Project no: FP6-031486
Project acronym: EFFORTS
Project title: Effective Operation in Ports

Instrument: Integrated Project
Thematic Priority: Integrating and Strengthening the European Research Area

Deliverable 3.1.1
Handbook of Process Modelling
- A guide to create process diagrams -

Due date of deliverable: 31/12/2006
Actual submission date: 20/01/2007

Start date of project: 01/05/2006                  Duration: 36 months

Organisation name of lead contractor for this deliverable: NUNDP

Revision: V2

<table>
<thead>
<tr>
<th>Dissemination Level</th>
<th>PU</th>
<th>Public</th>
<th>PP</th>
<th>Restricted to other programme participants (including the Commission Services)</th>
<th>RE</th>
<th>Restricted to a group specified by the consortium (including the Commission Services)</th>
<th>CO</th>
<th>Confidential, only for members of the consortium (including the Commission Services)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PU</td>
<td>Public</td>
<td>PP</td>
<td>Restricted to other programme participants (including the Commission Services)</td>
<td>RE</td>
<td>Restricted to a group specified by the consortium (including the Commission Services)</td>
<td>CO</td>
<td>Confidential, only for members of the consortium (including the Commission Services)</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PROJECT DELIVERABLE
DELIVERABLE 3.1.1 – HANDBOOK OF PROCESS MODELLING
A GUIDE TO CREATE PROCESS DIAGRAMS

Author(s): PHANTHIAN ZUESONGDHAM
Partner: HAMBURG UNIVERSITY OF TECHNOLOGY / MARITIME LOGISTICS - ISSUS
Document Code: EFFORTS-WP3.1.1-DEL-20061231-TUHH-FINAL V2.0
Partner Code: TUHH
Date of Issue: 31/12/2006
Status: FINAL
Revision: 1
No. of Copy

Contributing Partners / Authors
NUNDP / SVEN MATHES
NUNDP / NORMAN NOACK

Pages xx /57
Figures 28
Tables 6
Annexes 3

Signatures

Author:

Project Coordinator:

Document Code: EFFORTS-WP3.1.1-DEL-20061231-TUHH-FINAL V2.0
Date: [31/12/2006]
**Record of Changes**

This is a controlled document. The copy number is shown on page ii, the distribution list on page iv.

Additional copies can be obtained through the issuing authority.

In the event of copies being generated locally, each such document shall be numbered 0, indicating that it is an ‘Uncontrolled Copy’.

Amendment shall be by whole document replacement.

<table>
<thead>
<tr>
<th>Version</th>
<th>Status</th>
<th>Date</th>
<th>Authorized by</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Draft version – Consortium review</td>
<td>31.12.06</td>
<td>TUHH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Phanthian Zuesongdham</td>
</tr>
<tr>
<td>2.0</td>
<td>Final version including comment from peer review</td>
<td>20.01.07</td>
<td>TUHH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Phanthian Zuesongdham</td>
</tr>
</tbody>
</table>
# Distribution List

<table>
<thead>
<tr>
<th>Copy No</th>
<th>Recipient</th>
<th>Status</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Table of Contents**

PROJECT DELIVERABLE ..................................................................................................... II  
RECORD OF CHANGES ...................................................................................................... III  
DISTRIBUTION LIST .......................................................................................................... IV  
TABLE OF CONTENTS ...................................................................................................... V  

1 INTRODUCTION ...........................................................................................................  
1.1 Purpose of this handbook ...................................................................................... 1  
1.2 Why process modelling? ....................................................................................... 2  
1.3 The EFFORTS Roadmap – a procedure for process modelling .............................. 3  
1.4 How is the process modelling handbook structured? ............................................ 3  

2 PROCESS MODELLING ARCHITECTURE - CIMOSA ................................................ 4  
2.1 What is CIMOSA? ................................................................................................. 4  
2.1.1 CIMOSA Modelling Framework ...................................................................... 5  
2.1.2 Terminology .................................................................................................... 6  
2.1.3 Interoperability .............................................................................................. 6  
2.2 From modelling architecture to language ............................................................... 6  

3 PROCESS MODELLING LANGUAGE - UML ............................................................... 11  
3.1 What is UML? ....................................................................................................... 11  
3.2 UML diagrams in EFFORTS ................................................................................ 11  
3.2.1 Use case diagram .......................................................................................... 12  
3.2.2 Use case specification .................................................................................. 15  
3.2.3 Activity diagram .......................................................................................... 16  
3.2.4 Class diagram ............................................................................................... 17  

4 TOOLS AND PROGRAMS USED FOR PROCESS MODELLING .............................. 19  
4.1 Paper and pencil ................................................................................................. 19  
4.2 Microsoft Visio with additional UML shapes ..................................................... 19  
4.2.1 The program .................................................................................................. 19  
4.2.2 The shapes .................................................................................................... 20  
4.3 Mindjet MindManager ......................................................................................... 21  
4.4 Spreadsheet ......................................................................................................... 21  

5 THE EFFORTS ROADMAP FOR PROCESS MODELLING ........................................... 24  
5.1 Process modelling – Procedure ......................................................................... 24  
5.2 Roles and responsibilities .................................................................................. 25  
5.3 Milestones and deliverables ................................................................................ 25  

6 PROCESS MODELLING STEP-BY-STEP BY EXAMPLE OF VESSEL TRAFFIC SERVICE (VTS) PROCESS ........ 27  

Document Code: EFFORTS-WP3.1.1-DEL-20061231-ISSUS-FINAL V2.0  
Date: [31/12/2006]
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Set the scope of process</td>
</tr>
<tr>
<td>6.2</td>
<td>Gather the first information</td>
</tr>
<tr>
<td>6.3</td>
<td>Draft the first process in a global picture</td>
</tr>
<tr>
<td>6.4</td>
<td>Interview to get detailed information</td>
</tr>
<tr>
<td>6.5</td>
<td>Create use case diagram</td>
</tr>
<tr>
<td>6.6</td>
<td>Create use case specification</td>
</tr>
<tr>
<td>6.7</td>
<td>Create activity diagram</td>
</tr>
<tr>
<td>6.8</td>
<td>Prepare class diagram</td>
</tr>
<tr>
<td>6.9</td>
<td>Check plausibility of the program and specification with experts</td>
</tr>
<tr>
<td>6.10</td>
<td>Reviews the diagrams</td>
</tr>
<tr>
<td>6.11</td>
<td>Demonstration of development on process modelling</td>
</tr>
<tr>
<td>7</td>
<td>CONCLUSION</td>
</tr>
<tr>
<td>8</td>
<td>REFERENCE</td>
</tr>
<tr>
<td>ANNEX A</td>
<td>UML NOTATION OVERVIEW</td>
</tr>
<tr>
<td>ANNEX B</td>
<td>USE CASE SPECIFICATION TEMPLATE</td>
</tr>
<tr>
<td>ANNEX C</td>
<td>INTERVIEW MINUTE WITH VTS</td>
</tr>
</tbody>
</table>
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 2.1</td>
<td>Evaluation of Tool Selection in EFFORTS</td>
<td>7</td>
</tr>
<tr>
<td>Table 3.1</td>
<td>Official Diagram Types of the UML 2.0</td>
<td>12</td>
</tr>
<tr>
<td>Table 3.2</td>
<td>Use Case Specification</td>
<td>15</td>
</tr>
<tr>
<td>Table 3.3</td>
<td>Notations of Activity Diagram Used in EFFORTS</td>
<td>17</td>
</tr>
<tr>
<td>Table 4.1</td>
<td>Example of Use Case Specification</td>
<td>22</td>
</tr>
<tr>
<td>Table 6.1</td>
<td>VTS Process – Use Case Specification</td>
<td>37</td>
</tr>
</tbody>
</table>

LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.1</td>
<td>Process Modelling Meta-Model</td>
<td>2</td>
</tr>
<tr>
<td>Figure 2.1</td>
<td>The CIMOSA Modelling Approach</td>
<td>5</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>From Modelling Architecture to Language</td>
<td>8</td>
</tr>
<tr>
<td>Figure 2.3</td>
<td>CIMOSA Approach in EFFORTS</td>
<td>9</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>Example of a Simple Use Case Diagram</td>
<td>13</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>Include Relationship Notation</td>
<td>13</td>
</tr>
<tr>
<td>Figure 3.3</td>
<td>An Example of Include Relationship – Bank Services</td>
<td>14</td>
</tr>
<tr>
<td>Figure 3.4</td>
<td>Extend Relationship Notation</td>
<td>14</td>
</tr>
<tr>
<td>Figure 3.5</td>
<td>An Example of Extend Relationship – Process Payment</td>
<td>14</td>
</tr>
<tr>
<td>Figure 3.6</td>
<td>Single Class Diagram</td>
<td>18</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>Example of Visio</td>
<td>20</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>Instruction of Shape Selection</td>
<td>21</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>Example of Mind Map</td>
<td>22</td>
</tr>
<tr>
<td>Figure 5.1</td>
<td>VTS – Scope Map</td>
<td>28</td>
</tr>
<tr>
<td>Figure 5.2</td>
<td>Identifying Use Cases and Their Actors</td>
<td>32</td>
</tr>
<tr>
<td>Figure 5.3</td>
<td>Group of Use Cases</td>
<td>33</td>
</tr>
<tr>
<td>Figure 5.4</td>
<td>Use Case Diagram</td>
<td>36</td>
</tr>
<tr>
<td>Figure 5.4</td>
<td>VTS Process – Activity Diagram</td>
<td>38</td>
</tr>
<tr>
<td>Figure 5.5</td>
<td>VTS Process – Class Diagram Without Association and Operation</td>
<td>39</td>
</tr>
<tr>
<td>Figure 5.6</td>
<td>The First Use Case Diagram – VTS Process</td>
<td>41</td>
</tr>
<tr>
<td>Figure 5.7</td>
<td>The Second Use Case Diagram – VTS Process</td>
<td>42</td>
</tr>
<tr>
<td>Figure 5.8</td>
<td>The Third Use Case Diagram – VTS Process</td>
<td>43</td>
</tr>
<tr>
<td>Figure 5.9</td>
<td>The Fourth Use Case Diagram – VTS Process (Final Version)</td>
<td>44</td>
</tr>
<tr>
<td>Figure 5.11</td>
<td>The Second Activity Diagram – VTS Process</td>
<td>46</td>
</tr>
<tr>
<td>Figure 5.12</td>
<td>The Third Activity Diagram – VTS Process</td>
<td>47</td>
</tr>
<tr>
<td>Figure 5.13</td>
<td>The Fourth Activity Diagram – VTS Process (Final Version)</td>
<td>48</td>
</tr>
<tr>
<td>Figure 5.14</td>
<td>The First Class Diagram Without Association and Operation – VTS Process</td>
<td>49</td>
</tr>
<tr>
<td>Figure 5.15</td>
<td>The Second Class Diagram Without Association and Operation – VTS Process</td>
<td>50</td>
</tr>
</tbody>
</table>
LIST OF ABBREVIATIONS /GLOSSARY

A

**Action**
*Abbreviation: none*  
Source: Pilone (2005), p.104

A unit of work that needs to be carried out or a single step within an activity where data manipulation or processing occurs in a modelled system

**Activity**
*Abbreviation: none*  
Source: Pilone (2005), p.104

A behaviour that is factored into one or more actions

**Activity Diagram**
*Abbreviation: none*  
Source: Pilone (2005), p.104

A Technique to describe procedural logic, business process, and work flow with specific notations

**Activity Edge**
*Abbreviation: none*  
Source: Bennett, Skelton and Lunn (2005), p.238

A notation indicating the movements between actions in the activity diagram

**Activity Final Node**
*Abbreviation: none*  
Source: Bennett, Skelton and Lunn (2005), p.238

A notation indicating where the entire activity terminates in the activity diagram

**Activity Initial Node**
*Abbreviation: none*  
Source: Bennett, Skelton and Lunn (2005), p.238

A notation indicating where the activity begins in the activity diagram

**Actor**
*Abbreviation: none*  
Source: Pilone (2005), p.77

A person involved to the functionality or use case

**Architecture Views**
*Abbreviation: none*  
Source: N.A.

See View Dimension

**Association**
*Abbreviation: none*  
Source: Pilone (2005), p.77

A connecting line between object
**Attribute**

Abbreviation: none  
Source: Pilone (2005), p.12

A description explaining the details or characteristic of a class

**B**

**Building Blocks Instance**  
Abbreviation: none  

User defined individual of a Generic Building Block or a more generic Building Block Type, the characteristic of which (i.e. elements) have been partially (Partial Level) or fully (Particular Level) defined.

