump backward (double) begin detection signal control	remove movie	
Server Management Unit (SMU)	control & coord.	control & coord.	control & coord.	control & coord.	control & coord.	release resources
System Administration Console (SAC)						
New Content Gateway (NCG)						

Figure 3.15: Flowdown of functions to components of the video server.

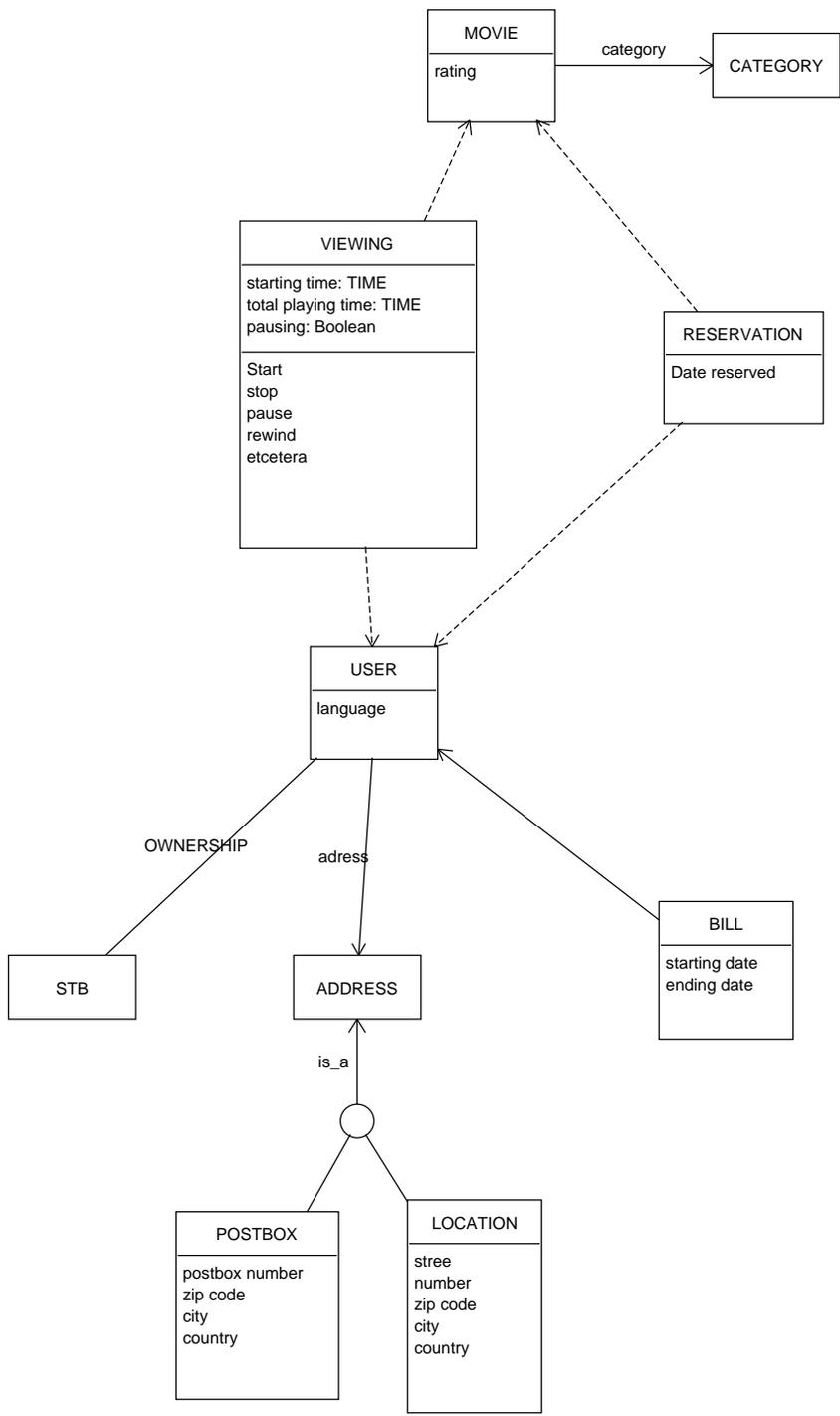


Figure 3.16: CRD of the subject domain.

	VIDEO SERVER	LEVEL 1 GATEWAY	SET TOP BOX
CATEGORY	X		
MOVIE	X		
VIEWING	X		
RESERVATION	X		
USER	X	X	
STB		X	
ADDRESS		X	
BILL		X	

**Figure 3.17:** Allocation of data about the subject domain, shown by means of a traceability table.

register the creation, change of state, and destruction of the subjects by creating, updating and deleting surrogates for these subjects. If any of the subjects is also a component of the system, it will appear in the context diagram or in some decomposition diagram of the system. Whether or not some subject is also part of the system is irrelevant and we can ignore that here.

