CATALOGUE BASED
COMPUTER AIDED ENGINEERING (CAE)
OF
PROCESS MODELS

MARKUS HOYER

A submission presented in partial fulfilment of the requirements of the
University of Glamorgan/Prifysgol Morgannwg for the degree of
Doctor of Philosophy

This research programme was carried out through a collaboration between the
University of Applied Sciences and Arts Hanover
and the
Faculty of Advanced Technology

May 2007

Certificate of Research

This is to certify that, except where specific reference is made, the work described in this thesis is the result of the candidate's research. Neither this thesis, nor any part of it, has been presented, or is currently submitted, in candidature for any degree at any other University.

Signed M. Hoyer
Candidate (Markus Hoyer)

Date 20-07-2007

Signed ....
Director of Studies (Dr. Giuliano C. Premier)

Date 12-09-2007

Abstract

In chemical industries, many hardware and software problems of chemical plants become obvious during start-up and commissioning and must be corrected in a way which is resource intensive in terms of time and money. The correction procedure continues until the desired operating performance has been achieved and efficient control has been established. In order to achieve at least an acceptable performance, often temporary solutions are established which may have drawbacks in terms of functionality and efficiency.

Many developments in modelling and simulation have already been undertaken to reduce such deficiencies during the early engineering phases. However, none of these approaches aim at the dynamic simulation of the plant design situation at the end of plant design, where in principle all components are completely specified –ready-to-order– and no parameterisation degrees of freedom are left as in the early design phases.

The subject of this work is the development of an integrated approach to automatically generate the required plant simulation models from a simulation model catalogue of physical components after completion of the engineering process, without requiring modelling expertise from the planning engineer to run the simulation. In this work, the approach is named ”Model^{CAT}“.

The availability of simulation models is a basic prerequisite for this approach and has led to a methodology for the definition of the simulation model requirements to be used at the end of plant design, by integrating them into CAE-plant design tool's databases. The core of this work is the model aggregation module (MAM), a methodology for the automatic aggregation of plant models based on the plant design information and including the simulation models. MAM contains a systematic analysis of the plant to be simulated and the aggregation of simulation models for their use in a simulation environment, which includes a process and a control simulator. In order to support planning engineers, who are assumed to be non-experts in the field of modelling and simulation, to establish simulations based on plant design, this work presents a concept for a smart graphical user interface (GUI), as an integral part of this Model^{CAT} approach.

To validate the proposed approach and associated methods, a prototype software environment has been established. It has shown that the concept is feasible and promising in establishing dynamic simulations on virtual plants at the end of detailed engineering. The experiences and knowledge gained from the prototype realisation and validation have used to extrapolate to the requirements for an industrial implementation.

Dedication

To all who have believed in me, in particular to Maren and to my family.

Acknowledgements

At first I wish to express my sincere gratitude to my supervisors Dr. Giuliano C. Premier, the Director of Studies of this research project, and Prof. John Ward for the many useful ideas, fruitful discussions and their support within my research work.

I am particularly appreciative of Prof. Dr.-Ing. Reimar Schumann for his enthusiastic supervision, help and encouragement during the course of my PhD programme, not forgetting the multitude of beneficial and productive discussions. In particular for the support at the final stage of this work I would like to thank him along with Dr. Giuliano C. Premier.

Many thanks are due to all my colleagues in the applied research programme AUBIOS, especially to Dr. Carsten Horn, Jan Mohr, Peter Hoffmann and Markus Messer. I would also like to thank the diploma students Marijke van der Meer, Khaldoun Halalo, Mohammed Sabatin as well as my project students Marc Lehmann and Nikolay Velichkov along with Markus Bellmann for their contribution and support in this research project.

I also like to thank the University of Glamorgan, the Faculty of Advanced Technology and the members of staff for supporting my work, not least financially. Additionally, the financial support from the Volkswagen foundation, the University of Applied Sciences and Arts Hanover and the British Council/DAAD through the Anglo-German Academic Research Collaboration Programme, ARC are gratefully acknowledged.

A special mention of thanks to Maren and Christian for the countless but always fruitful discussions and reviews of my work. I am deeply grateful for their interest and openness to my problems throughout this work and their continual encouragements.

I like to thank Taz, Laura and James for making especially my last stays at the Glamorgan Court so enjoyable and pleasant.

Furthermore, I like to thank my family, especially my parents and my sister Marion (and family) as well as all of my friends who supported me during this stage of my life, especially for their patience and encouragement throughout.

Last, but most certainly not least, very special thanks to the beautiful "M" at my side for her understanding, patience, support and especially her love in the last four years. Thank YOU!

Table of Contents

CERTIFICATE OF RESEARCH II

ABSTRACT III

DEDICATION IV

ACKNOWLEDGEMENTS V

TABLE OF CONTENTS VI

LIST OF FIGURES XI

LIST OF TABLES XV

NOMENCLATURE XVI

1 INTRODUCTION **1**

- 1.1 Aims and Objectives 4
- 1.2 Relevance to Science and Industry 5
- 1.3 Organisation and Structure of the Thesis 6

2 REVIEW OF MODELLING AND SIMULATION APPROACHES TO VIRTUAL START-UP WITHIN THE PLANT LIFE CYCLE 7

2.1 Mathematical Modelling 7
2.1.1 Classification of Mathematical Modelling 8
2.1.2 Development of Mathematical Models 8

2.2 Review Criteria 10
2.2.1 Life Cycle Phase Association 10
2.2.2 Modelling and Simulation Issues 10
2.2.3 User Support 11

2.3 Modelling and Simulation Tools for Process Domain 12
2.3.1 Process Flowsheeting Tools 12
2.3.2 Equation-Oriented Tools for Process Domain 15
2.3.3 Knowledge Based Modelling Approaches 17

2.4 Modelling and Simulation Tools for Control Domain 20
2.4.1 Block-Oriented Tools for Control Domain 20
2.4.2 Text-Oriented Tools for Control Domain 22

2.5 Domain Independent Modelling Tools 23

2.6 Integrated CAE Environments 24
2.6.1 épée and n-dim 25
2.6.2 Marquardt's Team, IMPROVE 26
2.6.3 ICAS 29
2.6.4 BASIS 30

2.7 Operator Training Simulation Tools 30

2.8 Life Cycle Modelling and Simulation Tools 32

2.9 Conclusions of Literature Review 34

3 PROPOSAL FOR CATALOGUE BASED CAE OF PLANT SIMULATION MODELS FOR VIRTUAL START-UP 41

3.1 General Concept of the Model^{CAT} Approach 41

3.2 Plant Design with CAE-Plant Design Tools 43
3.2.1 General Information on CAE-Plant Design Tools 43
3.2.2 Component Database 44
3.2.3 Project Database 45
3.2.4 P&I Diagrams and Function Block Diagrams 46
3.2.5 Results of Plant Design 49

3.3 Definition of the FSCS Model Catalogue ... 50
3.3.1 Differences between Generic and FSCS Models/ Definition of FSCS Models ... 50
3.3.2 Systematic Separation of FSCS Models into Two Domains ... 53
3.3.3 Requirements for FSCS Models in the Model Aggregation ... 54
3.4 Embedding of the FSCS Model Catalogue ... 59
3.5 Strategy for an Automatic Simulation Procedure Initiated from CAE-Plant Design Tools ... 61
3.5.1 Selection of Simulation Area ... 61
3.5.2 Systematic Analysis of the Selected Area in a P&I Diagram and Associated Function Block Diagrams ... 64
3.5.3 Automatic Model Aggregation ... 68
3.6 Methodology for Supporting Planning Engineers during Modelling and Simulation ... 71
3.6.1 Need for a GUI - a Concept for Non-Expert Users ... 71
3.6.2 Design Aspects of GUIs ... 72
3.6.3 GUI-Task 1: Selection of Simulation Area ... 74
3.6.4 GUI-Task 2: Specification of Boundary Conditions, Initial Values and Simulation Parameters ... 74
3.6.5 GUI-Task 3: Start of Simulation and Presentation of Simulation Results ... 79
3.6.6 Help System ... 80
3.7 Requirements on Materials for the Model^{CAT} Approach ... 80

4 PROTOTYPE REALISATION ... 82

4.1 General Considerations of the Model^{CAT} Prototype ... 83
4.1.1 Software Tools Used ... 83
4.1.2 Hardware Used ... 86
4.2 Plant Design Procedures in COMOS PT™ ... 86
4.2.1 General User Interface in COMOS PT™ ... 87
4.2.2 Component and Project Database ... 88
4.2.3 Design Utilities: P&I Diagrams and Function Block Diagrams ... 90
4.2.4 Results of COMOS PT™ ... 93
4.3 Definition of the FSCS Model Catalogue ... 94
4.3.1 Process FSCS Models in gPROMS™ Format ... 94
4.3.2 Process FSCS Model Examples ... 96
4.3.3 Control FSCS Models in Simulink™ Format ... 97
4.3.4 Control FSCS Model Examples ... 98

4.4 Embedding FSCS Models into COMOS PT™ 99

4.5 Automatic Simulation Procedure from COMOS PT™ including MAM and GUI Tasks 101

4.5.1 Help System (GUI Task) 103

4.5.2 Selection of Simulation Area in P&I Diagram (GUI Task) 104

4.5.3 Systematic Analysis of the Selected Area (MAM Task) 108

4.5.4 Specification of Boundary Conditions and Simulation Parameters (GUI Task) 112

4.5.5 Automatic Model Aggregation (MAM Task) 119

4.5.6 Start of Simulation and Presentation of Simulation Results (GUI Tasks) 125

5 VALIDATION OF MODEL^{CAT} – FRESH CHEESE APPLICATION EXAMPLE 126

5.1 Reference Plant 126

5.2 Virtual Start-Up Scenarios with Model^{CAT} 127

5.2.1 Automatic Simulation Procedure Based on CAE-Plant Design Results 128

5.2.2 Simulation Results 131

5.3 Comparison with Real Situation in an Industrial Fresh Cheese Production Plant 132

5.3.1 Simulation Results 133

5.3.2 Comparison of Simulation Results with Measurements at the Real Plant 135

5.4 Optimization of Plant Design with Model^{CAT} 136

5.4.1 Optimization of Process Design 137

5.4.2 Optimization of Control Design 139

6 DISCUSSION 145

6.1 Research and Development of a Proof of Concept for a FSCS Model Catalogue 146

6.2 Research and Development of the Integration of FSCS Models into CAE-Plant Design Tools 149

6.3 Research and Development of a Strategy for the Automatic Generation of Plant Models 150

6.4 Research and Development of a Strategy to Support the Planning Engineer with a GUI 152

6.5 Projecting the Prototypical Realisation of Model^{CAT} to an Industrial Realisation 155

6.6 Perceived Benefits for the Plant Life Cycle 156
6.7 Economic Aspects 159
6.8 The Overall Results Matched against Expectations 163

7 CONCLUSIONS AND RECOMMENDATIONS 165

7.1 Conclusions 166
7.2 Recommendations for Future Work 168

REFERENCES 170

APPENDIX OF THESIS 186

List of Figures

Figure 1-1: Phases and simulation support within plant life cycle 1

Figure 2-1: Classification of models 8

Figure 2-2: Process flow stream 12

Figure 2-3: Differences between sequential-modular and simultaneous-oriented simulation strategy 13

Figure 2-4: Example of equation-based approach for process domain (gPROMS™) 16

Figure 2-5: Model composition based on phenomenological building blocks of knowledge based modelling approaches 18

Figure 2-6: Measured signal from stream state 20

Figure 2-7: Example of block-oriented tools for control domain (Boris™) 21

Figure 2-8: IMPROVE's integrated environment (after Marquardt and Nagl, 2004) 27

Figure 2-9: Two types of operator training simulation tools (left: stimulated system; right: emulated system) including IS (Instructor System) and OS (Operator System) 31

Figure 2-10: Simulation in plant life cycle (after Cameron, 2005) 33

Figure 3-1: General concept of the Model^{CAT} approach 42

Figure 3-2: Component database hierarchy 45

Figure 3-3: Project database hierarchy 46

Figure 3-4: P&I diagram including components and function eyes 47

Figure 3-5: Two alternatives to indicate sensors in P&I diagrams: With sensor (left) and without sensor (right) 48

Figure 3-6: Function block diagram including control function blocks and input/ output signals 48

Figure 3-7: Plant design results at the end of detailed engineering 49

Figure 3-8: Inherent valve characteristics for quick opening (QO), linear (L) and equal percentage (EP) valve 51

Figure 3-9: Valve characteristics of equal percentage valve with different parameters 52

Figure 3-10: Natural interface between processes and control domains 53
Figure 3-11: Separation between process and control domains within Model^{CAT} approach 53
Figure 3-12: Dependencies of equation-oriented approach 55
Figure 3-13: Conventions of relationships between input and output system defining variables 55
Figure 3-14: Process FSCS models composed by sub-models 56
Figure 3-15: Process FSCS models: Single input and multi output 56
Figure 3-16: Process FSCS model: Sensor 57
Figure 3-17: Process FSCS model: Actuator 58
Figure 3-18: Embedding of FSCS models into the component database hierarchy 60
Figure 3-19: P&I diagram: Selecting section of plant for simulation 62
Figure 3-20: Distinction of the two domains in the P&I diagram 63
Figure 3-21: Overview of the generated lists of the systematic analysis of the selected area 64
Figure 3-22: List of process interconnections and list of open process connectors 65
Figure 3-23: List of interconnections between process and control domains 66
Figure 3-24: List of control interconnections and list of open control connectors/ parameters 67
Figure 3-25: Automatic model aggregation 68
Figure 3-26: Model aggregation (through streams) 69
Figure 3-27: Model aggregation (system defining variables) 69
Figure 3-28: Adaptation of system defining variables 70
Figure 3-29: Flow chart of GUI specification task 76
Figure 3-30: GUI task: Consistency check 78
Figure 4-1: Modules used in the Model^{CAT} approach 82
Figure 4-2: General user interface of COMOS PT™ 87
Figure 4-3: Component database ("Base objects") in COMOS PT™ 89
Figure 4-4: Project database ("Units") in COMOS PT™ 90
Figure 4-5: Function block diagram's location in project's database 91
Figure 4-6: Specification of component properties 92
Figure 4-7: Function block diagram within working area of COMOS PT™ 93
Figure 4-8: General gPROMS™ model 94
Figure 4-9: General Simulink™ function block model 97

Figure 4-10: Working area of COMOS PT™: Simulation model layer for embedding the FSCS model and variable information 99
Figure 4-11: Working area of COMOS PT™: Integration of FSCS models 100
Figure 4-12: General overview of MAM and GUI tasks 101
Figure 4-13: Logical sequence of developed programs (marked in blue colour) 102
Figure 4-14: GUI support: Model^{CAT} icon 103
Figure 4-15: GUI support: Help system 104
Figure 4-16: Selection of the section of plant to be simulated in COMOS PT™ 105
Figure 4-17: GUI support: Extended context menu of COMOS PT™ 106
Figure 4-18: Initialisation of simulation 106
Figure 4-19: Generated data sets of the selected area 107
Figure 4-20: Retrieval of connection information from the project database 109
Figure 4-21: Interconnection relationships for process components 110
Figure 4-22: Interconnection relationships between process and control domain 110
Figure 4-23: Interconnection relationships for control domain 111
Figure 4-24: Tool specific analysis of differential terms 112
Figure 4-25: GUI conventions (graphical and typographical aspects) 113
Figure 4-26: Areas of the GUI specification window 114
Figure 4-27: Converting the graphical set-up of P&I diagram into the set-up of the GUI 116
Figure 4-28: GUI specification window: Start of specification 117
Figure 4-29: GUI: Results of the consistency check 118
Figure 4-30: GUI: Specification of simulation parameters 119
Figure 4-31: Entire gPROMS™ model script 120
Figure 4-32: Entire Simulink™ script 122
Figure 4-33: gPROMS™ function block within Simulink™ 124
Figure 4-34: Entire Simulink™ scheme including the gPROMS™ function block ... 125
Figure 4-35: Simulation results presented in Simulink™ 125
Figure 5-1: Fresh cheese production line (with permission of Tetra Pak (1995)) 126
Figure 5-2: Cross section of fresh cheese separator 127
Figure 5-3: P&I diagram of the fresh cheese production plant 128
Figure 5-4: P&I diagram with the positioning of the NIR-sensor after the separator 129

Figure 5-5: Application example: Specification of boundary conditions within the GUI 130

Figure 5-6: Application example: Automatically generated Simulink™ scheme with the gPROMS™ block 131

Figure 5-7: Simulation results of application example 132

Figure 5-8: P&I diagram with the positioning of the NIR-sensor after the heat exchanger 133

Figure 5-9: Simulation results of the application example with sensor after heat exchanger 134

Figure 5-10: Measurements of an industrial fresh cheese production plant compared to simulation results 135

Figure 5-11: Process optimization: Selecting different process components from the component database 137

Figure 5-12: Simulation results of flow control example due to different FSCS models (valve): top) supplier A; bottom) supplier B 138

Figure 5-13: ICACSD modules 140

Figure 5-14: Integration of Model^{CAT} into ICACSD system 140

Figure 5-15: ICACSD integration: Extensions for ICACSD system 141

Figure 5-16: ICACSD integration: ICAI toolbox 142

Figure 5-17: ICACSD integration: ICAC toolbox 143

Figure 5-18: ICACSD integration: Extensions for ICACSD system for cascaded controller optimization 144

Figure 6-1: Extrapolation of results gained from the Model^{CAT} prototype realisation 145

Figure 6-2: Current state of Model^{CAT} prototype realisation 146

Figure 6-3: Industrial realisation of the Model^{CAT} approach 156

Figure 6-4: Qualitative overview of time savings with the Model^{CAT} approach 159

Figure 6-5: Economic savings by using the Model^{CAT} approach 162

List of Tables

Table 2-1: Modelling and simulation approaches applicable in plant life cycle 35

Table 3-1: Differences between generic and FSCS models 52

Table 3-2: Topics for the automatic simulation from CAE-plant design tools 72

Table 3-3: Software requirements 81

Table 4-1: Software tools used for the prototype realisation 85

Table 4-2: Hardware features used for the prototype realisation 86

Table 4-3: Short descriptions of the areas shown in Fig. 4-2 (COMOS PT™) 88

Table 4-4: Parts and short descriptions of process models (gPROMS™) 95

Table 4-5: Typical characteristics of a FSCS model (control valve) 96

Table 4-6: Parts and short descriptions of control models in Simulink™ 98

Table 4-7: Areas and short descriptions of the GUI specification window 115

Table 6-1: Effort savings resulting from Model^{CAT} 161

Nomenclature

ABBREVIATIONS

AE	Algebraic Equations
AI	Artificial Intelligence
ASCII	American Standard Code for Information Interchange
BC	Boundary Condition
CACSD	Computer Aided Control System Design
CAE	Computer Aided Engineering
CAEX	Computer Aided Engineering eXchange
CAPE	Computer Aided Process Engineering
CAPE-OPEN	Computer-Aided Process Engineering-Open
CEN	Comité Européen de Normalisation (European Committee for Standardization)
CFD	Computational Fluid Dynamics
CLIP	Conceptual Lifecycle Process Model
CO-LaN	CAPE-OPEN Laboratories network
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
CSSL	Continuous System Simulation Language
DAE	Differential Algebraic Equations
DCS	Distributed Control System
DOF	Degrees Of Freedom

EMCR	Term used in COMOS PT™, believed to be an acronym for electronic, measurement and control domain
EP	Equal Percentage
FSCS	Fully Specified Component Simulation
GMA	Gesellschaft für Mess- und Automatierungstechnik (German Measurement and Automation Society)
GUI	Graphical User Interface
HEX	Heat EXchanger
ICAC	Industrial Computer Aided Controller Design
ICACSD	Industrial CACSD
ICAI	Industrial Computer Aided Identification
IDM	Inductive Flow Meter
IEC	International Electrotechnical Commission
IFAC	International Federation of Automatic Control
IS	Instructor System
ISO	International Organization for Standardization
L	Linear
MAM	Model Aggregation Module
MIMO	Multi Input, Multi Output
Model^{CAT}	Catalogue based CAE of plant simulation models
NIR	Near-InfraRed
OS	Operator System
P&I	Piping and Instrumentation
PCB	Printed Circuit Board

PCS	Process Control System
PDE	Partial Differential Equations
PDXI	Process Data EXchange Institute
PFD	Process Flow Diagram
PID	Proportional plus Integral plus Derivative
PISTEP	Process Industries STandard for the Exchange of Product Data
PLC	Programmable Logic Controller
QO	Quick Opening
R&D	Research and Development
ROI	Return On Investment
SCADA	Supervisory Control and Data Acquisition
SISO	Single Input, Single Output
STEP	Standard for the Exchange of Product Data
VDI	Verein Deutscher Ingenieure (Association of German Engineers)
VDE	Verband der Elektrotechnik, Elektronik und Informationstechnik (German Association for Electrical, Electronic and Information Technologies)
UK	United Kingdom
XML	EXtensible Markup Language
3D	Three Dimensional

SYMBOLS

c_v	Valve coefficient [m³/h]
c_{v0}	Valve coefficient of a closed valve [m³/h]
c_{v100}	Valve coefficient of a fully opened valve [m³/h]
DM	Dry Matter [%]
DN	Diameter Nominal [mm]
Fr	Flow rate [m³/h]
k	Constant that depends on the units used in the valve's equation
N_{BC}	Number of boundary conditions
N_{DOF}	Number of degrees of freedom
N_E	Number of model equations
N_{Ei}	Number of model equations of each FSCS model
N_V	Number of the all used model variables
N_{Vi}	Number of the all used model variables of each FSCS model
Δp_{max}	Maximum pressure difference
p_{in}	Pressure at the inlet [bar]
p_{out}	Pressure at the exit [bar]
ρ	Density [kg/m³]
t	Time [s]
x_{100}	Maximum stem position path [mm]

1 Introduction

The development and design of chemical plants is a challenge for all chemical process, control and plant design engineers involved in their specific domains. Depending on the size of a chemical plant, an enormous technical and planning effort is required, in order to fulfil the technical and the growing temporal requirements. High competition pressures and open markets force shorter development periods with increasing requirements on the plant design process, see e.g. Lien and Perris (1996).

To meet the requirements of such growing economical and also ecological constraints with parallel growing demands on product quality and availability, modelling and simulation of chemical plants and processes have increased rapidly, especially due to the concurrent development of more powerful computers (Ponton, 1995).

During the plant life cycle, chemical plants pass through several phases until the plant design is accomplished. Fig. 1-1 illustrates the main phases during the plant life cycle, which are significant for this approach. Initialised by the idea for developing a new chemical plant, the conceptual phase is the first life cycle phase which comprises feasibility studies, process synthesis, including process hazards, analysis and risk assessments, see e.g. Biegler *et al.* (1997).

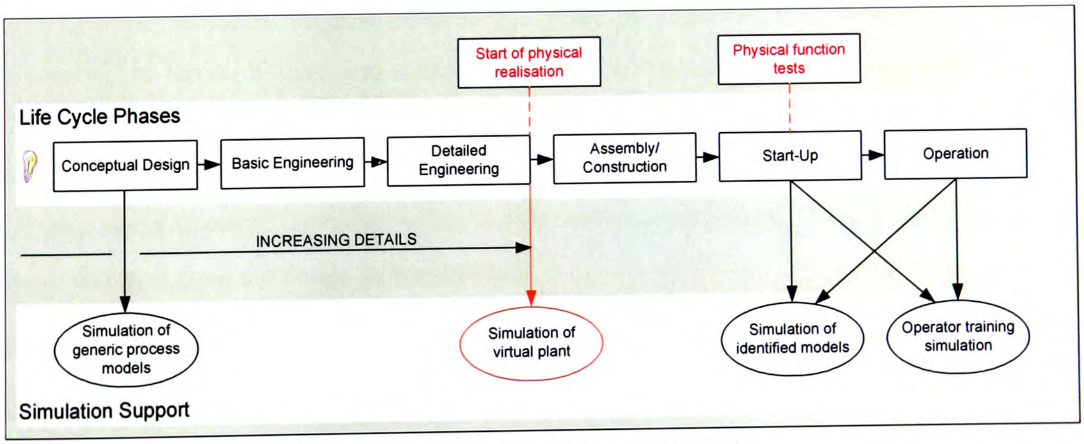


Figure 1-1: Phases and simulation support within plant life cycle

In the course of development, knowledge and detailing of the plant design increase. During basic engineering the design requirements are specified and main components are defined. Specification is completed at the end of detailed engineering, see e.g. Sinnott *et al.* (2005), which implies that all components and subsystems are completely specified – ready to order (= physical component) – and no latitude for specification remains, as in earlier design phases. All information that is required for construction, assembly, start-up and operation of the chemical plant is prepared and is available at the end of detailed engineering generally using large design databases.

By the end of the design phase, which has normally incurred 2 to 3 % of the total project costs, 80 % of the capital and operating expenses of the final plant are fixed, see Biegler and co-workers (1997), Plass (2001) talks about 70 %. In Bernecker (2001) and Couper *et al.* (2005) the costs for chemical plant design is estimated to be from 5 to 25 % of the overall costs depending on the kind of chemical plant. These costs illustrate the high importance of chemical plant design and the large potential of improvement.

After detailed engineering is complete, the embodiment of the plant begins and usually uncovers mistakes and deficiencies resulting from the plant design and engineering phases. Each component and subsystem of the chemical plant is ordered, built and assembled. After completing the assembly and construction phase, the start-up phase of the assembled plant takes place.

Role of Start-Up

The start-up phase presents the first possibility to test the plant's functionality realistically, including the interactions between components. During start-up of chemical plants hardware and software problems become obvious and must be corrected with intensive efforts and resources, in terms of time and money. The fault correction procedure continues until the desired performance has been achieved and efficient operation and control are established. Finally (and usually only) at this point in the plant life cycle it becomes apparent if the design of process components and control functions have been successful. Usually, the final reality looks different from the design in some details.

Frequency and reasons for delays caused by difficulties during start-up, are given by Holroyd (1967), Finneran *et al.* (1968) and Bernecker (2001) and may be summarised as follows:

- 26 to 30 % are due to design and engineering mistakes
- 56 to 61 % are due to malfunctions and breakdowns of equipment
- 13 to 15 % are due to operator errors

In total, more than 80 % of these malfunctions and shortcomings result from earlier phases of the plant life cycle and up to 30 % from plant design phases. Matley (1969) gives a reason for such shortcomings in the design phase. He argues that during chemical plant design compromises have to be made, a balance between the ideal plant and the economical aspects, i.e. the permissible costs.

An important factor for a successful chemical plant design is also a good interrelation between process and control domains, as stated by many authors, see e.g. Schuler (2004) and described in the following.

Interrelation between Process and Control Domains

Findings that illustrate the interrelationship between process engineering and process control are given by Aström and Häggelund (1995). They summarised two papers, Bialkowski (1994) and Ender (1993), documenting their experience on the quality of industrial control: about 80 % of the control loops do not fulfil their duties. The reasons for such deficiencies are that in 30 % of cases the sensors and actuators were incorrectly dimensioned, in another 30 % the control performance was insufficient due to inappropriate controller tuning and the last 20 % had other reasons. This reduces the product quality and can lead to unsafe process operation. Extra process staff is required to undertake frequent adjustments (Hahn and Nöth, 1997).

In most cases major malfunctions could have been avoided if process and control engineers could have worked closely together, in order to detect process design faults or equipment unsuitabilities for control, as described by Luyben *et al.* (1998) or Erickson and Hedrick (1999). Felleisen (2001) outlines this as an historical problem of separate development and defines it as a result from different methods of knowledge representation, especially in describing the processes.

Consequences for Start-Up and Operational Phase due to Inadequate Plant Design

Biegler and co-workers (1997) described the procedure for making changes to improve process performance during start-up as "debottlenecking". Due to time and money constraints, industrial companies have to cope with these debottlenecking procedures and could often only fulfil "quick and dirty" solutions with limitations in terms of functionality and efficiency, see Bernecker (2001). Such temporary solutions include changes to and expenditures on materials and equipment, which is only required to meet time and money constraints.

Furthermore, poor control performance can lead to critical situations, especially in the case of process disturbances. According to Froese (1995) it is industrial reality, that most companies run their processes with simple sub-optimal control strategies because they are deterred by the enormous costs of modelling.

Up to 20 % of all investment costs for a plant arise from the fault correction procedures, which result primarily from engineering faults and component breakdowns (Weber, 2002). According to Seider and co-workers (1999), the average start-up expenditure is 10 % of the fixed capital investment, but according to Peters and co-workers (1991), could also increase to 20 %.

Need for a New Modelling Approach for Chemical Plants

In order to overcome these time intensive and unpredictable fault correction problems during real start-up, novel approaches are required by the chemical industries. One possible answer is the establishment of a virtual start-up procedure with the ability to investigate scenarios at the end of the final design phase, the detailed engineering, allowing easy fault detection and correction at low costs. The main testing procedures could be done by computer based simulation of the virtual plant before its realisation begins and the plant is already built, thus reducing time and costs incurred during the real start-up.

The question of how such a virtual start-up procedure could be realised at the end of detailed engineering leads to the aims and objectives of this work.

1.1 Aims and Objectives

The main aim of this study is the development of a modelling and simulation approach to execute virtual start-up procedures based on plant design at the end of detailed engineering and to reduce time and money of the real plant start-up. The support of planning engineers, who are normally not experts in modelling and simulation, is another aim addressed in this work. A comprehensive scientific and systematic methodology in this field of modelling and simulation is still missing and could fill the gap of uncertainty between the design and start-up phases.

From these aims the following objectives have been derived:

- The first objective is the development of a proof of concept for a simulation model catalogue required for virtual start-up procedures scheduled at the end of detailed engineering. At this stage of design, the simulation models should reflect the behaviour of physical components, which have precise characteristics and parameters in order to establish simulations of realistic virtual start-up procedures. These "Fully Specified Component Simulation" models are hereafter referred to the abbreviation *FSCS* models in the whole thesis.
- The second objective comprises the integration of the FSCS models into CAE- (Computer Aided Engineering) plant design tools, which are powerful CAE-software tools for the support of basic and detailed engineering. A methodology to catalogue such FSCS models within the CAE-plant design tools for simple re/use within plant design is included in the second objective.
- The third objective is the development of a systematic strategy for the automatic generation of plant models based on the results of plant design making use of FSCS models.
- The fourth objective is the development of a systematic strategy for the support of the planning engineer with a suitable GUI (Graphical User Interface) to guide her/him from plant design to virtual simulation.

1.2 Relevance to Science and Industry

The theoretical development of component and plant models from physical and chemical laws is a difficult, time consuming and therefore expensive procedure. The variety and complexity of chemical processes and plants requires highly sophisticated expert knowledge of different domains, such as control domain, process domain, modelling and simulation domain. However, such experts are in general not available in small or medium sized industrial (planning and engineering) companies.

Although various approaches for modelling and simulation of chemical plants and processes are available, there is a need in industry and science for easy and satisfactory solutions for the automated generation of process models. Scientific and industrial researchers have discussed this topic for many years and the discussions are still going on, see conferences and workshops of the GMA-VDI/VDE Society for Measurement and Automatic Control from 1997, 2001, 2004, 2005 and 2006. The lack of meaningful models, i.e. process models, is also discussed in literature, as stated e.g. by Marquardt (1996) or Foss *et al.* (1998).

1.3 Organisation and Structure of the Thesis

The thesis has been organized in the following sequence, starting in Chapter 1 with a broad introduction for the reader into the background of this work: The development of chemical plants and its effects on the start-up phase. The deficiencies during start-up phases lead to the idea of catalogue based CAE of plant simulation models for a virtual start-up at the end of detailed engineering.

Chapter 2 reviews the current state of literature concerning the modelling and simulation approaches within the plant life cycle and how these tools relate to the aims and objectives of this thesis.

Chapter 3 contains a proposal for a modelling and simulation methodology suitable to realise virtual start-up procedures at the end of detailed engineering. It describes the definition of the required quality of FSCS models, the integration of such FSCS models into CAE-plant design tools and the methodology for an automatic procedure to generate plant simulation models based on the results of CAE-plant design. It also includes methods for the support of the non-expert, the planning engineer, during the execution of a virtual start-up.

Chapter 4 describes the prototypical realisation of the methods described in Chapter 3. The definition of FSCS models and integration into a commercial CAE-plant design tool is demonstrated. The realisation of a model aggregation module (MAM) for the analysis of the CAE-plant design, the aggregation of the process and control simulation models and the realisation of the smart GUI is the core of this chapter.

Chapter 5 describes a case study to demonstrate the feasibility of the developed approach. The case study considers a part of a fresh cheese production plant. Additionally, examples for the potential extension of the proposed approach as an optimization tool are given.

Chapter 6 presents the discussion of the results gained from the prototypical realisation. Furthermore, the requirements for an industrial realisation of the proposed approach are discussed. This chapter ends with considerations about the economic benefits of the proposed approach in an industrial environment.

In Chapter 7 the final conclusions are drawn and recommendations for future work are made.

2

Review of Modelling and Simulation Approaches to Virtual Start-Up within the Plant Life Cycle

This literature review serves to validate the outlined research area and the objectives with respect to existing literature. In the literature review current modelling and simulation approaches for process and control design are reviewed with respect to their potential use in a virtual start-up simulation. It does not address all aspects of modelling and simulation within plant design but it does indicate the need for modelling and simulation of virtual start-up scenarios after detailed engineering has been completed.

Simulation is used in many contexts within plant life cycle, including the modelling of particular systems in order to gain insight into their functioning. A model is always a simplification of the system itself and the accuracy required mostly depends on the usage and purpose of the model.

Before starting with the review of current approaches for process and control modelling and simulation, a brief review of essential aspects in mathematical modelling for use in process and control domain is given.

2.1 Mathematical Modelling

Mathematical models describe a system (plant subsystem or process) in a declarative and formal way with mathematical equations. Profound knowledge and understanding of the physical, chemical (or biological) phenomena is frequently required to abstract the process behaviour into a mathematical model, see e.g. Linninger (2001). However, mathematical modelling is the most common modelling method for describing systems such as process and plant units, see e.g. Marquardt (1992).

2.1.1 Classification of Mathematical Modelling

In order to classify mathematical models in the context of different model types, the model classification ideas of Geoffrion (1989) and Bogusch (2001) are used exemplarily. Fig. 2-1 illustrates the different classifications of model types.

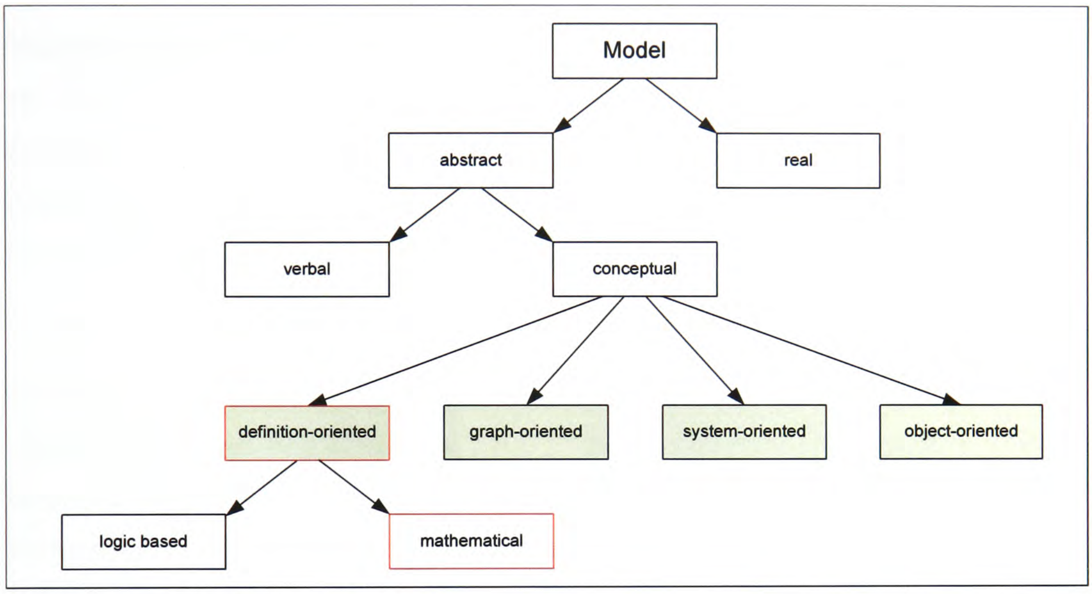


Figure 2-1: Classification of models

Mathematical models belong thereby to abstract conceptual and definition-oriented models. Other conceptual modelling aspects which are relevant for the model organisation and interconnections are object-, graph- and system-oriented models. For more information in these areas see e.g. Kim (1990) for object-oriented, David and Alla (1992) for graph-oriented and Kheir (1988) for system-oriented modelling.

2.1.2 Development of Mathematical Models

The development of mathematical models mainly depends on the later usage and purpose. In chemical plant design two different approaches have been developed: the theoretical and the experimental modelling approach.

Theoretical modelling is based on physical and chemical laws, also called first-principles based or rigorous models, see Marquardt (1996) and is predominantly used in the process domain. These models are usually based on balance equations for mass, energy, impulse and on phenomenal relationships. Stephanopoulos *et al.* (1990) were one of the first to present a formal framework for process engineering of process models. The result is a structured model described by DAEs (Differential Algebraic Equations), which contain apart from the input and output quantities also information about the internal states and relations of a system, see e.g. Brenan *et al.* (1989). DAEs describe the dynamic process behaviour. By contrast, stationary models are described by AEs (Algebraic Equations) without the time differential operator.

Modelling of process and control systems is mostly done with lumped models. In lumped models the state variables are invariant with respect to spatial dimensions. In distributed models, the dependent variables are a function of the spatial dimensions and are described in terms of PDEs (Partial Differential Equations). This kind of model is used in specific areas of process domain, i.e. in CFD (Computer Fluid Dynamics) investigations in order to investigate fluid dynamics.

For more details in the field of model classification see e.g. Jaako (1998) or Bogusch (2001).

By contrast, the experimental modelling investigates the system by measuring the input and output quantities and their correlations using a suitable mathematical approach to model the behaviour between the input and output quantities. This procedure is called identification and the resulting model is an empirical model. This kind of modelling is predominantly used in control system engineering, especially during start-up and operational phase, see Ljung (1987); however, it assumes that the system already exists. The result is often a linear unidirectional transfer function described by DAEs, which do not model internal states or relations, see Marquardt (1994).

In order to solve mathematical models several numeric integration methods are available, depending on the kind of model, the accuracy, the robustness and efficiency, see e.g. Schneider (2003).

2.2 Review Criteria

For the review of modelling and simulation approaches within the plant life cycle three main criteria have been defined and discussed in the following sections:

- Life Cycle Phase Association
- Modelling and Simulation Issues
- User Support

2.2.1 Life Cycle Phase Association

All modelling and simulation tools are categorised according to the phase of the plant life cycle, in which they are typically used. These phases are the conceptual design, basic and detailed engineering, real start-up and operational phase, see also Fig. 1-1.

2.2.2 Modelling and Simulation Issues

Here, all modelling and simulation tools are investigated with regard to specific modelling and simulation issues:

- **Kind of Model Provision**
  This issue shall answer the question about the kind of model provision, whether so-called generic or the FSCS models are provided in model catalogues. Whereas FSCS models reflect the real behaviour of the physical (ready-to-buy) components from specific suppliers (including their definitive properties), the behaviour of generic models, whose parameters can be freely chosen, is unspecific. A more detailed comparison of both model types is given Section 3.3.1.
  The size of model catalogues is investigated as well as the effort to integrate new models into the model catalogue.
- **Model Transfer from/to CAE-Plant Design Tools**
  The possibility for transferring plant models from CAE-plant design tools to the respective modelling and simulation tool is reviewed, with the question if their model and connection information is usable for generating simulation models. Within CAE-plant design tools, possible planning utilities of the process domain are block diagrams, PFDs (Process Flow Diagram) or P&I (Piping and Instrumentation) diagrams according to the International Standard ISO 10628:2000 (International Organization for Standardization, 2000). Planning utilities of the control domain are e.g. function block diagrams, standardised in the international standard IEC 61131 (International Electrotechnical Commission, 2003)

(Part 3 of the IEC 61131 contains, apart from the description of instruction lists, ladder diagrams and structured text, the continuous function block diagram and the sequential function block diagram).