**Building Block Type**  
Abbreviation: none  

Specialisation of a Generic Building Block or a more generic Building Block Type; has the nature of an object Class and therefore is defined by its name, a reference to its parent class, a list of elements and rules constraining the possible value for each element.

**Business Process**  
Abbreviation: BP  

Construct which describes pieces of enterprise behaviour at all levels of decomposition in the functional decomposition except the top and bottom levels.

**Business Process Interoperability**  
Abbreviation: BPI  

A State that exists when a business process can meet a specific objective automatically utilizing essential human labour only. It is present when a process conform to standards that enable it to achieve its objective regardless of ownership, location, make, version or design of the computer system used.

**C**

**CIMOSA**  
Abbreviation: none  
Source: CIMOSA (1995)

See Open Source Architecture for Computer Integrated Manufacturing
CIMOSA Models

Abbreviation: none  
Source: CIMOSA Glossary (1996), p. 3

Models used to describe the different phases of the enterprise life cycle (Requirement Definition, Designed Specification and Implementation). They contain set of instances of Generic Building Blocks and Building Block Types structured to present solutions to problem area in manufacturing enterprises in a process oriented language.

CIMOSA Modelling Framework

Abbreviation: none  
Source: CIMOSA Glossary (1996), p.3

A framework providing a common structure for semantic unification within enterprise in the form of Reference Architecture and Particular Architecture. It represents a three-dimensional structure whose orthogonal axes define the CIMOSA Model: Instantiation, Derivation and Generation.

CIMOSA Modelling Levels

Abbreviation: none  
Source: CIMOSA Glossary (1996), p.3-4

Definition of three modelling levels in CIMOSA: Requirement definition, Design Specification, and Implementation Description

CIMOSA View

Abbreviation: none  
Source: CIMOSA Glossary (1996), p.4

It defines along its Generation Dimension four views (Function, Information, Resource, and Organisation) which allow focusing on specific aspects of integrated enterprise models discarding others.

Class

Abbreviation: none  
Source: Pilone (2005), p.11

A group of things that have common state and behaviour

Class Level

Abbreviation: none  
Source: N.A.

See Partial Level

Class Diagram

Abbreviation: none  
Source: Bennett, Skelton and Lunn (2005), p.42

A diagram showing a static view of the classes in a model, or part of a model

Communication View

Abbreviation: none  
Source: N.A.

See Information View
Complexity

Abbreviation: none  
Source: EFFORTS D 3.1.1, v.1.0, p.2

In the use of this document, complexity means an environment with a complex system encompassing large number of stakeholders the number of processes and the amount of interactions between them.

Construct

Abbreviation: none  
Source: CIMOSA Glossary (1996), p. 4

A general term to cover components of Generic Building Blocks, Building Block Types, Partial Models, Particular Models as well as their instances

Decision Node

Abbreviation: none  
Source: Bennett, Skelton and Lunn (2005), p.239

A notation indicating where the flow on exit from an action may go in alternative directions, depending on a condition

Derivation of Models

Abbreviation: none  
Source: N.A.

See Modelling Dimension

Design Specification Models

Abbreviation: none  
Source: CIMOSA Glossary (1996), p.6

Models containing all functional, information, resource and organisational specifications for an optimised operational enterprise system design which verifies the business requirements of an enterprise

Domain Identification

Abbreviation: none  
Source: EFFORTS D 3.1.1, v.1.0, p.10

Identification of processes or problem areas to be analysed following CIMOSA-EFFORTS Model. This identification include name of the domain and its global characteristics of the problems

Estimated Time of Arrival

Abbreviation: ETA  

A measure of when a vehicle, vessel or aircraft in travel, cargo, or emergency services is expected to arrive at a certain place
Estimated Time of Departure  
**Abbreviation: ETD**  
Source: Answer.com (20.12.2006)

A measure of when a vehicle, vessel or aircraft in travel, cargo, or emergency services is expected to arrive at a certain place.

Event  
**Abbreviation: none**  
Source: CIMOSA Glossary (1996), p.8

Solicited and unsolicited real-world happening, timer or request to do something in the enterprise or its environment.

Event Process Chain  
**Abbreviation: EPC**  
Source: Wikipedia.org

A method developed within the framework of ARIS and used by many companies for modelling, analysing, and re-designing business processes. It is an ordered graph of events and functions.

Extend  
**Abbreviation: none**  
Source: Bennett, Skelton and Lunn (2005), p.29

A type of use case relationship used where one use case may optionally be extended by the functionality in another use case.

Extension Point  
**Abbreviation: none**  
Source: EFFORTS D.3.1.1 v.1.0, p.18

Text to define the alternative flows of events and also the open issue(s) which will not be captured by the relevant use case.

F  

Flow Final Node  
**Abbreviation: none**  
Source: Bennett, Skelton and Lunn (2005), p.238

A notation indicating where the path or flow terminates through an activity diagram.

Flow of Event  
**Abbreviation: none**  
Source: EFFORTS D.3.1.1 v.1.0, p.18

Text to describe the events normally happens in a use case.

Fork Node  
**Abbreviation: none**  
Source: Bennett, Skelton and Lunn (2005), p.241

A notation showing parallelism where there is one entry activity edge and two or more exit activity edges.
**Function View**

**Abbreviation: none**  
**Source:** CIMOSA Glossary (1996), p.9

A view describing the functional structure, the functionality, and behaviour (i.e. rules which define the process sequence or flow of action and control within an enterprise) as well as the inputs and outputs of the underlying functions required to satisfy the requirements of an enterprise.

**Generation of Views**

**Abbreviation: none**  
**Source:** N.A

Please see View Dimension

**Generic Building Block**

**Abbreviation: none**  
**Source:** CIMOSA Glossary (1995), p.9

Generalisation of all building blocks (Types, Partial, Particular) instantiated from it. It contains a minimum set of attributes common to those building blocks.

**Genericity Dimension**

**Abbreviation: none**  
**Source:** ESPRIT Consortium AMICE (1993), p.20

A dimension, in CIMOSA Modelling Framework, concerning with the degree of particularisation (It differentiates between Reference Architecture and Particular Architecture.)

**Generic Level**

**Abbreviation: none**  
**Source:** CIMOSA Glossary (1996), p.10

A sub-division of the CIMOSA Reference Architecture which defines the full reference set of CIMOSA Generic Building Blocks and Building Block Types, and their rules as well as constrains for use in creating models at the Partial Level and Particular Level via the process modelling

**Implementation Description**

**Model**

**Abbreviation: none**  
**Source:** CIMOSA Glossary (1996), p.10 and adapted by EFFORTS PMB (2006)

A model containing description of all business implementations (functional, information, resource and organisational) of a system including exception handling conditions. It derives from the design specification model adding implementation
relevant information, capturing deviation from the system design, and passing the validation specification model as the test bed or pilot station.

**Include**

Abbreviation: none  
Source: Bennett, Skelton and Lunn (2005), p.29

A type of use case relationship used where one use case imports the behaviour of the included use case which means the use case that includes another use cases is typically not complete on its own.

**Information**

Abbreviation: none  

Meaningful interpretation and correlation of some aggregation of data in order to allow one to make decisions or to communicate.

**Information View**

Abbreviation: none  

A view describing the information (data, messages, etc.) and information structure required by and in support of the functions defined in the Function View.

**Input**

Abbreviation: none  
Source: EFFORTS D.3.1.1 v.1.0, p.18

Resource(s) to enable the action in the use case. It can be information, materials, human forces, etc.

**Instance**

Abbreviation: none  

User-defined individual of Generic Building Block or Building Block Type. The characteristics of which have been partially or fully defined.

**Instantiation (of Building Blocks)**

Abbreviation: none  
Source: N.A.

See Genericity Dimension

**Interoperability**

Abbreviation: none  
Source: ISO/IEC 2382-01

The capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the users to have little or no knowledge of the unique characteristics of those units.
The International Ship and Port Facility Security Code (ISPS-Code)

A comprehensive set of measures to enhance the security of ships and port facilities, developed in response to the perceived threats to ships and port facilities in the wake of the 9/11 attacks in the United States.

The purpose of the Code is to provide a standardised, consistent framework for evaluating risk, enabling Governments to offset changes in threat with changes in vulnerability for ships and port facilities through determination of appropriate security levels and corresponding security measures.

J

Join Node

A notation showing parallelism where there are multiple entry activity edges and only one exit activity edge.

K

Key Process

A vital process to ensure system functionality and incorporates activities, inputs/outputs, and interfaces required within a system or sub-system.

L

Life Cycle

Definition of the model and system life cycle phase (Requirement Definition, Design Specification, Validation Specification and Implementation).

M

Meta-Model Level

See Generic Level.
Merge Node

Abbreviation: none  
Source: Bennett, Skelton and Lunn (2005), p.240
A notation indicating where the alternatives flow together

Modelling Dimension

Abbreviation: none  
Source: ESPRIT Consortium AMICE (1993), p. 20
A dimension in CIMOSA Modelling Framework which provides the modelling support for the system or work life cycle starting from requirements to implementation

Modelling Level

Abbreviation: none  
Level of abstraction corresponding to the main life cycle phases of enterprise model development. It contains Generic Building Block and Building Block Types at the Generic Level and models at the Partial and Particular Level

Modelling View

Abbreviation: none  
Abstraction viewpoint of one total model which emphasises some particular aspects of the model and disregards others for ease of analysis by a set of constructs

Monitoring and Control Model

Abbreviation: none  
Source: EFFORTS D 3.1.1 v.1.0, p.8
Quality assurance model of process modelling with permanent monitoring and control of activities accomplished in previous derivations of model. This will enable recognition of problems which were not yet discovered in the process and ability to revert to the relevant derivation to make the modification or adjustments of the problems according to the objectives of process.