All of the data represented in the CRD must be remembered in a database. We therefore have a second type of allocation table, viz. a table in which we allocate conceptual objects to physical components of the system. This table is shown in figure 3.17. We do not show the subtypes of ADDRESS, because they are allocated to the same physical component as ADDRESS. Because in this case the allocation is simple, we can also show it on the CRD (figure 3.18).

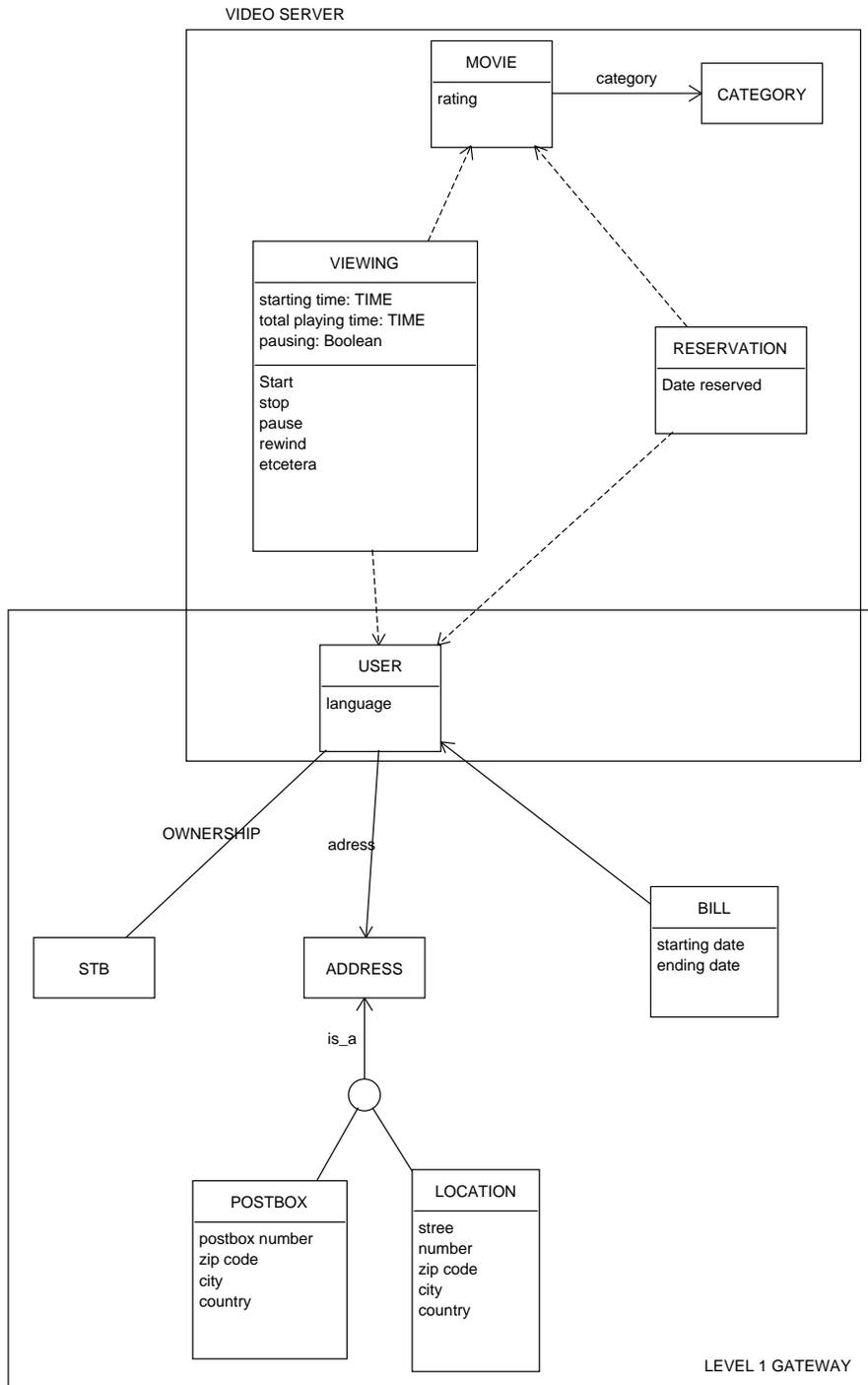


Figure 3.18: Allocation of data about the subject domain, shown in the CRD.