- **Interface to Modelling/Simulation Tools of other Domains (Co-Simulation)**
  The possibility for co-simulation is investigated, which is defined as an interface to modelling and simulation tools of the other domain (process/control domain).
- **Plant Modelling/Simulation Separated into Process and Control Domain**
  Special emphasis is placed on the possibility to perform plant modelling and simulation separated into process and control functions (as closely resembling a real plant as possible). The fictitious optimal solution would be the one-to-one model of the real process and the one-to-one image of the process control system (mostly realised as DCS (Distributed Control System) or SCADA (Supervisory Control and Data Acquisition) system).

2.2.3 User Support

This main criterion comprises the investigation of the support for non-experts during the modelling and simulation task. Therefore, special emphasis is laid on the use of GUIs (Graphical User Interface) with regard to support during parameterisation of models and during the specification of boundary conditions and simulation parameters.

In the following, all simulation and modelling tools are investigated by the criteria outlined above. For the sake of simplification these tools have been classified into the following groups:

- Modelling and Simulation Tools for Process Domain
- Modelling and Simulation Tools for Control Domain
- Domain Independent Modelling Tools
- Integrated CAE Environments
- Operator Training Simulation Tools
- Life Cycle Modelling and Simulation Tools

The review ends with a conclusion with emphasis on the aims and objectives of this work.

2.3 Modelling and Simulation Tools for Process Domain

Modelling and simulation tools for the process domain may be categorised into three groups, as outlined by e.g. Marquardt (1996):

- Process Flowsheeting Tools
- Equation-Oriented Tools for Process Domain
- Knowledge Based Modelling Approaches

These groups are reviewed in the following sections.

2.3.1 Process Flowsheeting Tools

Process flowsheeting tools, also called block-oriented tools, still are the dominant modelling and simulation tools in chemical industries, see reviews in Westerberg (1979), Biegler (1989) or Sinnot *et al.* (2005). The main aim is the definition and analysis of material balances during the early design phases. Therefore, process flowsheeting tools are accompanied by large physical property databases. Process flowsheeting tools provide the user with standardised, predefined blocks from model libraries at a block diagram level. The user can use these blocks and aggregate them to block schemes.

Another important attribute of all process modelling and simulation approaches is their representation of process media streams. Different from control domain approaches, where physical states of a media stream are separately regarded, in process domain a multitude of states (such as flow rate, temperature, pressure, enthalpy, etc.) is joined in one single stream, as presented exemplarily in Fig. 2-2.

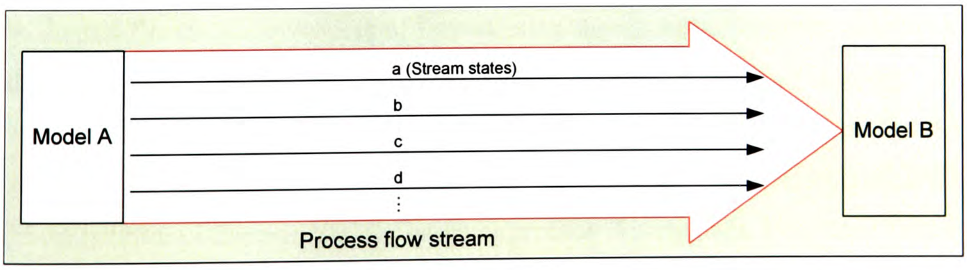


Figure 2-2: Process flow stream

Until the recent past process flowsheeting approaches were only solved by the sequential-modular strategy. The blocks (models) are evaluated one after another in a procedural manner. The output of an already computed block is used as input for the next block to be computed. This unidirectional characteristic results from the causal dependencies between two consecutive blocks. Problems arise, if recycles are in the information flow among the modules. Recycles have to be handled iteratively, which is not a trivial task (due to tearing problems) and has been the interest of several research groups, see e.g. Upadhye and Grens (1972) and Ollero and Amselem (1983).

As opposed to the sequential-modular strategy, the simultaneous strategy aims at calculating all constituent model equations at the same time simultaneously, *quasi* in one "super equation". By the non-causal dependencies the information flow between two models can vary depending on the models' states. Recycles do not have effects on the calculation efficiency. Fig. 2-3 illustrates the unidirectional dependencies of sequential-modular (on the left hand side) and bi-directional of simultaneous-oriented approaches (on the right hand side) resulting from their causal and respectively non-causal properties.

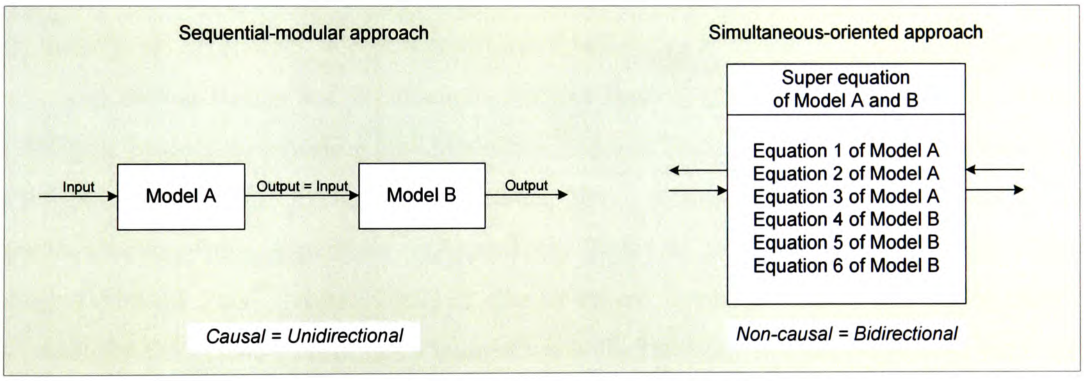


Figure 2-3: Differences between sequential-modular and simultaneous-oriented simulation strategy

In the last decade some vendors of flowsheeting approaches offered the possibility to combine the sequential-modular with the simultaneous-oriented simulation strategy. Examples are AspenPlus™ (AspenTech, 2002), Chemcad™ (Chemstations, 2002), CADSIM Plus™ (Aurel, 2002) and ProSimPlus™ (ProSim, 2002), together with Hysys™ (Hyprotech, 2002) or ProII™ (Invensys-Simsci-Esscor, 2002). The latter process flowsheeting tools have found widespread use in the chemical industry in the last 10-20 years. They are predominantly used to describe and simulate steady-state behaviour, however in recent years, dynamic behaviour has been integrated.

All tools listed are primarily used during the conceptual design phase and the early basic engineering, see Seider *et al.* (1999). During these early phases, process flowsheeting tools make use of generic models of standard components (or units) which are stored in model catalogues for easy use. The model catalogues of all reviewed process flowsheeting tools contain a large number of standard process components, such as mixers, splitters, separators, heat exchangers, columns, reactors, etc. Process flowsheeting tools are predominantly focussed on the process design, even though there are some basic controller models available. The design of own models, which are not in the model catalogue, or the extension of existing models is possible, but only by going into the depths of mathematic modelling and programming in the source code of the respective tool. E.g. within AspenPlus™, new units can be modelled with Fortran© code or dynamic models with C++© or C© code (within Aspen Customer Dynamics™). Other programming languages, such as Visual Basic© are used e.g. in Chemcad™. FSCS models are not provided as well as interfaces to other modelling/simulation tools of other domains. A clear separation into process and control domain is missing. Additionally, all tools investigated are used more or less intensively for process simulation and optimization of an existing plant and operator training, see e.g. Contreras and Ferrer (2005).

In recent years approaches to combine process flowsheeting tools with CAE-plant design tools appeared, such as Becker and Westfechtel (2003) or Syamlal (2003). In Becker and Westfechtel (2003) the process flowsheeting tool AspenPlus™ (AspenTech, 2002) was interconnected to the CAE-plant design tool COMOS PT™ (Innotec, 2002). Syamlal (2003) describes the possible interconnection from AspenPlus™ (AspenTech, 2002) to SmartPlant™ (Intergraph, 2000), while CADSIM Plus™ (Aurel, 2002) is able to import flowsheet drawings from AutoCad™ (Autodesk, 2003). The goal of these approaches is the information transfer from process flow diagrams of block-oriented simulation tools to CAE-plant design tools. The reverse direction is also possible but not favoured within the mentioned approaches. Vendors of CAE-plant design tools, such as Innotec (2005) and Intergraph (2005), also offers the interconnection to the process flowsheeting tool ProII™. However, the main emphasis is laid on the data exchange from block-oriented simulation tools to CAE-plant design tools. A systematic separation of the process and control domain in this context is missing.

For model aggregation at the block diagram level, no specific expert knowledge is required and can be done intuitively by the user. The parameterisation of the generic simulation models from the model catalogue is supported with specific dialogue windows (e.g. in Chemcad™) or with specific coloured graphical elements (e.g. in AspenPlus™ the icon of a specific model changes from red to blue if they are fully specified). Furthermore, all tools are supported by a help function. But, the generation and programming of new model blocks requires modelling experts which usually are not familiar with the knowledge of planning engineers. These modelling experts are generally not available in small and medium sized companies.

2.3.2 Equation-Oriented Tools for Process Domain

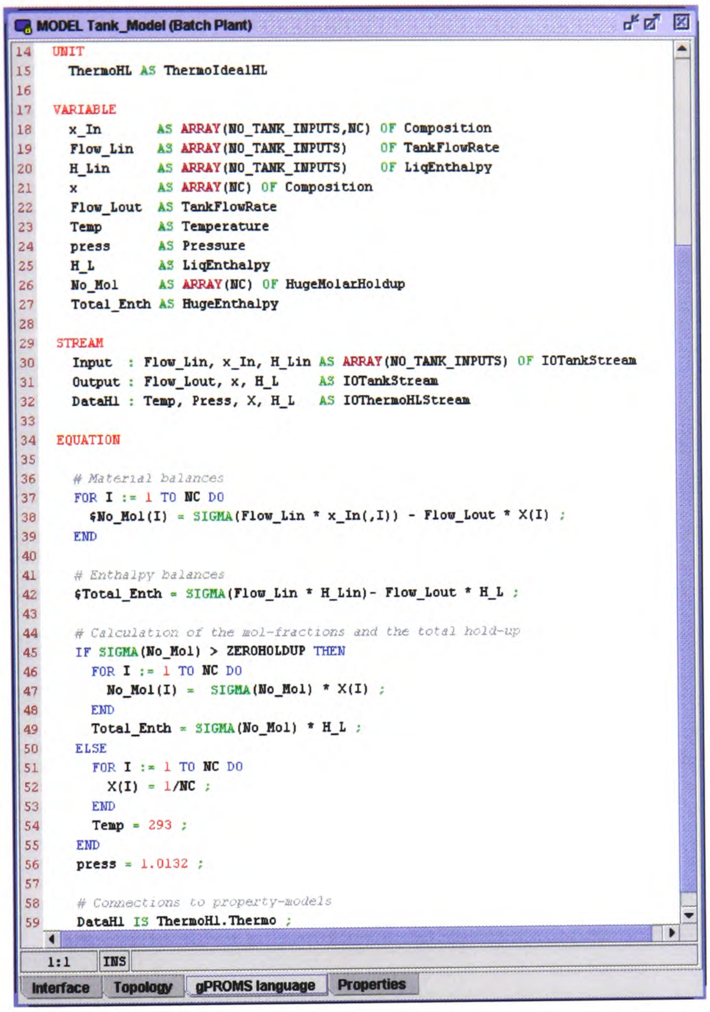
As opposed to process flowsheeting tools, equation-oriented approaches for the process domain provide graphical and textual modelling for the description of mathematical models, including dynamic behaviour. Their simulation strategy is generally simultaneous-oriented.

In process engineering several equation-oriented tools are available. SpeedUp (Sargent and Westerberg, 1964, Perkins and Sargent, 1982), gPROMS™ (Barton and Pantelides, 1993) and ABACUSSII (Barton, 2000) were developed at the Imperial College of London. In the meantime SpeedUp (later renamed in Aspen Custom Modeller™ (AspenTech, 2004)) and gPROMS™ (PSE, 2003b) are commercial products. Further examples of equation-oriented tools are company in-house solutions, such as Chemasim (Hegner and Schoenmakers, 1985) and Optisim (Burr, 1993).

All these tools are predominantly used during the conceptual phase using generic models. Some of these tools are used also as the basis for operator training simulation tools, such as gPROMS™ and Aspen Custom Modeller™.

All tools come with a limited number of standard models stored in model catalogues. Hence, the modelling effort requires also in this case profound knowledge in areas such as modelling and simulation, numerical mathematics and computer science (Marquardt, 1996).

Fig. 2-4 shows an example of an equation-oriented model to demonstrate the large programming effort which is required to define a model (exemplarily demonstrated for a tank model) for use in gPROMS™.



```
UNIT
  ThermoHL AS ThermoIdealHL

VARIABLE
  x_In        AS ARRAY(NO_TANK_INPUTS,NC) OF Composition
  Flow_Lin    AS ARRAY(NO_TANK_INPUTS)    OF TankFlowRate
  H_Lin       AS ARRAY(NO_TANK_INPUTS)    OF LiqEnthalpy
  x           AS ARRAY(NC) OF Composition
  Flow_Lout   AS TankFlowRate
  Temp        AS Temperature
  press       AS Pressure
  H_L         AS LiqEnthalpy
  No_Mol      AS ARRAY(NC) OF HugeMolarHoldup
  Total_Enth AS HugeEnthalpy

STREAM
  Input  : Flow_Lin, x_In, H_Lin AS ARRAY(NO_TANK_INPUTS) OF IOTankStream
  Output : Flow_Lout, x, H_L     AS IOTankStream
  DataHl : Temp, Press, X, H_L   AS IOThermoHLStream

EQUATION

  # Material balances
  FOR I := 1 TO NC DO
    $No_Mol(I) = SIGMA(Flow_Lin * x_In(,I)) - Flow_Lout * X(I) ;
  END

  # Enthalpy balances
  $Total_Enth = SIGMA(Flow_Lin * H_Lin)- Flow_Lout * H_L ;

  # Calculation of the mol-fractions and the total hold-up
  IF SIGMA(No_Mol) > ZEROHOLDUP THEN
    FOR I := 1 TO NC DO
      No_Mol(I) =  SIGMA(No_Mol) * X(I) ;
    END
    Total_Enth = SIGMA(No_Mol) * H_L ;
  ELSE
    FOR I := 1 TO NC DO
      X(I) = 1/NC ;
    END
    Temp = 293 ;
  END
  press = 1.0132 ;

  # Connections to property-models
  DataHl IS ThermoHl.Thermo ;
```

Figure 2-4: Example of equation-based approach for process domain (gPROMS™)

Additionally, the model aggregation requires profound knowledge of system- and object-oriented modelling. The development of new models is therefore restricted to a small group of experts. In the latest versions of gPROMS™ (PSE, 2006) the user can get support from a block-oriented model aggregation, however, this new feature is not as intuitively usable as the model aggregation of process flowsheeting tools.

The connection from CAE-plant design tools to equation-oriented modelling and simulation for the process domain has not been shown in literature. Therefore, an automatic model aggregation based on CAE-plant design tools is not available.

All equation-oriented tools investigated for process domain offer only a limited number of control models and limited support in designing sophisticated control schemes. However, some tools offer possibilities for co-simulation of process and control by standardised interfaces.

gPROMS™ provides co-simulation with Simulink™, see section Section 2.4.1, by the interface gO:Simulink™ (PSE, 2003a), and also SpeedUp was linked to Simulink™ in order to simulate and control cryogenic separation and liquefaction processes, see Mandler (2000). Mandler also describes the integration of AspenPlus™ which is used to develop block diagrams for steady-state simulation. This combination of block-oriented and equation-oriented tools is defined by Chen and Stadtherr (1985) as "mixed-mode" approach. Tolsma *et al.* (2002) outlines the possibility to interconnect ABACUSS and Matlab™, however, without detailed information about this interconnection.

Within Chemasim the user is supported in the specification of boundary conditions by a direct counting of the degrees of freedom. Tools, such as gPROMS™ or ABACUSS, provide support only after the simulation is started: Their solvers count the number of variables and equations at time zero and list incorrectly chosen boundary conditions. In gPROMS™ (PSE, 2003b) even alternative specifications are suggested. Nevertheless, expert knowledge is required for these specification procedures.

Detailed reviews of equation-oriented process tools are given by Biegler (1989) and Bogusch (2001).

2.3.3 Knowledge Based Modelling Approaches

In the past, the large variety of chemical process units and physical-chemical phenomena as well as increasing requirements on the sophistication of models led to an increasing interest in knowledge-based modelling environments. The general idea of knowledge-based modelling environments is a methodology to lift the equation-oriented modelling from the detailed mathematical representations to the level of process knowledge (in general by using phenomenological building blocks) (Bogusch *et al.*, 2001). The support for the modeller takes place by suitable dialogues, provision of information and modularisation of process models below the model unit level. Fig. 2-5 shows a possible decomposition in plant sections, the process units (such as the reactor), which itself can be decomposed into building blocks with increasing level of detail.

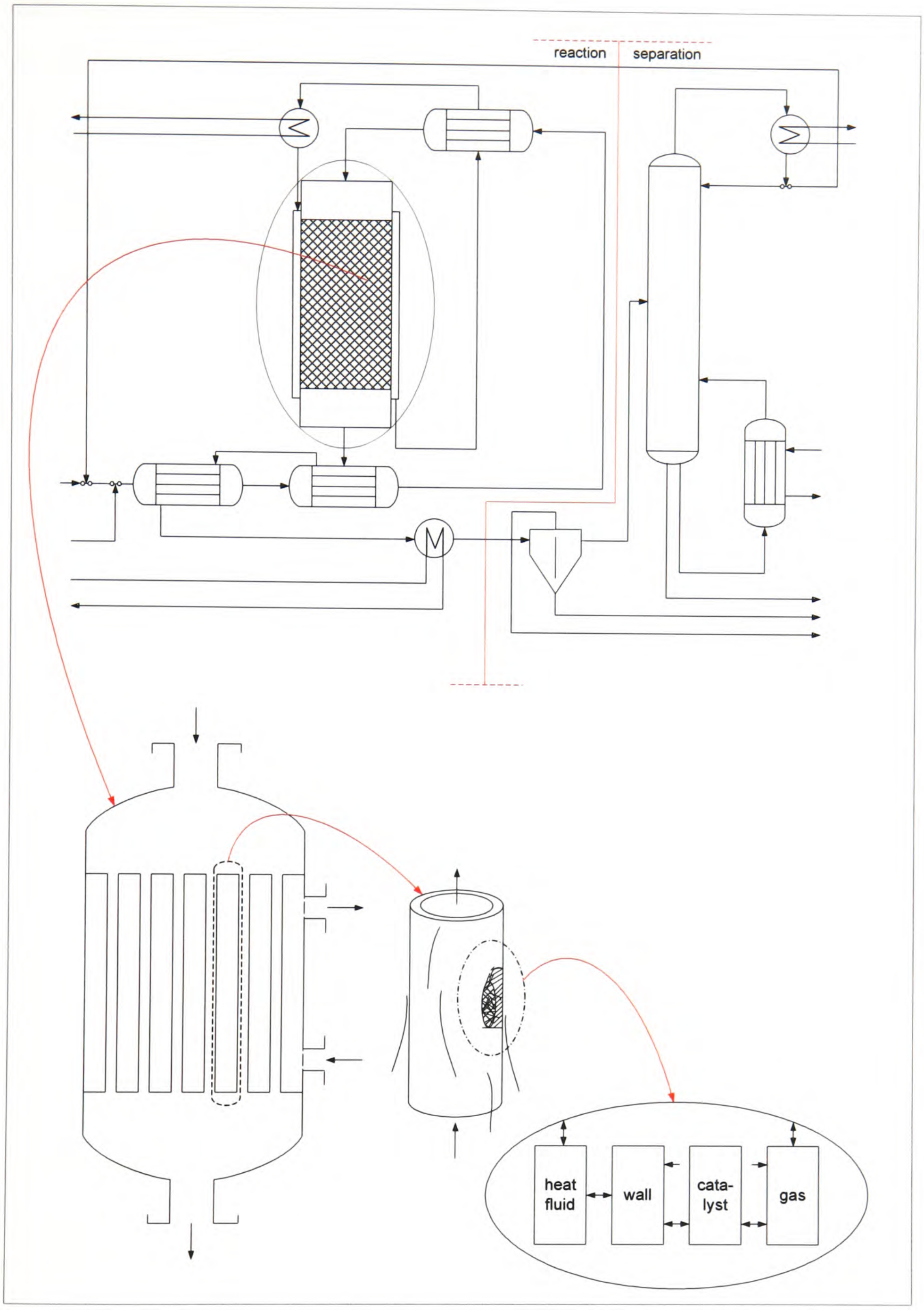


Figure 2-5: Model composition based on phenomenological building blocks of knowledge based modelling approaches

The modularisation aspect of process models has been investigated by several authors: Stephanopoulos *et al.* (1987), Marquardt (1992), Perkins *et al.* (1996) and Preisig (1996). All knowledge based modelling tools contain a general systematic methodology for structuring the model with respect to later reuse and modification. They provide two types of elementary units for giving a structural description of all kinds of chemical processes, called components (or devices) and coupling elements (or connections).

Based on such general structuring concepts, modelling tools and modelling languages have been developed which permit a modular and highly structured model formulation. Examples are Model.La (Stephanopoulos *et al.*, 1990), ModDev (Jensen and Gani, 1996), the modelling tool of Drengstig *et al.* (1997), Modeller (Westerweele *et al.*, 1999), TechTool (Linninger and Krendl, 1999), ProMot (Tränkle *et al.*, 2000) and ModKit (Bogusch *et al.*, 2001). All these modelling tools provide the development of model libraries based on simple, flexible and reusable modelling units (called fundamental building blocks, see Jensen and Gani (1996)). Some of these tools are established in integrated environments, as described in Section 2.6.

All knowledge based modelling tools discussed above are predominantly dedicated to conceptual design during the plant life cycle. The development of new processes is the main emphasis of these tools. Based on their generic model library, these tools are used for simplifying and supporting plant modelling from the very beginning. In order to extend the generic model catalogues of knowledge based modelling tools, the users have to be experts in the field of process engineering and even experts in mathematical modelling. During model design, these tools offer solution steps to be approved or rejected by the modeller, which provides expertise and knowledge in the related domains. Only ModKit offers an approach to support less experienced modellers by suitable guidance on request, which is outlined by Dömges *et al.* (1996).

All knowledge based tools investigated do not have any interconnections or model transfer possibilities to CAE-plant design tools so far. However, within the ModKit approach the user has the possibility to use ModKit's flowsheet generator in order to transfer the designed models to the equation-oriented simulation tools SpeedUp and gPROMS™, see Bogusch (2001).

Due to their direct dedication to the process domain, control tasks respectively the design and use of control models do not belong to the main tasks of knowledge based modelling approaches. Therefore, a separation into process and control domain is not desired. Model.La, ModDev and ModKit use external simulation tools, such as gPROMS™ or SpeedUp, but co-simulations with modelling/simulation tools of other domains have not been found within knowledge based modelling approaches.

The support of the user in specifying boundary conditions was realised within ModKit (Bogusch *et al.*, 2001), Model.La in Design-Kit (Stephanopoulos *et al.*, 1987) and ModDev (Gani *et al.*, 1997) with so-called incidence matrices. A detailed description of the analysis of degrees of freedom with the incidence matrix methods is given in Bogusch *et al.* (2001). However, expert knowledge is required to use and analyse such features.

2.4 Modelling and Simulation Tools for Control Domain

The control domain is mainly dominated by block-oriented modelling and simulation tools. These tools are reviewed in the following, and is followed by a review of text-oriented modelling and simulation tools for the control domain.

2.4.1 Block-Oriented Tools for Control Domain

In control domain, block diagrams composed of several blocks are used for a clear and structured representation of dynamic systems. Therefore, the physical behaviour of a dynamic system is split into single unidirectional blocks with inputs and outputs which are connected by signals. The use of signals is generally characteristic for control domain approaches, different from the process domain approaches, see Section 2.3. Approaches in control domain make use of signals (one definitive single stream state), as presented in Fig. 2-6. The blocks are solved in general separately in a sequential manner.

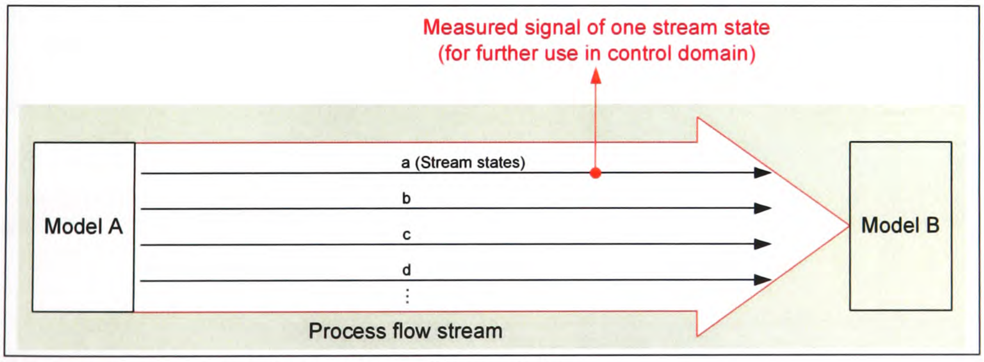
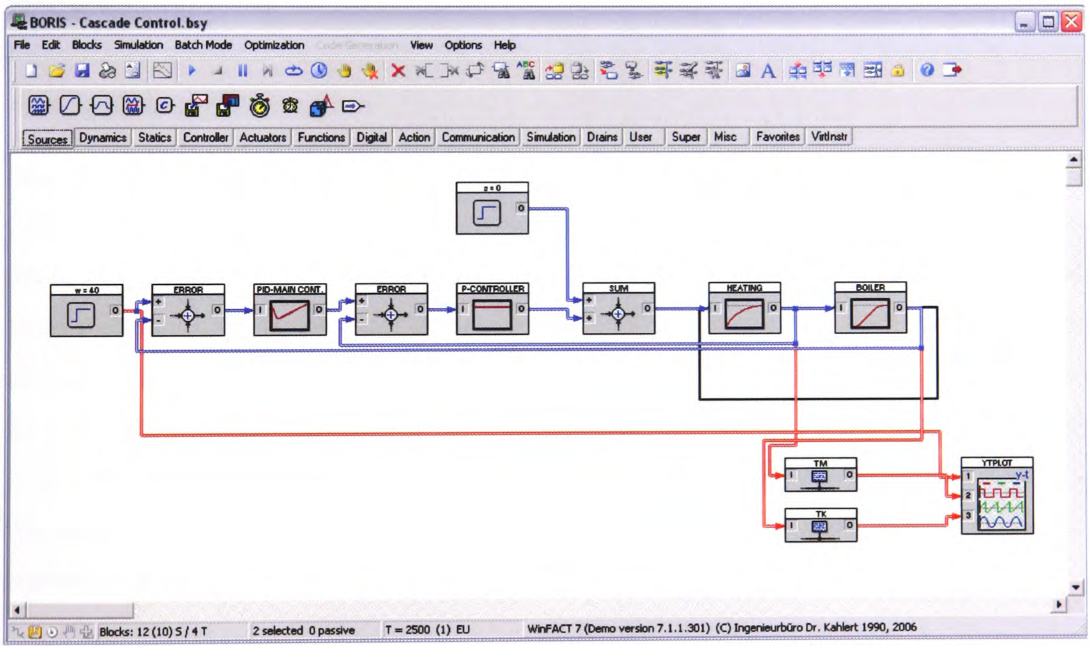


Figure 2-6: Measured signal from stream state

A variety of block-oriented simulation tools have been developed. Examples are Vissim, introduced 1990 by Darnell and Kolk (1990), FSIMUL (Gebhardt, 1990), DORA (Kiendl and Kahlert, 1991), PSI (Van Den Bosch, 1993), BORIS™ (Kahlert, 2006), Labview™

(National Instruments, 2003) or Simulink™ (Mathworks, 2003b), originally introduced by Grace (1991) as Simulab. Similar to the main features of Simulink™, an open source software is provided by the Scilab® Consortium (INRIA, 2007a), called SCICOS.

Block-oriented control simulation tools have generally in common that they are used during early design phases or for optimization procedures during real start-up and operation. These tools support a wide range of control system models, generally basic blocks, but also ranges of scopes, sinks, linear and non-linear transfer functions, etc. Exemplified for the block-oriented approach in the control domain, Fig. 2-7 illustrates a block scheme of BORIS™ (Kahlert, 2006). Process models within these tools are restricted to some general models and assume expert knowledge in mathematical modelling in order to parameterise such generic models. The generation of so-called "user-defined" function blocks is feasible but requires additional programming effort using specific programming languages, such as C©, Fortran or vendor specific languages.



Blocks: 12 (10) S / 4 T   2 selected 0 passive   T = 2500 (1) EU   WinFACT 7 (Demo version 7.1.1.301) (C) Ingenieurbüro Dr. Kahlert 1990, 2006

Figure 2-7: Example of block-oriented tools for control domain (Boris™)

The transfer of control schemes (including process blocks) from CAE-plant design tools to the block-oriented modelling and simulation tools for control domain is in general not available. Therefore, Geitner (2001) describes the interconnection from function block diagrams to Simulink™ by a self written toolbox, and also Labview™ (National Instruments, 2003) offers such a feature. Plant modelling and simulation systematically separated into process and control domain is only available for Simulink™ and SCICOS. Simulink™ provides the integration of gPROMS™ blocks within its block structure, as described in Section 2.3.2 and SCICOS allows the partial support of Modelica as described by Najafi *et al.* (2005).

In block-oriented tools for control domain, models are mostly pre-parameterised with basic settings. In order to change such parameter by the user, specific knowledge in control and mathematic understanding is required. A GUI for an "active" support is missing, however, in nearly all mentioned tools comprehensive help functions are available.

Detailed reviews of block-oriented simulation tools used in the control domain are given by Cameron (1983) or Aström (2003).

2.4.2 Text-Oriented Tools for Control Domain

Before block-oriented tools for control design became popular, text-oriented tools were predominantly used within the control domain. In the meantime, many of these tools have been enhanced by graphical visualisation. However, the text-oriented origin of these tools remain clearly visible.

One of the first text-oriented tools for the control domain is ACSL (Advanced Continuous Simulation Language), introduced by Mitchell and Gauthier (1976). It is based on the CSSL (Continuous System Simulation Language) standardisation, which was a milestone within the modelling and simulation development (Strauss *et al.*, 1967). ACSL was one of the most dominant continuous modelling and simulation tools. The Graphic Modeller toolbox for ACSL supports the user with graphical visualisation, see AEgis (2004). In Barker *et al.* (1991b) and Frederick *et al.* (1991), ASCL and TSIM 2 code were used to establish a library of process/plant simulation models for the widespread use in the UK control community. Several specific application examples including linear and non-linear models had been integrated into this model library.

One further example of text-oriented tools in control domain is Simnon (Elmqvist, 1977). Simnon is a command driven modelling and simulation tool developed at the Lund Institute of Technology, Sweden. Matlab™ (Mathworks, 2003b) is also a text-oriented mathematical and simulation tool with special emphasis on matrix manipulation, plotting of functions and data, implementation of algorithms and creation of user interfaces. One of Matlab's™ well known toolboxes is Simulink™, as reviewed in Section 2.4.1. These tools are predominantly used during early life cycle phases, but also for start-up and operational phase. Scilab® is an equivalent, but open source software, to Matlab™, and serves as a platform for other toolboxes, such as SCICOS, provided by the Scilab® Consortium (INRIA, 2007b).

All tools offer a small range of standard generic models for the control and even less models for the process domain, or in the case of Barker *et al.* (1991b) and Frederick *et al.* (1991) restricted to very specific process/plant models. For the development of new process models also profound knowledge is required, especially in mathematic modelling. The automatic integration of control schemes or the direct conversion of flowsheeting information based on CAE-plant design tools is in general not available. Matlab™ is the only tool which offers interconnections to the equation-oriented modelling and simulation tool ABACUSS (see Section 2.3.2) or to the integrated CAE environment BASIS, see Section 2.6.4. A support during the specification of boundary conditions and simulation parameters is not available in Matlab™, while for Simnon and ACSL any information about these features were not applicable from literature.

For detailed reviews on text-oriented simulation tools in the control domain see Cameron (1983), Cellier (1993) or Aström (2003).

2.5 Domain Independent Modelling Tools

Domain independent modelling tools, which cover different application areas, can be regarded as further developments of equation-oriented simulation tools for the process domain. Omola (Andersson, 1990), ASCEND (Piela *et al.*, 1991), Moses (Maffezzoni and Girelli, 1998) and Modelica (Elmqvist *et al.*, 1998) are some examples of general modelling languages which are based on object oriented data structures. The main goal of Modelica is the integration of the advantages of different existing languages and to provide a standard for the exchange of models between different applications. Modelica is only a modelling tool, several implementations to other simulation packages are available, see Modelica Association (2007). The most widespread implementation of Modelica is within Dymola (Elmqvist, 1978 and Dynasim, 2003). While ASCEND, Omola and Moses are textual-oriented, Modelica allows to design models textually and graphically (in a block-oriented way). The simulation strategies of all tools are simultaneous.

The main use of these tools is the design of new processes during conceptual design. Apart from some general examples in the libraries of Omola and ASCEND, the libraries within Modelica are suited to domains such as robotics, automotive or aerospace engineering and offer only some models for the process domain. Additionally, Nilsson (1989) created a model catalogue for the use in Omola with a very limited number of process and control models. Moses supports the user additionally with a range of mechatronic models.

All domain independent modelling tools provide the user with generic models. The integration of new models is quite difficult as in equation-oriented approaches for process domain, because all tools have their individual language characteristic. A strict separation of process and control domain is not available, only Modelica offers the co-simulation with Simulink™ (Otter and Elmqvist, 2001). Moses offers a clear systematic separation between the mechatronic and control domain; a separation between process and control domains is not available. In an additional paper, Maffezzoni *et al.* (1999) present a concept of the interconnection of a function block diagram design and simulation tool, called FBCad (Carpanzano *et al.*, 1998), but restricted to robotic systems.

In all four tools the model transfer of flowsheet information from CAE-plant design tools is missing. Furthermore, all tools support methods to analyse and reduce the index of differential equations, however, an active support during parameterisation of boundary parameters cannot be found in literature. The only exception is ASCEND. ASCEND offers help in debugging facilities which supports the user during parameterisation of boundary in order to reduce the degrees of freedom, see Piela *et al.* (1991).

In the following section integrated CAE environments are investigated.

2.6 Integrated CAE Environments

Over the last two decades the development of CAE-environments has become a major activity in chemical industries. Integrated CAE environments aim at the use of different tools and environments (e.g. modelling and simulation tools, design utilities, data repositories or management tools), even from different vendors, in order to achieve the standardisation of work flows, incorporating design and operational issues.

For tool integration different viewpoints of requirements for the development of process models have been distinguished, see e.g. Barker *et al.* (1993b) or Pohl *et al.* (1996):

- User interface integration (unique look and presentation of all tools)
- Platform integration (enabling communication and file access across system boundaries)
- Data integration (establishing common data exchange formats)
- Control integration (enabling service calls across tools)
- Process integration (guiding the usage of tools and the call of services according to explicit modelling process definitions)

Data models which intend to integrate all the required information in a standardised way have been developed by several research groups in recent years, see e.g. Jarke *et al.* (1999) or Bayer *et al.* (2000). Examples for these standards include e.g. STEP (Standard for the Exchange of Product Data, (International Organization for Standardization, 1994) or STEP-based approaches such as PDXI protocols (Process Data Exchange Institute (International Organization for Standardization, 2005)) for data integration. The CAPE-OPEN standard, which is summarised in the paper of Braunschweig *et al.* (2000) allows communication between software components from different sources (software and equipment vendors, universities and "self made"). The CAPE-OPEN standard is maintained by the CAPE-OPEN laboratories network (CO-LaN, 2006) to date. As part of data models, mathematical process models may be integrated for later use in modelling and simulation, as outlined by Barker *et al.* (1993b) or Bayer and Marquardt (2003).

Academic projects in this area include efforts to support the early design phases (conceptual design and the early phases of basic engineering). Surveys of integrated environments are described in Biegler *et al.* (1999) or Van Schijndel and Pistikopoulos (2000).

In the following some of the integrated environment tools are surveyed in detail with regard to their potential use in virtual start-up simulations based on the results of plant design.

2.6.1 épée and n-dim

A prototype for application integration is épée (Ecosse Process Engineering Environment developed by the University of Edinburgh, see Ballinger *et al.* (1994). It was developed to support process engineering in general, however with emphasis on the conceptual phase. The épée system is based on a modular architecture with the speciality that a server coordinates all requests from applications in a neutral form, see therefore Costello *et al.* (1996). Internal applications such as the design support system KDBS (Banares-Alcantara, 1995) or the flowsheet generator CHiPS (Fraga and McKinnon, 1994) are integrated as well as external tools such as SpeedUp or AspenPlus™ (Ballinger *et al.*, 1995). The main emphasis of épée is laid on the integration of KDBS for the design and decision support in the management of the design process.

A similar approach to épée is n-dim (n-dimensional information modeling) which is a design support environment developed at Carnegie Mellon University, see Levy *et al.* (1993). n-dim is a generic environment, permitting users to tailor the environment to suit their specific needs and to support a cross-disciplinary design team more effectively. In principle, any kind of information created during any phase of the plant life cycle can be represented within n-dim, even simulation models, which is realised with ASCEND, the general modelling and simulation tool, see Westerberg *et al.* (1997).

In both tools the data handling and storage of different kinds of data models are discussed including generic mathematical simulation models; however, their main emphasis is to organise the engineering process and to support its workflow. A systematic integration of simulation models or initialisation from CAE-plant design tools is not available. Furthermore, a separation into process and control domain within épée and n-dim is missing. The support during parameterisation of simulation models and during the specification of boundary conditions and simulation parameters is dedicated to the corresponding modelling and simulation tool. For ASCEND which is used in n-dim see Section 2.5 and for AspenPlus™ and SpeedUp which are used in épée, see Section 2.3.1 respectively Section 2.3.2.

2.6.2 Marquardt's Team, IMPROVE

Since 1997 the interdisciplinary collaborative research centre (IMPROVE) led by Marquardt (RWTH Aachen University) has set the focus on new concepts and software engineering solutions to support collaborative engineering design processes (Marquardt and Nagl, 2004). This support comprises an integrated software environment for the design, mathematical modelling, simulation and management of chemical processes. Research activities are concentrated on the early phases (conceptual design and basic engineering) of the plant life cycle.

Driven by the idea to integrate tools from different vendors, the concept comprises very different software tools, commercial as well as IMPROVE's in-house tools, as illustrated in Fig. 2-8.

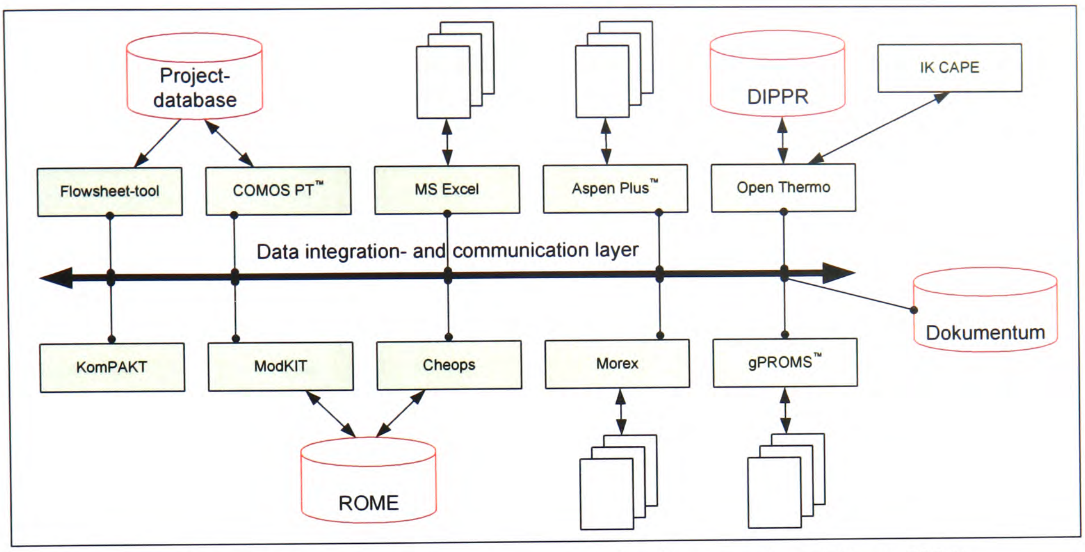


Figure 2-8: IMPROVE's integrated environment (after Marquardt and Nagl, 2004)

The approach contains, for example, process flowsheeting and equation-oriented modelling and simulation tools, such as Aspen Plus™ or gPROMS™ or Modelica, see Marquardt and Nagl (2004). It also contains dedicated simulation tools (e.g. Morex (Schlüter and Haberstroh, 2002)) for the simulation of extrusion processes. Furthermore, IMPROVE's in-house tools, as the knowledge-based ModKit, see Section 2.3.3, and Cheops (Component-based hierarchical explorative open process simulator) are implemented (von Wedel, 2003). Cheops allows the aggregation of models from different sources to a single flowsheet and the integration during runtime using existing dedicated simulators, such as Modelica or gPROMS™.