O

Object

Abbreviation: none  
Source: N.A
See Instance

Object Level

Abbreviation: none  
Source: N.A
See Particular Level

Objectives and Concept Definition

Abbreviation: none  
Source: EFFORTS D 3.1.1 v.1.0, p.10
Activity of defining the objectives of process analysis or domain as well as identification of views and roles relevant to the process.
Open Source Architecture for Computer Integrated Manufacturing

Abbreviation: CIMOSA 
Source: CIMOSA (1995)

A pre-normative development which provides a framework for process modelling based on the system life cycle concept together with a modelling language and definitions of a methodology and supporting technologies.

Organisational View

Abbreviation: none 
Source: CIMOSA Glossary (1996), p.15

A View describing an organisational structure of a group of enterprises or an enterprise, responsible and authorised for process, functions, control, information, and resources as defined in Function, Information and Resource View.

Output

Abbreviation: none 
Source: EFFORTS D 3.1.1 v.1.0, p.18

Expected result(s) from the use case.

P

Partial Level

Abbreviation: none 
Source: CIMOSA Glossary (1996), p.15

A sub-division of the CIMOSA Reference Architecture which defines a reference catalogue of partially instantiated models and the rules as well as constraints for their use in creating a model at the Particular Level.

Partial Model

Abbreviation: none 
Source: CIMOSA Glossary (1996), p.15

A model representing similar characteristics for common parts of manufacturing enterprises. It is a set of partially instantiated Generic Building Block or Building Block Types which are related to each other. It includes configuration rules defining wherein the combination of constructs a Partial Model is allowed to contain.

Particular Architecture

Abbreviation: none 
Source: ESPRIT Consortium AMICE (1993), p.20

A model architecture which serves the use of a specific case in process modelling and is not reusable for other models.

Particular Level

Abbreviation: none 
Source: CIMOSA Glossary (1996), p.16

A fully instantiated model of an enterprise or organisation adapted in CIMOSA.
Modelling Framework and created either from the Generic or Partial Level in process modelling

**Post-Condition**  
Abbreviation: none  
Source: EFFORTS D 3.1.1 v.1.0, p.15  
Text to define the significant condition(s) to check whether the use case is realised

**Pre-Condition**  
Abbreviation: none  
Source: EFFORTS D 3.1.1 v.1.0, p.15  
Text to define the significant condition(s) to enable action(s) in the use case

**Problem Domain**  
Abbreviation: none  
A domain where the parameters defining the boundaries of the domain and sufficient mappings into a set of ranges including itself are not well enough understood to provide a systematic description of the domain

**Process**  
Abbreviation: none  
A naturally occurring or designed sequence of changes of properties / attributes of a system or object. In technical perspective, process refers to operations or events which produce some outcomes

**Reference Architecture**  
Abbreviation: none  
Source: ESPRIT Consortium AMICE (1993), p.20  
A model architecture in CIMOSA Modelling Framework which resembles a catalogue of reusable building blocks containing generic and partial building blocks applicable to specific needs

**Relationship**  
Abbreviation: none  
Source: N.A.  
See Association

**Requirement Definition Model**  
Abbreviation: none  
Source: CIMOSA Glossary (1996), p.17  
Consistent instantiation of generic building blocks and/or building block types and/or partial models for requirement definition in order to express the business requirement of a particular enterprise or organisation
**Resource**

A capability set characterising the functional, object related, performance and operational attributes of the resource

**Resource View**

A view describing the required capabilities, the resources and their structure as well as location and the capabilities provided by the resources of the organisation which performs the functions defined in the Function View

**Stakeholder**

A person or organization that has a legitimate interest in a project or entity. Stakeholders in a company may include shareholders, directors, management, suppliers, government, employees, and the community.

**Unified Modelling Language**

A visual language with a specific notation and symbols expressing relationships, behaviours, workflows, communication, etc. It also can be applied to capture company organisation and business processes in order to develop customised enterprise software

**Use Case**

A technique for capturing the functions and requirements of a system in UML. It addresses the static view of a system and is important in organising and modelling the behaviours of a system.

**Use Case Specification**

A document with specific categories of information to describe all relevant aspects of a use case
V

Validation Model  

Abbreviation: none  
Source: EFFORTS D 3.1.1 v.1.0, p.8

An additional derivation of model in CIMOSA Modelling Framework adapted in EFFORTS. Its function serves as a test bed for the design specification model so that the feedback will be considered and correspondingly changed to the requirement defined in requirement definition model.

Vessel Traffic Service  

Abbreviation: VTS  
Source: IMO A.857(20)

A service implemented by a Competent Authority, designed to improve the safety and efficiency of vessel traffic and to protect the environment. The service should have the capability to interact with the traffic and to respond to traffic situations developing in the VTS area.

Very High Frequency  

Abbreviation: VHF  

The radio frequency range from 30 MHz to 300 MHz. Frequencies immediately below VHF is HF, and the next higher frequencies are known as Ultra high frequency (UHF). VHF is also commonly used for terrestrial navigation systems, Marine Communication, and aircraft communications.

View Dimension  

Abbreviation: none  
Source: ESPRIT Consortium AMICE (1993), p.20

A dimension – in CIMOSA Modelling Framework – which offers the users to work with partial-models representing different aspects of the enterprises: function, information (communication), resource and organisation.
1 Introduction

The integrated project EFFORTS aims at the improvement of the competitiveness of European Seaports by increased efficiency of the port processes and enhanced range of services provided by the ports. During specification of the EFFORTS objectives it became evident that a deep understanding of the underlying port processes is needed. Due to the large scope and size of the project it is important to have a uniform approach for process modelling in order to ensure consistency between the different work packages of the project.

This deliverable contains results of work package 3.1 ”Port Processes and Ship-Port Interface” within the framework of the RTD project EFFORTS. The objectives of this work package are two-folded:

1. Provide a methodology to conduct process modelling which is customised to port operations
2. Develop a ‘process ontology’ describing the top-level processes relevant within the scope of EFFORTS

The scope of this paper is to describe the methodology of process modelling applied within EFFORTS. Furthermore, it serves as a guideline to create a process model within the work packages of the project and beyond.

1.1 Purpose of this handbook

The integrated project EFFORTS consists of 3 sub-projects covering the entire scope of port operations. One of the biggest challenges is to ensure consistency within the project:

- avoid duplication of work,
- benefit from synergies between work packages,
- keep track of integration activities.

The purpose of this handbook is to define a common practice how to conduct process modelling within EFFORTS. In this sense it serves as a guideline to carry out modelling of port processes. Furthermore, it forms the basis for quality assurance of process modelling.

Another important aspect is to define a common terminology and to provide an efficient, easy-to-understand, consistent and comprehensive method to deal with port processes within EFFORTS related to both the ‘problem domain’ (i.e. port operations itself) as well as process modelling.
1.2 Why process modelling?
In general, there are three main reasons for process modelling:

- improve understanding of the subject,
- handle complexity of the problem domain,
- improve communication within the project.

**Improve understanding:** Process modelling within EFFORTS is mainly used to collect, understand and structure information during the analysis phase (fig. 1.1). Modelling is always an iterative process – thus, the result of a previous modelling step gives input to the following step. This leads to continued refinement of the model as illustrated in figure 1. The result is a complete picture of the subject modelled. To achieve this, a structured approach - which describes the subject by means of different 'views' covering different aspects of the problem domain e.g. stakeholders, functions, information exchanged, communication - is used.

**Handle complexity:** 'Complexity' in the sense of this document is given by the number of stakeholders, the number of processes and the amount of interactions between them. One of the main tasks of process modelling is to structure a complex system, i.e. to partition the system into smaller units with distinct interfaces between these units.

**Improve communication:** One of the biggest challenges within a large integrated project like EFFORTS is to gain a common understanding of the problem domain elaborated. A common process model serves as a 'map' to locate the different sub-projects and work packages within EFFORTS. Furthermore, interfaces between the different work packages can be identified clearly. Moreover, the definition of terms is a part of the process model. A unified language will help better communication among the partners within the project.

---

Figure 1.1: Process modelling meta-model

**Document Code:** EFFORTS-WP3.1.1-DEL-20061231-ISSUS-FINAL V2.0
**Date:** [31/12/2006]
1.3 The EFFORTS Roadmap – a procedure for process modelling

As already mentioned - the main objective of this handbook is to explain the methodology for process modelling to be used within EFFORTS as well as the language and tools supporting this methodology. The application of this methodology within EFFORTS will simplify the modelling tasks within the project, this making work more efficient and minimizing the workload needed for this task. To reach this goal, not only the methodology itself but also the application of this methodology needs to be defined clearly. The guideline how to apply the methodology is called ‘roadmap’; it comprises the following aspects:

- Definition of the sequence of activities to be performed during process modelling
  → What will be done?
- Definition of the role responsibilities within this procedure
  → Who does it?
- Determination of deliverables and milestones
  → What are the results?

Section 5 explains the EFFORTS roadmap for process modelling in detail.

1.4 How is the process modelling handbook structured?

The approach for process modelling defined in this handbook is based on the architecture defined by CIMOSA (Open Source Architecture for Computer Integrated Manufacturing). CIMOSA forms the basis of the European Pre-Standard for enterprise modelling, CEN/ISO 19439. The CIMOSA architecture is described in section 2 of this document. As language for process modelling, UML (unified modelling language) has been selected. This is a non-proprietary specification language for object modelling defined and hosted by the Object Management Group (OMG). Within EFFORTS, a customised subset of UML is used. Section 3 contains a detailed description of the UML elements used within the context of EFFORTS.

The document concludes with an example how process modelling is carried out according to this guideline. By means of modelling the process “VTS” (Vessel Traffic Services”), the steps to be carried out during process modelling are explained. Furthermore, the deliverables to be created for each step are described in detail.
2 Process modelling architecture - CIMOSA

If a shipbuilding engineer needs a blueprint to build a ship, a process modeller will require a modelling architecture as a basis what to be captured and described in the processes. In EFFORTS, WP 3.1 aims at searching for a possibility to capture the functional aspects as well as the business processes in one model.

As a result the architecture proposed by CIMOSA has been selected as the basis for process modelling in EFFORTS. The details of this architecture are explained in this section.

2.1 What is CIMOSA?

CIMOSA (Open Source Architecture for Computer Integrated Manufacturing) is an ESPRIT\(^1\) supported pre-normative development which provides a framework for process modelling based on the system life cycle concept together with a modelling language and definitions of a methodology and supporting technologies.

In the global business nowadays issues like change management, flexibility, or enterprise integration are not avoidable. Each business tries to get rid of inefficiencies and foster productivity in its organisation. Therefore, the business processes must be analysed and described according to the business and stakeholders’ requirements. Without any exception, increasing the competitiveness and the effectiveness among the European ports is the main task to be accomplished.

CIMOSA was designed to satisfy these issues and provides three inter-related concepts:

1) Modelling Framework (Reference Architecture, Particular Architecture and Enterprise Model)

2) System Life Cycle and Environments (Engineering and Operation)

3) Integrating Infrastructure

Since EFFORTS does not adopt all elements of CIMOSA, only the applicable parts will be described in this handbook. These components include

- modelling framework – reference architecture (views and hierarchies),
- terminology and
- interoperability concepts.

\(^1\) ESPRIT, the information technologies (IT) programme, is an integrated programme of industrial R&D projects and technology take-up measures. It is managed by the Directorate General for Industry of the European Commission. (Cordis, 10.11.2006)

Document Code: EFFORTS-WP3.1.1-DEL-20061231-ISSUS-FINAL V2.0

Date: [31/12/2006]
2.1.1 CIMOSA Modelling Framework

The CIMOSA Modelling Framework provides the necessary guidance to enable parties involved to model the enterprise and its processes in a coherent way. When modelling an enterprise or process there are many aspects, hierarchies and viewpoints to be examined that cannot be structured in a one-dimensional framework.\(^2\) Therefore, CIMOSA is developed to eliminate this weakness of previous models.