# Chapter 4

## Discussion and conclusions

### 4.1 The UoD of a requirements specification

Each narrative (written or spoken) has a *universe of discourse*, which is the part of the world referred to by the narrative. The narrative need not be a description of the UoD; it may also be a prescription how the UoD should be. For the reader or listener of the narrative, the UoD is the part of the world that must be understood in order to understand the narrative. The narrative presupposes knowledge about the UoD and provides extra information about the UoD.

In the case of the VoD requirements in chapter 2, the UoD of the natural language requirements specification is the VoD system to be developed as well as the part of the world that already exists and with which the VoD system will interact. That is, the UoD of the requirements specification consists of the SuD and its operating environment. In order to understand the requirements, a decomposition diagram of the operating environment should therefore be drawn. The decomposition diagrams in sections 3.2 (operating environment) and 3.6 (decomposition of video server) provide some of the information necessary to understand the requirements. These diagrams illustrate a number of assumptions made in the requirements about the operating environment, such as the allocation of tasks to ADSL and PON technology. However, not all assumptions are represented in the diagrams provided in this report. For example, there is, for the uninitiated, no clue as to the location and function of the ATM-LAN router, the Ethernet and the BOOTP server mentioned in requirement 3.1.1. To complete the requirements, this information should be added to the decomposition diagrams. This is useful even for the initiated, for one important property of implicit knowledge shared by experts is that there may be subtle but important differences in this knowledge, determined by the differences in personalities and background knowledge of the experts [24]. These differences may surface in different interpretations of abbreviations, different interpretations of where the system boundary lies, what an interface “normally” looks like, etc. The suggestion that all knowledge that “goes without saying” is truly common knowledge should therefore be regarded with suspicion.

We conclude that the operating environment and the parts of the SuD that are decided upon from the start, should be represented by means of decomposition diagrams. These diagrams illustrate the requirements and make them understandable. They also serve as a common frame of reference that avoids misunderstandings.

### 4.2 The UoD of a computer-based system

If the SuD contains an information system, as the VoD system does, then there is a second UoD that is important. An information system registers data about a part of the world and communicates this data to users of the system. There are two types of communication here: a message from the environment to the information system that the state of the world has changed and a message from the information system to its environment containing

information about the state of the information system; this state, in turn represents a state of the real world. Thus, the UoD of the information system is the part of the world about which it stores data and which is the subject of the communications between the system and its environment.

Once again, this UoD has a decomposition that just be represented in order to understand the communications between the information system and its environment. This decomposition is represented by a class diagram, which is a decomposition diagram at the type level.

### 4.3 Simulation

The essence of engineering is (1) that the product to be built is specified before it is built and that (2) predictions about the properties of the product are made on the basis of the specifications. In particular, engineers ensure that the properties of the product fall within the ranges prescribed by the requirements.

Prediction requires knowledge about the regularities in the predicted phenomena; for where there is no regularity, there can be no prediction. These regularities are laid down in empirically validated laws of nature or in empirically validated rules of thumb that have proven to be true even if it is not known exactly how these rules relate to the laws of nature. For example, nuclear engineers use the rule of thumb that one gram of uranium gives one megawatt day of energy. This is an empirically validated rule that has never been derived from the laws of nature.

Some of the predictive needs in the VoD requirements specification concern the real-time constraints cattered over the specification. Questions that arise include:

- Are all the real-time constraints in the specs consistent?
- What are the consequences for real-time constraints for components? In other words, how do the real-time constraints flow down to components?
- What are the design alternatives within the constraints set by the real-time constraints?

These answers must be answered by means of simulation, where we take this in a very general sense as prediction of the properties of the implementation from the specification. “Simulation” may include visiting the neighbors to see how they did it, building a scale model to see how it behaves, computing the properties of the product from a specification of the product, etc. In the case of software requirements, simulation can include the implementation of throw-away prototypes, making mock-ups, and writing a specification that can be executed. For example, Statemate includes a facility to simulate a real-time system on the basis of diagram specifications [13, 15, 16]. Statemate also offers reachability analysis to discover deadlock and to discover the real-time bounds within which a process can execute. In addition, Statemate can generate prototypes, that execute in real-time and thus can simulate the behavior of the product in real-time [14].

### 4.4 Formal specification

Formal specifications can play several roles in this engineering process.

- They can be used to predict properties of the finished product from the specification. The axioms of the formal logic used then play the role of the laws of nature and rules of thumb mentioned above. For example, real-time properties of the system and its possible decompositions can be derived, design options explored, consistency checked, etc. The derivations can be done by hand or by mechanical theorem-prover. Usually, formal specification is needed for a few selected parts of the requirements, where this is cost-effective, but not for the entire system. For example, a specification of the behavior of the video server, the level one gateway and the set top box all interact through the communication network. It is not immediately apparent that these are mutually consistent, i.e. avoid deadlock.