These tools make use of the CORBA object bus using the CAPE-OPEN standard interfaces, see von Wedel and Marquardt (2000). Apart from the integration of physical property databases the IMPROVE environment includes a storage platform for mathematical models, called ROME (Repository Of a Modelling Environment). ROME stores symbolic models in a neutral format (which is based on a system theoretical idea, outlined in Bayer *et al.* (2000)) or in any proprietary format of commercial simulators, in form of declarative equation-oriented models as well as executable block-oriented models; for more information see von Wedel and Marquardt (2000). All models stored in the model repository ROME are generic and dedicated to the conceptual phase of plant life cycle.

Additionally, this integrated environment supports the central storage of product data in the database of a CAE-plant design tool (in this case COMOS PT™ from Innotec) during a design project. In Becker *et al.* (2002), an interconnection from the CAE-plant design tool

COMOS PT™ to the block-oriented simulation tool ASPEN Plus™ is presented based on the general model framework CLIP (Conceptual Life cycle Process model) (Bayer *et al.*, 2001). CLIP takes existing standards such as PDXI or PISTEP into account with the intent of getting standardised interfaces for the whole plant life cycle. The established interface between COMOS PT™ and ASPEN Plus™ is based on the use and development of PFD. PFDs are limited in their level of detail. They mostly contain process units and a very limited representation of control functionality. The representation of every single process and control component cannot be described by PFDs. Furthermore, the data exchange from the PFD to ASPEN Plus™ is done only at the process domain level. For the design/engineering at the end of detailed engineering, the P&I diagram is essential, whereas the PFD is favoured during earlier engineering phases. A systematic consideration of the control domain, respectively a systematic separation of the process and control domain for the later start-up simulation is missing.

By the combination of a management system, which is used to serve as an archive for all design documents (Heller *et al.*, 2004) and a process data warehouse (Jarke *et al.*, 2000), the IMPROVE concept aims to produce accurate and timely management information and support data analysis through the whole plant life cycle. However, the data warehouses are dedicated to the storage of management and design information in order to improve the design activities themselves, and not for the storage of simulation models.

Summarising, it can be said that the IMPROVE concept implies a comprehensive environment for the support of the design process in the chemical process domain. However, all tools discussed use generic models in early design phases (conceptual and early basic engineering). The extension of this integrated environment with component models for use at the end of detailed engineering is conceivable, but not the emphasis of the IMPROVE approach, see Marquardt and Nagl (2004). Furthermore, all tools are designed for the support of model experts and for the exchange of data and documents between them, requiring profound knowledge of process, modelling and simulation domains.

Also the separation of process and control modelling simulation has not been a task of the IMPROVE concept, only Scharwaechter *et al.* (2002) started considering the integration of control simulation tools, such as Matlab, but it has not been realised within the IMPROVE concept so far.

2.6.3 ICAS

Another integrated CAE environment, named ICAS, was proposed by the Lund University by Gani and co-workers. ICAS combines computer-aided tools for modelling, simulation (including property prediction), synthesis/design, control and analysis into an integrated system (Gani *et al.*, 1997).

For modelling tasks, ICAS makes use of either the knowledge based tool ModDev, see Section 2.3.3, or the equation-oriented modelling tool MoT, see Sales-Cruz and Gani (2003). An interface to the simulation tool Dynsim is realised by a COM interface. Further toolboxes, such as the thermodynamic, synthesis, analysis, design and control toolbox are introduced in the paper of Gani and co-workers, see Gong *et al.* (1995). ICAS is dedicated to the conceptual phase of the plant life cycle.

Each toolbox solves a specific set of problems and is able to communicate with all other tools of ICAS using a main library, called "knowledge base", which includes specific models in the field of distillation, separation, reactors and crystallisation. It is possible to invoke the simulation engine to perform steady state and/or dynamic simulation for batch and/or continuous process operations from any toolbox.

The integration of CAE-plant design tools into the ICAS framework is not done nor considered. In the paper of Gani *et al.* (1997) the interrelation between process and control domains within ICAS is outlined. Here, process models have to be simplified (linearised) for use in the control toolbox (for process controllability and process sensitivity analysis and verification of the controller performance). The control functionality is limited to some general functions.

The ICAS environment is only dedicated to users with expert knowledge in the field of process and control modelling and simulation domains, even though a knowledge-based user interface for model preparation for later simulation is available. ICAS supports the user with the specification of boundary conditions, see Gani *et al.* (1997), however, detailed information about this feature is not available. The final version of the ICAS environment with special emphasis on molecular modelling is described in CAPEC (2004).

2.6.4 BASIS

The BASIS (Batch Simulation and Scheduling) approach of the University Dortmund is one example of an integrated environment for modelling and simulation of batch plants (Fritz and Engell, 1997). The environment contains four different editors, one with special emphasis on plant modelling for designing plant layout and one with special emphasis on recipes based on function block diagrams, three different simulators, one for the continuous, one for the discrete-event case and gPROMS™ to provide additional numerical solvers. The central element is called "Leitstand", operating as user interface and coordinates the modelling task for the respective simulator. Matlab™ (Mathworks, 2003b) is only used for the visualisation of simulation results.

Although BASIS has integrated plant modelling, the main emphasis is laid on recipe-driven production and product planning. Therefore, the model library contains only a very limited number of basic components, which are generic and dedicated to the conceptual phases of the plant life cycle. Additional models must be written in programming code "C++©" (Microsoft, 2003) which requires profound knowledge and modelling experts. Furthermore, a specific guidance for the user is not available, in order to parameterise the models, boundary and simulation parameters. The connection to CAE-plant design tools is not considered.

An overview of existing simulation tools for operator training is given in the following section.

2.7 Operator Training Simulation Tools

Due to the increasingly improving power to cost ratio of modern computers, operator training simulation using dynamic simulation tools is becoming more accessible and is gaining acceptance within the chemical industry, see Stawarz and Sowerby (1995).

As implied by their name, the purpose of operator training simulation tools is the training of operator personnel in operating the process, using a simulation of the plant's process and control system. Furthermore, these tools are used for specific process and/or control training and education not only for operators, but also for engineers and maintenance staff, as outlined in Cameron *et al.* (2002).

At a real plant, the operators communicate with the process through a process control system (either a DCS (Distributed Control System) or a SCADA (Supervisory Control and Data Acquisition system, where the process controllers are implemented by a number of PLCs (Programmable Logic Controllers)).

Depending on the process control system, two systems are distinguished for an operator training system:

- Stimulated systems
- Emulated systems

Stimulated systems contain a process simulator combined with a real process control system, see e.g. Gilles *et al.* (1990). Emulated systems make use of a process simulator and a control system simulator. Here, two different approaches can be found in literature, the use of a control simulator, see e.g. Sulc (2004) or a one-to-one emulation of a process control system (DCS or SCADA), see e.g. Lee *et al.* (2000) or Krause (2003). In Fig. 2-9, both types are presented, on the left the stimulated and on the right the emulated training system. A mix of both training systems is outlined in Ye *et al.* (2000), also called *quasi*-stimulated approaches, see Cameron *et al.* (2002). The main use of operator training simulation tools is during start-up phase and operation.

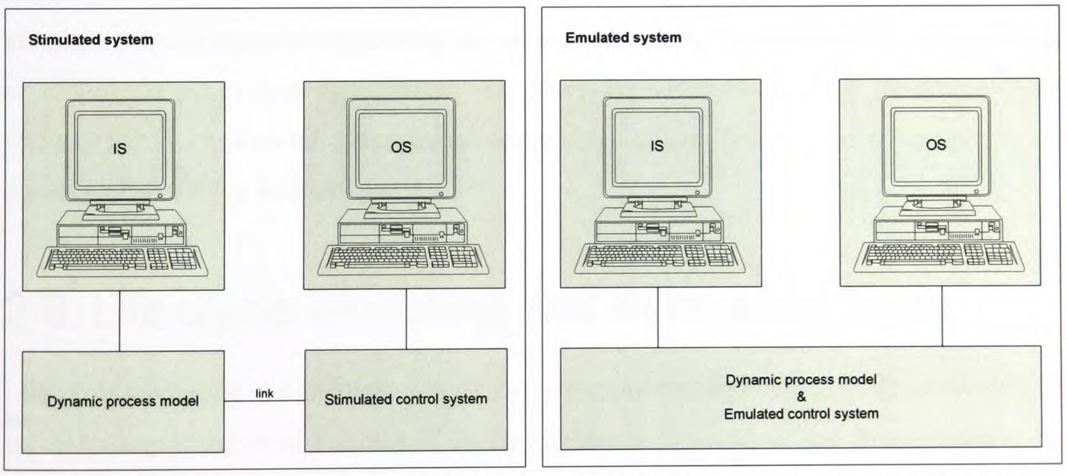


Figure 2-9: Two types of operator training simulation tools (left: stimulated system; right: emulated system) including IS (Instructor System) and OS (Operator System)

For the process simulation tool, either process flowsheeting tools, like in Kröner *et al.* (1990) and Holl (1994) or equation-oriented tools, such as SpeedUp (Kothe *et al.*, 1995) can be found for operator training simulators. Gilles *et al.* (1990), Kröner *et al.* (1990) and Holl (1994) describe their experiences with the process simulation environment DIVA, which is predominantly used as process simulator in operator training systems. Unlike the modelling and simulation approaches during design phases, process modelling and simulation for operator training does not aim to reflect the plant's behaviour with total fidelity, see Jaako (1998). Stawarz and Sowerby (1995) outlined that the models accuracy should lie within 2 % for critical and 10 % for non-critical parameters. It is more important for the operator training to run in real time rather than representing perfect process behaviour. Furthermore, companies such as Hyperion (2003) and Bayer (2003) have specialised in custom built operator training systems. However, these tools do not offer general solutions, and these companies have specialised in customer specific simulators for specific plants.

The integration of CAE-plant design tools has not been within the scope of literature references regarding operator training simulation tools. However, due to the clear separation of process and control domain (mainly through the interconnection of the process simulator to the real or emulated process control system), a strict separation of the modelling environment is realised. Many tools are available which model process and control inside the same tool, see e.g. Lee *et al.* (2000) or Sulc (2004). Integrated tools such as Karhela (2002) or Krause (2003) combine a process simulator with an emulator of a process control system.

Experts must be available to fulfil the modelling work especially for the process domain, modelling the process dynamics (including the sensor and actuator representations), see e.g. Ylén *et al.* (2005). Details of these specification procedures are not available. Only the process simulation tool DIVA supports the user with the specification of consistent initial values, see for more details Holl (1994) or Kröner (2003).

2.8 Life Cycle Modelling and Simulation Tools

Life cycle modelling and simulation tools, as a subset of the entire process life cycle and different in its objectives from Life Cycle Analysis, aim at the support and documentation of all phases of the process life cycle. It is proclaimed by several authors, such as Goldfarb and Bradley (1995) or Cameron (2005) that life cycle tools can be used for transmitting and retaining knowledge from conceptual design, through all design phases and start up to process operation. Furthermore, they can be used to transfer experience from operation back into the design of future plants and the retro fitting of existing plants, as illustrated in Fig. 2-10.

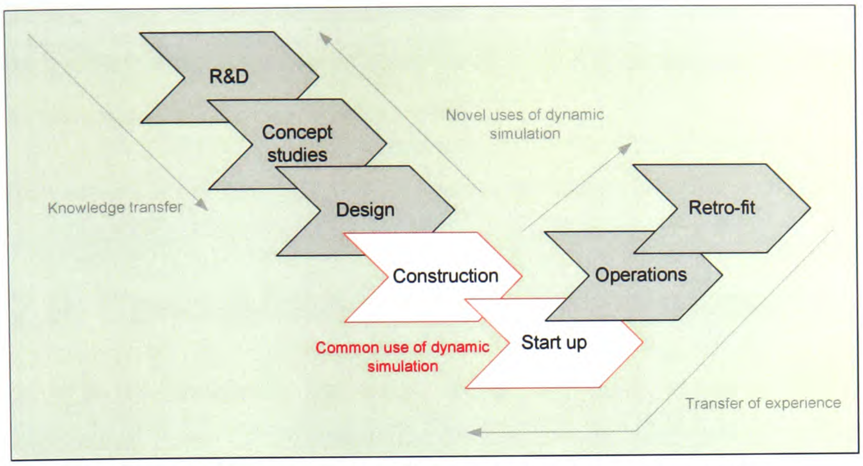


Figure 2-10: Simulation in plant life cycle (after Cameron, 2005)

Life cycle modelling and simulation implies, however, consistent and holistic standardisation through all phases of the plant life cycle and with all cooperating partners. Marquardt *et al.* (2000) indicate that the establishment of such a comprehensive approach results in a huge investment and the standardisation procedure is very complicated and time-consuming.

Goldfarb and Bradley (1995) introduced the PROTISS™ approach, a combination of two block-oriented simulation tools, the steady-state simulator ProII™ and the dynamic simulator OTISS™. ProII™ serves as a pre-simulator in order to simulate the steady state behaviour of the plant based on internal PFDs. It provides the initial conditions for the dynamic simulation in OTISS™. But, the simulation is restricted to the process domain only. Models for the control domain, including instrumentation are missing.

The commercial plant and process simulation system SIMIT™ promises a comprehensive approach (Seybold, 2000). Process simulation tools, used in SIMIT™, represent the behaviour of technological processes based on equations. SIMIT™ is used for operator training as well as for support during conceptual design, basic and detailed engineering (Fischer, 2005). During usage, specific simulation knowledge is not required, as long as the user can select from model libraries for some standard components. However, for modelling and simulation of specific components expert knowledge is still required. Additional changes by the user are possible, but require profound knowledge in mathematical modelling and is often only supported by the programming engineers of SIMIT™.

SIMIT^{TM} and the tools of Goldfarb and Bradley do not offer any connections to CAE-plant design tools. However, control data/models of a PLC (Simantic S7) configuration software can be transferred for the later work in SIMIT^{TM}.

Information about specifying the boundary parameters have not been found in the literature.

2.9 Conclusions of Literature Review

In order to summarise the results of the literature review of modelling and simulation approaches, Table 2-1 illustrates the most important criteria under which the considered tools have been reviewed.

All reviewed modelling and simulation approaches, namely the

- Modelling and Simulation Tools for Process Domain,
- Modelling and Simulation Tools for Control Domain,
- Domain Independent Modelling Tools,
- Integrated CAE Environments,
- Operator Training Simulation Tools and
- Life Cycle Modelling and Simulation Tools

have been reviewed with respect to the criteria described in Section 2.2:

- Life Cycle Phases Association
- Modelling and Simulation Issues
- User Support.

Table 2-1: Modelling and simulation approaches applicable in plant life cycle

Issues	Life Cycle Phase Association					Modelling and Simulation Issues					User Support	
Tools/Approaches	Conceptual Design	Basic Engineering	Detailed Engineering	Real Start-up	Operation	Model Catalogue of Generic Models	Model Catalogue of FSCS Models	Model Transfer from CAE-Plant Design Tools	Interface to Modelling/Simulation Tools of other Domains (Co-Simulation)	Plant Modelling/Simulation Separated into Process and Control System Domain	Support during Specification of Boundary Conditions	Usable by Non-Experts
Process Domain - Process Flowsheeting												
Hysys, ProSim, Chemcad	x	x	-	x	x	x	-	-	-	-	(X)	(X)
AspenPlus, ProII, CADSIM Plus	x	x	-	x	x	x	-	x	-	-	(X)	(X)
Process Domain - Equation-oriented												
ABACUSSII	x	x	-	-	-	x^1	-	-	x	-	(X)	-
gPROMS, SpeedUp	x	x	-	x	x	x^1	-	-	x	(X)	(X)	-
Chemasim	x	x	-	x	x	x	-	n/a	n/a	n/a	x	-
Optisim	x	x	-	x	x	x	-	n/a	n/a	n/a	n/a	-
Process Domain - Knowledge Based												
ModDev, Model.La, ModKit	x	x	-	-	-	x^1	-	-	-	-	x	-
TechTool, ProMot, Modeller Drengstig *et al.* (1997)	x	x	-	-	-	x^1	-	-	-	-	(X)	-
Control Domain - Block-oriented												
Vissim, FSIMUL, DORA, PSI, BORIS, Labview	x	x	-	x	x	x^1	-	-	-	-	(X)	-
Simulink, SCICOS	x	x	-	x	x	x^1	-	-	x	(X)	(X)	-
Control Domain - Text-oriented												
ASCL, Simnon, Scilab	x	x	-	x	x	x^1	-	-	-	-	n/a	-
Matlab	x	x	-	x	x	x^1	-	-	(X)	(X)	-	-

Table 2-1: Modelling and simulation approaches applicable in plant life cycle (Continued)

Issues	Life Cycle Phase Association					Modelling and Simulation Issues					User Support	
Tools/Approaches	Conceptual Design	Basic Engineering	Detailed Engineering	Real Start-up	Operation	Model Catalogue of Generic Models	Model Catalogue of FSCS Models	Model Transfer from CAE-Plant Design Tools	Interface to Modelling/Simulation Tools of other Domains (Co-Simulation)	Plant Modelling/Simulation Separated into Process and Control System Domain	Support during Specification of Boundary Conditions	Usable by Non-Experts
Domain Independent												
Omola	X	X	-	-	-	X[1]	-	-	-	-	-	-
ASCEND	X	X	-	-	-	X[1]	-	-	-	-	X	-
Modelica/ Dymola	X	X	-	-	-	X	-	-	X	-	n.a.	-
Moses	X	X	-	-	-	X[1]	-	-	X	(X)	-	-
Integrated Environments												
épée	X	X	-	-	-	X[1]	-	-	-	-	(X)	-
n-dim	X	X	-	-	-	X[1]	-	-	-	-	X	
IMPROVE Concept	X	X	-	-	-	X[1]	-	X	X	-	X	-
ICAS	X	X	-	-	-	X[1]	-	-	-	(X)	X	-
BASIS	X	X	-	-	-	X[1]	-	-	X	-	-	-
Operator Training Systems												
Hyperion, Bayer	-	-	-	X	X	n/a	-	-	X	X	n/a	-
Gilles *et al.* (1990), Kröner *et al.* (1990), Holl (1994)	-	-	-	X	X	X[1]	-	-	-	X	X	-
Sulc (2004), Lee *et al.* (2000), Krause (2003), Ye *et al.* (2000)	-	-	-	X	X	n/a	-	-	X	X	n/a	-
Life Cycle Tools												
SIMIT	X	X	X	X	X	X	-	-	X	X	n/a	-
Goldfarb and Bradly (1995)	X	X	X	X	X	X	-	-	-	-	(X)	-

1.) Model catalogues with a limited number of models.
(X) indicates that this criterion is partly fulfilled.
n/a- Not applicable

Life Cycle Phases Association

Nearly all reviewed modelling and simulation tools, except operator training systems and life cycle approaches, are dedicated to early phases of the plant life cycle, the conceptual and the early basic engineering phase. This is caused by the fact that nearly 80 % of all investment costs are fixed during these early design phases, see McGuire and Jones (1989).

The basic and detailed engineering phase is dominated by the generation of all necessary specifications, documents and manuals for process building, start-up and operation. Little emphasis is laid on modelling and simulation during these phases. Only the approach of Goldfarb and Bradley (1995) and SIMIT^{TM} (Seybold, 2000) is used to check major design changes by simulation during detailed engineering. However, even if basic and detailed engineering could be supported, the main emphasis of these tools are laid on the conceptual and operational phase.

Furthermore, many modelling and simulation tools are available for the (real) start-up and operational phase with generally two different intentions. The first group are the operator training systems whose main emphasis is laid on the education of operators. The process simulators of operator training systems are either process flowsheeting or sometimes equation-oriented modelling and simulation tools of the process domain. The other group of tools used during start-up and operational phases contains modelling and simulation approaches for the control domain. These tools mainly concentrate on the optimization of the control systems on the ready-built plants.

Modelling and Simulation Issues

All modelling and simulation tools make use of generic models, which are commonly stored in model catalogues. Especially the process flowsheeting, the equation-oriented modelling and simulation tools of the process domain and the domain independent modelling tool "Modelica" offers a multitude of generic models, however, none of the reviewed tools supports the integration and use of FSCS models within their model catalogues.

The model transfer from CAE-plant design tools to the reviewed modelling and simulation tools is limited to some process flowsheeting tools (AspenPlus^{TM}, CADSIM Plus^{TM} and ProII^{TM}). As described in Section 2.3.1, the main interest of these approaches is the data transfer from the simulation tool to the CAE-plant design tool in order to use the flowsheet data for the ongoing basic and detailed engineering phases. One of these approaches is used also within the integrated environment IMPROVE.

Several tools offer interfaces to modelling and simulation tools of other domains, however, with different qualities and directions:
The BASIS approach offers an interface to Matlab™ for the visualisation purposes only, but this interface is not used for the separated modelling and simulation. Moses is the only tool which offers co-simulation with a function block diagram design and simulation tool. A clear separation of the mechatronic and control domain is available, but co-simulation with process simulation tools is missing. The integrated environment ICAS and the tools gPROMS™, SpeedUp and Modelica offers co-simulation with Simulink™ and SCICOS. Here, the clear separation into both domains is evident, even if Simulink™ is not able to emulate all functions of a real process control system. Such functions can be simulated with the emulated process control systems (such as Lee *et al.* (2000) or Krause (2003)), which offer co-simulation with a process simulator. However, this co-simulation is only dedicated to the simulation of customer tailored plants for operator training purpose.

User Support

The literature review of modelling and simulation tools during the plant life cycle has shown that all tools are tailored to users with appropriate knowledge in their domains (control domain, process domain, modelling and simulation).

Only the process flowsheeting tools are partially usable for non-experts, as long as only standardised models from the library are selected and aggregated. If the models are custom-tailored, the process flowsheeting and also all other tools require knowledge from the respective domain and especially profound knowledge in mathematics and in modelling and simulation. Expert knowledge is required for model aggregation, parameterisation and setting of boundary conditions. This makes a permanent presence of experts necessary during all modelling and simulation procedures, see e.g. Sundquist *et al.* (2000), and may be realisable in large industrial companies or in scientific research. In small or medium sized industrial companies (planning and engineering companies) such experts are in general not available.

Furthermore, the specification of boundary conditions is supported by several tools, such as Chemasim, ModDev, ModKit, Model.La, ASCEND and DIVA and also the respective integrated environments. However, even in this case profound knowledge is required to use these features. Other tools, such as gPROMS™ and ABACUSS offer indirect solutions for the settings of boundary conditions. Here, the boundary conditions are checked after the simulation is started and displayed in an output window. Support during input and specification of boundary conditions is not available within such tools.

Finally and in summary, modelling and simulation tools/environments usable for virtual start-up procedures based on plant design at the end of detailed engineering are still missing up to date.

- **Objective 1: Development of a Proof of Concept for a Simulation Model Catalogue Required for Virtual Start-Up Procedures Scheduled at the End of Detailed Engineering**
  Nearly all reviewed tools are dedicated to the early phases of plant life cycle making use of generic models. The systematic provision of FSCS models which reflect the physical component's behaviour do not exist to date. Model catalogues which include such models of physical components are not available.
- **Objective 2: Integration of FSCS Models into CAE-Plant Design Tools**
  Within the reviewed tools only some process flowsheeting tools offer the possibility to establish plant models based on CAE-plant design tools. These tools are only dedicated to the PFD with limited information about physical components and process control schemes. Furthermore, a systematic separation of the process and control domain within the CAE-plant design tools is not available. In order to establish virtual start-ups at the end of detailed engineering, a systematic integration of FSCS models into CAE-plant design tools is required.
- **Objective 3: Strategy for the Automatic Generation of Plant Models Based on the Results of Plant Design Making Use of FSCS Models**
  Automatic model aggregation based on this separation in CAE-plant design tools does not exist. A systematic transfer of plant design knowledge separated into the process and control domains and based on the respective design utility is still missing. Co-simulations of process and control simulation tools are available, but neither based on CAE-plant design tools nor on FSCS models.
  Plant utilities, such as P&I diagrams or function block diagrams, which are used during plant design must be utilised as a starting point for model aggregation. A suitable simulation environment, which allows co-simulation must be chosen.
- **Objective 4: Strategy for the Support of the Planning Engineer with a Suitable GUI (Graphical User Interface) to Guide Her/Him from Plant Design to Virtual Simulation**
  All approaches investigated are dedicated to experts with profound knowledge in modelling and simulation domains as well as experts in process or control domains. Generally,

planning engineers do not possess all of these various expert skills. Guidance from plant design to simulation is not available in any of the approaches. Some of the tools offer support during model parameterisation, aggregation or the specification of boundary conditions. However, these tools are directed to experts in the respective domains.

The support of the planning engineer, as non-expert, is therefore one of the main tasks of this thesis, in order to guide her/him from plant design to the plant simulation. The support of planning engineers as non-experts in modelling and simulation was also the background of previous work in a collaboration between the University of Glamorgan, Wales and the University of Applied Sciences and Arts Hanover, Germany. Various approaches have focussed on easing the control design task for non-experts by making qualitative modelling available to the project engineer (Strickrodt, 1997), facilitating effective process identification (Körner, 1999) and assisting the design of complex control systems (Syska, 2004).

Research into GUIs in general, was very active in the late 1980's and the 1990's, notably the approaches developed by the Swansea University (see e.g. Barker *et al.* (1989) or Barker *et al.* (1993b)) and the University of Salford (Li and Gray, 1993). Many of their basic ideas are still significant in the design of GUIs currently. See also Section 3.6.2.

3

Proposal for Catalogue Based CAE of Plant Simulation Models for Virtual Start-up

Considering the limitations of the existing approaches for virtual start-up at the end of detailed engineering, a new proposal for catalogue based CAE (Computer Aided Engineering) of plant simulation models is introduced in this chapter. In the following this proposal is named the "Model^{CAT}" approach.

3.1 General Concept of the Model^{CAT} Approach

In order to allow testing and fault corrections at the end of detailed engineering, hence reducing time and costs of the real start-up phase, the Model^{CAT} approach has been developed. It aims to simulate the (virtual) plant before it is built.

Therefore, the Model^{CAT} approach intends to make use of the design situation at the end of detailed engineering, where the chemical plant – including its components and subsystems – is completely specified and no latitude for specification remains. For a plant simulation in this plant life cycle phase, simulation models of the physical components (ready-to-buy) are missing. These simulation models are named FSCS models in the following and aim to reflect the behaviour of physical components including their specific characteristics and properties, but also their limitations. This allows the simulation of chemical plants which reflect the real plant behaviour as closely as possible, which has primarily been reported in Hoyer *et al.* (2003a) and developed further in Hoyer *et al.* (2004) and Hoyer *et al.* (20005b).

The organisation of a FSCS model catalogue including the definition of requirements on FSCS models is therefore the first novel sub-task of the Model^{CAT} approach and is elaborated in Section 3.3 after a brief introduction of the main functionalities of CAE-plant design tools (Section 3.2). Two different possibilities of embedding the FSCS models into the database of CAE-plant design tools are described in Section 3.4.

The systematic "symbiosis" of CAE-plant design tools and FSCS models for the generation of plant simulation models have not been considered so far. This novel sub-task is elaborated in Section 3.5 by an automatic strategy for generating plant simulation models based on CAE-plant design data, separately for the process and the control domain. The scenario, where the Model^{CAT} approach provides an interface between the CAE-plant design and virtual start-up simulation, is illustrated in Fig. 3-1.

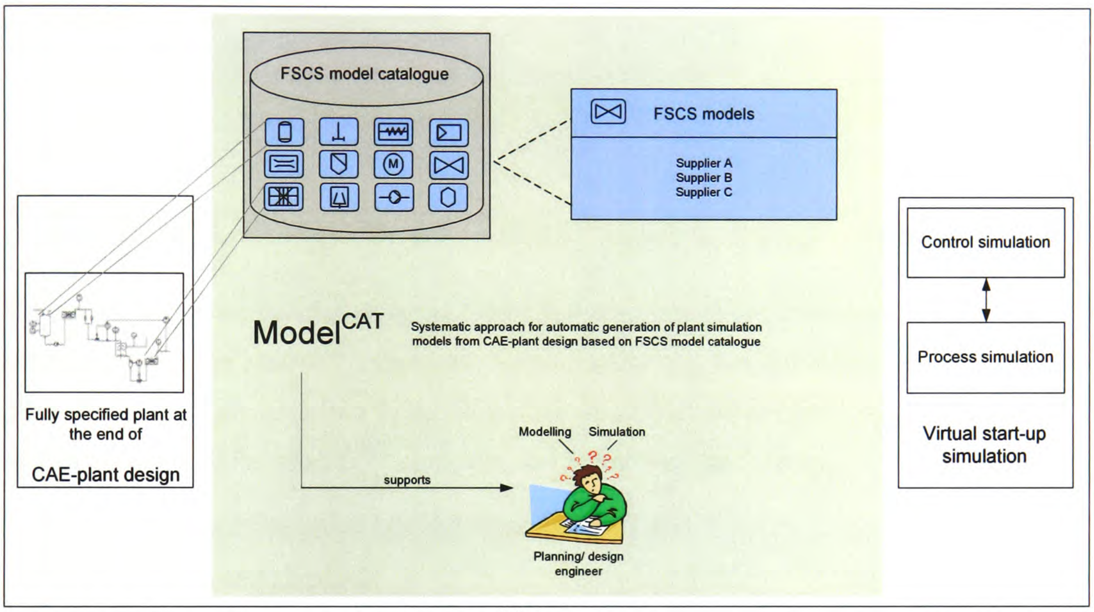


Figure 3-1: General concept of the Model^{CAT} approach

The intention to generate simulation models based on CAE-plant design implies that the planning/design engineers, who are usually non-experts in the field of modelling and simulation, have to be supported during the simulation procedure. The support of such non-experts in the field of modelling and simulation have not been investigated by existing approaches in the literature so far. This novel sub-task of the Model^{CAT} approach is elaborated in Section 3.6. Here, different functionalities supplying assistance are implemented, using a "smart" GUI (Graphical User Interface) module, which leads the planning/design engineer through the required specification process for the simulation of chemical plants, without requiring the assistance of specialists.

Summarised, the Model^{CAT} approach includes the following sub-tasks, which are required for an automatic generation of plant simulation models from CAE-plant design:

- Plant Design with CAE-Plant Design Tools (Section 3.2)
- Definition of the FSCS Model Catalogue (Section 3.3)
- Embedding of the FSCS Model Catalogue (Section 3.4)
- Strategy for an Automatic Simulation Procedure Initiated from CAE-Plant Design Tools (Section 3.5)
- Methodology for Supporting Planning Engineers during Modelling and Simulation (Section 3.6)

3.2 Plant Design with CAE-Plant Design Tools

The core of modern plant design is the CAE-plant design tool, which represents also the basic software tool of the Model^{CAT} approach. Some general and detailed information about CAE-plant design tools are described in the following which may be relevant for the understanding of the realisation of the Model^{CAT} approach, and is structured as follows:

- 3.2.1 General Information on CAE-Plant Design Tools
- 3.2.2 Component Database
- 3.2.3 Project Database
- 3.2.4 P&I Diagrams and Function Block Diagrams

3.2.1 General Information on CAE-Plant Design Tools

For more than two decades the plant design phases have been supported by powerful software tools, called CAE-plant design tools. CAE-plant design tools generally combine most of the following tasks/domains of plant design as outlined by Früh and Ahrens (2004):

- Process design
- Flowsheeting
- Plant layout
- Pipe work and apparatus design
- Control system design

- Infrastructure and logistic
- Purchase, assembly and start-up
- Maintenance

Examples for such tools are Smartplant™ (Intergraph, 2000), Cadison™ (Brückner, 2004) or COMOS PT™ (Innotec, 2002). A review of CAE-plant design tools is presented by Rauprich *et al.* (2002). The advantages of these CAE-plant design tools are the integrated workflow and dataflow management, highly flexible data modelling and integration, object-oriented database architecture.

The general organisation of CAE-plant design tools is presented in the next sections, starting with the component database.

3.2.2 Component Database

Component databases (also called master databases) of CAE-plant design tools are collections of base objects, which are required for use within plant design. These base objects serve as general templates that contain all kinds of data information such as properties, connections, captions or symbols separated into different domains (as listed in the previous section). All base objects are sorted in an object-oriented tree structure, with the domain category as top hierarchy class level.

The hierarchy class level is broken down from general classifications to sub-classes with increasing levels of detail. At the lowest class hierarchical level, component databases contain base objects of predefined components from specific vendors. The object-oriented structure of CAE-plant design tools is illustrated in Fig. 3-2 and shows the component hierarchy (in letters marked in grey) with typical examples (in normal letters). At the lowest hierarchical level the component database contains data from the geometrical, functional and technical data of components from a special component supplier.

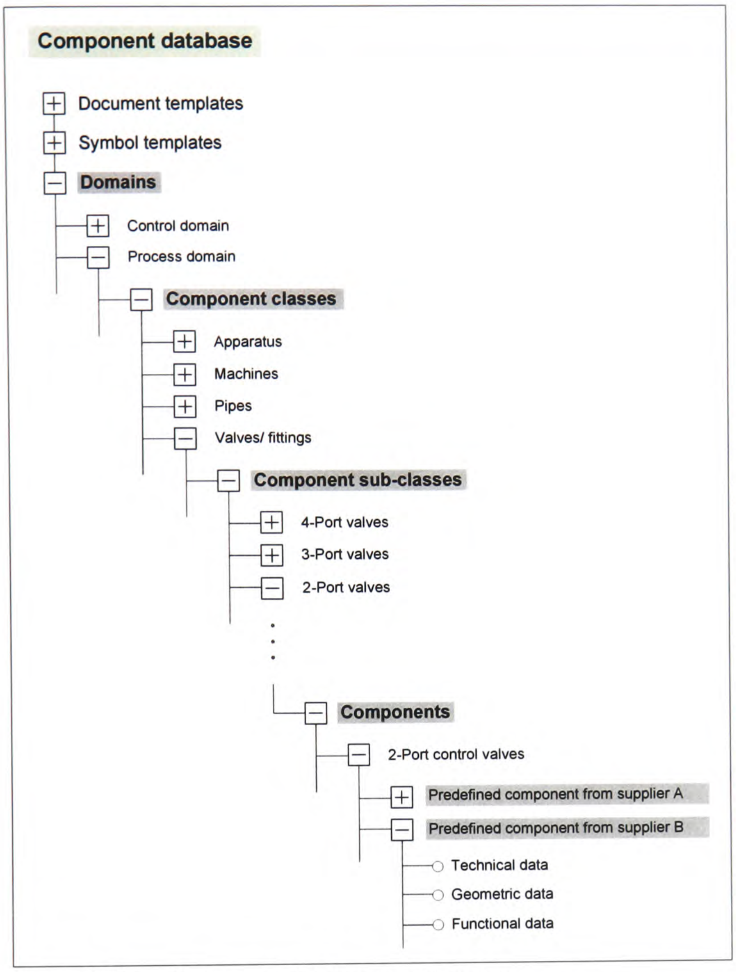


Figure 3-2: Component database hierarchy

3.2.3 Project Database

As opposed to the component database, the project database contains all data resulting from plant design. Here, the single set of data is called planning object. Planning objects are derived from base objects and have inherited all their defaults including properties, connections, etc. As illustrated in Fig. 3-3, the plant design project is subsequently structured from the overall chemical plant to plant units, from plant unit to components and further to their specifications. Even the planning and design utilities used are integrated within the project database. The most commonly used planning and design utilities are presented in the following section.

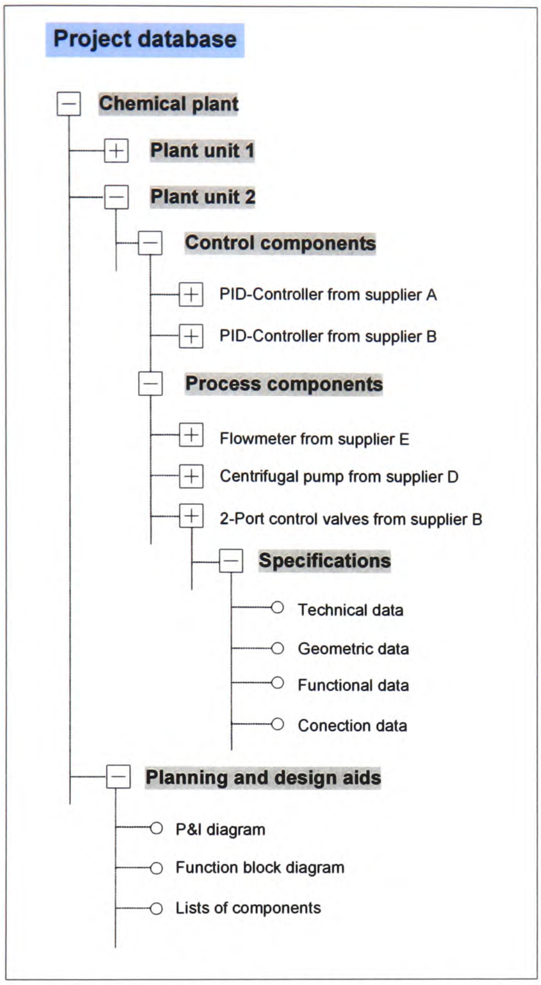


Figure 3-3: Project database hierarchy

3.2.4 P&I Diagrams and Function Block Diagrams

P&I diagrams are important planning and design utilities in chemical plant engineering. A typical example is presented in Fig. 3-4. P&I diagrams contain graphical information about process and control components (represented by function eyes), which are selected from the component database, placed on the diagram and connected to graphical editors by the planning engineer. Thereby, the planning objects are connected via their so-called input/output connectors: The output connector of one planning object is connected to the input connector of another planning object. This information and the specified parameters are automatically integrated into the structure of the project database hierarchy. Generally, in P&I diagrams the process domain is represented by components and pipes (through streams) and the control domain by function eyes and signal lines, as illustrated in Fig. 3-4.

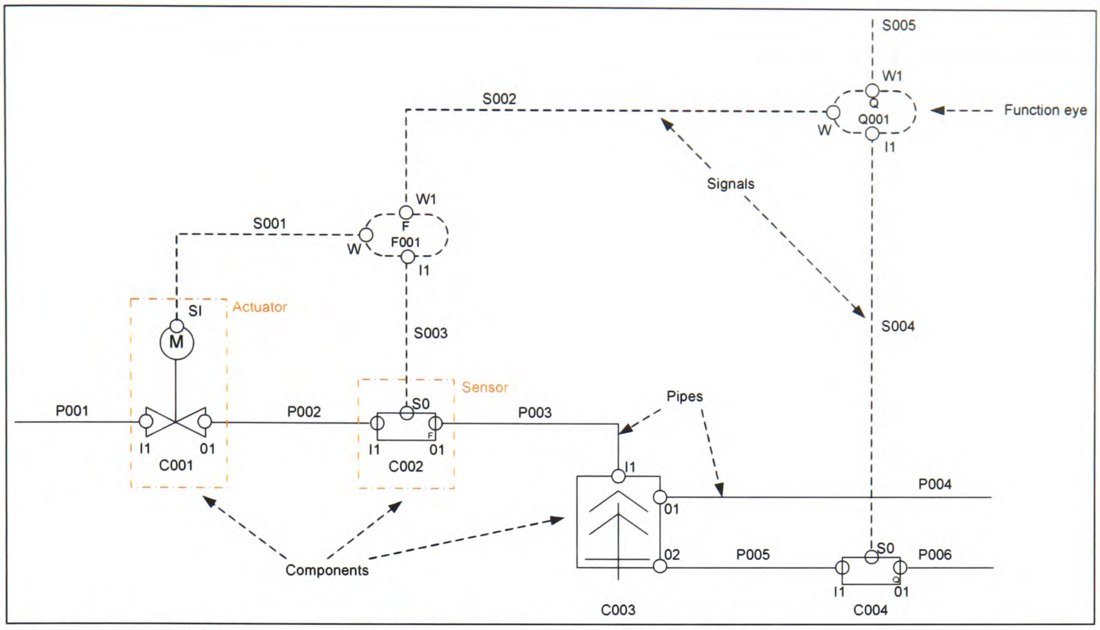


Figure 3-4: P&I diagram including components and function eyes

Remark:

According to the standard ISO 3511/2:1984 (International Organization for Standardization, 1984) two alternatives are possible in order to represent sensors in a P&I diagram. One way to show sensors in a P&I diagram is as a unique graphical symbol, whereas the other way, the most generally applied method, the sensor is hidden behind a function eye without any visible graphical information on the P&I diagram. Both alternatives are illustrated with a flow control example in Fig. 3-5. In both cases the function eye's function is indicated by the eye's function code, e.g. for monitoring or control of the respective measurement signal.