CIMOSA identifies a three-dimensional framework offering the ability to model different aspects and views of an enterprise.

From Figure 2.1, three dimensions encompass\(^3\):

- **The genericity dimension – so-called Instantiation of Building Blocks** - is concerned with the degree of particularisation. This dimension differentiates between Reference Architecture and Particular Architecture.
  - *Reference Architecture* resembles a catalogue of reusable building blocks which contains generic and partial building blocks applicable to specific needs.
  - *Particular Architecture* serves the use of a specific case in process modelling which is not reusable for other models.

- **The modelling dimension – so-called Derivation of Models** - provides the modelling support for the system or work life cycle starting from requirements to implementation. At this point, EFFORTS will only include Requirements and

\(^2\) Idem
\(^3\) Idem, p.21
Design Stage into consideration. The implementation is not in the scope of the project. However, as the demonstration of tools developed in EFFORTS should take place the Implementation Model will be complemented by a Validation Model as an extra level.

- The view dimension – so-called Generation of Views – offers the users to work with partial-models representing different aspects of the enterprises: function, information, resource and organisation.

2.1.2 Terminology
The second component derived from CIMOSA to be used in EFFORTS is terminology with respect to all aspects of process modelling. Prior to start modelling any process all relevant definitions must be clarified. According to this basis, on the one hand all modellers know what to include and exclude in their models. On the other hand, the readers can understand the process in a unified way.

The terminology applicable in EFFORTS is presented in the part of “abbreviation list and glossary” at the beginning of this document.

2.1.3 Interoperability
This term is normally used in the context of Information Technology and means the capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the users to have little or no knowledge of the unique characteristics of those units.⁴

There is another common term so-called Business Process Interoperability (BPI). It is a state that exists when a business process can meet a specific objective automatically utilising essential human labour only. Typically, BPI is present when a process conforms to standards that enable it to achieve its objective regardless of ownership, location, make, version or design of the computer systems used.⁵

Transferring the CIMOSA approach into the context of EFFORTS, interoperability means the capability to communicate and connect between each view and system in the modelling framework within the EFFORTS processes concerned where the users do not require having expertise in a modelling language.

2.2 From modelling architecture to language
A modelling language defines entities and relationships between them to describe and represent processes which are modelled according to the modelling architecture selected.

A number of languages have been evaluated applied within the areas of software development and design. The most appropriate was the object-oriented-approach. As a result, UML (Unified Modelling Language) has been selected.

⁴ ISO/IEC 2382-01
⁵ Wikipedia (20.10.2006)
There were several modelling tools which have been considered at this stage. The last three candidates were selected: Event Process Chain (EPC), CIMOSA Enterprise Modelling and Unified Modelling Languages (UML).

A value benefit analysis was conducted to ensure the objectivity of the tool selection and to maximise the benefit for process modelling in EFFORTS. The evaluation result is illustrated in table below.

<table>
<thead>
<tr>
<th>Evaluating criteria</th>
<th>UML</th>
<th>EPC</th>
<th>CIMOSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understandability</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Ability for hierarchical views</td>
<td>0</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Tool availability</td>
<td>+</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>Extensibility</td>
<td>+</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Standard</td>
<td>+</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Ability to model different views</td>
<td>+</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>Ability for eDocument generation</td>
<td>+</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Adaptability to business case</td>
<td>+</td>
<td>-</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 2.1: Evaluation of Tool Selection in EFFORTS**

The above table shows that UML can fulfil almost all significant criteria for EFFORTS’ requirement and therefore it was selected as the modelling language.
Figure 2.2 demonstrates the top-down approach to adapt CIMOSA Modelling Framework as the architecture of EFFORTS and the communication method by use of UML as the modelling language.

**Figure 2.2: From Modelling Architecture to Language**

To transfer this model in EFFORTS, some modifications have been made.

1. The two global levels to define the process domain and its objectives as well as its concept have been added. These are
   a. **Domain Identification** which includes to define domain name and its global characteristics of problem
   b. **Objectives and Concept Definition** which encompass the objectives of process analysis as well as identification of views and roles relevant to the process.

2. In EFFORTS the implementation level is not a part of the project result. Before the implementation could be started, the validation of the design model should be executed and it is concerned as an important part of EFFORTS. To make this stage explicitly, the final CIMOSA Model adapted to EFFORTS' requirements will show an additional derivation of model – the so-called **Validation Model**. Its functions serve as a test bed for the design specification model done in the previous level so that the feedback – both positive and negative – will be considered and correspondingly changed to the requirements defined in the first level.

3. Quality assurance is a very significant topic in any process analysis. This means after and during implementation, monitoring and control must be done permanently and consecutively. These activities help recognise some problem areas in the processes which was not discovered by the process analysis. Therefore, another modelling level – so-called **Monitoring and Control Model** - is added to this CIMOSA-EFFORTS Model. And since this is a permanent activity, the
graphic shows its circle as a loop which can be reverted to each derivation of model to enable modification to the process global objectives.

4. Moreover, to make the terminology more comprehensive among the project partners and practitioners, the terms used in original CIMOSA have been renamed as following:
   a. Generation of Views or View dimension – now so-called **Architecture Views**
   b. Information view – now so-called **Communication view** which include data, messages, and all items relating to communication aspects.
   c. Instantiation of Building Block – now so-called **Instantiation (Object-oriented Approach)**. This will help the understanding that this architecture is strongly related to object-oriented approach which is the significant reason for selection of UML as the modelling language.
   d. Relating to the previous point, the generic / partial / and particular level are also renamed to **Meta-Model / Class / and Object level** in order to link the relationship to an object-oriented approach. However, the meanings of these terms remain unchanged.

The modified model can be illustrated as below.

**Figure 2.3: CIMOSA Approach in EFFORTS**
The complete CIMOSA-EFFORTS Model will enable the process analysis within the project in efficient and sustainable ways.

According to this model, an example related to EFFORTS is provided in order to make this concept comprehensive. At the first stage of the project, identifying domain of the process is comparable to scope of the Mind Map and naming the process domains as well as defining their objectives and concepts. Afterwards, the modelling activities will be determined at the top of the model derivation, namely Requirement Definition Model. This means that the ports’ needs must be provided and formulated as requirements for modelling in business and technical aspects.

Since CIMOSA-Model is multi-dimensional, all dimensional elements must be taken into account of modelling. In our example, VTS is defined as the process domain with assumption that only functional view is concern for modelling. Followings demonstrate how each sub-cube or combination of three-dimensional model works (see also Figure 2.3):

**Sub-Cube 1: Meta-Model Level, Functional View, Requirement Definition Model**

For this sub-cube, the meta-model applied in modelling VTS process is the concept of modelling with UML. This will help illustrate the functions and activities taking place in this process with the requirements of finding out the inefficient or redundant actions.

**Sub-Cube 2: Class Level, Functional View, Requirement Definition Model**

The sub-cube with class level of instantiation illustrates the VTS process which will allow reusability of process contents – in this case VTS process in general. The activities or functions are similar in every port in the world or a specific region. Functional view and requirements are same as in sub-cube 1.

**Sub-Cube 3: Object Level, Functional View, Requirement Definition Model**

This sub-cube shows the VTS process which only takes place in a specific port, for instance, VTS Hamburg. The content of function and activity is not reusable for other ports. The last two dimensional element of the model remain unchanged like in sub-cube 3.

After knowing the basic architecture of process modelling, the next chapter will explain how the process model can be presented by a modelling language, i.e. UML. The details of use of UML in EFFORTS will be explained in the next section.
3 Process modelling language - UML

3.1 What is UML?

UML or Unified Modelling Language is one of the most famous modelling language standards being used in many industries nowadays. UML is a visual language with a specific notation and symbols expressing relationships, behaviours, workflows, communications, etc. It also can be applied to capture company organisation and business processes in order to develop customised enterprise software. Moreover, UML also supports in specifying, constructing, and documenting of a software development process.

This modelling language has been certified by Object Management Group (OMG) as a standard in 1997 and the first launch was UML v1.1. UML has been revised and refined several times leading to the current 2.1 release.

The main goal of UML is to be a general-purpose modelling language or a common language which can be understood not only within a computer community but also by the business.

3.2 UML diagrams in EFFORTS

A diagram is the graphical presentation of a set of elements, most often provided as a connected graph vertices (entities) and arcs (relationships). The diagram must be drawn to visualise a system from different perspectives.

In UML 2.0 and 2.1 there are 13 diagram types defined which are used for different specific purposes as shown in Table 3.1 below:

<table>
<thead>
<tr>
<th>Diagram Type</th>
<th>Purposes of the Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Use Case</td>
<td>To show how users interact with a system</td>
</tr>
<tr>
<td>2. Activity</td>
<td>To illustrate the procedural and parallel behaviours</td>
</tr>
<tr>
<td>3. Class</td>
<td>To classify the classes, features of each object and its relationship</td>
</tr>
<tr>
<td>4. Communication</td>
<td>To display the interaction between objects as well as the emphasis between their links</td>
</tr>
<tr>
<td>5. Component</td>
<td>To identify structure and connection of components</td>
</tr>
<tr>
<td>6. Composite structure</td>
<td>To specify runtime decomposition of a class</td>
</tr>
<tr>
<td>7. Deployment</td>
<td>To determine deployment of artefacts to nodes</td>
</tr>
<tr>
<td>8. Interaction overview</td>
<td>To demonstrate the mix of sequence and activity diagram</td>
</tr>
</tbody>
</table>

6 Pilone (2005), p.1
7 Rumbaugh, Jacobson and Booch (1999), p.14
8 Pilone (2005), p.2
9 Rumbaugh, Jacobson and Booch (1999), p.8
9. Object
To show examples of configurations of instances

10. Package
To group model elements together; packages can be nested

11. Sequence
To demonstrate the interaction between objects and the emphasis on sequence

12. State machine
To show how events change an object over its life cycle in a specific process

13. Timing
To illustrate the interaction between objects and its emphasis on timing aspect

<table>
<thead>
<tr>
<th>Table 3.1: Official Diagram Types of the UML 2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Source: Fowler (2004), p.11)</td>
</tr>
</tbody>
</table>

However, only three types of UML diagrams will be applied in EFFORTS: Use Case Diagram, Activity Diagram and Class Diagram (table marked in blue).
Beside these diagrams, a description template, the so-called Use Case Specification, will be also applied.

### 3.2.1 Use case diagram

Use cases are a technique for capturing the functional requirements of a system in UML. This kind of diagram addresses the static use case view of a system. It is especially important in organising and modelling the behaviours of a system.\(^{10}\)

The components of a use case include 1) functionality (use case); 2) persons and systems involved to the functionality (actors); 3) system or situation; and 4) connecting line (association).\(^{11}\)

The visual use case diagram can be demonstrated as shown in the Figure 3.1.

\(^{10}\) Rumbaugh, Jacobson and Booch (1999), p.25
\(^{11}\) Pilone (2005), p.77
In a simple use case (as shown in the figure 3.1) consists of the actors involved with the system and its related use cases.