- Even if a decomposition specification is consistent, it may specify the wrong decomposition. Formal specifications can be used to verify that a decomposition of a system is correct with respect to a specification of the system itself. For example, if the behavior of the video server, the level one gateway and the set top box have all been specified and are mutually consistent, then it can be shown that the overall behavior of the entire VoD system that flows from this is, indeed, the desired behavior. In general, if the properties of a system have been specified in the same language as those of its components, then a language that allows formal deduction allows verification that the system properties follow from the component properties and the way in which the components are connected.
- If a specification is to be executed, then a formal semantics is needed, if only because the interpreter of the specification is itself a formal system that makes all choices in the semantics explicit. Again, Statemate is an example where this was done in a thorough way.
- Formal specification theory has an important role to play in the definition of semiformal, diagram-based specification techniques. Each such semiformal technique should have a formal counterpart. This way, ambiguities in the diagrams are avoided and this in turn makes the diagrams easier to use. The TRADE toolkit is defined in this way, using LCM as formal specification language.

## 4.5 Semiformal specifications

We already observed that the semiformal specifications help in explaining and interpreting the informal requirements. However, they also help in explaining and interpreting the formal specifications. (This is not illustrated in this report.) Formal specifications should always be accompanied with semiformal specifications and comments that explain their meaning.

In addition, semiformal specifications help structuring the requirements. This helps making requirements manageable and also makes blank spots visible. For example, drawing a function refinement tree made clear that a number of functions had been omitted in the informal requirements (because they “went without saying” or because nobody had thought about them). Also, drawing a context diagram of the system, it becomes clear that the requirements are stated in terms of the components of the VoD system and not in terms of the system as a black box with a number of users.

It should be observed that the semiformal specification in this report uses some ideas that also play a role in the SDL world - decomposition in blocks, representation of communication by message sequence chart (MSC). One promising line of research is to see in whether and how these ideas can be incorporated in the TRADE toolkit. In particular, TRADE could be extended as a toolkit with concepts and heuristics to be used in combination with the SDL and MSC techniques.

# Bibliography

- [1] P. du Bois. *The ALBERT II Language*. PhD thesis, Facultés Universitaires Notre-Dame de la Paix, Namur, 1995.
- [2] J. Broersen, R. Feenstra, and R.J. Wieringa. Minimal semantics for transaction specifications in a multi-modal logic. Technical report, Faculty of Mathematics and Computer Science, *Vrije Universiteit*, De Boelelaan 1081a, 1081 HV Amsterdam, 1996.
- [3] P.G. Brown. QFD: echoing the voice of the customer. *AT&T Technical Journal*, pages 18–32, March/April 1991.
- [4] Digital Audio-Video Council. *DAVIC 1.0 Specification part 00*, 1996.
- [5] A.M. Davis. *Software Requirements: Objects, Functions, States*. Prentice-Hall, 1993.
- [6] F. Dehne and R.J. Wieringa. Toolkit for conceptual modeling (tcm): User’s guide. Technical report, Faculty of Mathematics and Computer Science, *Vrije Universiteit*, De Boelelaan 1081a, 1081 HV Amsterdam, 1996. <http://www.cs.vu.nl/~tcm>.
- [7] M. Dorfman and R.H. Thayer, editors. *Standards, Guidelines, and Examples on System and Software Requirements Engineering*. Computer Science Press, 1990.
- [8] ESA. *ESA Software Engineering Standards*, pages 101–120. In Dorfman and Thayer [7], 1987.
- [9] R.B. Feenstra and R.J. Wieringa. LCM 3.0: a language for describing conceptual models. Technical Report IR-344, Faculty of Mathematics and Computer Science, *Vrije Universiteit*, Amsterdam, December 1993.
- [10] D.C. Gause and G.M. Weinberg. *Exploring Requirements: Quality Before Design*. Dorset House Publishing, 1989.
- [11] T. Gilb. *Principles of Software Engineering Management*. Addison-Wesley, 1988.
- [12] J.R. Hauser and D. Clausing. The house of quality. *Harvard Business review*, 66(3):63–73, May–June 1988.
- [13] i-Logix Inc., 22 Third Avenue, Burlington, Mass. 01803, U.S.A. *STATEMATE Analyzer Reference Manual Version 5.0*, 1993.
- [14] i-Logix Inc., 22 Third Avenue, Burlington, Mass. 01803, U.S.A. *STATEMATE Prototyper Reference Manual Version 5.0*, 1993.
- [15] i-Logix Inc., 22 Third Avenue, Burlington, Mass. 01803, U.S.A. *STATEMATE User Reference Manual Volume I, Version 5.0*, 1993.
- [16] i-Logix Inc., 22 Third Avenue, Burlington, Mass. 01803, U.S.A. *STATEMATE User Reference Manual Volume II, Version 5.0*, 1993.
- [17] International telecommunication union. *Criteria for the Use and Applicability of Formal Description Techniques: Message Sequence Charts (MSC)*, 1994. Z.120 (03/93).
- [18] M. Jackson. *Software Requirements and Specifications: A lexicon of practice, principles and prejudices*. Addison-Wesley, 1995.
- [19] I. Jacobson, M. Christerson, P. Johnsson, and G. Övergaard. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Prentice-Hall, 1992.
- [20] R. Jungclaus, R.J. Wieringa, P. Hartel, G. Saake, and Thorsten Hartmann. Combining Troll with the Object Modeling Technique. In B. Wolfinger, editor, *Innovationen bei Rechen- und Kommunikationssystemen. GI-Fachgespräch FG 1: Integration von semi-formalen und formalen Methoden für die Spezifikation von Software*, pages 35–42. Springer, Informatik aktuell, 1994.