In the case without the graphical sensor representation, the sensor's information must be virtually added between the corresponding pipes (see Fig. 3-5 between pipe P002 and P003), in order to obtain the required information for later simulations. Therefore, the alternative with the graphical sensor representation was chosen for this work, because the sensor is "physically" connected at the right position in the P&I diagram, which is relevant to later interconnection analysis, see Section 3.5.2.

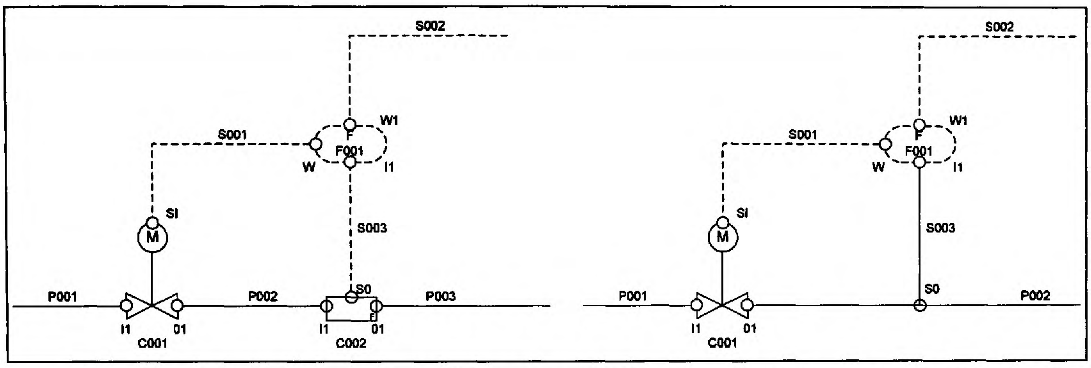


Figure 3-5: Two alternatives to indicate sensors in P&I diagrams: With sensor (left) and without sensor (right)

While the interaction between the process components is comprehensively covered by the P&I diagram, the information about the control components within P&I diagrams is limited to a general representation by function eyes.

From Function Eyes to Function Block Diagrams

Function block diagrams are associated to function eyes and used to describe the control functions. This information is separately stored in the project database under the "Control components" hierarchy level. An example of a function block diagram, which includes two function blocks and a multitude of input and output signals, is presented in Fig. 3-6. These signals are connected to the function blocks by connection lines.

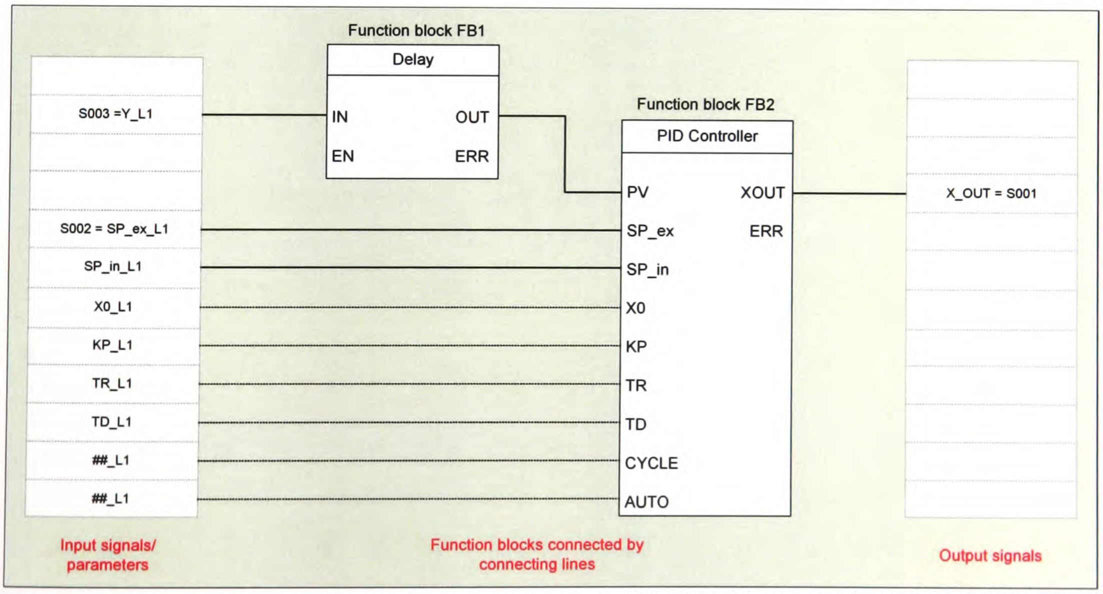


Figure 3-6: Function block diagram including control function blocks and input/output signals

3.2.5 Results of Plant Design

At the end of plant design (= end of detailed engineering), all data for building the plant have been prepared and specified. These data include the fully specified components, which are aggregated within P&I diagrams and function block diagrams. Furthermore, all data sheets, manuals, lists, etc. for the ongoing plant life cycle have been completed. These plant design results are illustrated in Fig. 3-7. This situation of plant design may be used as an initial point for the Model^{CAT} approach. For an automatic simulation strategy based on plant design data, the prerequisite for any simulation is the availability of (mathematical) component models. The requirements for designing these simulation models, here named FSCS models, is described in the following section.

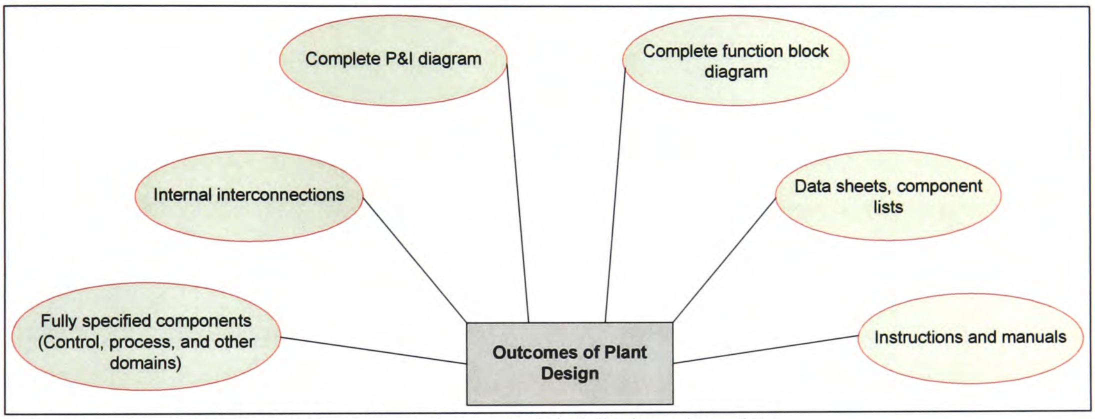


Figure 3-7: Plant design results at the end of detailed engineering

3.3 Definition of the FSCS Model Catalogue

The methodology for the organisation of the FSCS model catalogue addresses the following topics described in the following sections:

- 3.3.1 Differences between Generic and FSCS Models/ Definition of FSCS Models
- 3.3.2 Systematic Separation of FSCS Models into Two Domains
- 3.3.3 Requirements for FSCS Models in the Model Aggregation

3.3.1 Differences between Generic and FSCS Models/ Definition of FSCS Models

A prerequisite for this approach is the availability of FSCS models for all plant components. As mentioned before, FSCS models required for simulation of the fully specified plant at the end of detailed engineering are models of its specific physical components. FSCS models reflect the real behaviour of the physical (ready-to-buy) components from specific suppliers (including their definitive properties). This behaviour is essential for the composition (aggregation) of the virtual plant, thus allowing a realistic test on process and control functions by its simulation.

During prior engineering phases these realistic component specifications are not available. Detailed characteristics from the physical components which are chosen at the end of detailed engineering are mostly not the matter of interest and not relevant to the investigations at this stage. The models used at this stage are generic models, whose parameters can be freely chosen (which is generally done according to the requirements). This, exemplified by a control valve, is illustrated and the characteristics of the valves are predominantly defined by its inherent characteristic curve, as presented in Fig. 3-8, exemplified by three different valve characteristics: a linear, an equal percentage and a quick opening valve characteristic. Each characteristic has an influence on the relevant model equation.

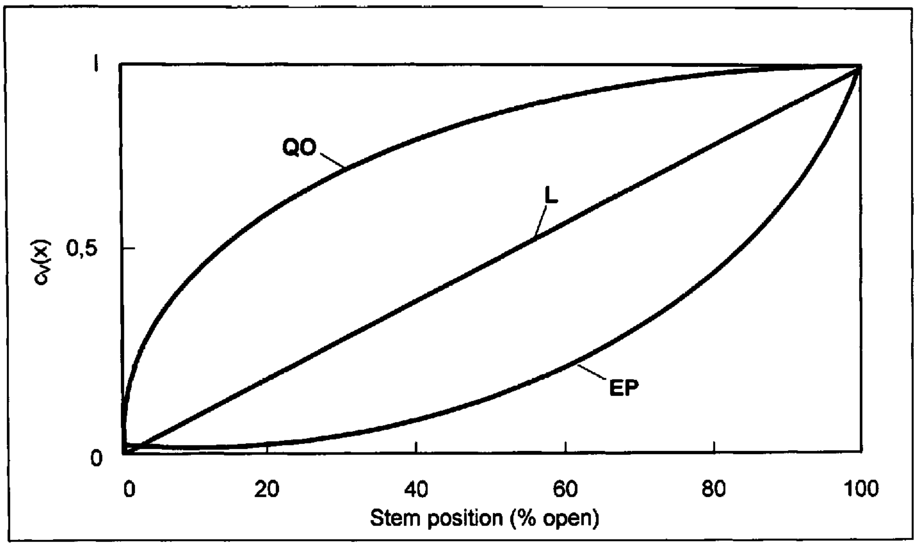


Figure 3-8: Inherent valve characteristics for quick opening (QO), linear (L) and equal percentage (EP) valve

Only at the end of detailed engineering, by choosing a specific physical component, can the real parameters be precisely defined. This is exemplified by an equal percentage valve characteristic in the following.

The equation which defines the characteristics of an equal percentage valve is shown in Equation (1). The valve coefficient c_v, depends upon the stem position x and the component specific properties, i.e. the valve coefficient of a fully opened valve is c_{V100}, the valve coefficient of a closed valve is c_{V0} and the maximal stem position path is x_{100}.

$$c_V(x) = c_{V0} \cdot e^{\ln\left(\frac{c_{V100}}{c_{V0}}\right) \cdot \frac{x}{x_{100}}} \qquad (1)$$

This equation represents on example of a generic model. All parameters can be freely chosen. The graphs in Fig. 3-9 shall illustrate the general situation using generic models, such that at early design phases different specification variations (of the above mentioned parameters c_{V100}, c_{V0} and x_{100}) are possible, with the result that very different valve characteristics are obtained. At these early phases, the target properties of components are specified in order to best fulfil the specifications at this stage of design. Whether these target properties can be met by physical (ready-to-buy) components specified later, is not the issue at these early design phases. Models used in this stage reflect only the desired properties and do not represent the behaviour of the

physical components specified at the end of detailed engineering. Only at the end of the design stage, when the physical component is chosen, will the definitive characteristics of the valve be available, in Fig. 3-9 this is exemplified with the thick red graph.

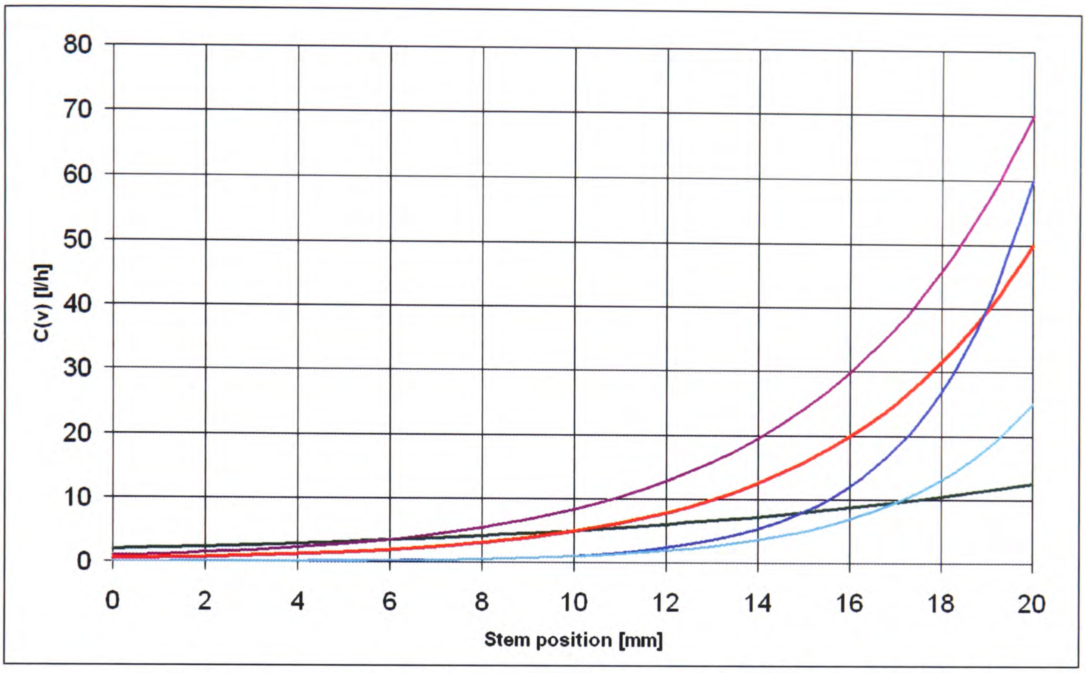


Figure 3-9: Valve characteristics of equal percentage valve with different parameters

This final characteristic should meet the requirements from the early design phases as precisely as possible, but is unique and much closer to the real behaviour of the physical component. This behaviour, which reflects the behaviour of the physical (ready-to-buy) component, is modelled in a FSCS model.

Table 3-1 summarises and compares the main differences between generic and FSCS models.

Table 3-1: Differences between generic and FSCS models

	Generic model	FSCS model
Life cycle phase	Early design phases	End of detailed engineering
Brief description	Parameters of equations must be chosen according to the target properties.	FSCS models represent the properties of the physical components from one specific component supplier.
Result	Required properties	Properties of physical (ready-to-buy) component: Catalogue number e.g. 1234

This approach of defining such unique FSCS models must be taken into account for setting up the catalogue of FSCS models, either for process and control FSCS models.

Because of the conventional and intrinsic differences between process and control models, the separation of these two domains is essential for the proposed simulation approach and is described in the following.

3.3.2 Systematic Separation of FSCS Models into Two Domains

The basis for the Model^{CAT} approach is a clear separation of the process and control domains, which was presented by Hoyer *et al.* (2004). In order to fulfil the requirements of the process domain including process components, e.g. valves, tanks, reactors, pipe work, etc. with their interconnecting media flow streams, specific modelling and simulation tools have been established which provide limited usability within the control domain. The control domain is characterised by components including control functions and equipment in their control schemes. Here, signals play the predominantly role. In large plants the control domain is represented by process control systems (DCS or SCADA/PLC).

Generally, sensors and actuators represent the natural interface between the process and the control domains, as illustrated in Fig. 3-10, marking the distinction line between two worlds: streams and signals.

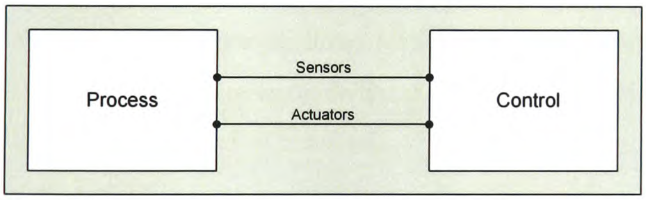


Figure 3-10: Natural interface between processes and control domains

The physical part of sensors and actuators belongs to the process domain and the signal from the sensors and to the actuators belongs to the control domain, as illustrated in Fig. 3-11.

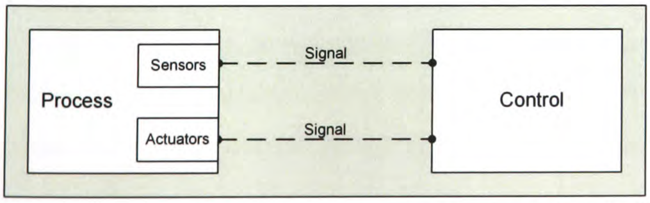


Figure 3-11: Separation between process and control domains within Model^{CAT} approach

But also from the modelling and simulation point of view, different technical requirements have been established (and evolved over time) for the modelling and simulation approaches for both domains, as described in Section 2.3 and Section 2.4. The causal block-oriented control simulation tools are commonly used in control domain, whereas for dynamic simulation of process models, simultaneously solved equation-oriented process simulation tools are established using non-causal dependencies. The strengths of both classes of simulation tools should be retained within the Model$^{\mathrm{CAT}}$ approach.

3.3.3 Requirements for FSCS Models in the Model Aggregation

In this section essential requirements which are relevant for the model aggregation and the model quality are described, separated between the requirements of the process and control domain.

3.3.3.1 Requirements for Process Domain

In order to allow the model aggregation of an entire process model, each FSCS model of the process domain – in the following shortened to "process FSCS models" – has to fulfil the general model structure requirements.

The prerequisite for automatic model aggregation is the standardisation of all process FSCS models with regard to the stream characteristics. Streams contain state information (in the following named *system defining variables*) of the respective medium, as described in Section 2.3.1. In order to aggregate different process FSCS models, such variables have to be compatible with all process FSCS models.

For the model definition itself, such standardisation implies that each process FSCS model has an effect only on these system defining variables. This implies that all process FSCS model equations only depend on the input and output system defining variables, as illustrated in Fig. 3-12.

At this stage it must be pointed out, that the flow direction of the media depends on the actual system defining variables of the process FSCS model. So, "input" system defining variables could change into "output" system defining variables and *vice versa*.

Fig. 3-12 shows an example of system defining variables describing the media flow from left to right.

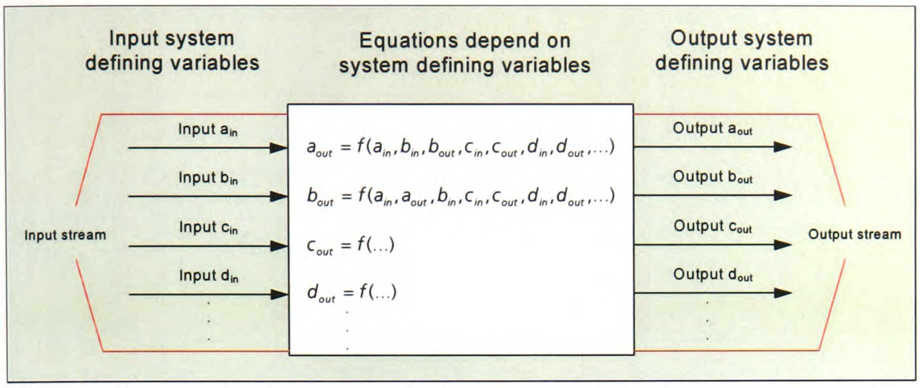


Figure 3-12: Dependencies of equation-oriented approach

For later model aggregation, the process FSCS models require definitions of the right type, number and order of system defining variables. This implies that every process FSCS model contains system defining variables of the same type, in the same order. Consequently, also the number of input and output system defining variables must be equal. This correlation is illustrated in Fig. 3-13.

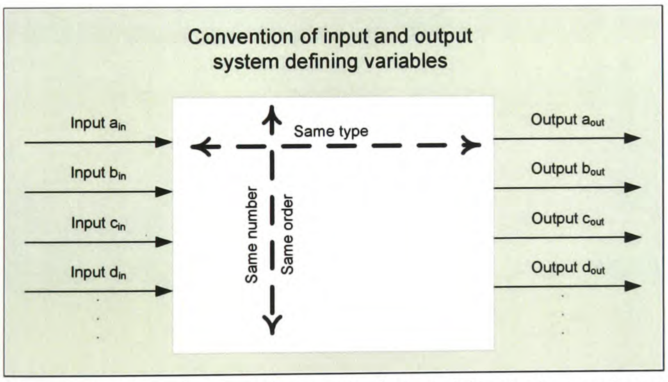


Figure 3-13: Conventions of relationships between input and output system defining variables

It is possible, that process FSCS models contain a multitude of submodels and internal streams, when representing a modelled component. This composition hierarchy is presented in Fig. 3-14 and could be further decomposed into "sub-sub" models.

However, only the input and output stream of the composed model is relevant for the ongoing model aggregation.

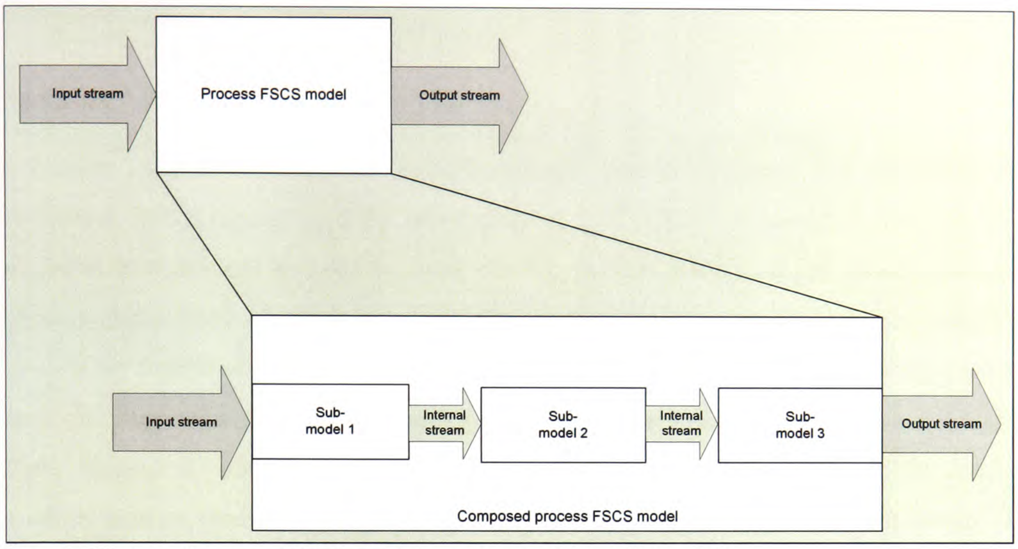


Figure 3-14: Process FSCS models composed by sub-models

In order to model physical (ready-to-buy) components, even multi-input and multi-output connectors, including their system defining variables have to be modelled.

Fig. 3-15 shows an example of process FSCS model with single input and multi (two) outputs.

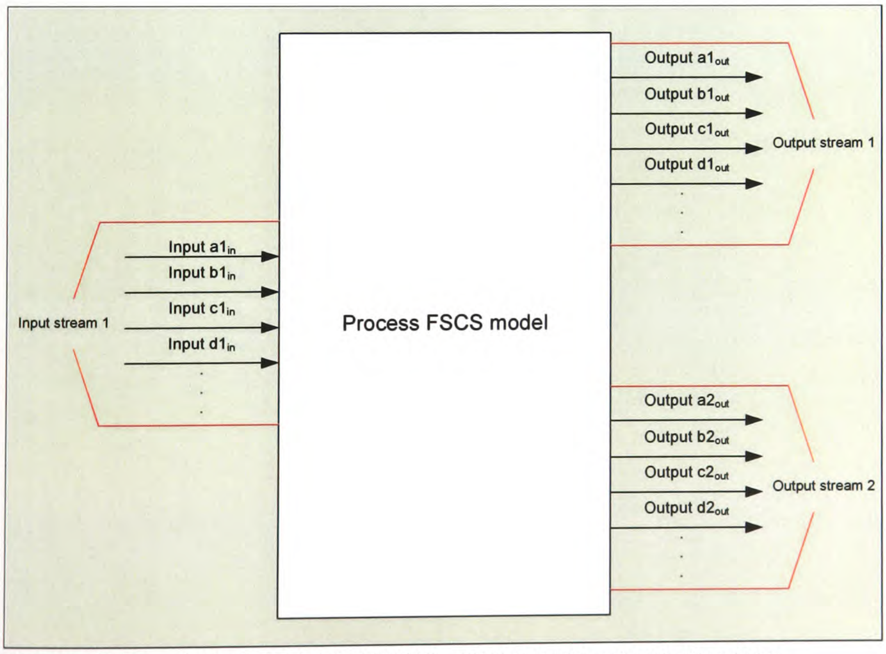


Figure 3-15: Process FSCS models: Single input and multi output

3.3.3.2 Requirements for Sensors and Actuators

Sensors and actuators play an essential role for the interconnection between the process and the control domain.

Sensors

Generally, sensors measure one specific system defining variable and transform the state dimension into a signal, typically into voltage (e.g. 0...10 V) or current (e.g. 4...20 mA) or as digital state value with units of measure, see Fig. 3-16. Similarly, to the real interconnection of process and control components, these signals from sensors are connected to control FSCS models for further use in the control domain. Two types of sensors may be distinguished, sensors with medium contact and contactless sensors. Contactless sensors (such as IDM (Inductive Flow Meter)) do not have an effect on the system defining variables, while sensors with medium contact (such as an inline measuring device) may have an effect on system defining variables (such as pressure drops). Further examples of sensors are level sensors in a vessel or temperature sensors.

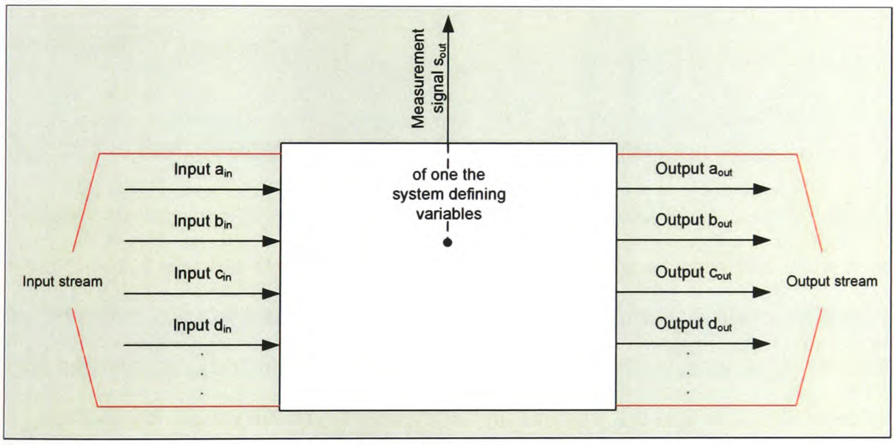


Figure 3-16: Process FSCS model: Sensor

Actuators

As opposed to sensors, actuators make use of signals from the control domain and manipulate some system defining variables of a stream. Fig. 3-17 shows an example, where the signal "s_{in}" may be part of the model equation of the process FSCS model.

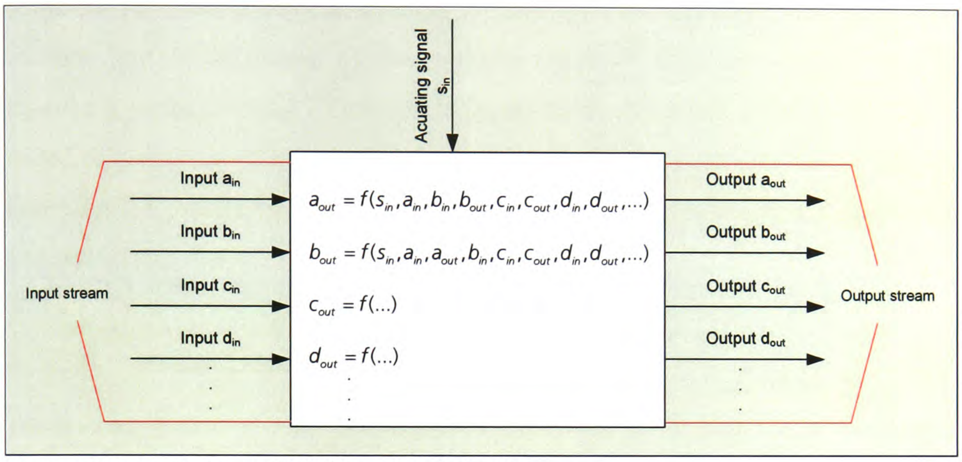


Figure 3-17: Process FSCS model: Actuator

This separation of process and control domain is used in the definition of FSCS models within the Model^{CAT} approach.

3.3.3.3 Requirements for the Control Domain

Control models in the context of FSCS models (control FSCS models) must represent the feedback control and the logic control as closely to reality as possible. This representation is done by so-called control function blocks. Every function block contains at least one or more inputs and outputs, as illustrated in Fig. 3-6. Generally, two kinds of input parameters of a function block may be distinguished: Parameters and signals. Parameters must be explicitly specified or are set by default values and are dedicated to one specific function block. Changeable and unchangeable parameters can be differentiated. The latter cannot be changed online by the user while changeable parameters could be altered by the user, such as the P-I-D parameters of a PID controller or the reference value.

Signals are time-varying quantities and generally available in analog or digital format. They are available as connections between two function blocks, where the inputs of one function block may be connected to outputs of other function blocks, between sensors (process components) and function blocks as well as between function blocks and actuators.

Every function block has its specific function written in a specific programming language, e.g. IEC 61131-3 (International Electrotechnical Commission, 2003). Most simulation tools provide a multitude of standardised function blocks, but also user defined blocks can be generated.

Function blocks may be decomposed into sub-function blocks, as described within the requirements for the process domain. In order to fulfil the functional aspect of these blocks, it is conceivable that the functional blocks could be provided either in a neutral description, paying regard e.g. to the German VDI/VDE 3696 standard (VDI/VDE, 1995) or the CAEX (Computer Aided Engineering eXchange)-standard IEC/PAS 62424, published by the International Electrotechnical Commission (2005), or might be in a specific description from the respective control system supplier.

3.3.3.4 Model Documentation and Quality

The documentation of every FSCS model is essential for its later use in simulation, as outlined e.g. by Barker *et al.* (1991b). Only by a correct and consistent documentation will it be possible to retrieve information about the background, purpose, characteristics and validity ranges of each component and its parameters and variables (e.g. differential term of differential equation) for which the FSCS models were derived. Additionally, the information about suitable numerical solvers may be included.

All information should be an integral part of the FSCS model description and should be supplied by the model developer (e.g. component supplier) as part of the FSCS model.
As described in Section 3.2.2, CAE-plant design tools make use of centralised component databases. The extension of component databases with FSCS models is described in the following.

3.4 Embedding of the FSCS Model Catalogue

For the integration of the FSCS model catalogue into the component database of the CAE-plant design tools two methods may be distinguished:

1. Separate model catalogue detached from the planning objects. Here, the FSCS models are linked to the planning objects.
2. The physical embedding of the FSCS model on the component hierarchy level of the planning object.

Both alternatives are possible and would not change the outcome of this study; however, the direct integration of the FSCS models into the component database was chosen, in order to illustrate the direct link from the component to its FSCS model. Therefore, the object-oriented structure of the component database will be used to store the simulation model information as an additional specification, apart from the other component information supplied by the component supplier. This way of physical embedding of the FSCS models into the component database hierarchy is presented in Fig. 3-18.

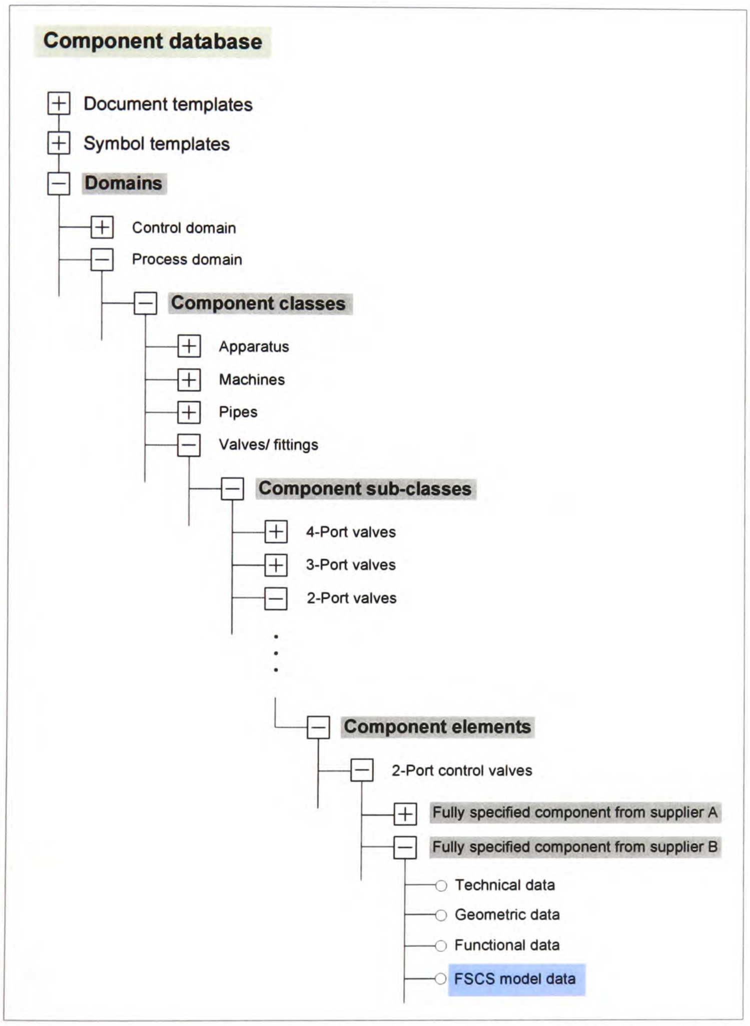


Figure 3-18: Embedding of FSCS models into the component database hierarchy

However, the essential prerequisite for the realisation of the Model^{CAT} approach is the free accessibility on the data of the component (master) database in order to integrate the FSCS models.

3.5 Strategy for an Automatic Simulation Procedure Initiated from CAE-Plant Design Tools

The strategy for an automatic simulation initiated from CAE-plant design tools is summarised under the following topics, which were presented in Hoyer *et al.* (2005a) and Hoyer *et al.* (2005b):

- 3.5.1 Selection of Simulation Area
- 3.5.2 Systematic Analysis of the Selected Area in a P&I Diagram and Associated Function Block Diagrams
- 3.5.3 Automatic Model Aggregation

3.5.1 Selection of Simulation Area

As discussed in Section 3.2.3, the project database of the CAE-plant design tool contains all data after the completion of detailed engineering that have been specified by the planning engineer within plant design, as illustrated in Fig. 3-3.

The P&I diagram is taken as the basis to select a plant section for simulation. The graphical representation of the P&I diagram helps the planning engineer to intuitively select the section of plant to be simulated. Fig. 3-19 illustrates a simple procedure used to select the simulation area within the P&I diagram. Here, a part of a fresh cheese production plant is presented, including the process components: Control valve (C001), flow rate sensor (C002), separator (C003), quality (dry matter) sensor C004, some pipework (P001-P006) and the control functions: Flow control loop (F001) and the cascaded quality (dry matter) control loop (Q001). In Fig. 3-19 the control valve (C001), the flow sensor (C002) and some pipeworks with the flow control loop (F001) have been selected for simulation.

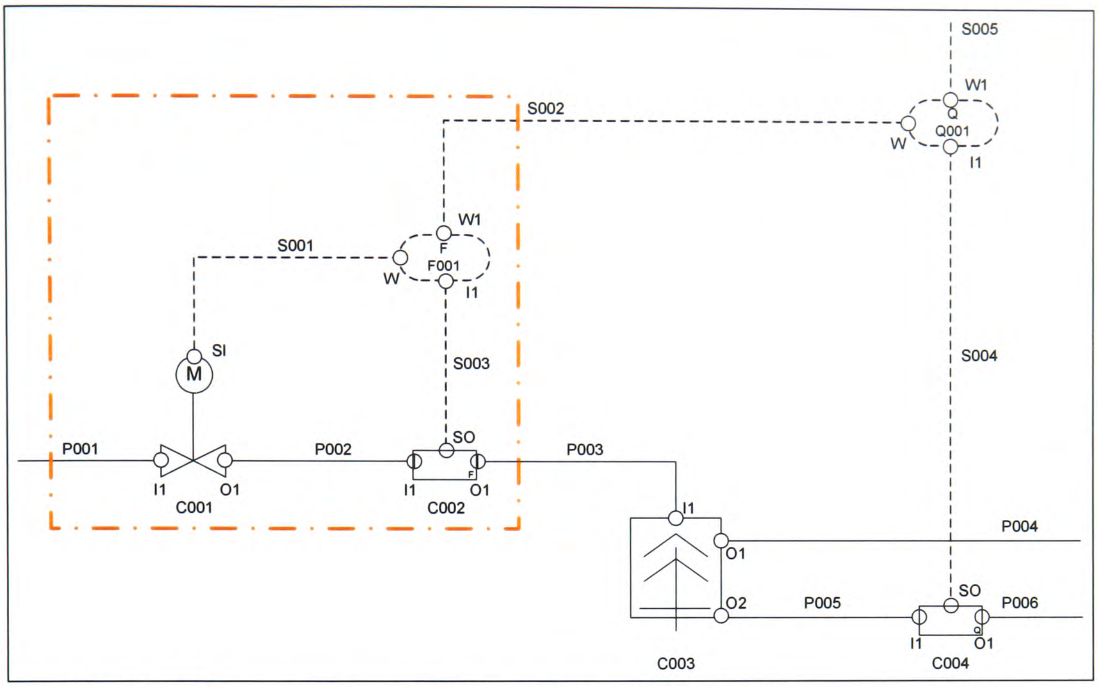


Figure 3-19: P&I diagram: Selecting section of plant for simulation

With the selection of a specific area on the P&I diagram the component and connection information can be extracted for the process components. The systematic separation of the process and control domains, as described in Section 3.3.2, is a precondition for a general approach to analyse the information of the selected area. Within the P&I diagram the separation between the process and control domain is done at the distinction line, as illustrated in Fig. 3-20, between:

- Sensors and function eyes and
- Function eyes and actuators.

All process components (C001 and C002) including the pipes (P001-P003) with their media flows belong to the process domain whereas the function eye (F001) - with its underlying diagrams - and signals (S001-S003) belong to the control domain.

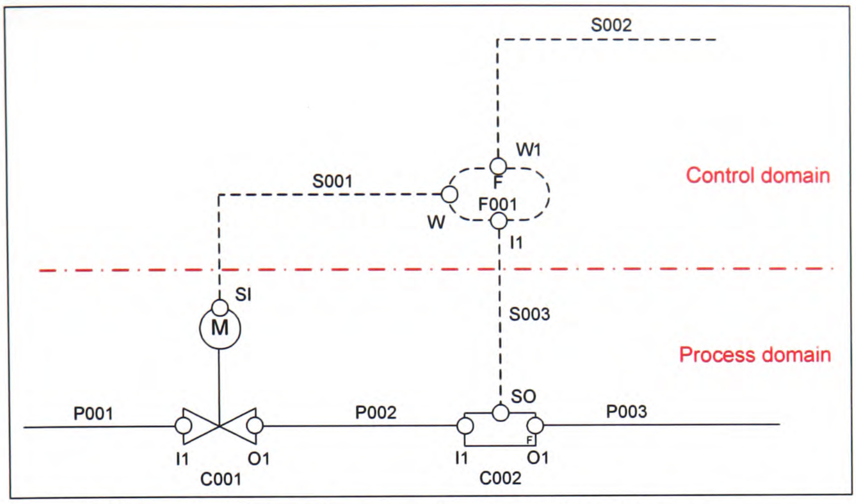


Figure 3-20: Distinction of the two domains in the P&I diagram

The graphical information on the P&I document is not sufficient to retrieve the complete information associated with the control components. As stated in the previous section, function eyes only contain "rough" information about the internal signals and chosen control function blocks. Therefore, the underlying function block diagram has to be evaluated, see Fig. 3-6.