Each use case represents functionality for a system. It must be named with a clear verb plus a noun (object) or substantive to capture the action of each function, for instance, make offers, withdraw money, etc. The relationship between the use cases and the relevant actors are bound with associations.

Moreover, a use case diagram offers the possibility to demonstrate dependencies between use cases. These relations are called “include” and “extend”.

“Include” is used when one use case imports the behaviour of the included use case. In other words, the use case that includes another use case is typically not complete on its own. The included functionality is not considered optional; it is factored out simply to allow for reuse in other use cases.\(^\text{12}\)

The *include* relationship is drawn as an open arrow with a dashed line that points towards the use case that is being included. The stereotype «include» is written alongside the relationship arrow.\(^\text{13}\) (See figure 3.2)

An example of services offered in a bank can be demonstrated by use case diagram with include relationships as shown in figure 3.3

---

\(^{12}\) Pilone (2005), p.82

\(^{13}\) Bennett, Skelton and Lunn (2005), p.29

\(^{14}\) Idem

Document Code: EFFORTS-WP3.1.1-DEL-20061231-ISSUS-FINAL V2.0

Date: [31/12/2006]
Unlike the include relationship, “extend” is used where one use case may optionally be extended by the functionality in another use case. Similarly, to illustrate the extend relationship, the same notation as use case inclusion is used. The only difference lies at the stereotype «extend» which is written alongside the relationship. (See fig. 2.3)

For example, a use case “process payment” can be demonstrated by use case diagram with extend relationship as shown in Figure 3.5

It has to be pointed out that a use case diagram shows only the functionalities and the involved parties relevant to the system in view. But it could not demonstrate the flow of works or processes. Therefore, the activity diagram will take this role to demonstrate the process flow.

---

15 Roques (2004), p.33
16 Idem
17 Idem, p.30
3.2.2 Use case specification

Since a use case diagram can deliver only specific details of a system, the use case specification plays a significant role to describe the use case in detail. The use case specification is a document with specific categories of information to describe all relevant aspects of a use case. There is no standard of this document given by OMG. However, some certain information categories must be provided in the use case specification to enable modelling in other kinds of diagrams such as activity diagrams.

In EFFORTS a template of use case specification has been created so that all documentations concerning use case modelling can be provided in the same format. The template of the use case specification is given in Annex B of this handbook. In short, the required information to this document is listed below:

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
<th>Information formatted as</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. No. of use case</td>
<td>Number to identify the use case</td>
<td></td>
</tr>
<tr>
<td>2. Name of use case</td>
<td>Name to identify the use case</td>
<td>Verb(s)</td>
</tr>
<tr>
<td>3. Short description</td>
<td>Text to shortly explain the use case</td>
<td>Flow text of roughly 25 words</td>
</tr>
<tr>
<td>4. Actor(s)</td>
<td>Involved party/parties of the use case</td>
<td>Numbered substantive(s) or noun(s)</td>
</tr>
<tr>
<td>5. Pre-condition(s)</td>
<td>Text to define the significant condition(s) to enable action(s)</td>
<td>Numbered sentence(s)</td>
</tr>
<tr>
<td>6. Input</td>
<td>Resource(s) to enable the action(s) in the use case</td>
<td>Numbered substantive(s) or noun(s)</td>
</tr>
<tr>
<td>7. Flow of events (standard flow)</td>
<td>Text to describe what normally happens in the use case</td>
<td>Numbered “if, then” sentence(s) or specific action description(s)</td>
</tr>
<tr>
<td>8. Post-condition(s)</td>
<td>Text to define the significant condition(s) to check whether the use case is realised</td>
<td>Numbered sentence(s)</td>
</tr>
<tr>
<td>9. Output</td>
<td>Expected result(s) from the use case</td>
<td>Numbered substantive(s) or noun(s)</td>
</tr>
<tr>
<td>10. Extension points</td>
<td>Text to define the alternative flows of event and also the open issue(s) which will not be captured by this use case</td>
<td>Numbered sentence(s)</td>
</tr>
</tbody>
</table>

Table 3.2: Use case specification

Document Code: EFFORTS-WP3.1.1-DEL-20061231-ISSUS-FINAL V2.0
Date: [31/12/2006]
3.2.3 Activity diagram

To model a process, it is necessary to illustrate the execution and flow of the behaviour of a system. In UML the activity diagram is created to serve this function.

An activity diagram is a technique to describe procedural logic, business process, and work flow with specific notations. It seems functionally alike to flowcharts, however, the main difference between them is that an activity diagram supports parallel activities.\(^{18}\)

This type of diagram can be used throughout the project, from business analysis through program design.\(^{19}\) Normally it is used to describe the process along the use case diagrams. However, its uses are still numerous and can be concluded below\(^{20}\):

- To model business workflow
- To identify candidate use cases, thought the examination of business workflows
- To identify pre- and post-conditions for use cases
- To model workflows between use cases
- To model workflows within use cases
- To model complicated workflows in operations on objects
- To model complex activities in a high-level activity diagram in more detail

The basic activity diagram only describes what happens, but it does not depict who does what. Therefore, an extended activity diagram is selected in EFFORTS as an option to enable visibility of this information.

An activity diagram consists of many elements which could be illustrated and explained below\(^{21}\):

<table>
<thead>
<tr>
<th>Elements of Activity Diagram</th>
<th>Notation</th>
<th>Functions / Uses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Actions with extension(^ {22})</td>
<td><img src="image" alt="Actor:" /></td>
<td>Indicate the unit of work that needs to be carried out with additional information of actors involved in this action</td>
</tr>
<tr>
<td>2) Activity initial nodes</td>
<td><img src="image" alt=" " /></td>
<td>Indicate where the activity begins</td>
</tr>
</tbody>
</table>

---

\(^{18}\) Fowler (2004), p. 117  
\(^{19}\) Bennett, Skelton and Lunn (2005), p.236  
\(^{20}\) Idem  
\(^{21}\) Idem, p.237-242  
\(^{22}\) The extension allows us to put the information involved to the action which we would like to demonstrate or describe in the model such as actors, input, output, pre-condition, post-condition, etc.
3) Activity final nodes

Indicate where the entire activity terminates

4) Flow final nodes

Indicate where the path or flow terminates through an activity diagram

4) Activity edges

Indicate the movements between actions

5) Decision nodes

Indicate where the flow on exit from an action may go in alternative directions, depending on a condition

6) Merge nodes

Indicate where the alternatives flows together

7) Fork and join nodes

Indicate a number of actions to run in parallel
(An activity edge can be split into multiple paths and multiple paths combined into a single activity edge)

Table 3.3: Notations of activity diagram used in EFFORTS

Note: There is one main difference between activity final nodes and flow final nodes. Any token arriving at activity final node the whole activity will be terminated without results, while the flow final node terminates only the executions directly involve to it. ²³

3.2.4 Class diagram

A basis of CIMOSA architecture can be executed in an object-oriented modelling language like UML. To repeat, in CIMOSA there are three different instantiation of building blocks. They could be interpreted in the object-oriented modelling languages as follows:

- Generic Level ➔ Meta-Model Level
- Partial Level ➔ Class Level
- Particular Level ➔ Object Level

The first phase of EFFORTS aims at gathering information and requirements of the port needs as well as the description of existing port processes. The system design or elements of software design are not yet included in this version of the handbook.

²³ Pilone (2005), p.117
Since the concept of classes can represent the generic level in the CIMOSA architecture, we therefore cannot ignore the class concept and its importance in UML within our specific modelling description. This will be useful for the design of the model in the later phase of the project.

Therefore, in the first phase of the project, only two important elements from class diagram will be executed, namely class and its attributes. The associations or relationship between classes as well as the operations will be included in the second phase.

A class represents a group of entities that have common properties and behaviour. For example, Toyota, Mercedes, BMW, VW and Honda are cars, so we can represent them using a class named “Car”.

We represent a class with a rectangular box divided into compartments where the relevant information will be filled in. The first compartment holds “the name of the class”, the second holds “attributes”, and the third is used for “operations”.

![Figure 3.6: Single Class Diagram](image)

Attributes encompass the details or characteristics of a class. If we raise the example of port authority, its attributes could be name, location, facilities, type of port, organisational structures, etc.

This section has explained the application of UML as modelling language to demonstrate the concepts specified within the CIMOSA architecture. The next section explains in detail how this could be applied in practice. As example the modelling of the process VTS has been selected.
4 Tools and programs used for process modelling

4.1 Paper and pencil
These basic tools will be the first one for a specific reason. Working on the project showed that it does not make sense to start with detailed applications while writing down first thoughts. It is better to keep these detailed applications on a separate piece of paper. This procedure helps figure out whether these thoughts can be realised the way it was planned or not. It is not only true for the beginning of this project but also throughout the whole project because there are always new thoughts and ideas being added or removed, new aspects being found and much more.

The way of working with paper and pen is not supposed to be a part of this section. It should just show that the usage of these “antique” tools can be an advantage for the whole work process.

4.2 Microsoft Visio with additional UML shapes

4.2.1 The program
Microsoft Visio is a drawing program with provided shapes that can help create business and technical diagrams that document and organise complex ideas, processes, and systems. Unlike the demonstration of text and numbers, the Visio diagrams are able to visualise and communicate information in the clear, concise and effective ways.

Visio is a program from the company ”Microsoft”. It is possible to order directly from Microsoft Website.

Here you have the opportunity to choose many different kinds of production versions. Besides this you still have the choice to receive the software through a distributor or a business, whose service you normally use for buying software.

26 http://www.microsoft.com/office/visio/prodinfo/overview.mspx
27 http://www.microsoft.com/office/visio/howtobuy/default.mspx

Document Code: EFFORTS-WP3.1.1-DEL-20061231-ISSUS-FINAL V2.0
Date: [31/12/2006]
Visio offers the user sets of different shapes for various diagram types. In the left field you will find these shapes or search for shapes you are looking for. Visio simply works with drag-and-drop function. Choose the shape you want to use, click on it with the left mouse button and move it over to your work space by holding the left mouse button. The symbols in your work space can be named, moved or changed in size or formatted somewhere else.

Like other drawing programs, we use the mouse to move an object or a symbol. However, moving those objects with precision, we have to click on the symbol with the mouse and mark it. Now the symbol can be moved with the arrow keys on your keyboard. For even more precise movements, keep holding the shift key while using the arrow keys.

How each symbol could be used in process modelling was explained in the section 3.4.2.2 The shapes

Visio offers in its basic configuration a variety of shapes. For this project solely the symbols for the use-case-diagram, the activity diagram and the class diagram are necessary. For the first two diagrams the project team makes in each case two templates available with the most required symbols. For the class diagram, the professional version of Visio makes templates available to meet the requirements of the project.
To use the offered shapes you have to choose in the file menu the point Shapes and there the point “Open Stencil”.

### 4.3 Mindjet MindManager

Mind Mapping is a powerful technique for quickly generating, capturing, and organizing ideas, tasks, and activities. Utilizing this technique, MindManager ensures success by empowering you to focus on the details, mitigate risks, and exploit opportunities. Mind Maps represent complex information in an organised, easy-to-understand visual format. Furthermore, they enable you to easily grasp connections, obstacles, and paths so you can quickly choose the best course of action.  