- [21] S.M. McMenamin and J.F. Palmer. *Essential Systems Analysis*. Yourdon Press/Prentice Hall, 1984.
- [22] Oblog Software. *OBLOG Case VI.2*, 1995.
- [23] D.L. Parnas and J. Madey. Functional documents for computer systems. *Science of Computer programming*, 25:41–61, 1995.
- [24] M. Polanyi. *Personal Knowledge*. Chicago University Press, 1958.
- [25] Quality Systems and Software Ltd. *DOORS Unix Reference Manual, Version 2.1*, 1995.
- [26] P.A. Spruit, R.J. Wieringa, and J.-J.Ch. Meyer. Regular database update logics. Submitted for publication.
- [27] P.A. Spruit, R.J. Wieringa, and J.-J.Ch. Meyer. Axiomatization, declarative semantics and operational semantics of passive and active updates in logic databases. *Journal of Logic and Computation*, 5(1):27–50, 1995.
- [28] R.J. Wieringa. Equational specification of dynamic objects. In R.A. Meersman, W. Kent, and S. Khosla, editors, *Object-Oriented Databases: Analysis, Design, and Construction (DS-4)*, pages 415–438. North-Holland, 1991.
- [29] R.J. Wieringa. A formalization of objects using equational dynamic logic. In C. Delobel, M. Kifer, and Y. Masunaga, editors, *2nd International Conference on Deductive and Object-Oriented Databases (DOOD'91)*, pages 431–452. Springer, 1991. Lecture Notes in Computer Science 566.
- [30] R.J. Wieringa. Object-oriented analysis, structured analysis, and Jackson System Development. In F. van Assche, B. Moulin, and C. Rolland, editors, *Object Oriented Approach in Information Systems*, pages 1–21. North-Holland, 1991.
- [31] R.J. Wieringa. LCM 3.0: Specification of a control system using dynamic logic and process algebra. In C. Lewerentz and T. Lindner, editors, *Formal Development of Reactive Systems — Case Study Production Cell*, pages 333–355. Springer, 1994. Lecture Notes in Computer Science 891.
- [32] R.J. Wieringa. Combining static and dynamic modeling methods: a comparison of four methods. *The Computer Journal*, 38(1):17–30, 1995.
- [33] R.J. Wieringa. *Requirements Engineering: Semantic, Real-Time, and Object-Oriented Methods*. Course notes., 1995.
- [34] R.J. Wieringa. *Requirements Engineering: Frameworks for Understanding*. Wiley, 1996.
- [35] R.J. Wieringa. A toolkit for requirements and design engineering (TRADE): Tool definitions and directions for use. Technical report, Faculty of Mathematics and Computer Science, *Vrije Universiteit*, De Boelelaan 1081a, 1081 HV Amsterdam, 1996.
- [36] R.J. Wieringa and R.B. Feenstra. The university library document circulation system specified in LCM 3.0. Technical Report IR-343, Faculty of Mathematics and Computer Science, *Vrije Universiteit*, Amsterdam, December 1993.
- [37] R.J. Wieringa, W. de Jonge, and P.A. Spruit. Using dynamic classes and role classes to model object migration. *Theory and Practice of Object Systems*, 1(1):61–83, 1995.
- [38] R.J. Wieringa and G. Saake. A formal analysis of the Shlaer-Mellor method: towards a toolkit for formal and informal requirements specification techniques. To be published, *Requirements Engineering Journal*, 1995.