The results from this selection procedure are two data sets, one for each domain:

- "Process data set" including the entire component and connection information from the P&I diagram with the interconnection information from function eyes.
- "Control data set" which includes the information about the function eyes and the associated function block diagrams.

3.5.2 Systematic Analysis of the Selected Area in a P&I Diagram and Associated Function Block Diagrams

At this stage, the "Process and Control data set" of the selected area must be analysed with respect to their interconnection relationships with each other, to the open connectors for the specification of boundary conditions and to the differential terms for the specification of initial conditions. An overview of the lists generated is presented in Fig. 3-21, which are derived from this systematic analysis, divided into the following issues:

- Interconnection Relationships within the Process Domain ("List of process interconnections" and "List of open process connections")
- Interconnection Relationships between Process and Control Domains ("List of interconnections between process and control" domain)
- Interconnection Relationships within the Control Domain ("List of control interconnections" and "List of open control connections/parameters")
- Differential Terms ("List of differential terms")

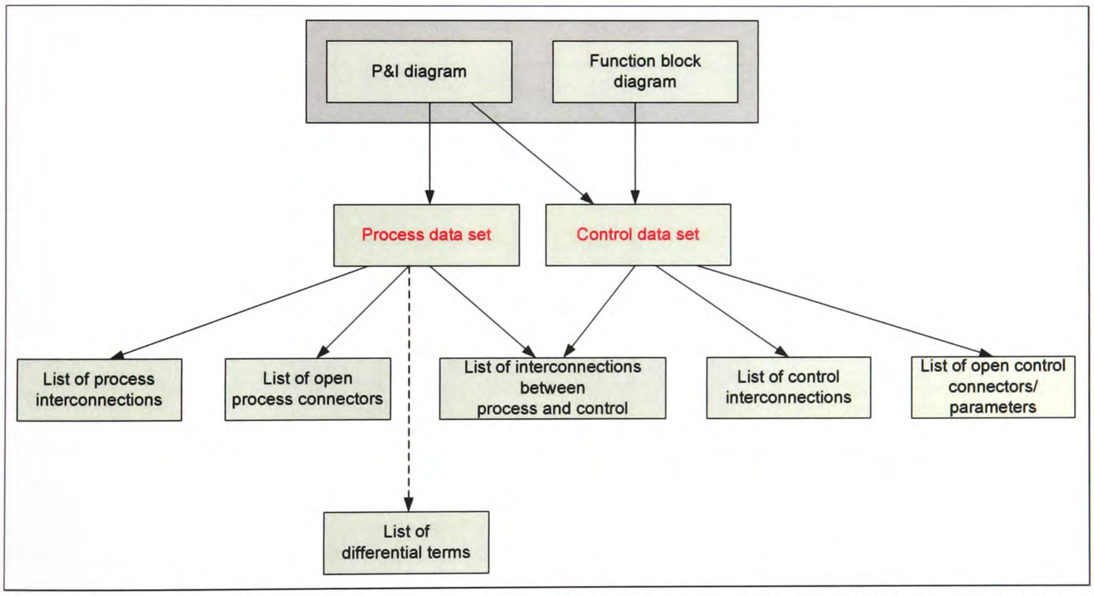


Figure 3-21: Overview of the generated lists of the systematic analysis of the selected area

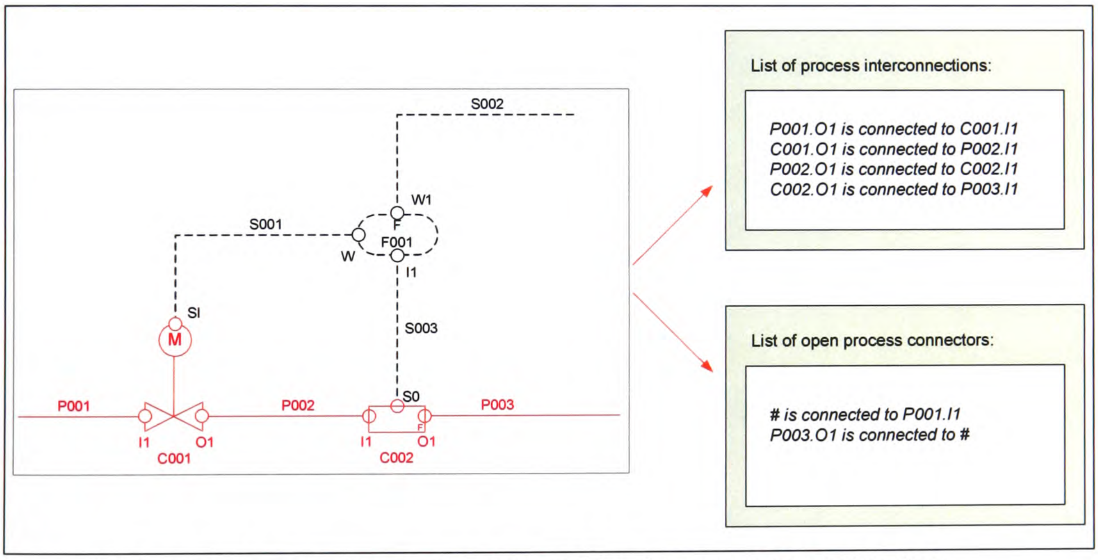
The systematic analysis of the four categories presented above is presented in the following sections.

3.5.2.1 Interconnection Relationships within the Process Domain

The interconnection information among the process components of the P&I diagram is required for process model aggregation.

Therefore, the interconnection relationships of the process components must be evaluated. Connections from an output connector of a planning object to the input connector of the connected object must be investigated. The results are stored in a neutral format (e.g. "P001.O1 is connected to C001.I1") in the "List of process interconnections". The procedure for retrieving the information from the project database of the CAE-plant design tool depends heavily on the chosen CAE-plant design tool and is described in more detail in the prototype realisation in Section 4.5.3.2.

Open connectors, which result from the selected area of the P&I diagram, must be marked e.g. with the symbol "#" and stored in the "List of open process connectors". Both lists derived from the selected area of the sample P&I diagram are presented in Fig. 3-22.



List of process interconnections:

P001.O1 is connected to C001.I1
C001.O1 is connected to P002.I1
P002.O1 is connected to C002.I1
C002.O1 is connected to P003.I1

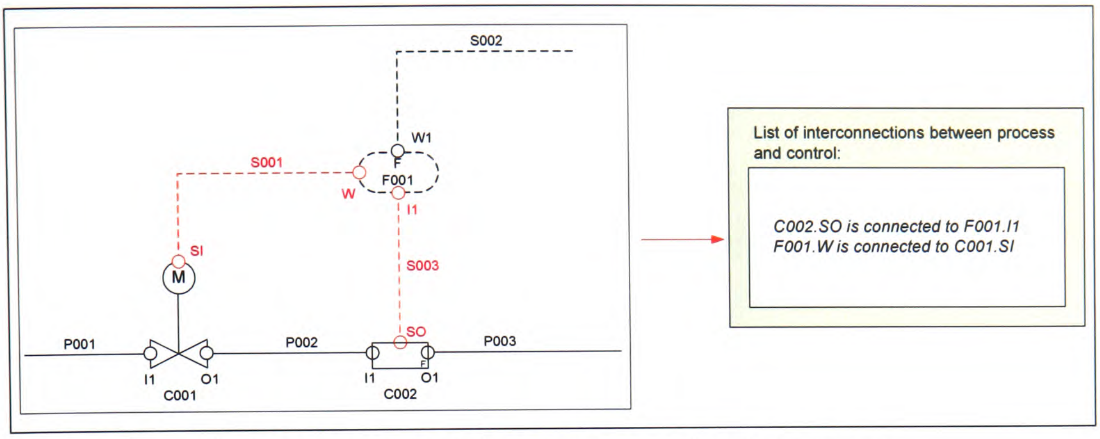
List of open process connectors:

 # is connected to P001.I1
P003.O1 is connected to #

Figure 3-22: List of process interconnections and list of open process connectors

3.5.2.2 Interconnection Relationships between Process and Control Domains

For the later interconnection of the process and control simulation tools, the interconnection relationships between process and control domains are required. Therefore, sensor signals to function eyes and actuating signals from function eyes must be detected. This procedure is illustrated in Fig. 3-23.



C002.SO is connected to F001.I1
F001.W is connected to C001.SI

Figure 3-23: List of interconnections between process and control domains

The measured signal "S003" is defined as the connection from the sensor's connector "SO" (Signal Out) to the corresponding function eye's input "I1". By contrast, the actuator signal "S001" is defined as the connection from the function eye's output "W" to the actuators input connector "SI" (Signal In). The results of this analysis are collected in the "List of interconnections between process and control".

3.5.2.3 Interconnection Relationships within the Control Domain

The function block diagram which is located under each function eye must also be analysed with respect to the internal interconnections of the control components and signals.

The signals between the process and control domain from the P&I diagram, see previous section, build the starting and end point for the control domain within the function block diagram. Therefore, the prerequisite within the CAE-plant design tool must be fulfilled such that the signal in the P&I diagram is the same as the signal in the function block diagram.

The interconnection relationships for the control domain must be defined analogically to the analysis of the process domain. But here, only the signals must be taken into account, which are relevant to the functional behaviour of the control system. According to Fig. 3-24 the interconnections are described by the following rules, exemplified by the following two examples:

- "Y_L1 is linked to input of function block FB1"
- "Output of function block FB1 is linked to PV-input of function block FB2"

The result from this analysis must be stored in the "List of control interconnections", as illustrated in Fig. 3-24. Furthermore, the connections from/to other function block diagrams must also be taken into account, as would be the case for a ratio control or a cascaded control.

Finally, the required signal and changeable parameters must be checked to determine, if they have been specified within plant design. If they are not specified, the parameters have to be added into the "List of open control connectors/parameters", as presented in Fig. 3-24.

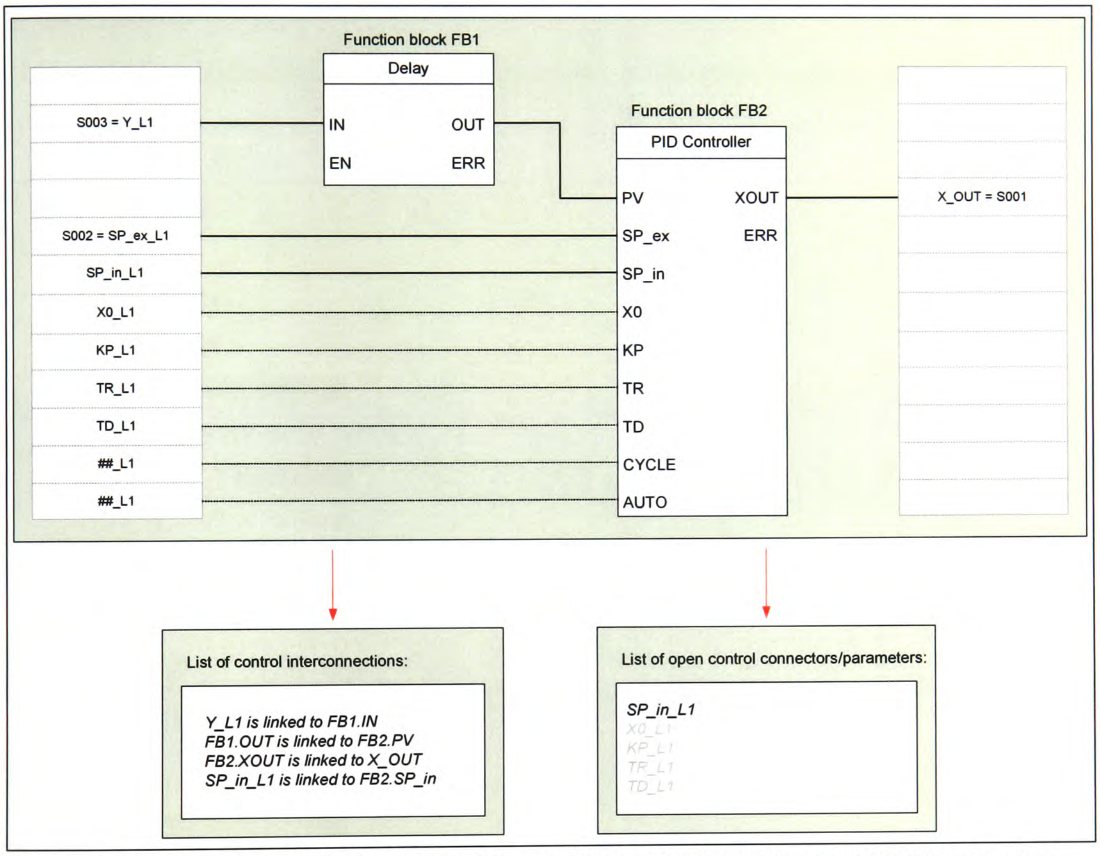


Figure 3-24: List of control interconnections and list of open control connectors/parameters

3.5.2.4 Differential Terms

In order to initialise the process simulation, the differential terms of the component model equations are required. In process models, differential terms are generally indicated by specific symbols, e.g. "$" or "£". All selected component models have to be analysed and the variable names of the differential terms must be collected in a "List of differential terms".

3.5.3 Automatic Model Aggregation

By the previous analysis and the specification of boundary conditions (see Section 3.6.4 (GUI-Task 2)), all information of the selected components and their interconnections are available for aggregating the plant simulation models, namely the entire process model and the entire control model.

All FSCS models must be retrieved from the project database together with the interconnection dependencies of Section 3.5.2.1 and Section 3.5.2.3, as illustrated in Fig. 3-25. Based on the "List of process interconnections", the entire process model must be aggregated. An equivalent control simulation script must be generated based on the "List of control interconnections".

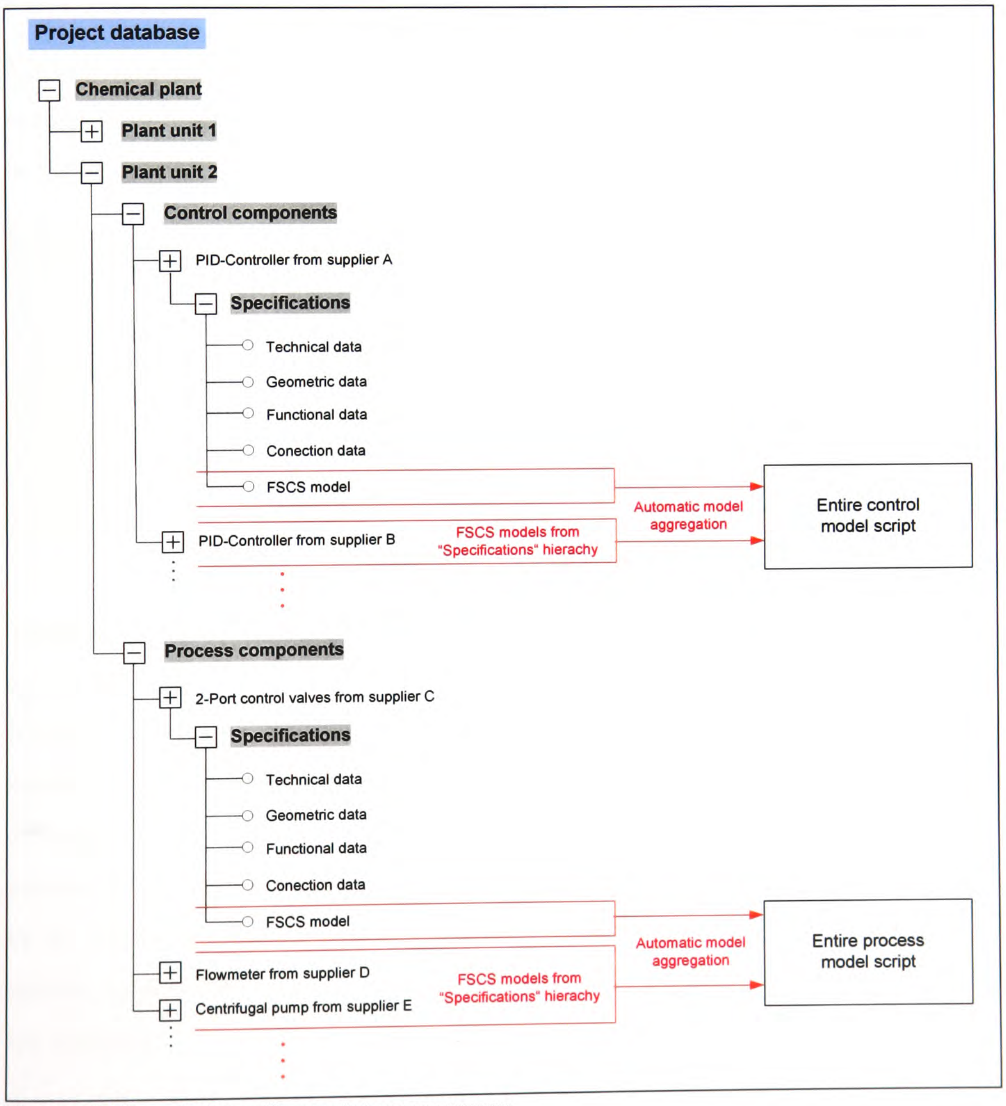


Figure 3-25: Automatic model aggregation

Adaptation of System Defining Variables of Process FSCS Models

The aggregation of process models is generally done by connecting the output flow streams of a *Model A* with an input flow stream of a *Model B*. This is illustrated in Fig. 3-26.

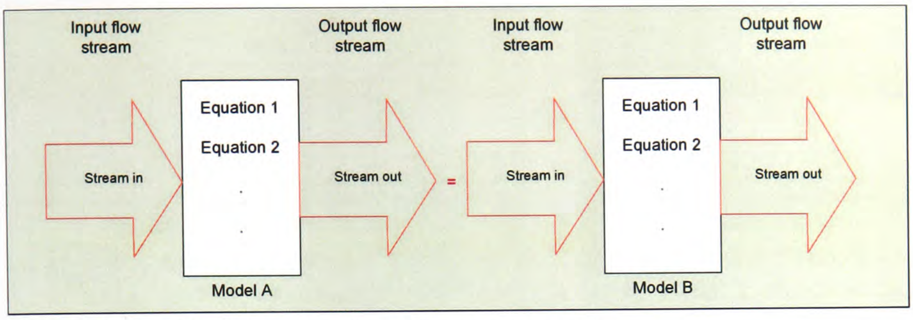


Figure 3-26: Model aggregation (through streams)

The rule that the output flow of *Model A* is connected to *Model B* implies that both flow streams must be compatible with respect to the system defining variables, as described in Section 3.3.3.1 and illustrated in Fig. 3-27.

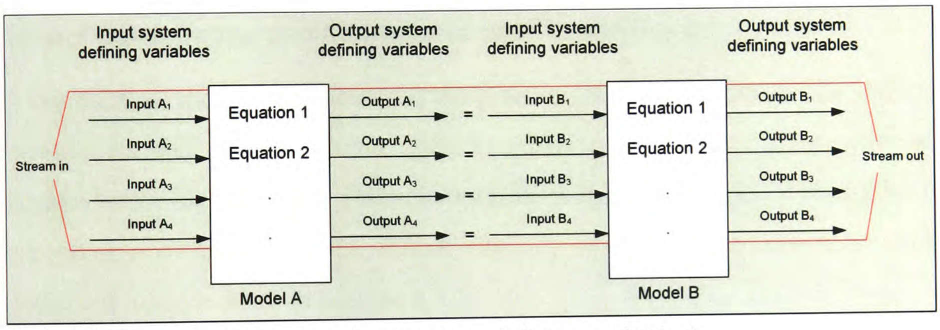


Figure 3-27: Model aggregation (system defining variables)

Therefore, every process FSCS model must be checked for the system defining variables. All system defining variables used must be part of every aggregated process model. This procedure is shown by example in Fig. 3-28. All process FSCS models must be extended to include the missing system defining variables. In Fig. 3-28 the process *FSCS model A* contains system defining variables *a*, *b* and *c*, whereas process *FSCS model B* contains *a*, *b* and *d*. Therefore process *FSCS model A* is extended by the system defining variable *d* and process *FSCS model B* by the system defining variable *c*. For such an extended artificial system defining variable, the input is equal to the output. As a result both process FSCS models have the same system defining variables.

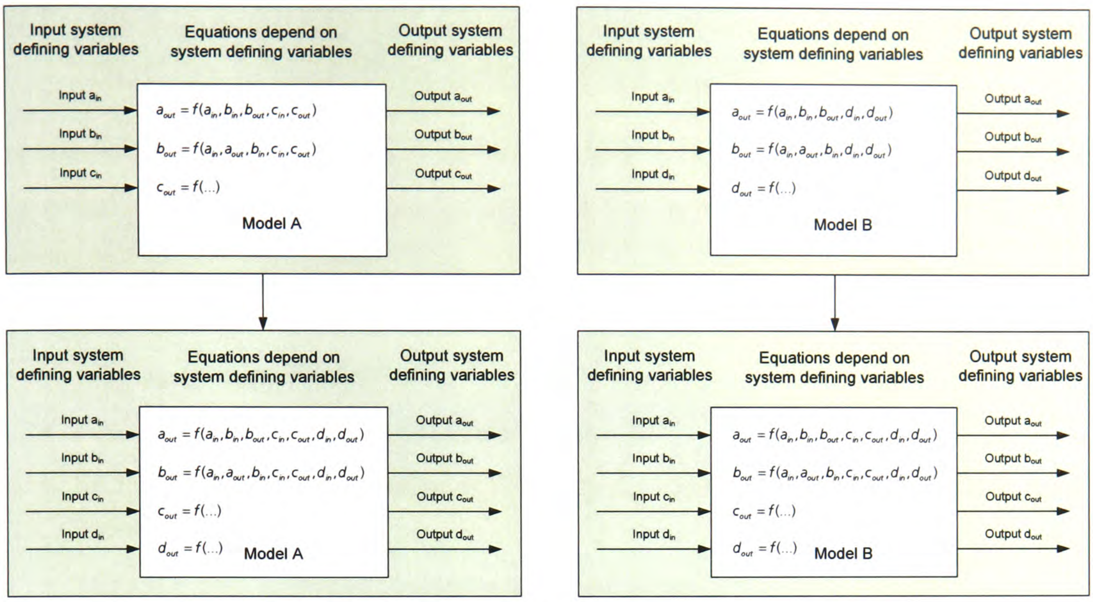


Figure 3-28: Adaptation of system defining variables

Model Compilation and Transfer to Simulation Tools

After the final model aggregation of the process and control domain, the simulation scripts for the process and control domain must be compiled. Each simulation script contains all the required information to start a simulation in the respective simulation tool. Also in this case, the compilation of the simulation scripts depends on the chosen simulation tools and will be described in more detail in Section 4.5.5.

3.6 Methodology for Supporting Planning Engineers during Modelling and Simulation

In order to support the planning engineer in the field of modelling and simulation, a smart GUI (graphical user interface) has been proposed. The following sub-topics are described in the following sections:

- 3.6.1 Need for a GUI - a Concept for Non-Expert Users
- 3.6.2 Design Aspects of GUIs
- 3.6.3 GUI-Task 1: Selection of Simulation Area
- 3.6.4 GUI-Task 2: Specification of Boundary Conditions, Initial Values and Simulation Parameters
- 3.6.5 GUI-Task 3: Start of Simulation and Presentation of Simulation Results
- 3.6.6 Help System

3.6.1 Need for a GUI - a Concept for Non-Expert Users

As became apparent in Section 2.9, the planning engineer, as target user of the proposed approach, is not an expert in the field of modelling and simulation. For that reason a user-oriented help support is very important and consequently, it is essential that the software development is based on the users` needs.

In order to specify the tasks for the planning engineer, all topics for the realisation of an automatic simulation from CAE-plant design tools, as described in Section 3.5, are analysed with respect to the active role of the planning engineer, see Table 3-2.

Table 3-2: Topics for the automatic simulation from CAE-plant design tools

Topics for an automatic simulation from CAE-plant design tools	Planning engineer	System (MAM)
1. Selection of simulation area in P&I diagram	X	
2. Systematic analysis of selected area		X
3. Model aggregation		X
4. Specification of boundary conditions, initial conditions and simulation parameters	X	
5. Generation of simulation code		X
6. Start of simulation	X	

Whereas topics 2, 3 and 5 in Table 3-2 can be automatically accomplished, see Section 3.5, topics 1, 4 and 6 require an active role of the planning engineer.

Before describing the proposals for the support of the planning engineer in the topics 1, 4, and 6, general design aspects of the GUI are presented.

3.6.2 Design Aspects of GUIs

The functional preparation of knowledge is essential for the user's understanding within different kinds of media, such as documents and graphical representations.

Several authors have established guidelines for the interaction between users and computers, such as the "Ten Usability Heuristica“ of Nielson (1993), the "Eight Golden Rules“ of Shneidermann (2002) or approaches which propose the "Look and Feel“ criterion, such as Barker *et al.* (1993a). There are, indeed, international standards available for the implementation of GUIs, e.g. DIN EN ISO 9241 standard (International Organization for Standardization, 2006).

The basic principles underlying all of these approaches are functionality, simplicity and consistency, which are summarised as key elements of the didactical design, a particular applied cognition science, introduced by Ballstaedt (1997). These principles must be regarded in the graphical design, the wording and especially in the order of supported events to assist the user in an intuitive way. The headline must be:

Consistent but simple functionality!

Functionality

Functionality apart from simplicity is the most essential principle in the design of a GUI. With respect to the graphical layout/design, a clearly and functionally structured scheme helps the planning engineer to get a good overview of the simulation job and its tasks.

The functionality can be enhanced by utilising appropriate colours and graphical aids for readability of the GUI and indication of important issues.

A typical example of using colour and graphical aids is the grouping by colour in combination with shapes or boxes in order to recognise similar functionality and to reduce the recognition time of the user. E.g. the same colours and fonts which are used in the CAE-plant design tool may be selected for the design of the GUI for Model^{CAT}, so providing easy recognition.

The aim should be to provide wording which is simple but functional and therefore easy to understand for the planning engineer. Charwat (1996) investigated aesthetic aspects of the GUI and came to the conclusion that it should be *"Better functional than beautiful"*.

Simplicity

Simplicity of design and good ordering of events are essential to ease access to the GUI. The reduction of required information is important to let the planning engineer concentrate on the main tasks.

Furthermore, it is helpful to provide graphical input of values, whenever it is possible instead of numerical input, which may be prone to errors.

Consistency

Consistency begins with the graphical representation of the GUI. The user should be supported with the same layout as in the CAE-plant design tool throughout the simulation tests. This applies to colours, buttons, windows and so on, but also to the graphical representation of contents. Consistent representations of GUIs and standardised placement of interactive GUI elements help to increase the workflow and acceptance by the user.

For more details of the three didactical design aspects and their application, refer to Franzkowiak (2004) or Fischer (2007). The didactical aspects of this section are taken into account in the following outline of the main three tasks of the GUI.

3.6.3 GUI-Task 1: Selection of Simulation Area

The planning engineer has to be guided into the CAE-plant design tool in order to start a simulation. Thus, an execution button is placed in the menu bar of the CAE-plant design tool such that, when clicked, displays an instruction text within the help system (see Section 3.6.6) to the planning engineer in order to lead them to select the area of the P&I diagram to be simulated. In contrast to simulations of the entire plant, the simulation of a selected area gives the planning engineer the opportunity to simulate only a selected plant section. However, the selection of the entire plant is also possible with this selection method.

The selection of a subsystem has to be established by an intuitive procedure, considering the P&I diagram, e.g. by drawing a square area with the computer mouse. After this selection, the analysis of the selected area (see Section 3.5.2) should be triggered automatically.

3.6.4 GUI-Task 2: Specification of Boundary Conditions, Initial Values and Simulation Parameters

The specification of boundary conditions, initial values and simulation parameters is a non-trivial task in most simulations. Only by the specification of adequate boundary conditions, initial values and simulation parameters, can the simulation be initialised meaningfully. While the knowledge about inputs and outputs of the entire plant are generally known, it is more difficult to specify the boundary conditions of a selected subsystem, because the knowledge of internal state information of subsystems and the respective parameters is often not known to the planning engineer. Additionally, initial values of differential terms must be specified at $t = 0\ s$. Furthermore, simulation parameters must be set before starting a simulation, e.g. simulation time.

Because of these difficulties the planning engineer must be led to the specification task in a systematic way. Therefore, this task is split into five sub-topics:

- 3.6.4.1 Methodology to Specify Boundary Conditions for the Process Domain
- 3.6.4.2 Methodology for Specifying Variables of the "List of Interconnections between Process and Control"
- 3.6.4.3 Methodology for Specifying Boundary Conditions for the Control Domain
- 3.6.4.4 Methodology for Specifying Initial Values of Differential Terms
- 3.6.4.5 Methodology for Specifying Simulation Parameters

These sub-topics are described in the following.

3.6.4.1 Methodology to Specify Boundary Conditions for the Process Domain

This section describes a methodology to guide the planning engineer through the task of specification of the boundary conditions for all input and output system defining variables for the process domain, which have been collected in the "List of open process connectors". With the use of simultaneously solved equation-oriented process simulation tools, the process part of the selected area is solved as one entire "system of equations" ("super equation"). This super equation contains the entire process FSCS model including the component model equations and the interconnection relationships.

In order to solve this super equation, its degrees of freedom must be eliminated. Because the internal degrees of freedom of each model are eliminated beforehand by the model developer (as defined in Section 3.3.3.1) and every FSCS model depends only on its input and output system defining variables, the super equation can only be solved after consistent boundary conditions have been specified. Only when the correct number and combination of boundary conditions for input and output system defining variables are available, can the super equation be completely specified. The general difficulty with the specification of boundary conditions is finding the right combination of boundary conditions.

In the following, an automatic procedure is proposed to support the planning engineer in this task. Fig. 3-29 gives an overview of the different tasks in this procedure. Each task is explained, starting with the determination of the number of degrees of freedom and ending with a consistent and fully specified super equation for the process domain.

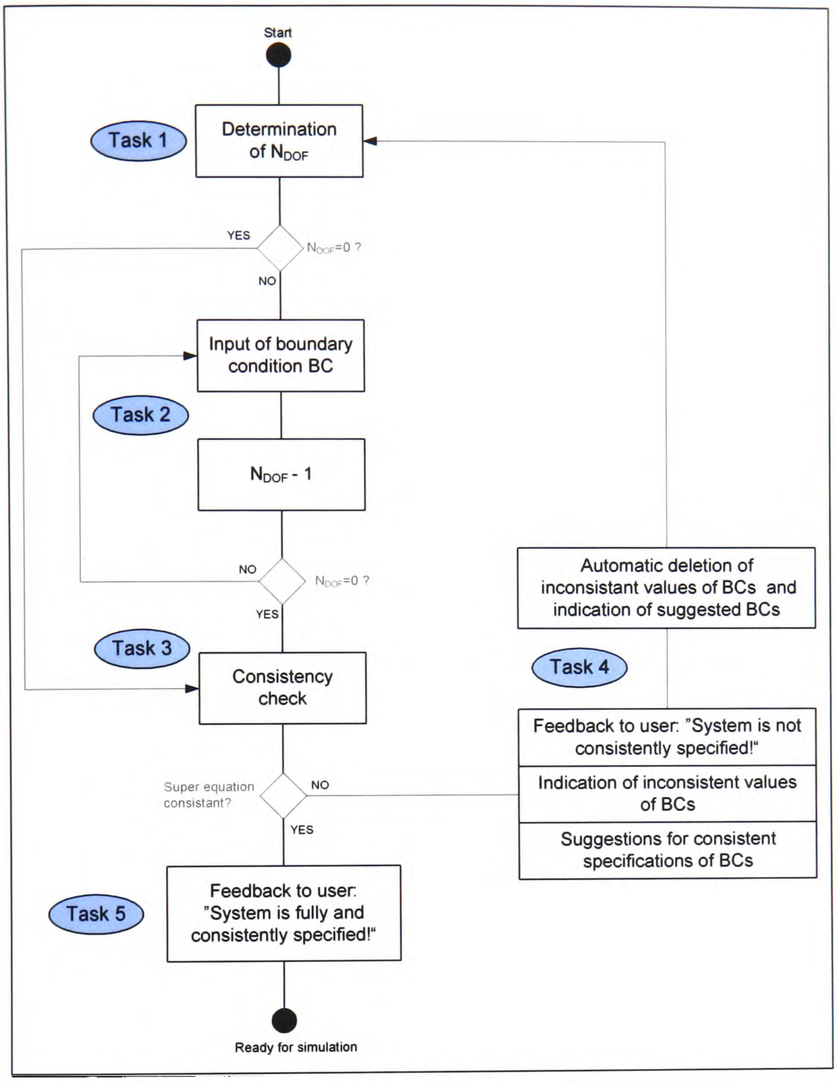


Figure 3-29: Flow chart of GUI specification task

Task 1: Determination of the Number of Required Boundary Conditions

The number of degrees of freedom (n_{DOF}), which have to be specified, is calculated by subtracting the total number of used model variables (n_V) from the number of equations (n_E), see Equation (2).

$$n_{DOF} = n_V - n_E \tag{2}$$

Therefore, each process FSCS model must be investigated with respect to the number of model variables (n_{Vi}) in the model equations and the number of equations (n_{Ei}). The total number of the model variables and the number of equations must be determined according to Equation (3) and Equation (4).

$$n_V = \Sigma(n_{V1} + n_{V2} + \ldots) \tag{3}$$

$$n_E = \Sigma(n_{E1} + n_{E2} + \ldots) \tag{4}$$

Task 2: Input of Boundary Conditions

Definition of boundary condition reduces the number of degrees of freedom (n_{DOF}). This input procedure must be continued until all degrees of freedom are eliminated ($n_{DOF} = 0$).

Task 3: Consistency Check

In the next step, these specified boundary conditions must be checked for consistency, because not every combination of boundary conditions is possible. Therefore, the "consistency check" feature of existing process simulation tools is used. This feature makes use of the entire system of equations and the set of specified boundary conditions, as illustrated in Fig. 3-30, and investigates if the specified boundary conditions are sufficient to initialise the super equation. If the consistency check succeeds, the super equation is fully specified, if not, the incorrectly specified boundary conditions are indicated. Suggestions for eliminating these deficiencies are given by the consistency check module. These results can be used for supporting the planning engineer with additional information.

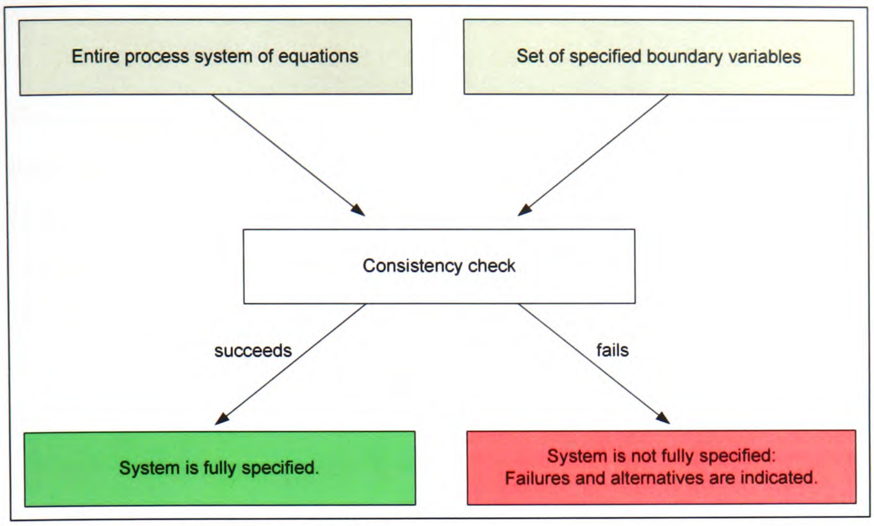


Figure 3-30: GUI task: Consistency check

Task 4: Support for Planning Engineer after Failed Consistency Check

If the consistency check fails, the user must be informed that the specification has failed, e.g. through an information window and by additional listings of the incorrect boundary conditions. The values of the incorrectly specified boundary conditions must be deleted and the input fields must be disabled from further use (e.g. provided with red colour) automatically. Input fields of suggested boundary conditions can be indicated with the colour green indicating assistance functionality. The overall procedure starts again with Task 1, this time however, with the recommended boundary conditions for the simulation tool.

Task 5: Support for Planning Engineer after Successful Consistency Check

If the consistency check is successful, the user must be informed that the system is fully specified (also in this case using an information window) and that the specification task for the boundary conditions of the process domain is completed.

3.6.4.2 Methodology for Specifying Variables of the "List of Interconnections between Process and Control"

The actuator variables of the "List of interconnections between process and control" must be specified at $t = 0\ s$. At the start of simulation it is important if the valve's stem is either in an opened, closed or a position in between. The project database must be checked, if such variables have already been set within the plant design. In this case, the value from the plant design data has to be taken as default, but may still be adjustable by the planning engineer.

3.6.4.3 Methodology for Specifying Boundary Conditions for the Control Domain

The boundary conditions for the control domain contain the data of the "List of control interconnections", such as the reference value for a PID controller. These values have to be specified in the same way as the boundary conditions of the process. However, the boundary conditions for the control domain should be checked in the plant design data, even if they have already been set during planning. As in the previous section, the value from the plant design data has to be taken as default.

3.6.4.4 Methodology for Specifying Initial Values of Differential Terms

With the use of differential equations in the FSCS models, it is necessary to define initial values (of the differential terms of the equation) at time $t = 0\ s$. Only by the correct specification of initial values, can the dynamics of the process FSCS model be simulated correctly. Therefore, every model equation of each process FSCS model must be checked for differential terms. The planning engineer must be supported in order to specify the required initial values for every differential term. For this purpose, the value ranges of all differential terms must be determined. In principle, these terms could be set to zero, or, predefined terms can be selected from the model description, if provided by the model developer, and must be suggested to the planning engineer in the GUI specification window.

3.6.4.5 Methodology for Specifying Simulation Parameters

In order to start a simulation, it is necessary to also specify some simulation parameters. These simulation parameters, like the simulation time, numerical solver or the sample time, must be provided to the simulators. Whereas the simulation time may be specified by the planning engineer, more sophisticated specifications, such as the choice of the numerical solver or the best sample time, should be automatically derived from the FSCS model descriptions, if possible.

3.6.5 GUI-Task 3: Start of Simulation and Presentation of Simulation Results

After the specification of all boundary conditions, initial parameters and simulation parameters, with all degrees of freedom eliminated, the user must get an indication that s/he can start the simulation.

Presentation of Simulation Results (GUI)

Simulation results should be directly presented in the context of the CAE-plant design tool. The respective simulators may be running in the background in order to produce the simulation results, not distracting the planning engineer with additional simulation environments.

It could be advantageous to design strategies to support the planning engineer with the interpretation of the simulation results. This topic is outlined in the discussion Section 6.4.

3.6.6 Help System

For software tools - in this case Model^{CAT}- it is essential to support the user with corresponding information or help functions. Barker and co-workers (1993a) postulate access to help files from any point in the program's execution. This position is also taken for this study. With regard to easy use, an online help document may be used. One advantage of the online help is direct access during work using context sensitive help functions.

Online help functions can be easily provided via the Internet and can provide more detailed information about the respective topics. Within an online help, planning engineers can get information at any time about any specific topic. Furthermore, online help functions can be easily altered and extended.

3.7 Requirements on Materials for the Model^{CAT} Approach

The materials and resources are defined predominantly by the nature of this work: computer facilities and software components. The requirements for the simulation procedures predominantly depend on the software tools used and must be adjusted to their requirements.

In Table 3-3 the requirements for the software tools used for the Model^{CAT} approach are described.