There is no necessity to buy MindJet MindManager since the export functions, which are included in the delivery, offer the possibility to all project members to review the results of the leading team on the EFFORTS-Homepage through an internet browser. Herewith a possibility was created to visually show project members in which responsible part of the project they are involved. Through clicking on the corresponding links of the several sub-menus the user can access further project levels.

4.4 Spreadsheet

For every Use Case a specification or description has to be written. The presentation of these specifications should be shown in a spreadsheet which makes it clearly laid out. Following image shows an example:

<table>
<thead>
<tr>
<th>Use Case No.</th>
<th>VTS.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Inform about arrival</td>
</tr>
<tr>
<td>Actors</td>
<td>Terminal Operation, Shipping Agent, Port VTS</td>
</tr>
<tr>
<td>Aim (Why is it done?) (Short Description)</td>
<td>Identify significant information for arrival of the vessel so that the Port Administration and Port VTS can prepare the navigational assistance</td>
</tr>
<tr>
<td>Pre-Condition</td>
<td>1. The vessel has been scheduled to call the port. 2. The vessel has reached the identified port territory line.</td>
</tr>
<tr>
<td>Required Input</td>
<td>Ship particulars, Berthing position, ETA/ETD, Cargo details, Pilotage Demand, Tugging Demand, Towing Demand, Crews</td>
</tr>
<tr>
<td>Flow of Events (What is done?)</td>
<td>Case Flow 1. Shipping Agent send information of the arriving vessel to Port Administration and Port VTS</td>
</tr>
<tr>
<td>Post-Condition</td>
<td>Port VTS are informed</td>
</tr>
<tr>
<td>Resulting Output</td>
<td>Arrival information</td>
</tr>
<tr>
<td>Extension Points</td>
<td>Alternative none</td>
</tr>
</tbody>
</table>

Table 4.1: Example of use case specification
There are several programs in which such a presentation is possible. The most well-known is probably Excel from Microsoft. But also Microsoft Word, openOffice, WordPerfect or other programs can also be the alternative to present the data in this way. The choice of program is dependent on the user because screenshots will be made of the finished specifications and uploaded to the server.
5 The EFFORTS Roadmap for process modelling

To make the methodology described in the previous sections usable within a project, a guideline how to apply this methodology is needed. Such a guideline is called roadmap. The purpose of the EFFORTS roadmap on process modelling is to give a clear definition of following aspects:

- Sequence of activities to be performed during the modelling process
- Roles and responsibilities during the modelling process
- Milestones and expected results (deliverables) per milestone

5.1 Process modelling – Procedure

This section gives a general overview of the activities to be carried out during process modelling in EFFORTS. In order to achieve high efficiency during the modelling process, these activities need to be distributed among the experts involved. Whereas roles and responsibilities are defined in section 5.2, the objective of this section is to define areas of activity which can be later assigned clearly to one certain role. A detailed description of all activities to be performed during the modelling process can be found in section 6.

The first activity area to be carried out is specification of the scope of the “system” to be analysed. This is most probably the most critical phase of the analysis process. During this phase the boundaries of the “system” to be analysed are specified. For instance, it will be clearly defined, what lies ‘within’ the scope of the modelling process and what is ‘outside’. Additionally, it needs to be agreed in this phase, which level of detailed is required. The level of detail is dependent of the application of the model. Furthermore, possible interfaces with the environment are defined within this phase. This phase is mainly driven by the work package (WP) analysts and business experts. Due to the criticality of this phase close co-operation with the methodology expert is required.

The second phase (activity area) of the modelling process aims in creating a first draft of the process specification. Actors and processes forming the system are defined by means of a first use case diagram (see section 4). The specification of the processes is done by means of use case specifications (section 4). It needs to be pointed out that the result of this phase shall be on a very general level. The objective is not to create a comprehensive and detailed picture of the entire system but to structure the object of investigation in order to get a common understanding and allow systematic refinement of the model in the following steps. This task is mainly carried out by the work package analyst in close co-operation with the business expert.

Objective of the third phase of the modelling process is to create a comprehensive and detailed model of the system in view. This task is mainly performed by the EFFORTS process modelling team in close co-operation with the business experts. Within this phase, the process model will be created on a level of detail as specified in phase 1 (see above). The documentation of the model is done with the different UML diagrams as described in section 3.
The roles involved in the modelling process as well as their responsibilities during the phases of the analysis are described briefly in the following section.

5.2 Roles and responsibilities

One of the challenges of process modelling is that expertise from very different fields is required:

- Project analysts with a deep knowledge of the project requirements
- Business experts
- Experts on the methodology for process modelling applied
- Experts on Quality assurance

The WP analyst is the point of entry within a work package requiring modelling activities. This role requires a deep knowledge of the WP requirements. The WP analyst co-ordinates all analysis activities with respect to his work package, in particular co-operation between business experts and modelling experts. The WP analyst is responsible for the ‘model specification’, i.e. definition of scope and level of detail of the model to be applied within the WP.

Business experts involved will be mainly industrial partners within the particular WP, i.e. port authorities, terminal operators etc. Their main task is to provide detailed business information and to monitor that the model complies with their understanding of the business processes. The team of business experts will approve the final process model from a business point of view.

The modelling experts give support during the modelling process. After providing a first draft of the model (draft use cases and use case specifications, see 5.1) the team of modelling experts will carry out further refinement of the process model as specified by the WP analyst.

The process quality manager approves the process model formally. This means that the process model complies with the rules specified within this handbook.

5.3 Milestones and deliverables

To structure the activities and achieve measurable results, the procedure of process modelling is divided into phases as outlined in 5.1. The end of each phase is defined by a milestone. The results of each phase are documented by clearly defined deliverables.

It is recommended to start the modelling activities with a kick-off meeting hosted by the EFFORTS process modelling team. The objectives of this kick off are as follows:

- Provide a common understanding of the EFFORTS process modelling procedure
- Give an overview of the underlying methodology and the tools applied for process modelling
- Agree on roles and responsibilities during the modelling procedure
The result of phase 1 of the modelling process is the definition of the scope of the model, documented by a mind map (see 4.3). This deliverable needs to be agreed between the WP analyst and the business experts. Formal acceptance by the process quality manager finalises this phase.

Drafting of the first process specification in phase 2 is mainly carried out between the WP analyst and the team of business experts. Support by the modelling experts can be accessed as appropriate. The deliverable of this phase is a first use case diagram of the system and the specification of the corresponding use cases. It is recommended to finalise this phase with a workshop to achieve a common understanding of the use cases between WP analyst, business experts and modelling experts.

Within phase 3 further refinement of the model is done by the modelling experts. The results are documented by the appropriate diagrams according to the unified modelling language (UML, see section 3). The refined model is discussed during a workshop between all responsible involved. The final model needs to approve from a business point of view by the business experts and formally by the process quality manager.
6 Process modelling step-by-step by example of Vessel Traffic Service (VTS) process

After becoming familiar with the fundamentals of CIMOSA and the UML notations as well as possible tools or programs used for modelling in the previous chapter, this section explains how process modelling can be done step-by-step in practice. It describes the transition from the modelling framework – CIMOSA into the modelling language - UML. In order to demonstrate this clearly, the modelling of the process Vessel Traffic Service (VTS) will be described exemplary.

By practising process modelling, the following steps should be carried out in sequence:

1. Set the scope of process
2. Gather first information
3. Draft the first process in global picture
4. Interview to get detail information
5. Create use case diagram
6. Create use case specification
7. Create activity diagram
8. Prepare class diagram (without associations and operations)
9. Check plausibility of diagrams and specifications with an expert
10. Review the diagrams

Note: Step 4 will not be implemented in case that the modeller is an expert and knows the process very well.

6.1 Set the scope of process

Similar to other activities, the first thing to be done is to set the scope of work. For modelling a process, this is a critical starting point. At this stage, the boundaries of the “problem domain” are defined. I.e. it will be specified clearly, what is part of the system (problem domain) to be modelled.

Within EFFORTS, so-called “scope maps” are used to specify the scope of a problem domain. A scope map is a diagram which groups related topics around a central keyword. Relations between topics are indicated by lines. The result is a network consisting of different branches with one common root (the central keyword). Topics near to the centre are general; depending on the distance from the centre they become more specific.

To set the scope of a process does not only mean to define the begin and end of the process. Moreover, it concerns all three dimensions of the CIMOSA modelling framework which must be covered by the specification of the process. In other words, the modeller should, at this stage, be able to answer following questions:

1) What process is to be described?
2) Which requirements must be fulfilled after the process has been modelled and described?
3) Why is the process modelling necessary for this selected process?
4) What is the begin and the end of the process?
5) To which abstract level (generic, partial or particular) should the process be modelled?
6) Which views should be concerned / presented in the process according to the requirements in 2)?

In the example of VTS process, we can answer the above-mentioned questions correspondingly.

1) The process to model is Vessel Traffic Service or VTS.
2) and 3) In EFFORTS, the main objectives of process modelling are
   - To find out the activities which could be done automatically by an existing technology in maritime research area
   - To find out the VTS activities which are redundantly done by other involved parties
   - To find out how the VTS process can be run more effectively
4) The beginning point of the process is when the vessel or its agents has informed VTS about the port call of the vessel. The ending point of the process is when the vessel arrives at the berth or terminal.
5) The level of abstraction will be kept generic.
6) At this stage the functional and organisational views are selected to present in the process description.

At the end, we can draft the scope of the process in a map as below. This must not be drawn in logical order. It aims at showing what to be considered as the scope of our process description.

![Figure 6.1: VTS – Scope Map](image-url)
6.2 Gather the first information

This step aims at accomplishing an overview of the process and its rough activities to be modelled. It could be skipped when you hold the experience of the profession in the area or process you are going to model.

If you are not the specialist in that field, the on-desk research is recommended to get an overview of the process. Following are the possible sources where you can get the first information to process description.

1. Traditional literature: This includes books which are normally obtained from libraries, bookshops, archives, etc.
2. Internet: This media seems to be the fastest and the most effective one. We can use the search engines - such as Google, Yahoo, Lycos, etc. - to find the organisations, institutes, or articles relevant to the process we would like to describe or model.
3. Magazines: Nowadays there are numbers of journals in the field of transport and logistics which sometimes include the practical operational articles by scripting the interview from the expert in that area. Moreover, we can also obtain changes or new information in the specific topic which probably pertinent to the process or may take effect to it in the future. This information is useful to help broaden the perspectives of how the process could be effectively optimised in the short- to middle-run-period.

The information found from on-desk research will be the basic for the next step, namely “drafting” the process.

6.3 Draft the first process in a global picture

There are some activities\textsuperscript{29} to be completed in this step. The information captured in the first step will be categorised according to the scope we have set at first.

**Firstly**, we have to **identify the active partners** who initiate or trigger the process. Meanwhile, the **passive partners** of the process will be defined. Active partners include the key persons who initiate the activity in the process. Meanwhile, passive partners are persons who receive the message or action from the active partner.

\textbf{Main question:}
- Who are the key persons or systems who are responsible for or involved in the process? (both active and passive)

From our example VTS, the answer includes VTS, Terminal Operator, Shipping Agents, Vessel, Tug Service and Pilot. The active and passive role of each partner cannot be concretely defined since some partners sometimes are active and sometimes are passive. The definite role must be identified to each activity in the process.

**Secondly**, after having the active and passive partners, we **identify the use cases** involved to the process. Here the order of the use cases is not an essential point.