Table 3-3: Software requirements

Software tool	Requirements
CAE-plant design tool	– Object-oriented database for specification input without redundancy – Open system structure – Integrated P&I diagram and function block diagram – Easy import/export of data – Provision of all required design documents (tables, instructions, bills of material, etc.) – Easy integration of proposed FSCS model catalogue
Process simulation tool	– Simultaneously solved equation-oriented approach – Open system structure (interface) – Dynamic model simulation – Organised model catalogue including basic processes and apparatuses – Interface to control simulation (co-simulation)
Control simulation tool	– Block-oriented approach – Open system structure (interface) – Dynamic model simulation – Interface to process simulation (co-simulation)

Programming language requirements for the development of the model aggregation module and GUI depend on the choice of the CAE-plant design tool.

4 Prototype Realisation

In this chapter the prototype realisation of the methods outlined in Chapter 3, is presented. The prototype realisation is implemented using existing (commercial) software tools, namely a CAE-plant design tool and a simulation environment, including a process simulation tool and control simulation tool. General consideration of the choice of the software tools is given in Section 4.1.

In order to establish plant simulation models, based on the results of the plant design, for use in the simulation environment, the Model Aggregation Module (MAM) and the smart GUI were developed by the author and presented *inter alia* in Hoyer *et al.* (2005b) and Hoyer *et al.* (2006b). These interfacing coded modules enable the connection between the user, CAE-plant design and the simulation environment, as illustrated in Fig. 4-1.

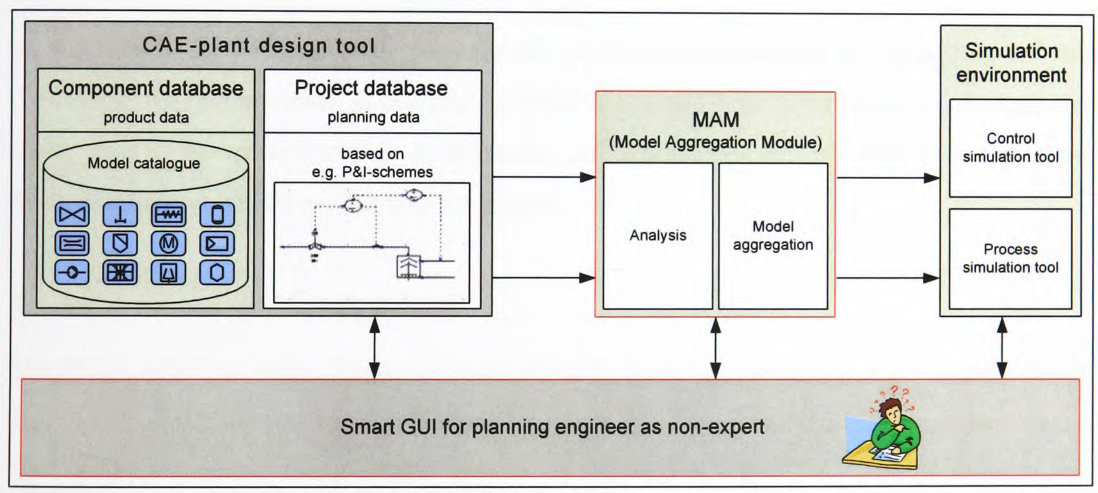


Figure 4-1: Modules used in the Model^{CAT} approach

The MAM was developed to analyse the results of the CAE-plant design tool and to use this information (including the FSCS models) to aggregate the required plant simulation models separately for process and control domain. The smart GUI module was realised to support the planning engineers in the selection of the plant to be simulated, the specification of boundary conditions and starting the simulation.

Similarly to Chapter 3, this chapter is subdivided into subtasks. After general considerations on the choice of the software tools and hardware used for the prototype realisation (Section 4.1), the essential features of the chosen CAE-plant design tool are described (Section 4.2). In this case, the definition of the FSCS models is demonstrated using an illustrative example in the chosen simulation tool syntax (Section 4.3), followed by the realisation of the application used to integrate the FSCS models into the CAE-plant design tool (Section 4.4). Finally, based on the results of the CAE-plant design and the integrated FSCS models, the methods for the MAM and GUI tasks were implemented using a concise example (Section 4.5) and the feasibility of this comprehensive approach is demonstrated.

4.1 General Considerations of the Model^{CAT} Prototype

4.1.1 Software Tools Used

In this section the software tools used for the prototypical realisation are described. For this realisation, the requirements, as outlined in Table 3-3, which have been placed on the software tools, have to be accommodated. Additionally, the availability and the costs were important, due to the limited budget available to the project.

4.1.1.1 CAE-Plant Design Tool

The choice of a CAE-plant design tool was primarily based on the review of CAE-plant design tools by Rauprich and co-workers (Rauprich *et al.*, 2002). They investigated several approaches from different vendors using similar criteria, as outlined in Table 3-3. The information presented by Rauprich *et al.* (2002) was sufficient to select the appropriate CAE-plant design tool; only SmartPlant™ (Intergraph, 2000) and COMOS PT™ (Innotec, 2002) fulfilled the essential requirements for a CAE-plant design tool. COMOS PT™ was chosen, because this tool fulfils all requirements listed in Table 3-3.

4.1.1.2 Control Simulation Tool

As a control simulation tool, the state of the art software tool in the control domain Matlab/ Simulink™ (Mathworks, 2003b) was selected. It fulfils the basic requirements of Table 3-3. In research, Matlab/Simulink™ is almost a *de facto* standard because of its flexibility and its open interfaces. The block-oriented simulation approach of Simulink™ allows a hierarchical approach and rapid generation of control schemes. Furthermore, this software tool enables the use of commercial toolboxes such as the System Identification Toolbox and the Control System Toolbox (Mathworks, 2003a) or 3rd party toolboxes such as ICAI (Körner, 1999) and ICAC (Syska, 2004). Additionally, Simulink™ allows co-simulation with equation-oriented process simulation tools, such as SpeedUp (Aspen Custom Modeller™) and gPROMS™.

4.1.1.3 Process Simulation Tool

The selection of the process simulation tool was based on evaluations of the literature review (Section 2.3) and tests by van der Meer (2002). Some of the widely used equation-oriented approaches are adopted in SpeedUp (Aspen Custom Modeller™), ABACUSSII, gPROMS™ and company in-house solutions, see Section 2.3.2. The company in-house solutions were not available for the author's use and only SpeedUp and gPROMS™ fulfil the requirements completely. SpeedUp and gPROMS™ offer a standardised interface to Simulink™ for co-simulation, a requirement that ABACUSSII does not support. After this assessment, the process simulation environment gPROMS™ was selected because gPROMS™ and Simulink™ can be interconnected via the standard interface (called gO:Simulink™), see PSE (2003a).

The chosen software tools fulfilled all the essential requirements that were defined by the author, as listed in Table 3-3.

4.1.1.4 Self-Developed Software Modules (MAM and GUI)

Generally, the choice of software languages heavily depends on the selection of the CAE-plant design tool. With the selection of COMOS PT™, the choice of the software programming package was predetermined. Therefore, the software programming package Visual Basic© (Microsoft, 2000) was used for the prototype realisation of MAM and GUI. Visual Basic offers a wide range of design and functional features for designing GUIs.

4.1.1.5 Help System

In designing the help system, the software WinCHM from Softany (Softany, 2006) was used because of its common and easy usage. It is accessible by standard browsers such as Microsoft Explorer, Firefox or Netscape.

All software tools that were used for the prototype realisation of the Model^{CAT} approach are outlined in Table 4-1.

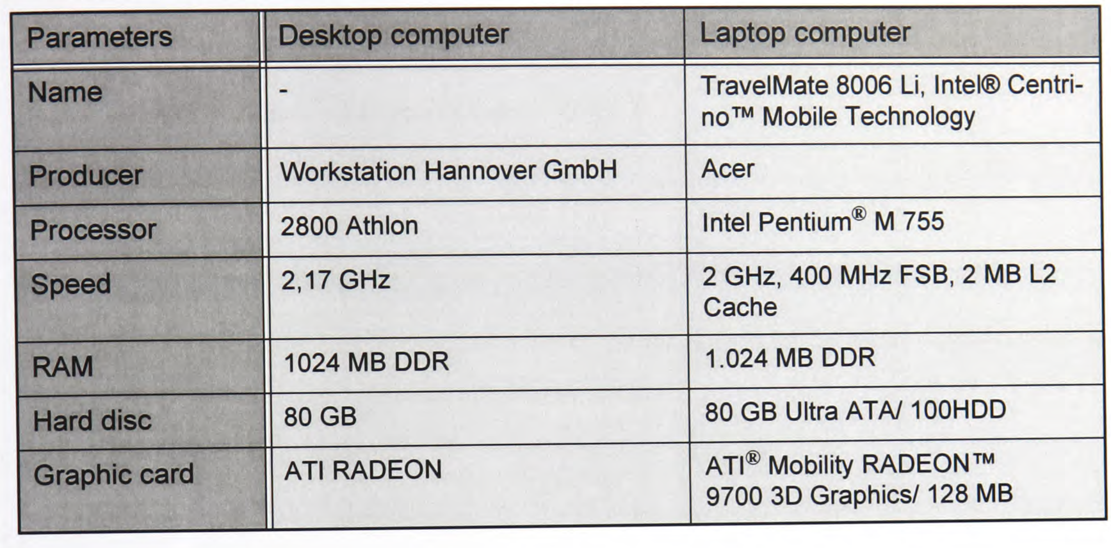
Table 4-1: Software tools used for the prototype realisation

Type of tool	Software	Producer	Version
CAE-plant design tool	COMOS PT™	Innotec (Innotec, 2002)	Version 7.0.4
Control simulation tool	Matlab/ Simulink™	Mathworks (Mathworks, 2003)	Version 6.5.0 Release 13.0.1
Process simulation tool	gPROMS™	Process System Enterprise (PSE, 2003b)	Version 2.2.6 Version 2.3.1
Interface between process and control simulation tool	gO:Simulink™	Process System Enterprise (PSE, 2003a)	Version 2.2 Version 2.3.1
MAM	Proprietary solution	Author	Version 2
GUI	Proprietary solution	Author	Version 2
Programming language	Visual Basic©	Microsoft (Microsoft, 2000)	Version 6.0, SP6 2003
Help system	WinCHM	Softany (Softany, 2006)	3.22 (demo version)

4.1.2 Hardware Used

Two computers were used in the execution of the work presented in this thesis. The hardware features are listed in Table 4-2.

Table 4-2: Hardware features used for the prototype realisation



Parameters	Desktop computer	Laptop computer
Name	-	TravelMate 8006 Li, Intel® Centrino™ Mobile Technology
Producer	Workstation Hannover GmbH	Acer
Processor	2800 Athlon	Intel Pentium® M 755
Speed	2,17 GHz	2 GHz, 400 MHz FSB, 2 MB L2 Cache
RAM	1024 MB DDR	1.024 MB DDR
Hard disc	80 GB	80 GB Ultra ATA/ 100HDD
Graphic card	ATI RADEON	ATI® Mobility RADEON™ 9700 3D Graphics/ 128 MB

4.2 Plant Design Procedures in COMOS PT™

In this section the main features (functionalities) of plant design within COMOS PT™, which are required for the realisation within the Model^{CAT} approach, are outlined. These features are summarised in the following sections:

1. General User Interface in COMOS PT™
2. Component and Project Database
3. Design Utilities: P&I Diagrams and Function Block Diagrams

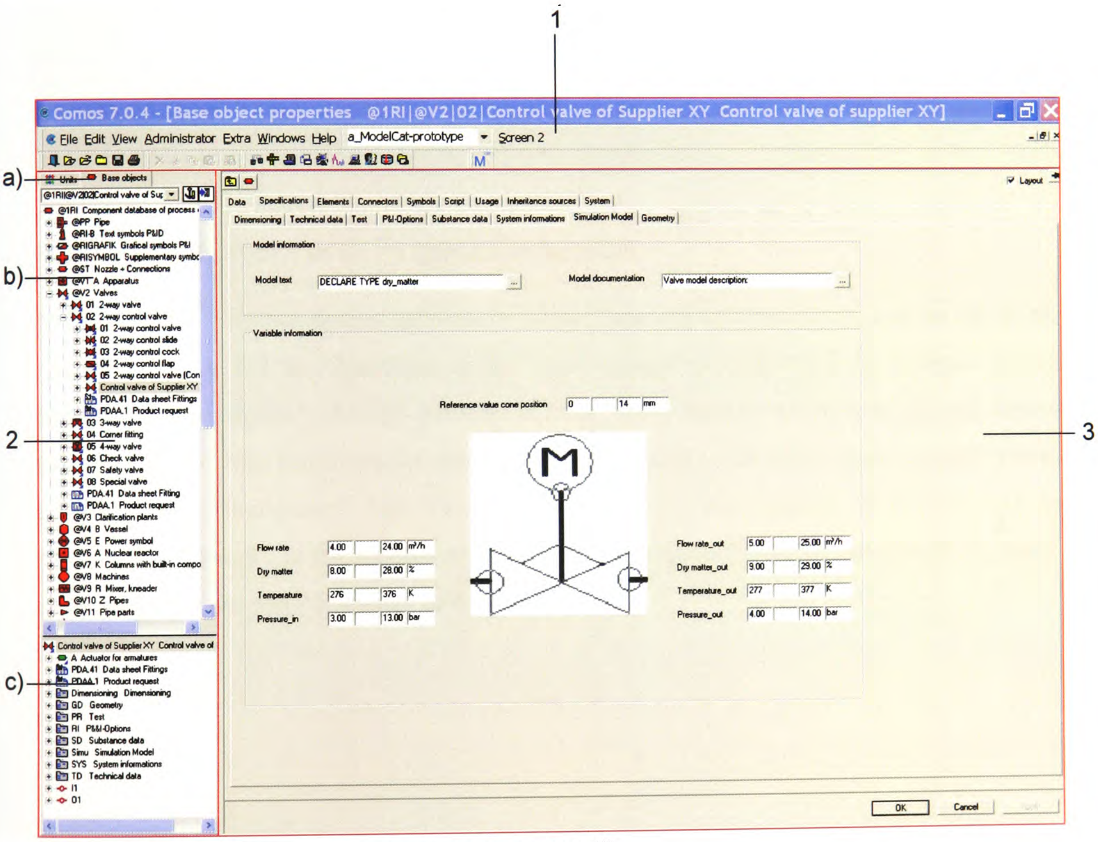
Finally, an overview of the results of CAE-plant design using COMOS PT™ is given.

4.2.1 General User Interface in COMOS PT™

Generally, the COMOS PT™ window may be separated into three areas, as illustrated in Fig. 4-2:

- Area 1: Menu and toolbar
- Area 2: Navigator with object tree
- Area 3: Working area

Short descriptions of these areas are outlined in Table 4-3.



a_ModelCat-prototype

Model text: DECLARE TYPE dry_matter

Model documentation: Valve model description:

Reference value cone position: 0 … 14 mm

Flow rate: 4.00 … 24.00 m³/h
Dry matter: 8.00 … 28.00 %
Temperature: 276 … 376 K
Pressure_in: 3.00 … 13.00 bar

Flow rate_out: 5.00 … 25.00 m³/h
Dry matter_out: 9.00 … 29.00 %
Temperature_out: 277 … 377 K
Pressure_out: 4.00 … 14.00 bar

Figure 4-2: General user interface of COMOS PT™

Table 4-3: Short descriptions of the areas shown in Fig. 4-2 (COMOS PT™)

Area	Name of area	Short description
1	Menu and toolbar	For navigating via menu or toolbar to the functions of COMOS PT™.
2	Navigator with object tree	a) "Units" or "Base objects" (called "Chapters") for switching between project and component database. The selected "chapter" is shown within the navigator of COMOS PT™. b) "Main navigator" for visualisation of the object trees of the respective database. c) "Detailed navigator" for visualisation of specific information of the respective database, e.g. specification and interconnection (connectors) data.
3	Working area	For parameter specification of objects, operation property windows, P&I diagrams and function block diagrams, etc.

4.2.2 Component and Project Database

In COMOS PT™ the component database is called "Base objects" and can be opened within the navigator. In Fig. 4-3 the object tree of the "Base objects" of COMOS PT™ is demonstrated, including "Base objects" of other domains and the "Base objects" of the process and control domain. Fig. 4-3 also illustrates the object-oriented hierarchy within the "Base objects" exemplified by the "Fluid pump". The "Fluid pump" belongs to the category "Machines" and the sub-category "Pump". In the detailed window of the "Navigator" window, are placed the predefined specifications and connectors of the "Fluid pump".

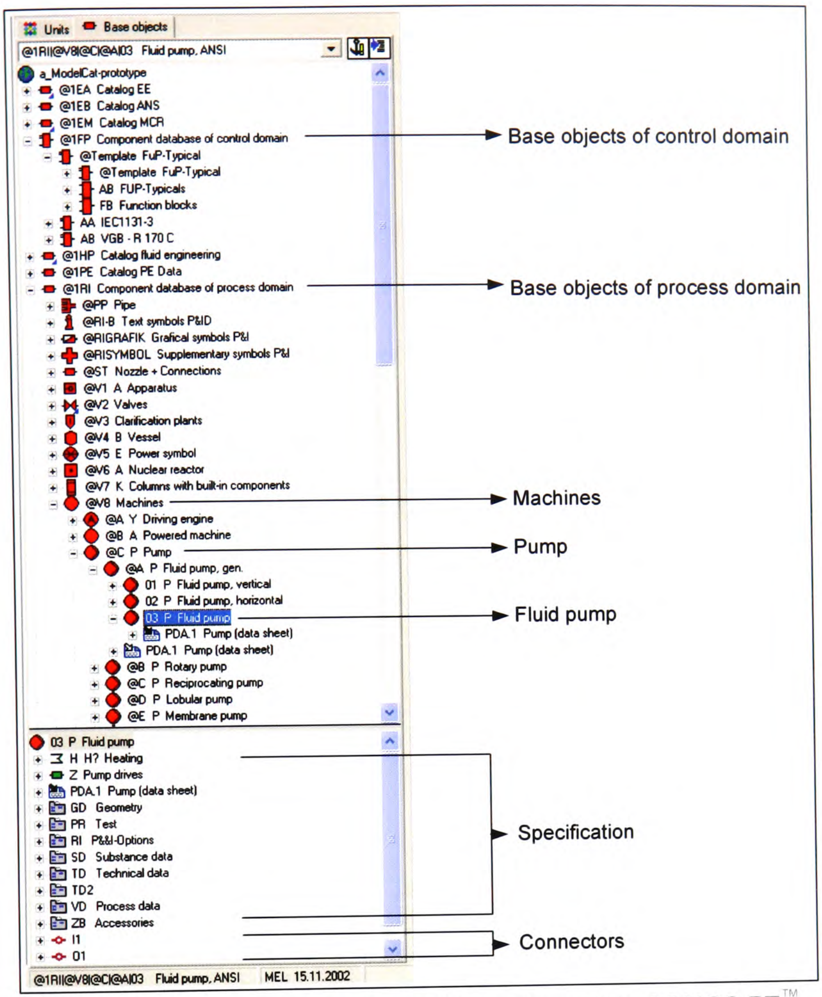


Figure 4-3: Component database ("Base objects") in COMOS PT™

The project database is organised in "Units" in COMOS PT™. The project database is generated by the transfer of component models from the component database, including all specifications, connectors and graphical properties. Fig. 4-4 shows an example of a project database in COMOS PT™. The plant project is subdivided into several hierarchical levels. The hierarchy outlined, which includes "Factory/Building/Production" to "Main plant" and to "Part unit" is a default classification hierchary within COMOS PT™. The planning objects are located below the "Part unit", which are clearly arranged by folders, divided into "Apparatus", "Fittings", "Pipe work" and "EMCR technique" (Control functions). Furthermore, the diagrams (i.e. P&I diagrams) used and lists of components are generally located below the "Part unit" hierarchy level. Function block diagrams are automatically located below the function eye, see Fig. 4-4. The specifications and interconnections which have been defined within the plant design are located in the "Detailed navigator".

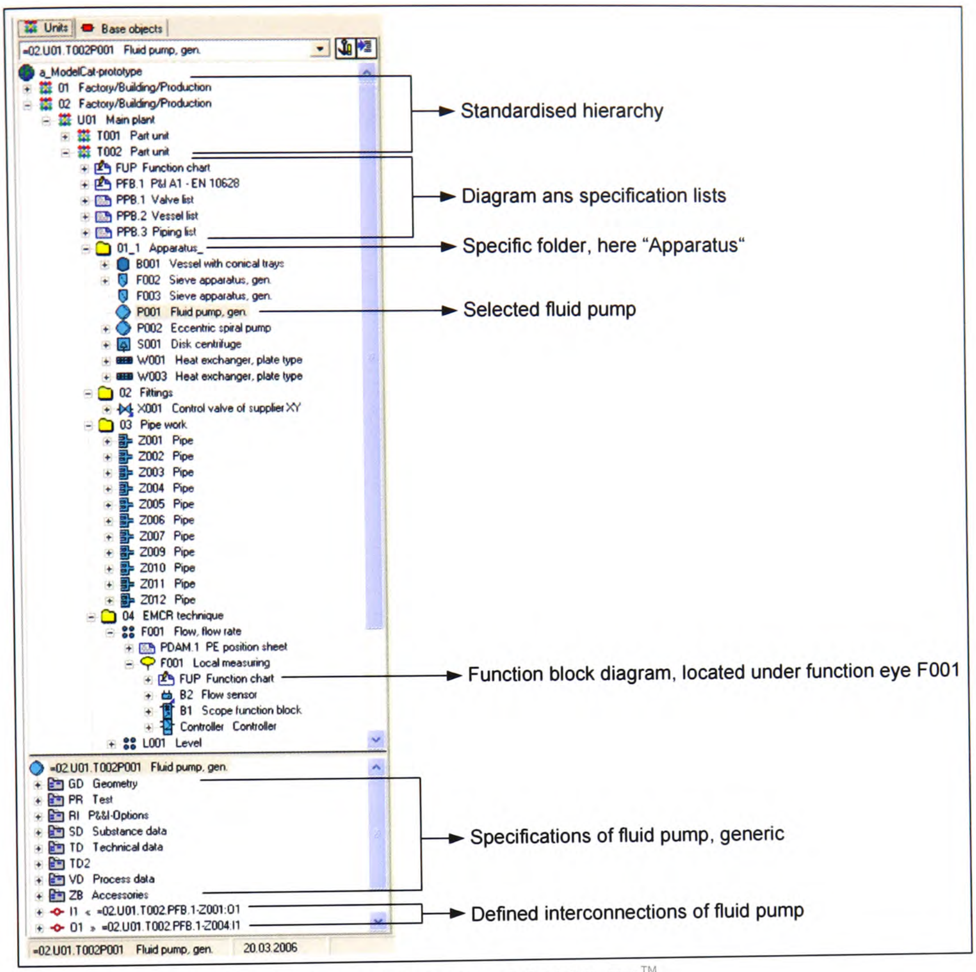


Figure 4-4: Project database ("Units") in COMOS PT™

Within plant design, the most convenient and widespread way to integrate components into the project database is done by using P&I diagrams or function block diagrams. The design utilities within COMOS PT™ are outlined in the following section.

4.2.3 Design Utilities: P&I Diagrams and Function Block Diagrams

In COMOS PT™ process components and function eye's can easily be aggregated in P&I diagrams. For this purpose "Base objects" of the component database are selected and placed via "Drag and Drop" in the P&I diagram and connected.

Parallel to the carry over and aggregation within the P&I diagram, the information about the component (or function eye) is automatically stored in the project database, see Fig. 4-5. The function eyes are connected via signals from the sensor and to the actuator.

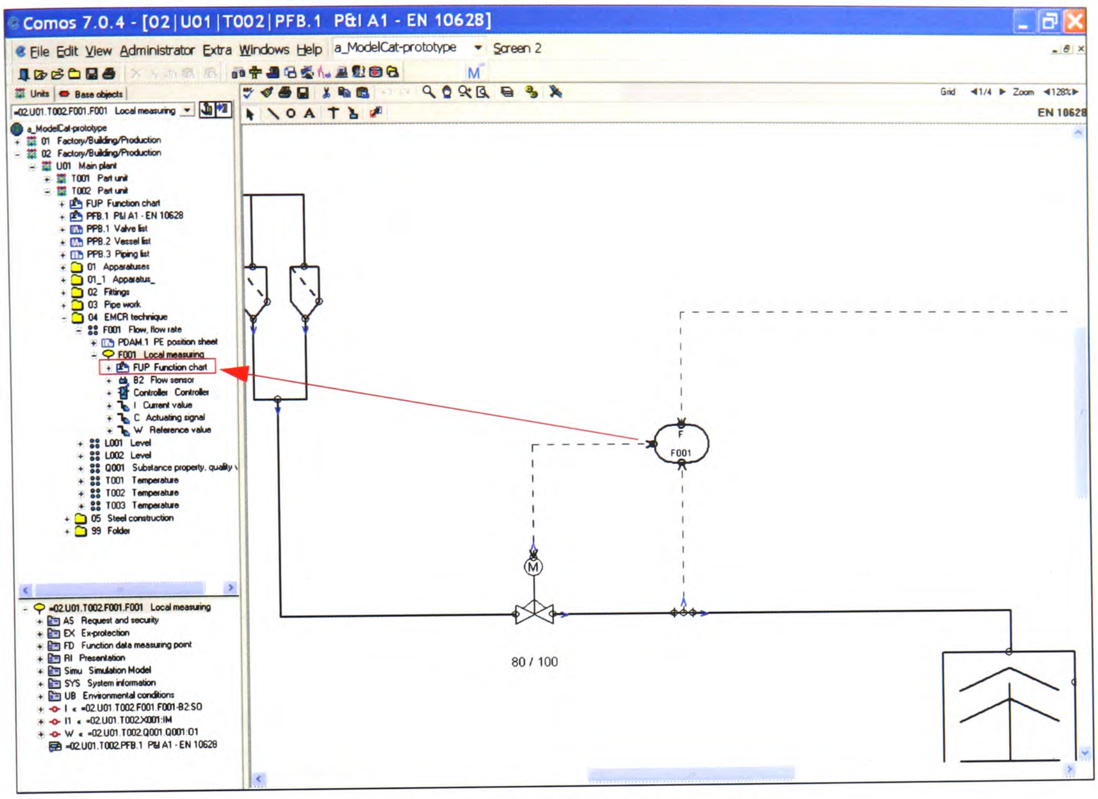
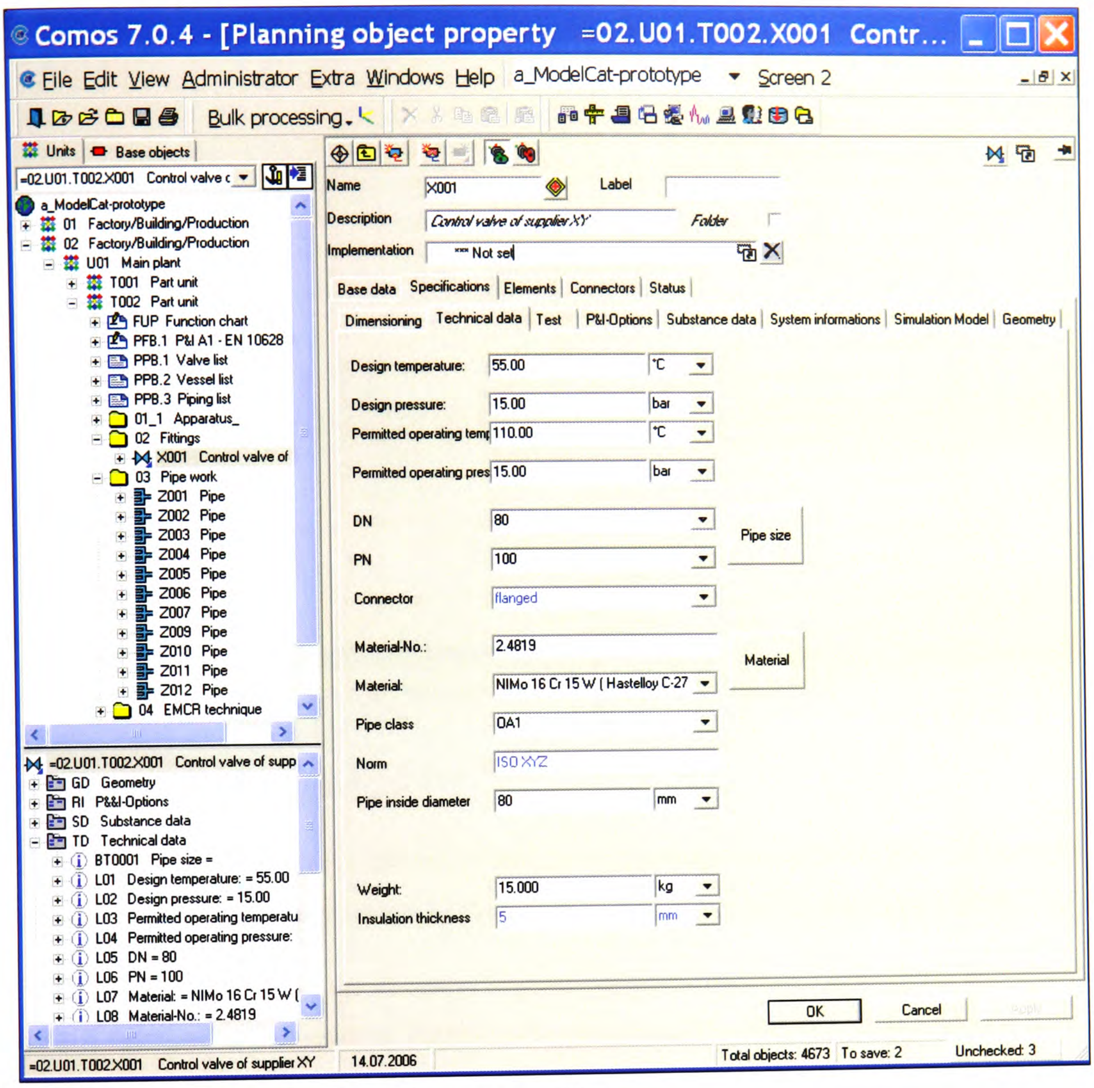


Figure 4-5: Function block diagram's location in project's database

The specification of a single component can be done by the selection of the respective object either within the object tree of the project database ("Navigator") or within the P&I diagram. An example of a specification window (only the limited working area) is shown in Fig. 4-6. Here, the technical data of a control valve (X001) is presented, which is partly predefined in the "Base objects" and which must be specified by the planning engineer within the plant design process.



Name: X001
Label:
Description: Control valve of supplier XY
Folder
Implementation: *** Not set

Design temperature: 55.00 °C
Design pressure: 15.00 bar
Permitted operating temp: 110.00 °C
Permitted operating pres: 15.00 bar
DN: 80
PN: 100
Connector: flanged
Material-No.: 2.4819
Material: NiMo 16 Cr 15 W (Hastelloy C-27
Pipe class: 0A1
Norm: ISO XYZ
Pipe inside diameter: 80 mm
Weight: 15.000 kg
Insulation thickness: 5 mm

=02.U01.T002.X001 Control valve of supplier XY | 14.07.2006 | Total objects: 4673 | To save: 2 | Unchecked: 3

Figure 4-6: Specification of component properties

Defining a Corresponding Function Block Diagram

The corresponding function block diagram must be defined using function blocks in the control component database. This procedure is equivalent to the generation of P&I diagrams. Control function blocks are dragged from the control component database, dropped on the function block diagram and connected via signals. Furthermore, the function block can be specified in the same way as the procedures within P&I diagrams. An example of a function block diagram is shown in Fig. 4-7.

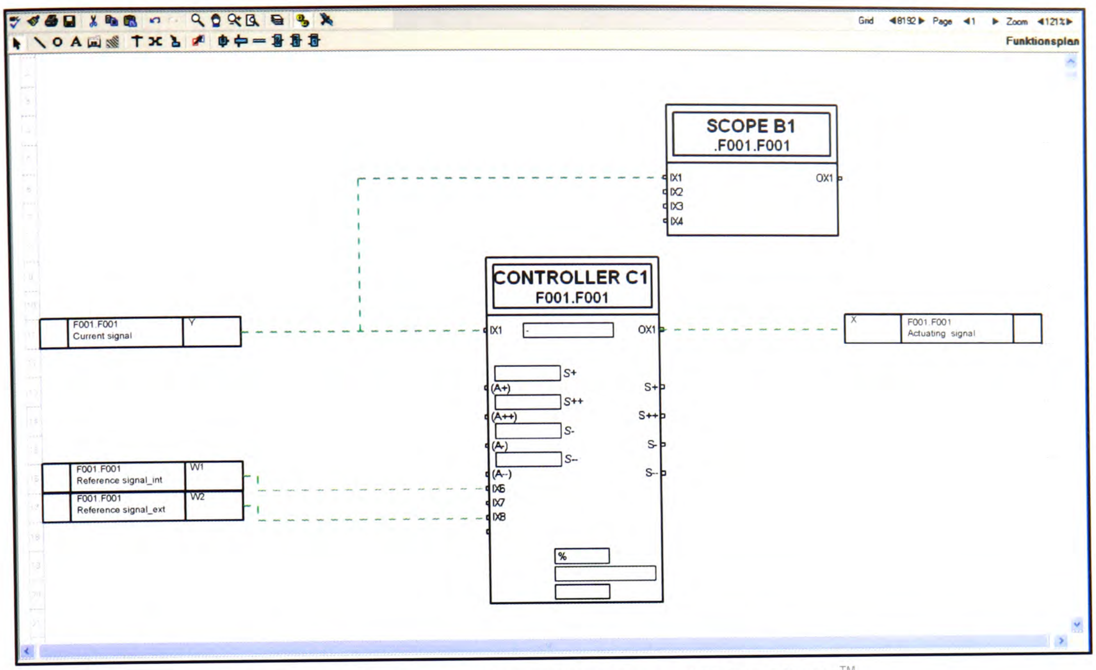


Figure 4-7: Function block diagram within working area of COMOS PT™

4.2.4 Results of COMOS PT™

By the end of plant design (i.e. the end of detailed engineering) the chemical plant with all of its subsystems and components should be fully specified and ready to be procured and built.

The definitions (including the choice of components/function blocks, aggregation and specification) described in the previous sections, present the basis for the plant realisation and for the generation of all the required documents, such as P&I diagrams, function block diagrams, manuals, component lists, etc., which can automatically be generated with COMOS PT™.

The fully specified plant, including the FSCS models, is the prerequisite for the automatic simulation strategy. The definition of the FSCS models is described in the following chapter.

4.3 Definition of the FSCS Model Catalogue

The prerequisite for an automatic generation of plant models at the end of detailed engineering, is the availability of FSCS models. For the prototypical realisation, the FSCS model catalogue contains process FSCS models in gPROMS™ format and control FSCS models in Simulink™ format. Before presenting some examples, the general structures of both simulation tool's formats are described in the following.

4.3.1 Process FSCS Models in gPROMS™ Format

The general data structure of gPROMS™ process models can be categorised into seven parts: TYPE declaration, MODEL name, PARAMETER declaration, VARIABLE declaration, STREAM declaration, SET values and EQUATION. These parts are shown graphically in Fig. 4-8 and are described in more detail in Table 4-4.

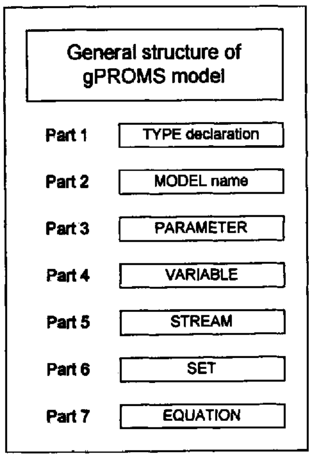


Figure 4-8: General gPROMS™ model

Table 4-4: Parts and short descriptions of process models (gPROMS^{TM})

Part	Name of Part	Short description
1	TYPE declaration	The "TYPE" declaration part is categorised into variable and stream types: – "Variable TYPE" declaration The variable types used for the model description must be declared. Therefore, the variable's lower bound, upper bound and the default value must be defined. – "Stream TYPE" declaration By the declaration of the stream types, the system defining variables are defined, corresponding to the media flow stream. In the prototype realisation, the system defining variables are flow rate, pressure, temperature and dry matter.
2	Definition of MODEL name	The appropriate model name (XYZ) must be set by the following expression "MODEL XYZ".
3	PARAMETER declaration	The "PARAMETER" part is used to declare the parameters of a "MODEL". Parameters are time-invariant quantities that will not, under any circumstances, be the result from a calculation. These parameters indicate the specific behaviour of the FSCS model of a specific component supplier.
4	VARIABLE declaration	The "VARIABLE declaration" part is used to declare the variables of a "MODEL". These variables represent quantities that describe the time-dependent behaviour of the process. By the "VARIABLE declaration" the name and the "Variable type" are defined.
5	STREAM declaration	The types of the input and output system defining variables are combined to "Flow Streams". Additional external inputs or outputs, like signals from sensors or actuating signals, are implemented as "CONNECTIONS". For this prototype realisation all outgoing signals from sensors are defined by "SO" connections and all incoming actuator signals by "IM" connections.
6	SET values of the parameters	All parameters must be assigned with appropriate values. The values of the declared parameters are dedicated to the FSCS model of a specific component supplier, e.g. the specific number of the nozzles or the nozzle diameter.
7	EQUATION	The "EQUATION" part is used to define the equations which determine the models behaviour with the parameters and variables already declared in the "PARAMETER" and "VARIABLE" part. All equations of each FSCS model are defined under the prerequisite that all equations only depend on the input and output system defining variables. In gPROMS™, the symbol "$" before a variable name denotes the differential term with respect to time of that variable. Each of these "differential terms" must be specified for t = 0 s. In order to start dynamic simulation, these differential terms must be specified with appropriate values, see therefore Section 4.5.4.

4.3.2 Process FSCS Model Examples

In order to illustrate the specific and unique characteristic of a physical (ready-to-buy) component, the FSCS model must fulfil its essential requirements (in this case only the functional aspects which are required to establish simulations are investigated). For the valve example outlined in Section 3.3.1, these functional characteristics contain the valve cone characteristic, the valve specific c_{V0}, c_{V100}, x_{100}, the maximal pressure difference or the drive mechanism. Other technical aspects, such as the diameter of the connectors or the allowed density range, are specific to this component and should be used for plausibility checks (see Section 6.6 for more details). Demonstrating this on a real physical component example, the control valve "VARVIVENT" (GEA Tuchenhagen, Büchen, Germany), these functional and also technical data are summarised in Table 4-5 and outline the uniqueness of the "VARVIVENT" control valve, including its specific characteristics.

Table 4-5: Typical characteristics of a FSCS model (control valve)

Abbreviation/ Symbol	Name	Specific parameters
EP	Equal percentage characteristic of the valve cone	See Equation (5)
c_{V0}	Valve coefficient of closed valve	0.014
c_{V100}	Valve coefficient of fully opened valve	25
x_{100}	Maximal stem position path	20 mm
	Drive mechanism	Pneumatic
Δp_{max}	maximal pressure difference	10 bar
DN	Diameter of connector	50 mm

From the functional data, the characteristic curve of the control valve can be determined, by using Equation (1). The definitive function for the valve coefficient of the control valve "VARVIVENT's" characteristic is presented in Equation (5).

$$c_V(x) = 0,014 \cdot e^{\ln\left(\frac{25}{0,014}\right) \cdot \frac{x}{20}} \qquad (5)$$

Using such specific characteristic, simulation models can be lifted to another level of detail. This is demonstrated using a simplified valve flow equation, based on incompressible flow, as given by Equation (6). *FR* is the flow rate through the valve, *K* is a constant that depends on the units used in this equation, $c_v(x)$ is the valve coefficient, which is dependent upon the stem position *x*, see Equation (5). ρ is the density of the fluid, p_{in} is the pressure at the inlet to the control valve, and p_{out} is the pressure at the exit of the control valve.

$$FR = k \cdot c_V(x) \cdot \sqrt{\frac{abs(p_{in} - p_{out})}{\rho}} \cdot \mathrm{sgn}(p_{in} - p_{out}) \qquad (6)$$

The code of the control valve is written in gPROMS™ language and presented in Appendix B, along with other FSCS models, which have been developed for the prototype realisation. These include the separator, heat exchanger, pipe work, flow sensor and dry matter (quality) sensor.

4.3.3 Control FSCS Models in Simulink™ Format

Generally, Simulink™ function block models are structured into five parts, very similar to the gPROMS™ process model. These parts are: "Block type", "Name", "Ports", "Position" and a "Block specific function", as shown in Fig. 4-9 and described in the corresponding Table 4-6.