\textsuperscript{29} Adapated from Oestereich, Weiss, Schröder, Weilkiens and Lenhard (2003), p.51-83
As a result from the on-desk research, we could summarise the use cases for VTS process as 1) inform about arrival of the vessel; 2) coordinate information during the whole services; 3) give pre-entry report; 4) give notifications; 5) give security reports; 6) issue deviation report; 7) issue incident report; 8) monitor traffic and support vessel; 9) acquire pilotage; 10) acquire towage; 11) check plausibility of the report; 12) issue position report; 13) monitor pilot admittance; 14) approve entry to port and departure from port; and 15) check and approve berthing entrance and position.

Next is to match the partners with the use cases we have identified in the first two steps. This is carried out by mean of Use Case Diagrams. Here we have to consider all use cases and identify the roles of the partners involved.

Main questions:
- What could be the activities or use cases occurring in the process?
- Which use cases are important in the view of the active partners?
- What could the partners expect as a result of a use case?
- What should each partner do to make the process complete?

As a result, we will have:

![Use Case Diagrams]

Document Code: EFFORTS-WP3.1.1-DEL-20061231-ISSUS-FINAL V2.0
Date: [31/12/2006]
The step four is to identify the key processes in the system VTS by grouping the use cases we had in from the third step. The criteria for grouping can be done by looking for the use cases which could belong together and represent the group as business processes. After that we give each group a name and classify them to key processes or supporting processes.

**Figure 5.2: Identifying use cases and their actors**

As a result, we have
After we have grouped the use cases, they must be sorted out as the key processes and supporting-processes. We will get result as follow:

<table>
<thead>
<tr>
<th>Key Processes</th>
<th>Supporting Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notify and Inform</td>
<td>Issue Reports</td>
</tr>
<tr>
<td>Monitor</td>
<td>Acquire Service</td>
</tr>
<tr>
<td></td>
<td>Approve and Check</td>
</tr>
</tbody>
</table>

### Figure 5.3: Group of use cases

6.4 Interview to get detailed information

Until now we have a global draft of how the VTS process should look like. However, some detailed and essential information is still omitted. Therefore, we need to contact the specialist in the profession of VTS to get that information. This stage will be done through an interview.

Here we have divided the interview in three sub-steps: 1) preparing the interview; 2) conducting the interview; and 3) post-editing the interview minute.
1. **Preparing the interview** includes some activities as following:
   a. Find out an appropriate expert as interviewee
   b. Look for a fellow to go with you for the interview
   c. Make a list of which information you are still missing or is not clear for you
   d. Prepare the process map you have drafted so far and bring it with you
   e. Bring a laptop or note block to make a minute
   f. Bring a recorder in case that you are allowed to record the interview
   g. Define the role between you and your buddy: who will conduct the interview and who will minute

2. **Conducting the interview** is the crucial part of the interview. There are specific activities you have to do and some more recommendations you can adapt.
   a. Ask the interviewee first whether you could record the interview. This will help you and your fellow know and be prepared of how precise your minutes will be
   b. One only asks and conducts the interview and another only makes the minutes
   c. Show the draft you have prepared and ask the interviewee to the specific point you have in your question list
   d. As the interview conductor, you can let the interviewee explain in detail. This will help you to get real information from the practice which will be useful to find out where the process still has a problem and where could be optimised. However, you have to be careful that his/her explanation must not be really out of the scope of the interview topic.

3. **Post-editing the interview minute** means that you will need to review on your minutes whether what your fellow has used the terminology appropriately and made the minutes with correct content. This can be done by a discussion round between your fellow and you. Importantly, you should also send the minute to the interviewee so that he/she can review and confirm the correctness of the details of the content in that minute.

**Some more recommendations:**

- It is good to pause once in a while to test your understanding, to practice the terminology and to give the interview partner a comforting feeling.
- Let the interview partner always define technical terms you do not understand. Do not hesitate to acknowledge your ignorance. The reason why you are there is to gain knowledge and to learn the terminology.
- Frequently you might be able to pose a question about a theme which you discovered in the answers of previous questions. Keep your eyes peeled for the opportunity to ask such like questions. The logic of business often arises out of these answers.
- When you get the feeling that a part of the process becomes complicated and too complex think about the possibility to outsource the complication as a separate process in business. Instead of trying to merge all processes into one single process you can easily shape that process and the resulting model becomes clearer.
- Try to get a feedback for your diagram from your interview partner. Include the changes he proposed to you.

(Source: Schmuller, 2003, p. 256)

The minutes for interview regarding the VTS Process can be found in the Annex C
6.5 Create use case diagram

From the basic about the use case diagram in 3.2.1, we will start drawing the use case diagram based on the information from on-desk research and the interview. Below are the steps to follow:

1. Start with the system and name it “VTS Process”
2. Draw the use cases we have listed
3. Try to match the use cases with their relevant actors
4. Look for the use case which could be an “extend” or “include” use case

As a result, the use case of VTS process would look like as below:
Figure 5.4: Use case diagram
6.6 Create use case specification

This step is about to draft description of the use cases. Here we will describe each use case with some short sentences in natural language in the template of use case specification explained in 3.2.2. Therefore, following information must be clearly defined:

1. the beginning and the end point of each use case as well as the pre-/post-conditions
2. The data input and output for each use case
3. The involved or relevant actors to each use case

**Main questions:**
- What is triggering each use case?
- What are the results of each use case?
- What are the inputs/outputs of each use case?
- Who are the relevant actors (active and passive) to each use case?

At this point we have the information from on-desk research and the interview which are categorised and defined into a specific type for the description of the use case. As a result, we will demonstrate drafting the use case specification of “Inform about arrival of the vessel” will be drafted as below:

<table>
<thead>
<tr>
<th>Process Domain</th>
<th>VTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case No.</td>
<td>VTS.1</td>
</tr>
<tr>
<td>Name of Use Case</td>
<td>Inform about arrival</td>
</tr>
</tbody>
</table>
| Actor(s)       | 1) Terminal Operator (active actor)  
                 2) Shipping Agent (active actor)  
                 3) Port VTS (passive actor) |
| Short Description | Identify significant information for arrival of the vessel so that the Port Administration and Port VTS can prepare the navigational assistance |
| Pre-Condition(s) | 1. The vessel has been scheduled to call the port.  
                         2. The vessel has reached the identified port territory line. |
| Required Input(s) | 1) Ship particular  
                           2) Berthing position  
                           3) ETA/ETD  
                           4) Cargo details  
                           5) Pilotage Demand  
                           6) Tugging Demand  
                           7) Towage Demand  
                           8) Crews |
| Flow of Event(s) | Basic Flow  
                           1. Shipping Agent send information of the arriving vessel to Port Administration and Port VTS |
| Post-Condition(s) | Port VTS are informed |
| Resulting output(s) | Arrival Information |
| Extension point(s) | None |

**Table 6.1: VTS Process – Use case specification**
Note: The use case specification for other use cases must be completely done in the same way as above.

6.7 Create activity diagram

We now arrive the main part of process description “creating an activity diagram”. Like in the use case diagram, here we start at drawing the system and also name it “VTS Process”.

At this stage, firstly, we have to put the processes and activities in the correct order since the diagram represents the flow of process. Each activity is bound to its involved actor as we have set in the scope to model only functional and organisational view. Secondly, we have to consider if at some points require a decision with “yes” or “no” and a parallelism of activities.

Attention: The notations used here must comply with the explanation in Chapter 3 so that the communication in the process will be unified.

According to the information we have got so far, we can describe the VTS process in the activity diagram as shown below:

![Activity Diagram of VTS Process](image-url)
Please note that what you have modelled at first, its result must not be the most correct one. In the process of process modelling, the modeller normally tries to model in various versions to get the most comprehensive one which is also close to the practice world.

6.8 Prepare class diagram

The Chapter three explained briefly about Class Diagram and how deep we will work with it in the beginning phase of process modelling in EFFORTS. Therefore, we will only concentrate on extract the classes and their attributes from the diagrams we have modelled so far. Some activities for this step are to:

1. Take the use cases and the actors from the use case diagrams
2. Name the items in No 1. in a substantive form without space between words
3. Read the information including your interview minute and highlight all words which are noun or substantive
4. Try to group the items or now so-called the “objects” which belong together or have closed relationship according to their characteristics

From our information, we will have:

**Actors**

<table>
<thead>
<tr>
<th>Class</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilot</td>
<td>Name, Id</td>
</tr>
<tr>
<td>Vessel</td>
<td>VesselId, Tonnage, Draught, Crews, Cargo, Bears, Length, LDG, Speed, SailingList</td>
</tr>
<tr>
<td>ShippingAgent</td>
<td>Name, AgentCode</td>
</tr>
<tr>
<td>TerminalOperator</td>
<td>Name</td>
</tr>
<tr>
<td>TugService</td>
<td>Name</td>
</tr>
</tbody>
</table>

**Information**

<table>
<thead>
<tr>
<th>Class</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hydro.Info.</td>
<td>Name, Current</td>
</tr>
<tr>
<td>Meteor.Info.</td>
<td>Temperature, WindSpeed, Pressure</td>
</tr>
<tr>
<td>Notifications</td>
<td>PilottageDemand, TowageDemand, TugAssistantDemand, Incident</td>
</tr>
<tr>
<td>TrafficInformation</td>
<td>BerthingPosition, ETA, ETATD, ATWATD, PilotageSchedule</td>
</tr>
</tbody>
</table>

**Reports**

<table>
<thead>
<tr>
<th>Class</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>IncidentReport</td>
<td>VesselInfo, CauseOfIncident, DateOfIncident, TimeOfIncident, DamageDetails, LossDetails, InvolvedParties</td>
</tr>
<tr>
<td>PositionReport</td>
<td>VesselInfo, PositionInfo</td>
</tr>
<tr>
<td>Pre-EntryReport</td>
<td>VesselInfo, ETA, PortOfCall</td>
</tr>
</tbody>
</table>

**Communication Devices**

<table>
<thead>
<tr>
<th>Device</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td>VHF</td>
</tr>
<tr>
<td>Phone</td>
<td>Paper</td>
</tr>
</tbody>
</table>

Figure 5.5: VTS Process – Class diagram without association and operation
6.9 Check plausibility of the program and specification with experts

This step is one of the most critical in the modelling process and happens 
repeatedly after you have drawn the diagram or issue the specification. In this step 
we have to give all diagrams and specification we have created to the experts and let 
them check for correctness.

The experts, in this place, mean both specialists in the profession of the process we 
model – VTS Officer - and the IT specialists who are familiar with UML notations. 
Moreover, if possible, we should distribute these documents not only to management 
level but also operational one to check the plausibility. This is because sometimes the 
perspectives on the process in these two levels are different. We then are able to find 
out the problems in the process easier which are really relevant to the practice.

You should not wait so long to let the expert look on your diagram. The faster the 
feedback you get, the better you can continue modelling the process.

6.10 Review the diagrams

After you have got the comments from both sides of experts, you have to analyse and 
take them into consideration for modification of your diagrams. From point 5.5 until 
5.10 these activities happen as a recursive activity meaning that after reviewing 
and modifying your diagrams you will send to your experts again to check for the 
plausibility and correctness of the diagram according to specialists’ comments, and if 
necessary you have to modify your diagrams again.