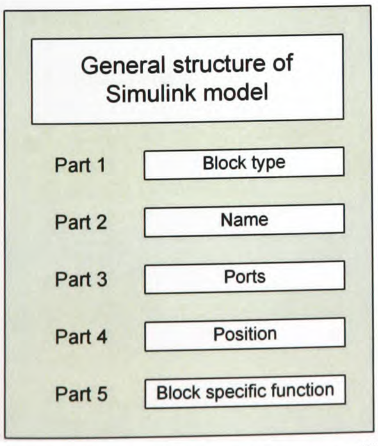


Figure 4-9: General Simulink™ function block model

Table 4-6: Parts and short descriptions of control models in Simulink™

Part	Name of Part	Short description
1	Block type declaration	In this part, Simulink™ specific block types are declared. For the prototype realisation only standard Simulink™ function blocks, such as "Scopes" or "Controller" blocks, are used.
2	Name declaration	The name of the Simulink™ function block model is declared, e.g. "PID Controller".
3	Ports declaration	The number of ports of the Simulink™ function block is declared and realised in the following format [number of input port, number of output ports], e.g. [2,1] comprises two input ports and one output port.
4	Position declaration	The declaration of the position is required to define the position of the function block on the later entire Simulink™ scheme, see therefore Chapter 4.4.4.
5	Block specific function	Block specific functionality is declared, e.g. the P-I-D (Proportional-Integral-Derivative) terms of the PID controller block or the expression of "function" block.

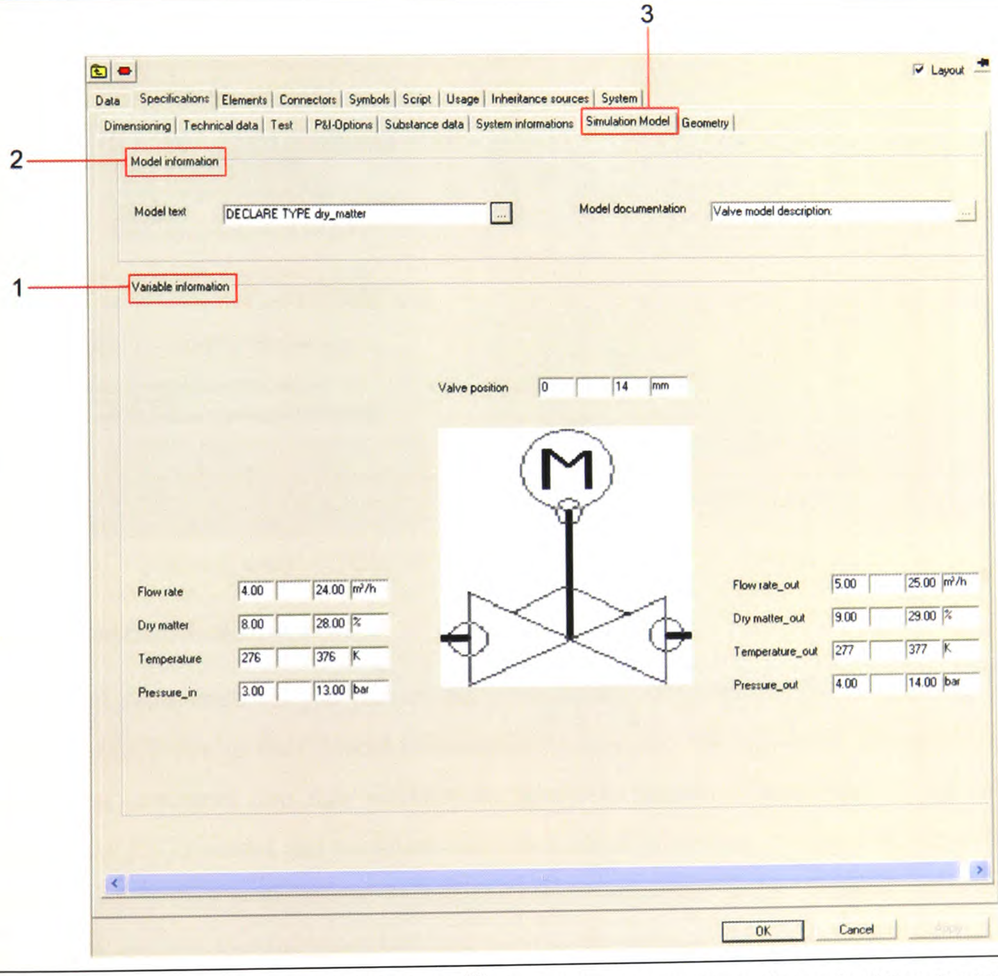
4.3.4 Control FSCS Model Examples

As in process FSCS models, control FSCS models are unique with regard to the behaviour and parameters of the chosen process control system (i.e. its function blocks). In the prototype realisation the function blocks of Simulink™ play the role of the process control system function blocks. The following Simulink™ function blocks are taken into account for this prototype realisation:

- PID controller
- Sinks
- Sources
- Subsystem
- User defined functions
- Math operations

4.4 Embedding FSCS Models into COMOS PT™

The FSCS models are integrated in the component database hierarchy in COMOS PT™, as additional object specifications in the form of ASCII-files. Therefore, the specifications of a particular planning object are extended by an extra layer called the "Simulation model", see Fig. 4-10 (3). This layer contains the "Model information" (2) and "Variable information" (1) frames. The "Model information" frame (2) is subdivided into the "Model text" and the "Model documentation" of the FSCS models.

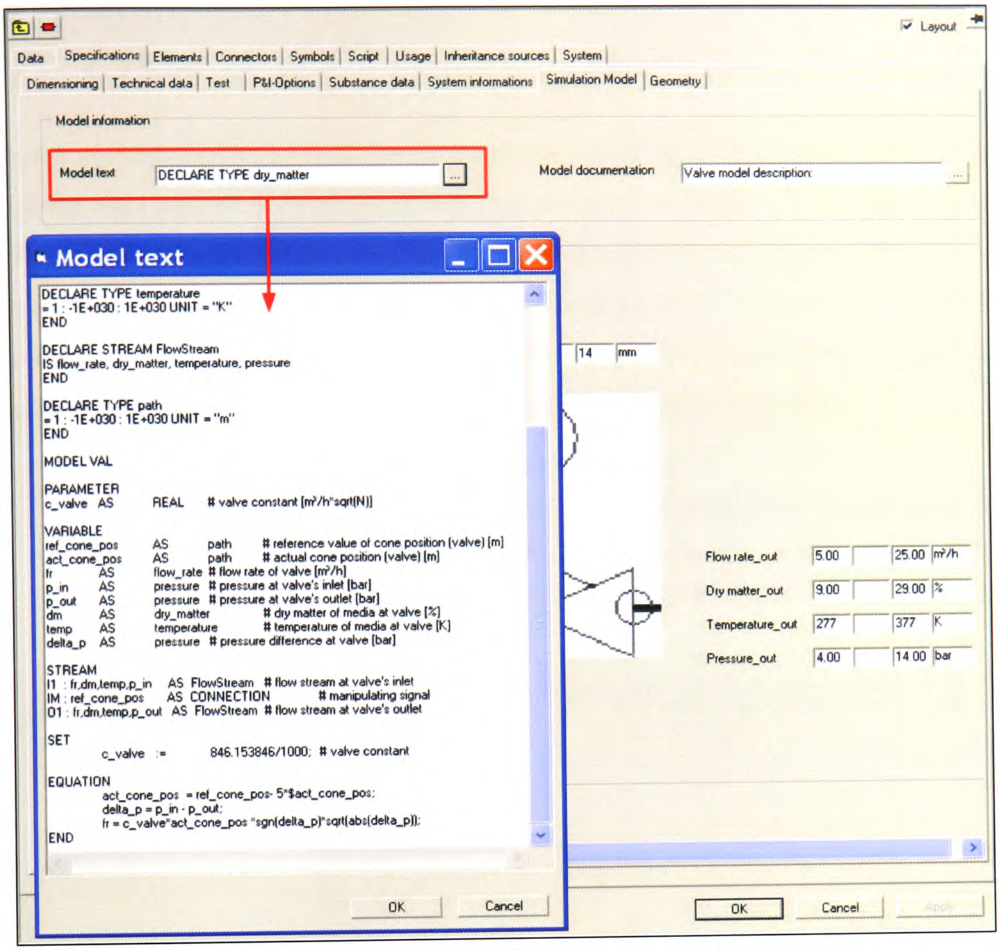


Model text: DECLARE TYPE dry_matter
Model documentation: Valve model description:

Valve position: 0 | | 14 | mm
Flow rate: 4.00 | | 24.00 | m³/h
Dry matter: 8.00 | | 28.00 | %
Temperature: 276 | | 376 | K
Pressure_in: 3.00 | | 13.00 | bar
Flow rate_out: 5.00 | | 25.00 | m³/h
Dry matter_out: 9.00 | | 29.00 | %
Temperature_out: 277 | | 377 | K
Pressure_out: 4.00 | | 14.00 | bar

Figure 4-10: Working area of COMOS PT™: Simulation model layer for embedding the FSCS model and variable information

The integration of the simulation code as "Model text" (in ASCII) within COMOS PT™ is demonstrated in Fig. 4-11. For this prototype, the collection of model and variable information has been realised manually within COMOS PT™, but may be automatically integrated in future, see Section 6.2.



Model text: DECLARE TYPE dry_matter

Model documentation: Valve model description

14 mm

Flow rate_out 5.00 25.00 m³/h
Dry matter_out 9.00 29.00 %
Temperature_out 277 377 K
Pressure_out 4.00 14.00 bar

Model text

```
DECLARE TYPE temperature
= 1 : -1E+030 : 1E+030 UNIT = "K"
END

DECLARE STREAM FlowStream
IS flow_rate, dry_matter, temperature, pressure
END

DECLARE TYPE path
= 1 : -1E+030 : 1E+030 UNIT = "m"
END

MODEL VAL

PARAMETER
c_valve   AS        REAL      # valve constant [m³/h*sqrt(N)]

VARIABLE
ref_cone_pos        AS        path       # reference value of cone position (valve) [m]
act_cone_pos        AS        path       # actual cone position (valve) [m]
fr          AS        flow_rate # flow rate of valve [m³/h]
p_in        AS        pressure  # pressure at valve's inlet [bar]
p_out       AS        pressure  # pressure at valve's outlet [bar]
dm          AS        dry_matter           # dry matter of media at valve [%]
temp        AS        temperature          # temperature of media at valve [K]
delta_p     AS        pressure  # pressure difference at valve [bar]

STREAM
I1  : fr,dm,temp,p_in    AS  FlowStream   # flow stream at valve's inlet
IM : ref_cone_pos       AS  CONNECTION            # manipulating signal
O1 : fr,dm,temp,p_out   AS  FlowStream  # flow stream at valve's outlet

SET
          c_valve  :=          846.153846/1000;  # valve constant

EQUATION
          act_cone_pos  = ref_cone_pos- 5*$act_cone_pos;
          delta_p = p_in - p_out;
          fr = c_valve*act_cone_pos *sgn(delta_p)*sqrt(abs(delta_p));
END
```

Figure 4-11: Working area of COMOS PT™: Integration of FSCS models

Model Documentation

The ”Model documentation“, as part of the FSCS model, is integrated in COMOS PT™ as a structured ASCII-file in the ”Model information“ frame (2), see Fig. 4-10. The model documentation is structured into five sections: Background, purpose, characteristics, and validity ranges of the FSCS model and variables used, as defined in Section 3.3.3.4. For the prototype realisation, some data in the model documentation are presented in the ”Variable information“ frame (1) in order to highlight specific data, such as the validity ranges, for the planning engineer, see Fig. 4-10.

By embedding all FSCS models and their documentation into the component database of COMOS PT™, the path is prepared for an automatic model aggregation and simulation strategy.

4.5 Automatic Simulation Procedure from COMOS PT™ including MAM and GUI Tasks

In this section, significant tasks for the automatic model aggregation and simulation procedures derived from the results of the preceding plant design, are described. These tasks, separated into GUI and MAM tasks and including the essential programs developed, are indicated by numbers in ascending order in Fig. 4-12. In order to point out the internal dependencies of the programs developed and their resulting data sets and lists, Fig. 4-13 summarises the main programs, which have been developed for the realisation of the GUI and MAM tasks.

1. Simulation area selection on P&I diagram (GUI task)
   This GUI task allows the planning engineer very easily to select the section of plant to be simulated. The result is the "Process and Control data set", which includes the selected process components, and control functions, derived from the "Diagram_analysis" program.

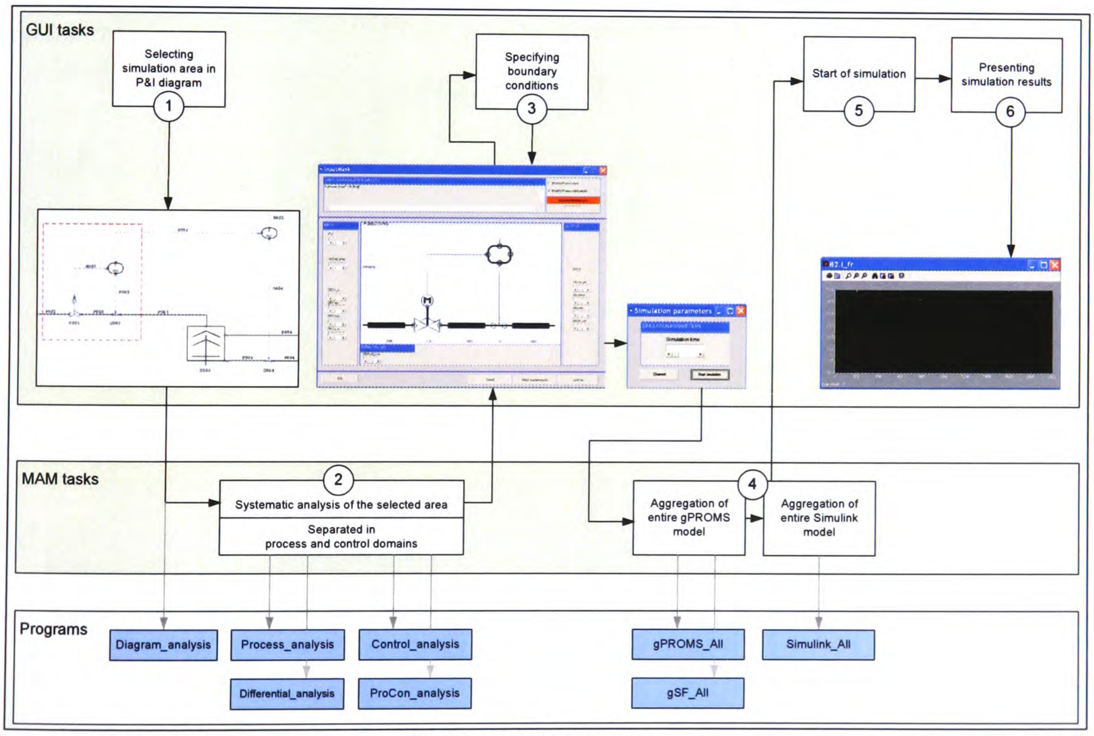


Figure 4-12: General overview of MAM and GUI tasks

2. Systematic analysis of the selected area (MAM task)
   The "Process and Control data sets" are analysed with regard to internal interconnections and open boundaries for the process and control domains (by the programs: "Process_analysis" and "Control_analysis"), with regard to interconnection between the process

and control domain ("ProCon_analysis“) and with regard to the analysis of differential terms ("Differential_analysis“). The results are corresponding lists, as shown in Fig. 4-13.

3. Specifying boundary conditions (GUI task)

   The GUI specification window has been developed in order to support the planning engineer to specify missing boundary conditions, initial values and simulation parameters.

4. Aggregating models automatically (MAM task)

   After the successful specification of boundary condition, initial values and simulation parameters, the "Entire gPROMS“ and "Entire Simulink“ models are aggregated automatically (by the programs "gPROMS_all“ and "Simulink_all“). This also includes the definition of the co-simulation interface between gPROMS™ and Simulink™, the gSF-file ("gSF_all“).

5. Start of simulation (GUI task)

   After the successful compilation of all required simulation scripts, the planning engineer is supported to initialise the simulation procedure.

6. Presenting simulation results (GUI task)

   Finally, the simulation results are presented within the simulation environment.

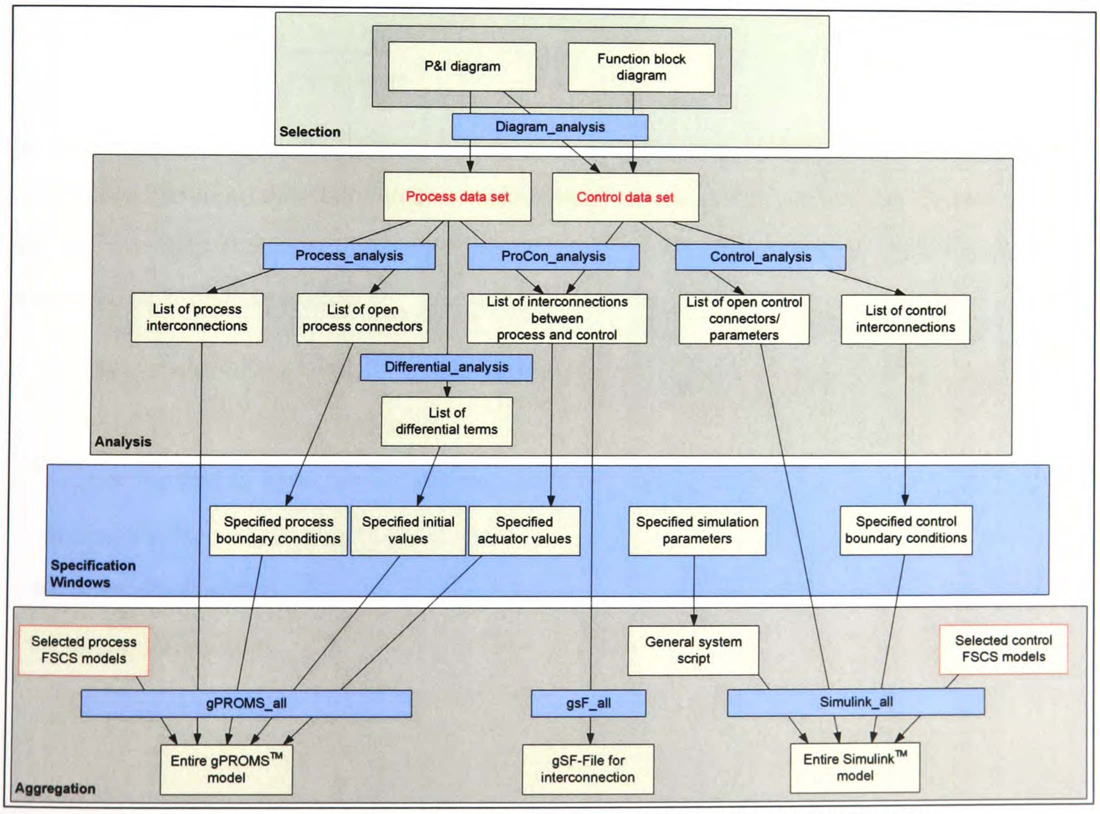


Figure 4-13: Logical sequence of developed programs (marked in blue colour)

In order to support the planning engineer with additional information during the GUI and MAM tasks and detached from the tasks, aimed at an automatic simulation strategy, a help system has been established with access from COMOS PT™ and the communication window used later.

The help system is described in the following section before presenting the realisation of the six tasks including their executable programs required for an automatic simulation strategy.

4.5.1 Help System (GUI Task)

In order to support the planning engineer during all GUI tasks, a help system was realised. For free access to the help system, the toolbar of COMOS PT™ was extended by a new icon ”M^{CAT}“, which stands for the Model^{CAT} approach, see Fig. 4-14.

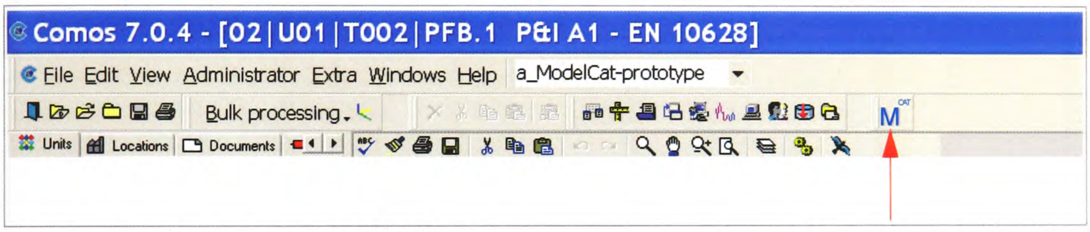
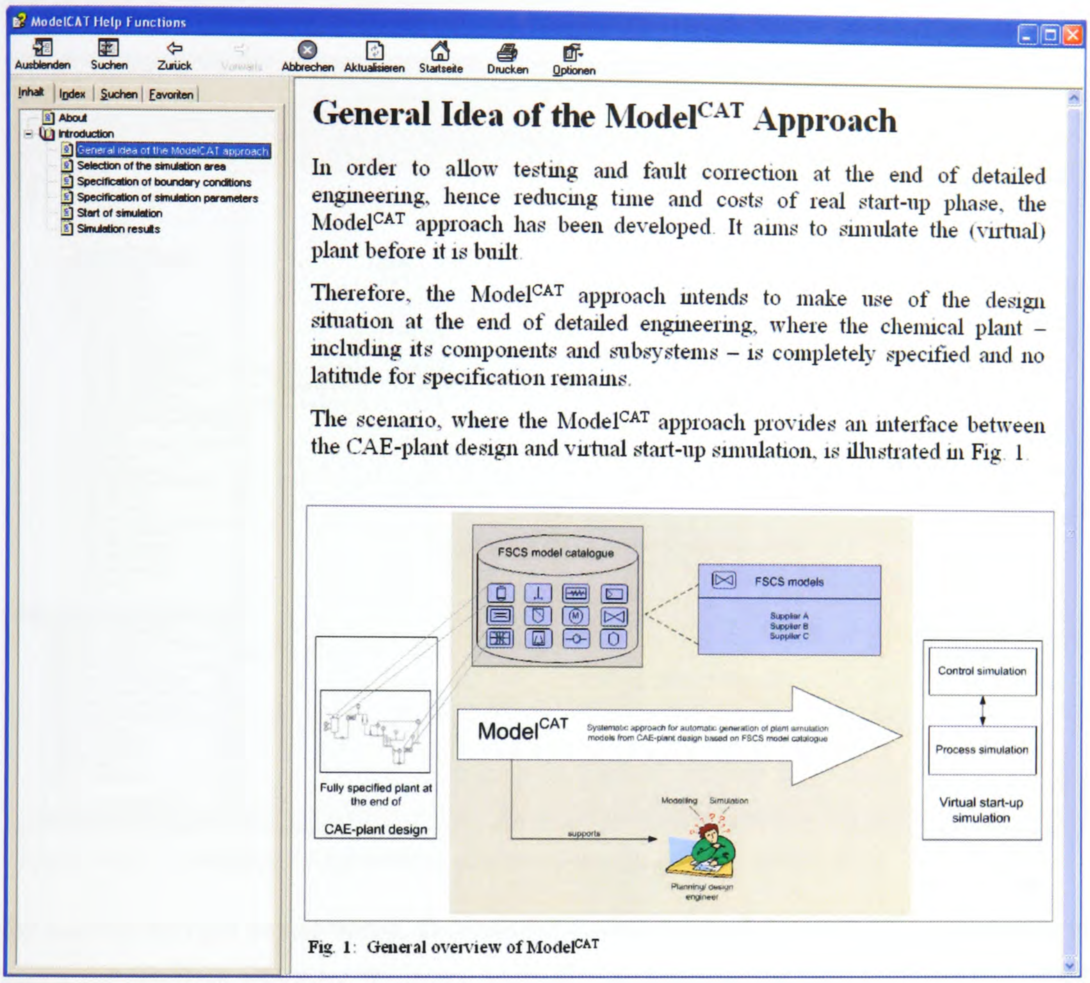


Figure 4-14: GUI support: Model^{CAT} icon

By clicking on this icon, the Model^{CAT} help system opens with an introduction to the next step of selecting the simulation area. Fig. 4-15 shows an example of the current help system. On the left, the user may choose help instructions for several tasks. The following tasks are integrated within the help system, to date:

1. General idea of the Model^{CAT} approach
2. Selection of the simulation area
3. Specification of boundary conditions and initial values
4. Specification of simulation parameters
5. Start of simulation
6. Simulation results

The help system can also be launched from all user communication windows, see Section 4.5.4.1.



General Idea of the Model^{CAT} Approach

In order to allow testing and fault correction at the end of detailed engineering, hence reducing time and costs of real start-up phase, the Model^{CAT} approach has been developed. It aims to simulate the (virtual) plant before it is built.

Therefore, the Model^{CAT} approach intends to make use of the design situation at the end of detailed engineering, where the chemical plant – including its components and subsystems – is completely specified and no latitude for specification remains.

The scenario, where the Model^{CAT} approach provides an interface between the CAE-plant design and virtual start-up simulation, is illustrated in Fig. 1.

Fig. 1: General overview of Model^{CAT}

Figure 4-15: GUI support: Help system

4.5.2 Selection of Simulation Area in P&I Diagram (GUI Task)

When the plant design phase is completed and the plant's components are completely specified, the planning engineer can select the area of the plant within the P&I diagram which she/he requires to be simulated.

In order to extract the information of the section of plant to be simulated from the P&I diagram graphically, the functionality of COMOS PT™ has to be extended. This was realised by a program written in Visual Basic©, called "Diagram_analysis".

Therefore, the plant area to be simulated must be selected, which is done using the standard screen area definition mouse functionality. By pressing the left mouse button, a square window can be drawn, which includes the section of plant to be simulated, as shown in Fig. 4-16.

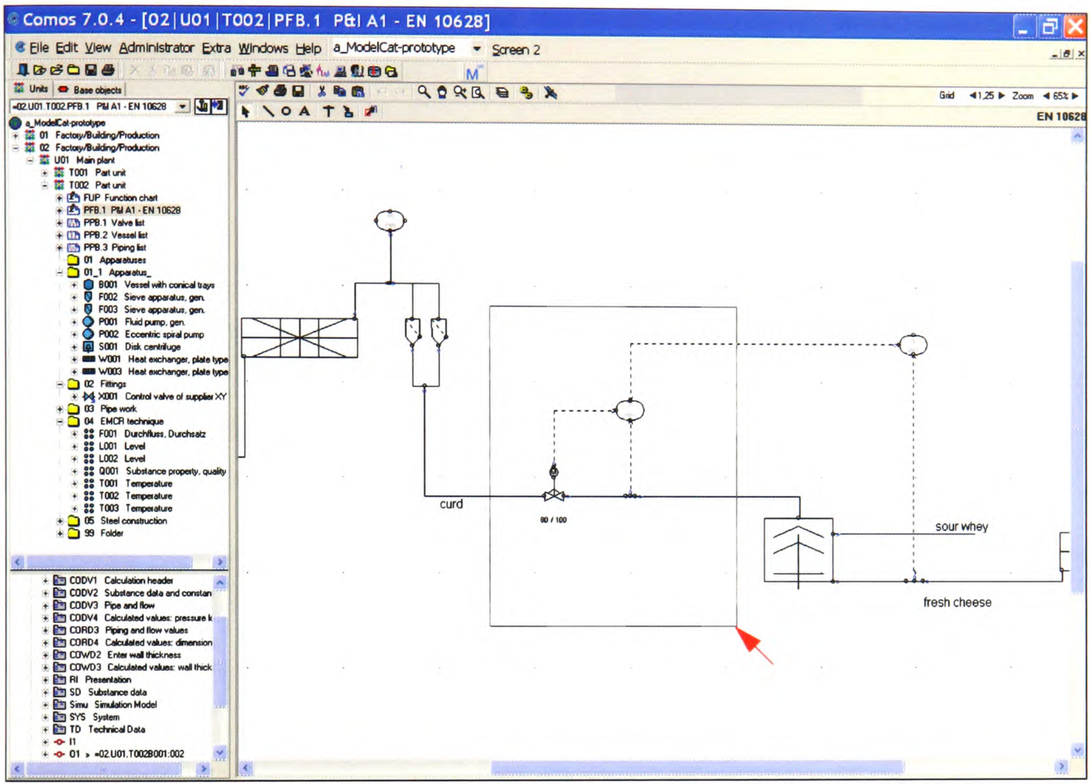


Figure 4-16: Selection of the section of plant to be simulated in COMOS PT™

By clicking the right mouse button, the extended context menu of COMOS PT™ is opened with two extra functions added by "Diagram_analysis". As illustrated in Fig. 4-17, the first added function "Components of selected area" lists every component of the selected area. The planning engineer can navigate from this list down to the "Simulation model" layer of each component.

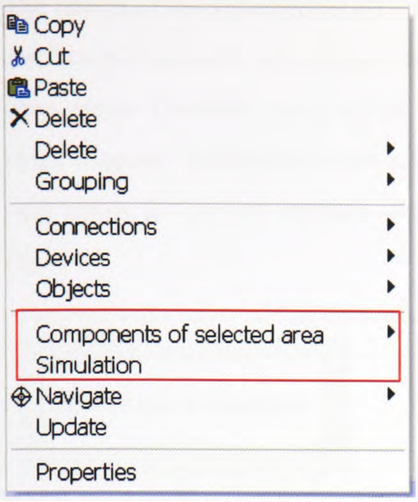


Figure 4-17: GUI support: Extended context menu of COMOS PT™

The second added function is the initialisation function for the simulation task, which is realised by a "Simulation" function.

By clicking the "Simulation" function the systematic analysis of the selected plant section is initialised, as illustrated in Fig. 4-18.

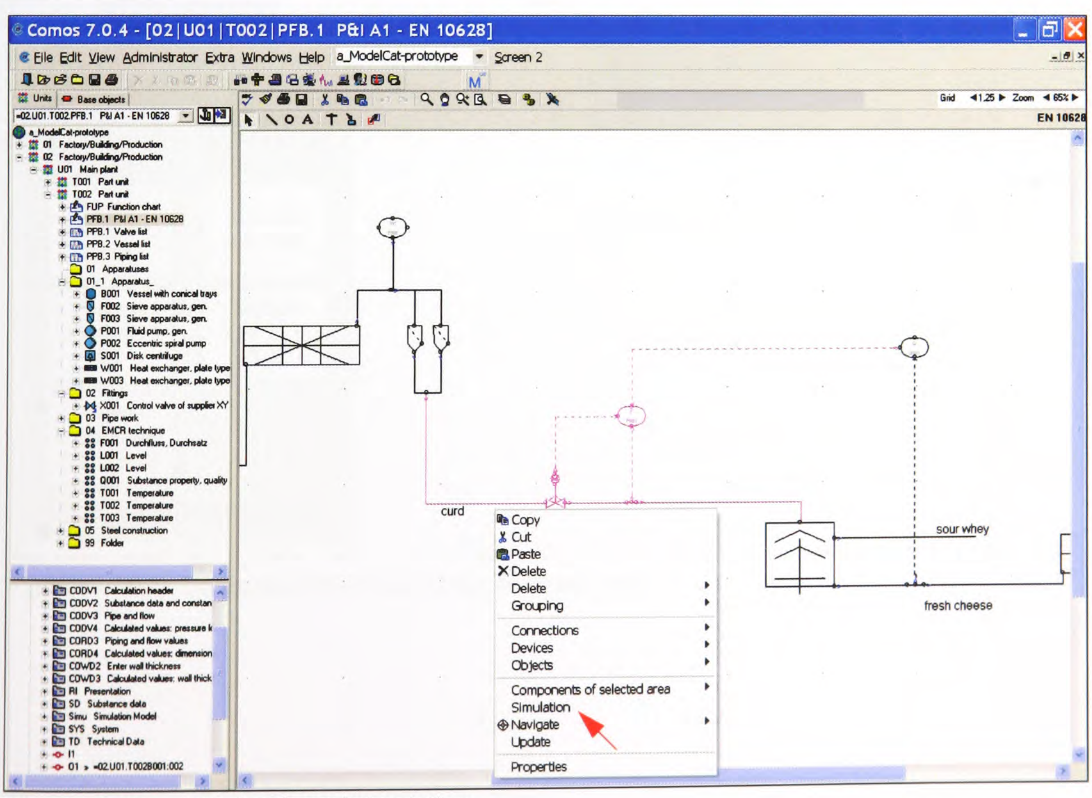


Figure 4-18: Initialisation of simulation

The results of the selection of the simulation area are two component data sets. The first "Process data set" includes all process components and the second, the "Control data set", all control components (function eyes) of the P&I diagram and the link to the corresponding function block diagram. Furthermore, the information of process and control components related to the path within the project database hierarchy level is collected in these data sets, as illustrated in Fig. 4-19.

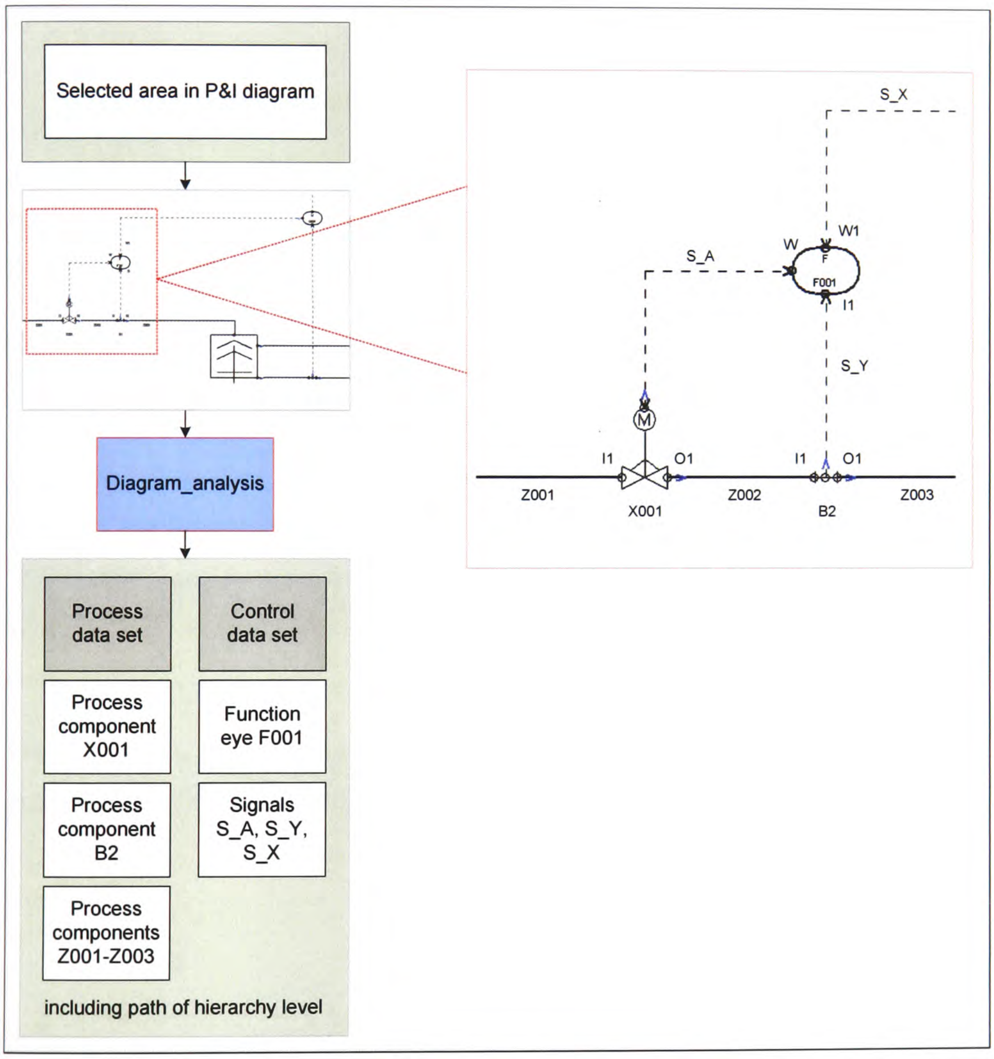


Figure 4-19: Generated data sets of the selected area

4.5.3 Systematic Analysis of the Selected Area (MAM Task)

The generated "Process and Control data sets" of the selected area with the information of the hierarchy level in the project database form the basis for the analysis in order to obtain following information about the ongoing specification of the boundary conditions and about the model interconnection relationships.

- Analysis of Interconnection Relationships for the Process Domain
- Analysis of Interconnection Relationships between Process and Control Domains
- Analysis of Interconnection Relationships for the Control Domain
- Analysis of Differential Terms

Before going into the details of these topics, the general procedure in COMOS PT™ to retrieve the connection information from the project database is described.

4.5.3.1 General Procedure for the Retrieval of Connection Information from the Project Database

Using the information of each component and its path to the hierarchical level in the project database, the interconnections within the selected area are retrieved from the COMOS PT™ "connectors" specifications for each component. Therefore, the object-oriented data structure of COMOS PT™ is used to navigate to the connectors' specifications of the respective components.

Fig. 4-20 shows an example of how such interconnection data are retrieved for a specific component, in this case the control valve X001, in the selected P&I diagram area.

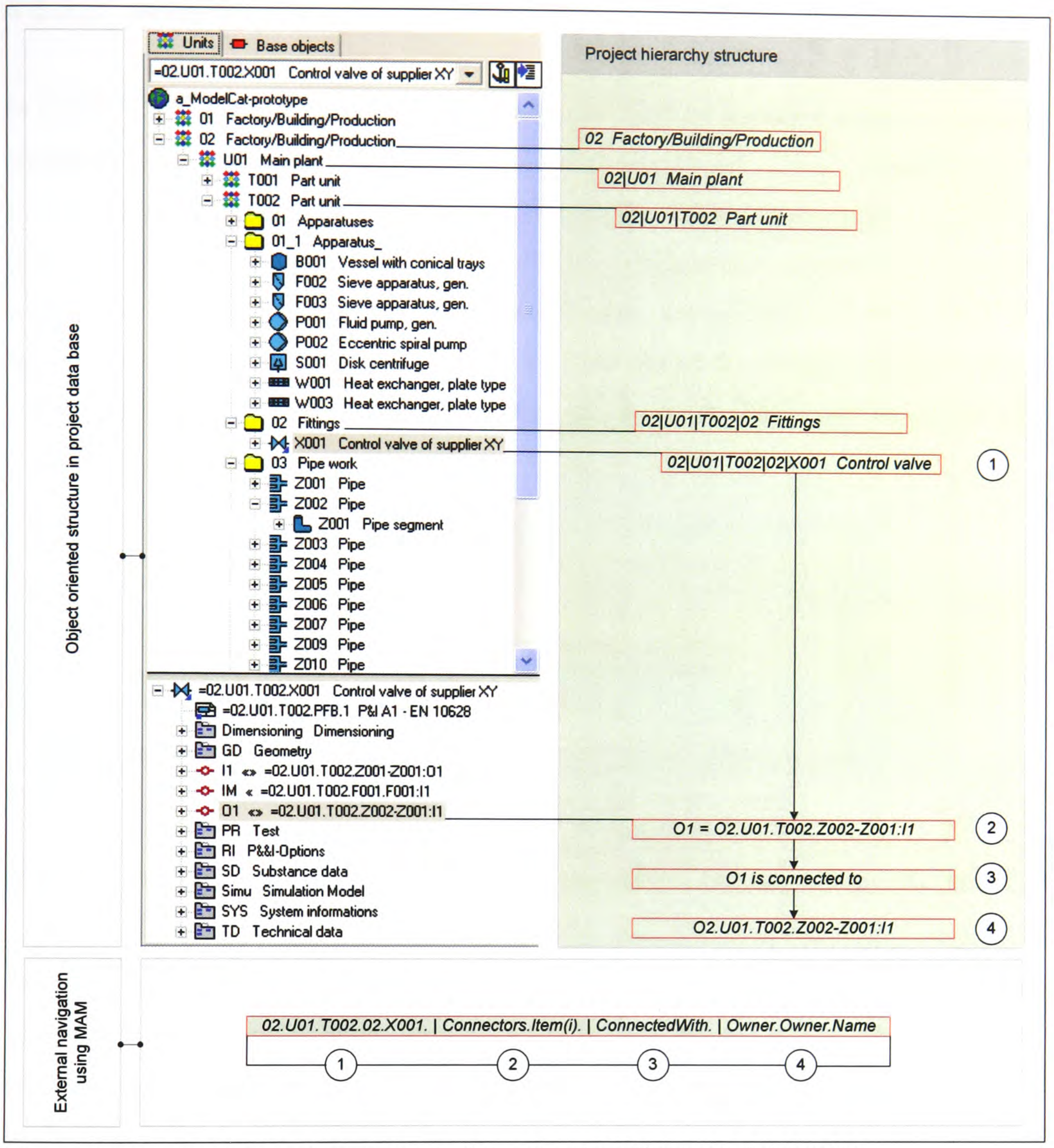
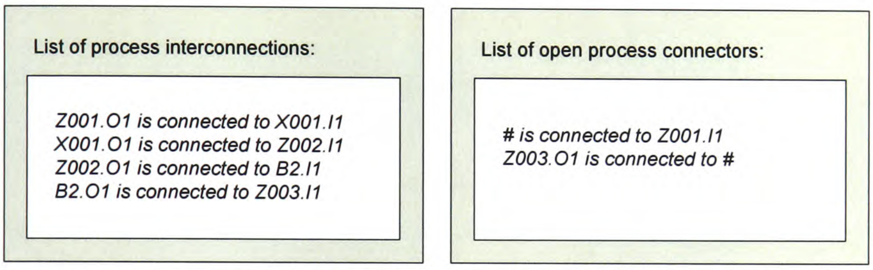


Figure 4-20: Retrieval of connection information from the project database

At the connector hierarchy level (2) of the control valve (1), all connections of the component along with the "connected to"-information used, to subsequently connected components, are available. Fig. 4-20 illustrates the required programming path (for external navigation using MAM), which is required to navigate from the output connector "O1" of the control valve (X001) to the input connector "I1" of the connected component (Z002-Z001). In COMOS PT™ the "connected to" functionality is called "ConnectedWith". By this analysis, all required interconnection relationships are made available.