6.11 Demonstration of development on process modelling

The diagrams and their modifications until the final result of the VTS process can be 
classified into each type of diagrams: use case-, activity- and class diagram; and 
demonstrated in this chapter. However, not every change is illustrated and explained 
here. To each diagram, only one example of change and the development of diagram 
in each step are presented.
Use Case Diagrams

Figure 5.6: The first use case diagram – VTS Process

The first use case diagram is created according to the on-desk research and existing information about VTS process found by the modeller. After the first visit at VTS office to get the overview of its activities, it can be summarised that admitting pilot is a direct action between pilot and vessel. Therefore, this use case is not included within
the scope of VTS process, instead VTS helps monitor whether the pilot is admitted on-board. Therefore, the use case “Admit pilot” and the actor “Pilot” are eliminated and replaced by the use cases “Monitor pilot admittance” from the second use case diagram as shown in figure 5.7.

Figure 5.7: The second use case diagram – VTS Process
Due to the further discussion with VTS officer, the modeller has found out that the "pilot" could be relevant in the VTS process in "acquiring pilot" and "acquiring towage". Moreover, the use case "check plausibility of report / information / order" does not exist in the reality. Therefore, these changes were re-illustrated in the diagram again as shown in figure 5.8. This third version has been finally presented to the Head of VTS Office for comments.

Figure 5.8: The third use case diagram – VTS Process

After the review of diagram by the Head of VTS office, it can be concluded some activities are not directly involved with VTS, including "acquire pilotage" and "acquire towage". Therefore, these two use cases as well as the actors "Pilot" and "Tug Service" can be excluded from the diagram. Moreover, some other activities are not implemented in the reality, even though many handbooks or descriptions have explained because the changes of regulations from time to time. (See figure 5.9)
Figure 5.9: The fourth use case diagram – VTS Process (Final Version)
Activity Diagrams

The first activity diagram (shown as figure 5.10) is created based on the second use case diagram (figure 5.6) since the irrelevance of the use case "admit pilot" has emerged before the activity diagram was created. More detailed information is captured in the activity diagram in the term of activities or orders and their logical flow.

Figure 5.10: The first activity diagram – VTS Process

Concerning the change of the second use case diagram, namely “check plausibility of report / information / order”, the figure 5.11 demonstrates how this change was treated in the second activity diagram. This is a good example showing that only an elimination of a use case can radically change the structure and flow of the activity diagram.

The changes from the second to the third activity diagram can be easily noticed by the usage of notation in “decision node” and the disappearance of the activities "acquire pilotage" and “acquire towage”. Moreover, the time line (at the most link of the diagram) was added. (See figure 5.11 and 5.12)

Last but not least, a slight modification in figure 5.14 has been made by deleting the time line since it is not really relevant in this stage. The time and sequence will be later explained in the next phase.
Figure 5.11: The second activity diagram – VTS Process
Figure 5.12: The third activity diagram – VTS Process
Figure 5.13: The fourth activity diagram – VTS Process (Final Version)
Figure 5.14: The first class diagram without association and operation – VTS Process

The only change which has been done for the class diagram is to delete the actors “pilot” and “tug service” since they are not directly relevant to the VTS process, according to Head of VTS office.
Figure 5.15: The second class diagram without association and operation – VTS Process
7 Conclusion

As mentioned in the first chapter, this handbook is created for specific uses within the project “EFFORTS” to be a guideline for all project partners using in process modelling purpose. The document started with some objectives and situation of the projects as well as the importance of process modelling. It is followed by an explanation of CIMOSA modelling framework which is applied as process modelling architecture in this project. Along with this concept a modelling language used for this purpose – UML - was agreed and it has been described and demonstrated from the fundamentals to the execution.

An example of VTS Process has been provided according to the guideline of this handbook. After the verification of the project, the partners who are responsible for work packages will follow this guideline and model the involved process in their work tasks.

In case that you have questions in regards with this handbook, you can contact EFFORTS Process Helpdesk:
Phanthian Zuesongdham
ISSUS / Maritime Logistics
Tel. +49-40-42878-6134
E-Mail. zuesongdham@tu-harburg.de
8 Reference

Bibliography

Websites
Cordis
URL: http://cordis.europa.eu/esprit/src/intro.htm
Wikipedia.org
URL: http://en.wikipedia.org/wiki/Business_process_interoperability
Microsoft Visio
URL: http://microsoft.com/office/visio/prodinfo/overview.mspx
Mindjet Mindmanager
URL: http://www.mindjet.com/
Annex A – UML notation overview

Unified Modelling Language (UML) 2.1 – Notation Overview for EFFORTS (1st Phase)

Activity Diagram

Use Case Diagram

Class Diagram

---

Document Code: EFFORTS-WP3.1.1-DEL-20061231-ISSUS-FINAL V2.0
Date: [31/12/2006]
# Annex B – Use case specification template

## EFFORTS Use Case Specification

| **Process Domain** | [Please enter the process domain] i.e.  
| VTS |
| **No of Use Case** | [Please enter the number of the use case] i.e.  
| VTS.1 |
| **Name of Use Case** | [Please enter the name to identify the use case with the format: a verb + noun or substantive] i.e.  
| Book the theater ticket |
| **Short Description** | [Please enter a short description to the use case with flow text of roughly 25 words] |
| **Actor(s)** | [Please enter all involved parties only for this use case with numbered nouns or substantive] i.e.  
| 1. Customer Service  
| 2. Customer |
| **Pre-Condition(s)** | [Please enter significant condition(s) to enable the action in this use case with numbered sentence(s)] i.e.  
| 1. The customer must be a member of the Theater Club  
| 2. The customer asks for availability of the ticket |
| **Input** | [Please enter all resource(s) for the use case including materials, information, documents, software, etc.] i.e.  
| 1. Customer ID  
| 2. Customer Name  
| 3. Computer  
| 4. Date  
| 5. Number of Tickets |
| **Flow of Event(s) - Standard Flow** | [Please enter text to describe the sequence of action in the use case with numbered conditional sentences (if...then)] i.e.  
| 1. if...then...  
| 2. if...then... |
| **Post-Condition(s)** | [Please enter significant condition(s) to check whether the use case is realised with numbered sentence(s)] i.e.  
| 1. The theater ticket is booked  
| 2. The customer received the ticket |
| **Output** | [Please enter expected result(s) for the use case] i.e.  
| 1. Ticket  
| 2. Ticket ID  
| 3. Seat Number  
| 4. Amount of Payment |
| **Extension Point(s)** | [Please enter description to define the alternative flows of events as well as the open issues which is not (yet) captured by this use case with numbered sentences] |

**Original Author**  Phanthian Zuesongdham / ISSUS

**Edited by**  Sven Mathes / N+P

**Version**  1.0
Annex c – Interview minute with vts

<table>
<thead>
<tr>
<th>Minute of Interview for Process Modelling</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Topic:</strong> VTS Process</td>
</tr>
<tr>
<td><strong>Place / Date / Time:</strong> VTS Center Hamburg Port Authority / 21.09.2006 / 10:00 a.m.</td>
</tr>
<tr>
<td><strong>Participants:</strong></td>
</tr>
<tr>
<td>Mr. Uwe Claassen (VTS – Hamburg Port Authority)</td>
</tr>
<tr>
<td>Mr. Norman Noack (N+P)</td>
</tr>
<tr>
<td>Ms. Phanthian Zuesongdham (ISSUS)</td>
</tr>
</tbody>
</table>

Inform about Arrival/Pre-Entry-Report

- All vessels outside the North Sea or Canal in Germany must inform the VTS 24 hours before they call Hamburg Port. Thereafter they will receive a confirmation of berthing position and entry of vessel from the VTS center.
- For some specific berthing position, the vessel must follow port obligation.
- A significant factor for berthing allocation is the draught of the vessel and the river Elbe. Therefore, the preliminary confirmation of the berthing position will be given. As soon as the draught is ensured, the final berthing position will be confirmed.
- The shipping agent or terminal operator informs the VTS about port call of the vessel.
- Feeders and other small ships inform about their calls at VTS shortly before the arrival.

Security Report

- The vessel and the berthing position must be certified according to the ISPS Code.
- All vessels passing through the German Bight must submit “Security Report” at VTS Center in Wilhemshafen as the Point of Contact. This document includes the details of Vessel names, last 10 port calls, security category, etc.
- The vessel will then be classified according to its security regulation.
- Normally vessels are classified as level 1 which means the vessel is non-dangerous.
- Vessels with security level 2 and 3 must submit a separate registration or notification to VTS.
- However, the security level has no influence on confirmation of berthing position.
- The Point of Contact Report will be sent per E-Mail by Central VTS.
- This report must be submitted also 24 hours before the arrival of the vessel.
Position Report

Arrival:
- There are many points of contact (POC) along the waterway in Elbe until the Hamburg Port territory.
- The POC notify VTS through the traffic channel 47 about the positions of the vessel.
- No reply or correspondence by VTS is needed to the POC notification.
- In exceptional case, the VTS can contact the vessel with supporting information from POC.
- After receiving POC notification, a VTS officer enters the information in its system. Alternatively, if the technology is available, the information flows automatically into the system.
- The system detects the information of the vessel such as time, place, speed and vessel position.

Departure:
- In case that the vessel is berthed at a terminal, it must inform its departure details.
- This activity will be done as soon as the vessel starts gathering way.

Aquire Pilotage

Arrival:
- The pilotage is normally acquired by the shipping agents few hours before the vessel arrives.
- The vessel must again request the pilot directly per VHF from pilotage station.
- There are many pilot stations along the way to Hamburg Port. The sea pilot station is responsible only for the area from the North Sea until Brunsbüttel. The Elbe pilot station support in pilotage from Brunsbüttel until Blankenese. At Blankenese, the Elbe pilot will be replaced by port pilot.
- The vessel only requires the pilot at the first point, namely sea pilot station. The other pilots will be requested by previous pilot onboard.
- The pilot takes over the navigational task from the captain and is the direct contact person for the vessel.
- VTS is informed about the pilot admittance and enter the information into its system.
- Vessels which its length is less than 90 Metre and/or its beam is less than 30 Meter are not obligated to acquire pilotage.
- There is also a list of vessel types which the pilotage can be exempt.
- This exemption will be applied when the vessel has called Hamburg Port in a specific period and the captain can communicate comprehensively in German.

Departure:
- Here the large vessel must acquire the pilotage 4 hours before its departure, while the smaller vessel 2 hours.
Issue Deviation Report (Only done in the port area)

- This report will only be issued if there is significant change to the vessel plan.
- The changes in regards with berthing position are not counted into this category. In this case, the port pilot will be informed. No report must be created.
- In case that the berthing position and the conditions of the vessel to call the port, Port VTS will issue an “Order” and send to the VTS Central Office, so that the communication between vessel and pilot can be created. (Pre-Condition.: Important Details)

Incident Report

- The vessel or the pilot onboard contacts VTS. Then, VTS will undertake all required measures and inform all relevant parties.
- In case that any occurrence which has affects to the vessel, the terminal operator must inform VTS.

Approve entry to port

- Already done together with confirmation of the berthing position.

Check and approve

- No such activities are practised.
Note: Security, facility and guarantee of environmental aspects must be highly concerned.

Aquire Tug

- Tug service will be acquired by the pilot on the vessel. There is no official instruction or regulation whether a vessel must deploy the tug assistance.