4.5.3.2 Analysis of Interconnection Relationships for the Process Domain

In COMOS PT™, output connectors are labelled as "O1" of a process component X and are connected to the input connector "I1" of the process component Y. These connections represent the flow streams between two process components. The results are stored in a neutral format (e.g. Z001.O1 is connected to X001.I1) in the "List of process interconnections". Open connectors, which result from the selection on the P&I diagram, are indicated with the symbol "#" and stored in the "List of open process connectors". Both lists of the selected area of the flow rate control example are shown in Fig. 4-21, as a result from the execution of the "Process_analysis" program.



List of process interconnections:

Z001.O1 is connected to X001.I1
X001.O1 is connected to Z002.I1
Z002.O1 is connected to B2.I1
B2.O1 is connected to Z003.I1

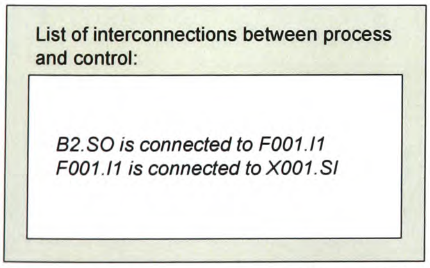
List of open process connectors:

 # is connected to Z001.I1
Z003.O1 is connected to #

Figure 4-21: Interconnection relationships for process components

4.5.3.3 Analysis of Interconnection Relationships between Process and Control Domains

The realisation of the analysis of the interconnection relationships between the process and control domain was embedded in the "ProCon_analysis" program and is demonstrated by the flow control example, but in this case with the components derived from COMOS PT™. The result is the "List of interconnections between process and control", as shown in Fig. 4-22.



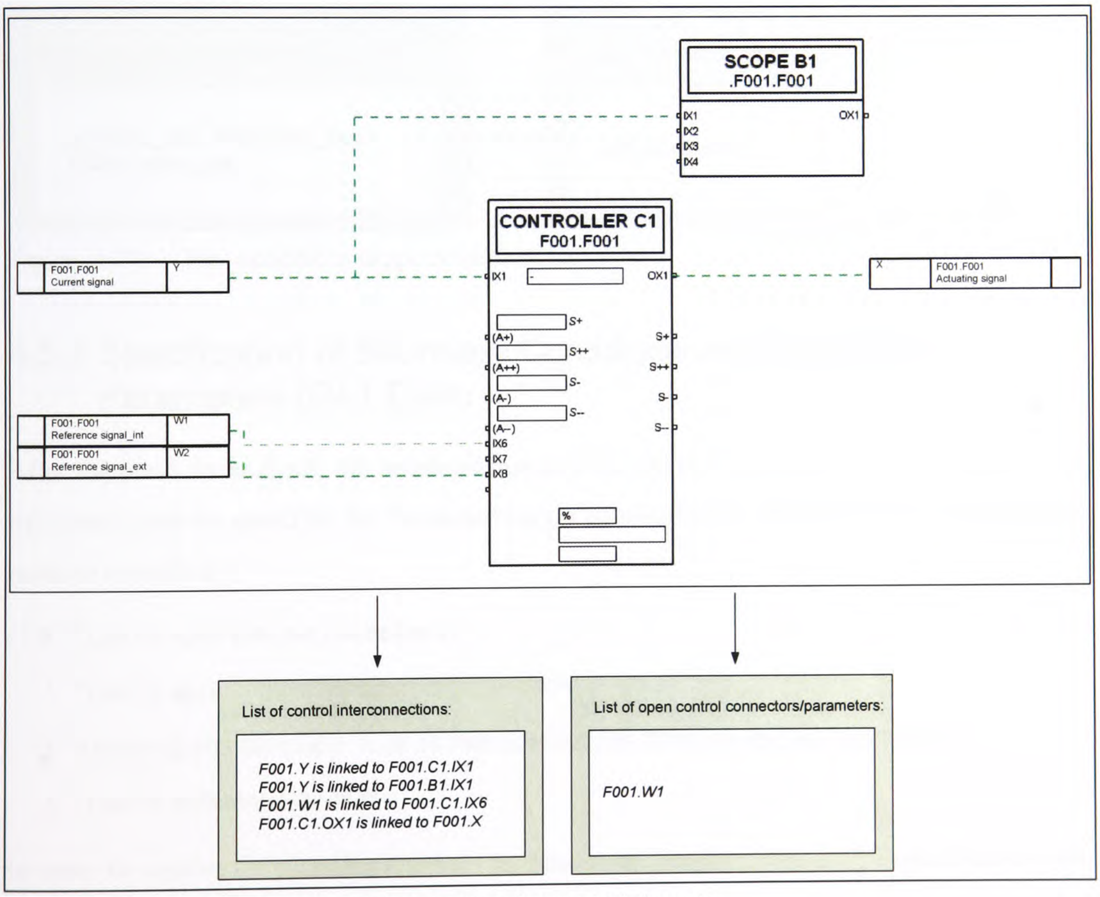
List of interconnections between process and control:

B2.SO is connected to F001.I1
F001.I1 is connected to X001.SI

Figure 4-22: Interconnection relationships between process and control domain

4.5.3.4 Analysis of Interconnection Relationships for the Control Domain

The corresponding function block diagram is analysed in order to retrieve the interconnection relationships within each function block diagram ("Control_analysis" program). The signal from the sensor to the function eye is transferred to the function block diagram as "Current signal Y" and builds the starting point of the analysis of the function block diagram. The "Current signal Y" is connected to the controller "C1" and the scope "B1". Every function block has to be checked for further connector links, similarly to the connectors and "connected to" analysis of the process domain. The results of this analysis are collected in the "List of control interconnections", as shown in Fig. 4-23, together with changeable variables and signals, which have not been specified within plant design, stored in the "List of open control connectors/parameters".



List of control interconnections:

F001.Y is linked to F001.C1.IX1
F001.Y is linked to F001.B1.IX1
F001.W1 is linked to F001.C1.IX6
F001.C1.OX1 is linked to F001.X

List of open control connectors/parameters:

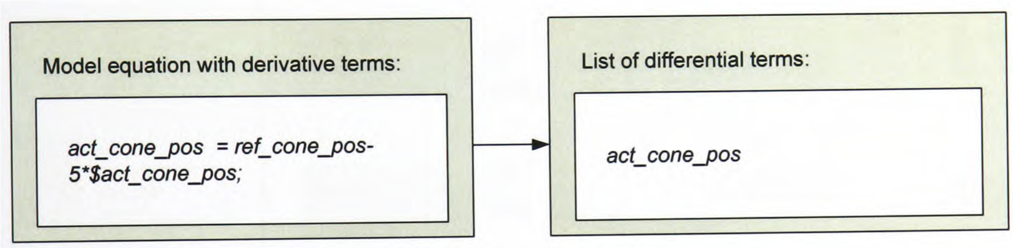
F001.W1

Figure 4-23: Interconnection relationships for control domain

The choice of the internal or external reference value depends on the results from the selected area. If the second function eye of the cascaded control (see Fig. 4-16) had also been selected for simulation, the external reference value would have to be taken into account. If not (as in this case), only the internal reference value is taken.

4.5.3.5 Analysis of Differential Terms

In gPROMS™, the differential term is indicated by a ”$“. Therefore, all selected gPROMS™ models have to be analysed and the variable names of the differential terms must be collected in a ”List of differential terms“ (derived by the ”Differential_analysis“ program). In the flow rate control example, one differential term is found in the model equations of the control valve (X001) and included in the ”List of differential terms“, see Fig. 4-24.



```
act_cone_pos = ref_cone_pos-
5*$act_cone_pos;
```

Figure 4-24: Tool specific analysis of differential terms

4.5.4 Specification of Boundary Conditions and Simulation Parameters (GUI Task)

After the previous analysis, the missing boundary conditions and initial values of the differential terms must be specified for the initialisation of simulation. Therefore, the following lists must be evaluated:

- ”List of open process connectors“
- ”List of open control connectors/parameters“
- Actuating signals of the ”List of interconnections between process and control“
- ”List of differential terms“

In order to support the planning engineer in these specification tasks, a GUI specification window was designed. Before going into the specification task in detail, the graphical representation of the GUI specification window is described in the following section.

4.5.4.1 Graphical Representation of the GUI Specification Window

In order to simplify the modelling and simulation task for the planning engineer, the following GUI conventions, with regard to graphical and typographical aspects, are defined and shown in Fig. 4-25.

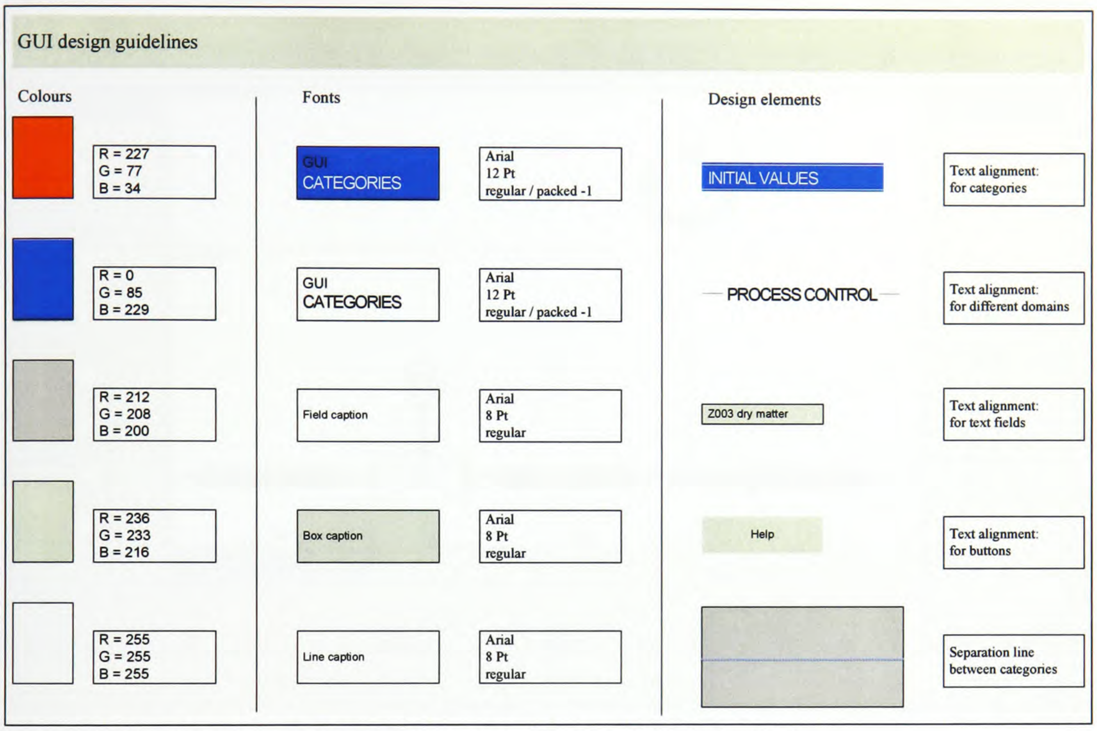
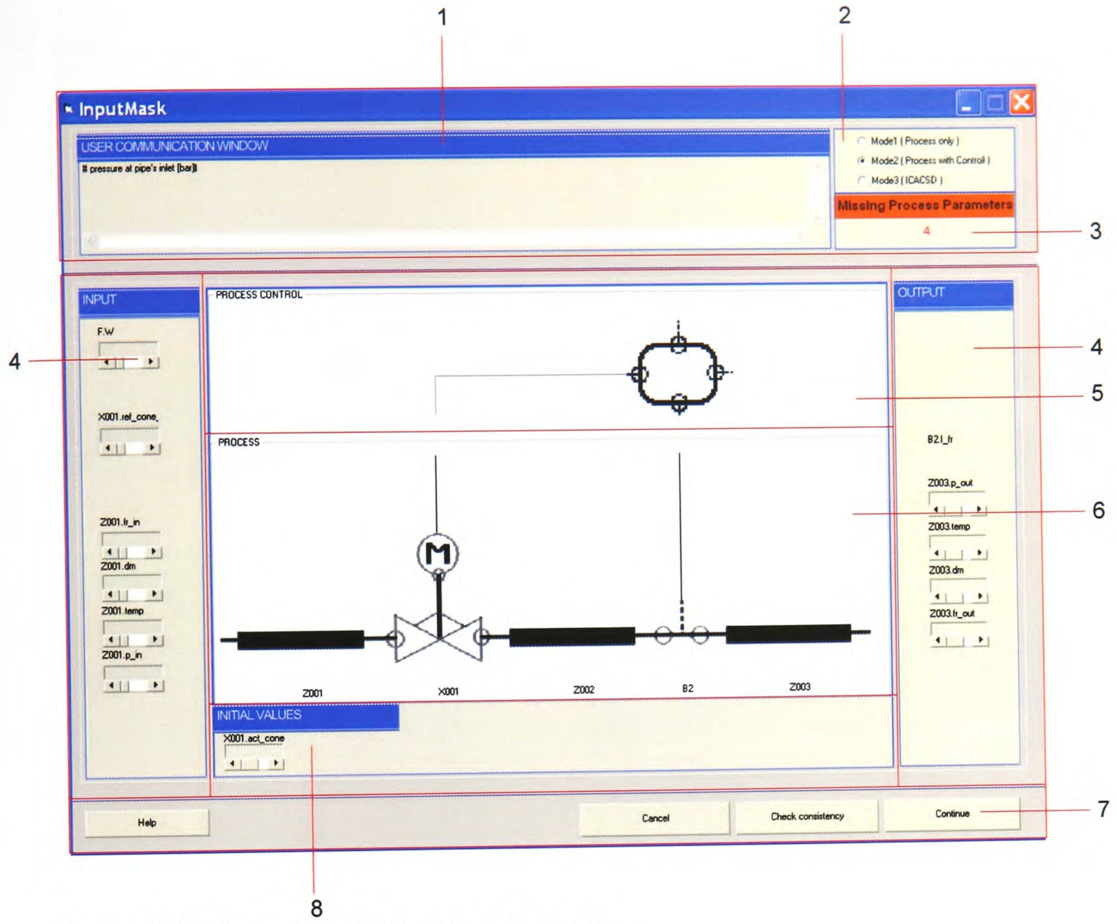


Figure 4-25: GUI conventions (graphical and typographical aspects)

The GUI specification window for the specification of boundary conditions and initial values (GUI specification window) was designed based on didactic or preferably heuristic design considerations described in Section 3.6.2, which were presented by Hoyer *et al.* (2006b). Generally, the GUI specification window may be separated into the areas illustrated in Fig. 4-26 and described in more detail in Table 4-7.



InputMask

USER COMMUNICATION WINDOW

pressure at pipe's inlet [bar]

○ Mode1 (Process only)
● Mode2 (Process with Controll)
○ Mode3 (ICACSD)

Missing Process Parameters
4

Figure 4-26: Areas of the GUI specification window

Table 4-7: Areas and short descriptions of the GUI specification window

Area	Name of area/function	Short description
1	User communication window	The "User communication window" displays explanatory and help information - instructions, parameter descriptions, value ranges allowed etc.
2	Simulation mode	The user can choose between the following simulation modes: – Mode 1 (Process only) – Mode 2 (Process with control) – Mode 3 (Process for use in ICACSD)
3	Missing process parameters	Displays the number of missing boundary conditions, indicated by colours: – Green: All required parameters are set (fully specified = no degrees of freedom). – Red: A definitive number of required boundary conditions are not specified.
4	Input and output variables	These frames define the system defining variables on both sides, indicated as "INPUT" and "OUTPUT" states in order to support the planning engineer during specification of boundary conditions. Control variables are also integrated within these frames.
5	Process control area	Illustrates the control schemes by function eyes.
6	Process area	Illustrates the process components by graphical symbols, similar to those from P&I diagrams.
7	Buttons	By clicking buttons: – The user can get support from the help system ("Help-button") implementation. – The simulation can be cancelled ("Cancel-button"). – The consistency of parameterisation can be checked ("Check consistency-button"). – The simulation can be started/continued ("Continue-button").
8	Initial values	Initial values are the differential terms of differential equations and have to be specified at *t* = *0 s* in order to start a simulation.

The GUI design was based on the nominal conditions of functionality, simplicity and consistency, reviewed in Section 3.6.2:

- Functionality
  The main functionalities are listed in Table 4-7 and incorporated during the specification procedures, described in Section 4.5.4.2.
- Simplicity
  In order to ease the demands from the input fields for every state and control variable, scroll bars were used for easy graphical changes of the input/output variables, thus avoiding mistakes during editing if every state value were to be edited manually. The value ranges for the scroll bars were selected from the "Variable information" frame of the model description and represent the limits of the input field, see Section 4.4 (Model Documentation). Scroll bars can be activated by clicking on the bar's surface. The typographical conventions (fonts and colours) are similar to COMOS PT™ in order to support recognition by the user and allow concentration on the specification task. This also allows consistency, through the recognition of similar styles.
- Consistency
  Separation of process and control elements into two domains was realised by using a frame for the process and a frame for the control domain (see number 5 respectively 6 of Fig. 4-26). The graphical scheme of the P&I diagram was assumed to ease problems of recognition in the GUI for the planning engineer, through a consistent graphical context, see Fig. 4-27.
  Fig. 4-27 illustrates the flow control example in the P&I diagram which is then used as a basis for the schematic set-up in the GUI. In order to preserve the well known, recognisable and consistent scheme, each component in the P&I diagram can be found in the GUI scheme.

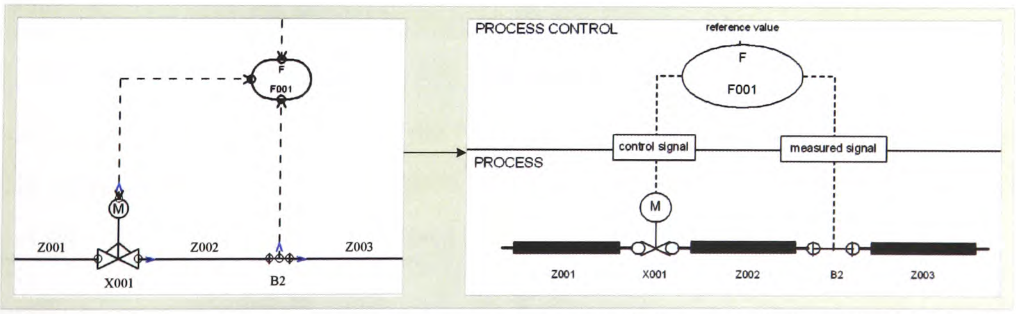
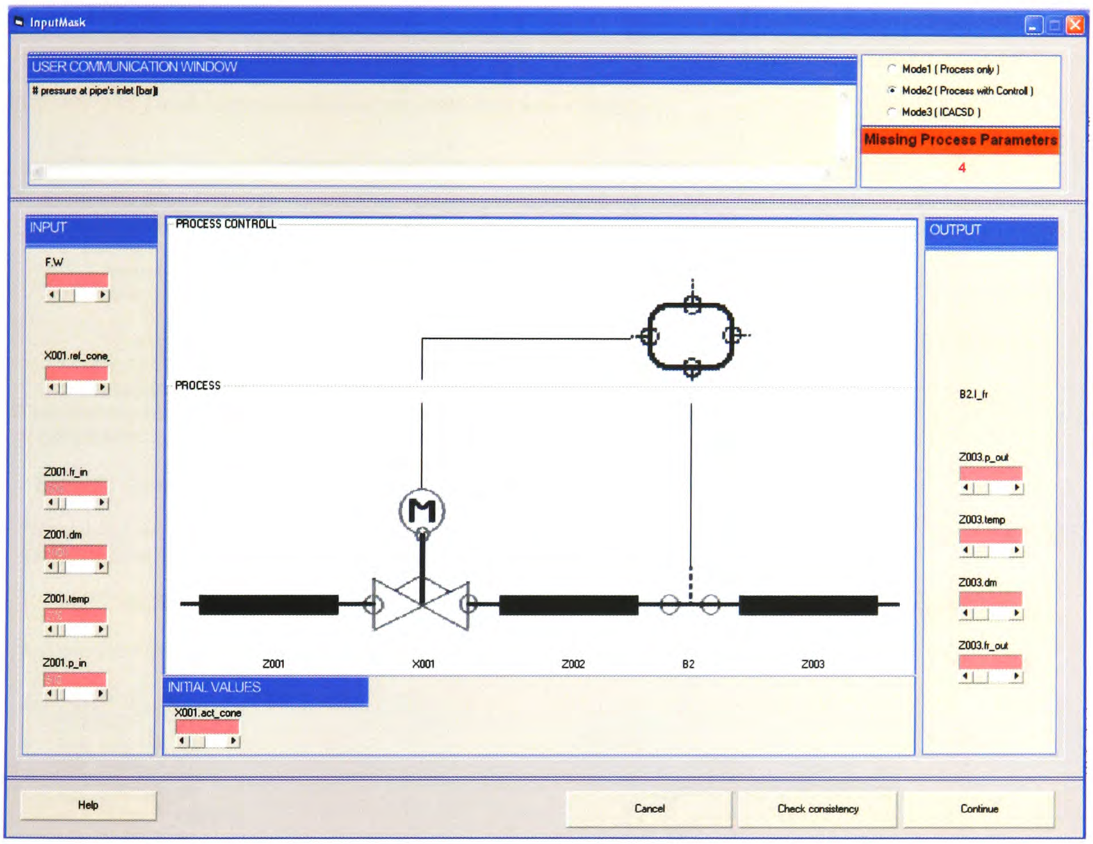


Figure 4-27: Converting the graphical set-up of P&I diagram into the set-up of the GUI

4.5.4.2 Specification Procedure

The specification procedure for boundary conditions was realised as defined in Section 3.6.4. The number of the "Missing Process Parameters" is counted down by each process boundary condition specified. In the user communication window, the planning engineer is provided with the instructions to "Specify all boundary conditions red labelled in order to continue with simulation". Fig. 4-28 shows a GUI window at the very beginning of the specification procedure.



USER COMMUNICATION WINDOW
pressure at pipe's inlet [bar]

○ Mode1 (Process only)
● Mode2 (Process with Control)
○ Mode3 (ICACSD)

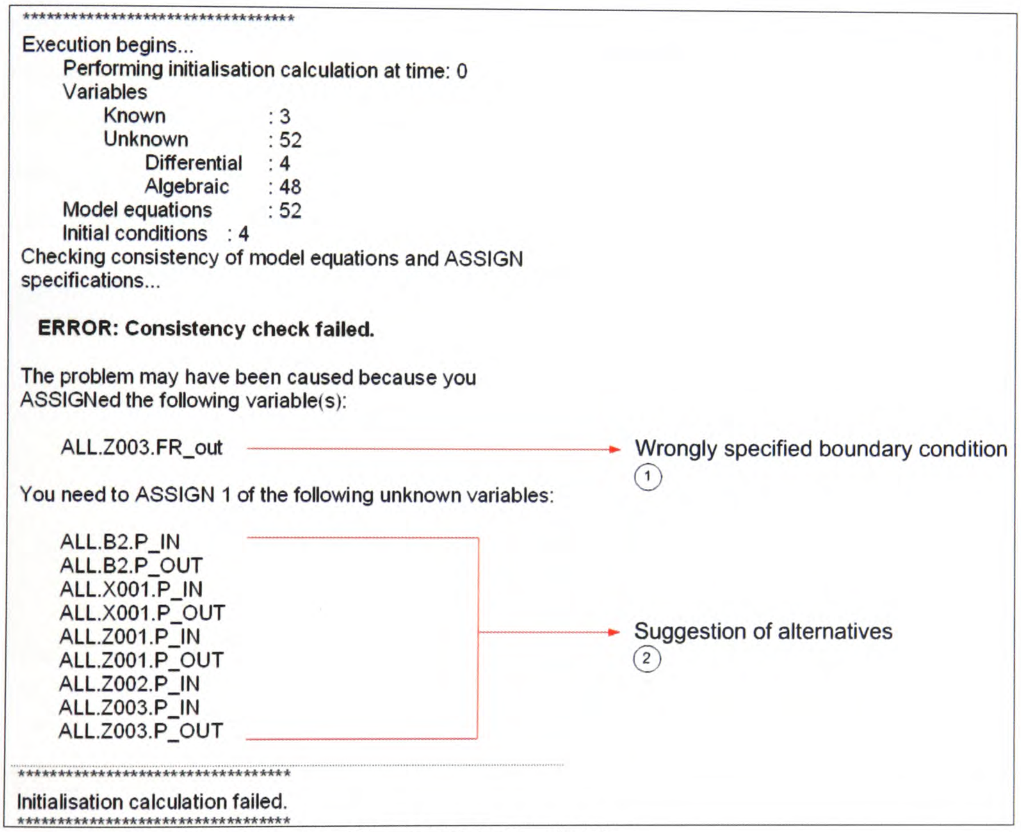
Missing Process Parameters
4

Figure 4-28: GUI specification window: Start of specification

Additionally, boundary conditions from the "List of open control connectors/parameters", the actuating signal of the "List of interconnections between process and control" and the initial values of the "List of differential terms" must be specified.

The number of process boundary conditions, which must be specified, depends on the analysis defined in Section 3.6.4.1 (Task 1). The specification of process boundary conditions by the planning engineer is initialised by firstly specifying process input/output variables and must be continued until the number of "Missing Process Parameters" is "*0*" (Task 2) and all red input fields on the control input and initial values are specified. In this case, the "Check consistency"-button is enabled to establish the consistency check when being clicked.

The consistency check is realised within gPROMS™ using the entire gPROMS™ model (see therefore Section 4.5.5.1). In gPROMS™ also the initial values and the actuator signals are taken into account for the consistency check. The result from the consistency check is an ASCII-file which includes information about the results of the test. Fig. 4-29 shows a possible output example of a failed consistency check. The incorrectly specified boundary condition is indicated (1) and suggestions for alternatives are made (2).



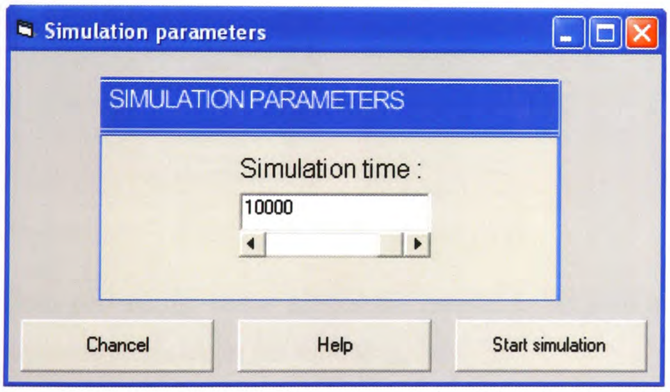
```
************************************
Execution begins...
    Performing initialisation calculation at time: 0
    Variables
        Known           : 3
        Unknown         : 52
            Differential  : 4
            Algebraic     : 48
    Model equations     : 52
    Initial conditions  : 4
Checking consistency of model equations and ASSIGN
specifications...

  ERROR: Consistency check failed.

The problem may have been caused because you
ASSIGNed the following variable(s):

    ALL.Z003.FR_out

You need to ASSIGN 1 of the following unknown variables:

    ALL.B2.P_IN
    ALL.B2.P_OUT
    ALL.X001.P_IN
    ALL.X001.P_OUT
    ALL.Z001.P_IN
    ALL.Z001.P_OUT
    ALL.Z002.P_IN
    ALL.Z003.P_IN
    ALL.Z003.P_OUT

************************************
Initialisation calculation failed.
************************************
```

Figure 4-29: GUI: Results of the consistency check

gPROMS™ provides the user not only with suggestions about the boundary conditions, but includes suggestions about internal variables. Therefore, these suggestions must be filtered to get only the suggestions for the boundary conditions. The output results show the information that the consistency check has failed and the names of the incorrectly specified boundary conditions are displayed in the communication window of the GUI specification window. Automatically, the values of the incorrectly chosen boundary conditions are deleted and disabled for further use. The suggested alternatives are also presented in the communication window and the colour of the input field is changed to green. After repeating the specification tasks (1-2) and executing the second now successful consistency check, the system is fully and consistently specified.

The following message then appears in the communication window: "All boundary conditions and initial values are defined, please continue with specifying the simulation parameters by clicking the "Continue-button"". The "Continue button" is enabled after a successful parameterisation check.

After clicking the "Continue" button, the planning engineer has to define the simulation parameters in the "Simulation parameters" window in order to initialise simulation. In the prototype realisation, only the simulation time is left to be defined, see Fig. 4-30. This is done by using a scroll bar with a range from *0* to *10,000 s*. By clicking the button "Start simulation", the automatic model aggregation for the entire Simulink™ and the entire gPROMS™ model is initialised, which is described in the following chapter.


Simulation parameters

SIMULATION PARAMETERS

Simulation time : 10000

Figure 4-30: GUI: Specification of simulation parameters

4.5.5 Automatic Model Aggregation (MAM Task)

In the following, the automatic model aggregation of the entire gPROMS™ model, the entire Simulink™ scheme and the additional scripts for embedding the entire gPROMS™ process model into Simulink™ are outlined. Both scripts are generated as ASCII-text by the "gPROMS_all" and "Simulink_all" programs, listed in Appendix C.

4.5.5.1 Entire gPROMS™ Model

The general structure of the entire gPROMS™ model script is shown in Fig. 4-31. It may be separated into the "Aggregated model" and the "Executing process" sections. The "Aggregated model" section (Part 1-6) comprises all aggregated process FSCS models including their model texts and defines the internal interconnections based on the "List of process interconnections". The "Executing process" section (Part 7-11) is required to initialise the entire gPROMS™ process model at $t = 0\ s$, including the specified boundary conditions and initial values (specified in Section 4.5.4.2). All sections are generated by MAM automatically.

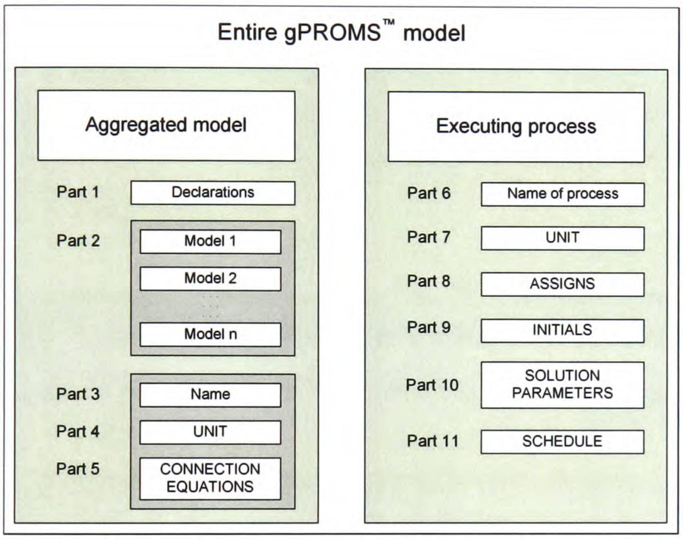


Figure 4-31: Entire gPROMS™ model script

Each part of the entire gPROMS™ model is outlined and addressed with special regards to model aggregation in the following.

- Part 1: Declarations
  The declaration sections of individual gPROMS™ models (see Section 4.3.1) are collected. An extra program was developed to sort out all conflicting declarations.
- Part 2: List of individual gPROMS™ models
  All individual gPROMS™ models are collected (without the declaration section).
- Part 3: Name of entire process simulation model
  The name of the entire gPROMS™ model is defined. In the prototype realisation the entire gPROMS™ model was named "*ALL*".

- Part 4: UNIT

  The component names of the plant design are referred to the names of the FSCS models, such as ”*X001 AS VAL*“ (Valve), which means, the valve X001 is associated with a ”VAL“ model.

- Part 5: CONNECTION EQUATIONS

  In the definition of the internal interconnection relationships, the ”List of process interconnections“ is used, which was determined during the systematic analysis of the selected area (see Section 4.5.3.2). However, the syntax has to be adapted in order to meet the requirements of gPROMS™ syntax. The neutral description ”*Z001.O1 is connected to X001.I1*“ is changed to ”*Z001.O1 IS X001.I1*“.

- Part 6 and 7: Name and Unit of process

  In order to execute the entire gPROMS™ model, gPROMS™ requires the definition of the process’ name. In this case, an instance of the entire gPROMS™ model has to be declared in the Unit section by the syntax ”*PROCESS AS ALL*“.

- Part 8: ASSIGNS

  The boundary conditions of the process are defined for $t = 0\ s$. The specified process boundary conditions from the ”List of open process connectors“ are assigned, e.g. ”*Z001.fr_in := 50*“ which means that the flow stream variable fr_in (flow rate_in) of pipe ”Z001“ receives the value 50.

- Part 9: INITIALS

  The differential terms of the differential equations, as specified in the GUI specification window, are prepared using the following syntax: *X001.s = 8.*

- Part 10 and 11: SOLUTION PARAMETERS and SCHEDULE

  Both sections are a prerequisite, necessary to establish simulation within gPROMS™, but from the functional point of view within Simulink™ they have no relevance for the simulation procedure.

4.5.5.2 Entire Simulink™ Scheme

The general structure of the entire Simulink™ script is shown in Fig. 4-32.

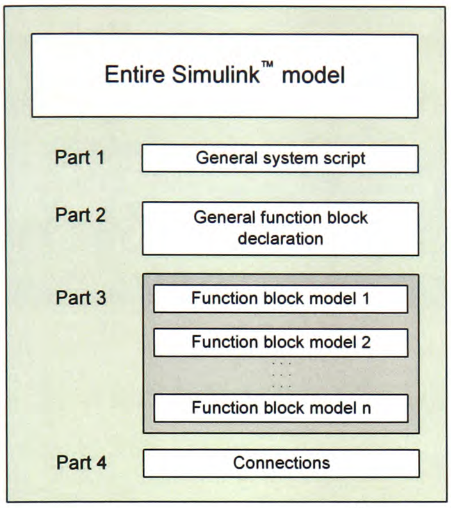


Figure 4-32: Entire Simulink™ script

It contains four parts, which are outlined in the following:

- Part 1: General system script
  The "General system script" of Simulink™ defines all required parameters, which are relevant to initialise a simulation within Simulink™, such as file name and design properties, but also the simulation time and the solver mode are defined. The simulation time is taken from the specification of the simulation parameters. In the prototype realisation, the standard solver mode is used.
- Part 2: List of general function block declarations
  For all different types of function blocks used, a general function block declaration is required in order to call the corresponding function block within Simulink™.
- Part 3: List of function block models
  To define the entire Simulink™ scheme, all required function block models are collected from the specifications ("Simulation model layer") of each control component and parameterised with the specified control boundary conditions from the "List of open control connectors/parameters".
- Part 4: List of connections
  The data of the "List of control interconnections" is used to connect all function blocks from port to port of the respective function blocks.

The integration of the entire gPROMS™ model into Simulink™ is described in the following section.

4.5.5.3 Script for Embedding the Entire gPROMS™ Model into Simulink™

For embedding the entire gPROMS™ model into Simulink™ the gO:Simulink™ interface is used, see Section 4.1.1.3. Therefore, two prerequisites must be fulfilled: The function block of the entire gPROMS™ model (in the following named "gPROMS™ function block") and the so-called gSF-file must be defined. Both tasks are briefly outlined in the following. More detailed information about the gO:Simulink™ interface can be found in PSE (2003a).

gPROMS™ Function Block

The script for the gPROMS™ function block represents a specific function block within Simulink™ and has to be integrated into the "List of all function block models", see Part 3 (Fig. 4-32) of the "Entire Simulink™ model". The structure of the gPROMS™ function block is provided by PSE (2003b).

gSF-File

The gSF-file defines the input and output ports of the gPROMS™ function block within Simulink™, based on the "List of interconnections between process and control". Therefore, at least the actuator and sensor signals are required to interconnect the gPROMS™ function block with the Simulink™ scheme. However, in the prototype realisation, the specified boundary conditions are also taken as input ports, while all output boundary conditions are declared as output ports of the entire gPROMS™ process model.

Fig. 4-33 shows a gPROMS™ function block within Simulink™, including a set of input and output ports.

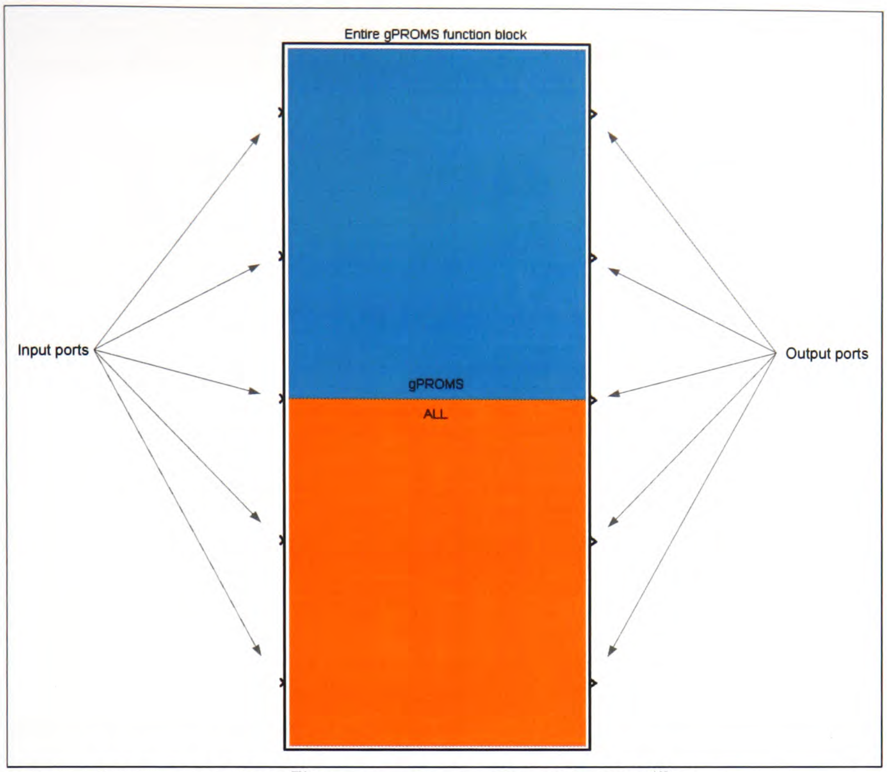


Figure 4-33: gPROMS™ function block within Simulink™

After the automatic generation of all required simulation scripts, the MAM tasks are completed by generating the entire Simulink™ scheme script as a "mdl"-file, including the gPROMS™ function block, as illustrated in Fig. 4-34. This scheme mirrors the P&I diagram of the CAE-plant design into the Simulink™ environment.

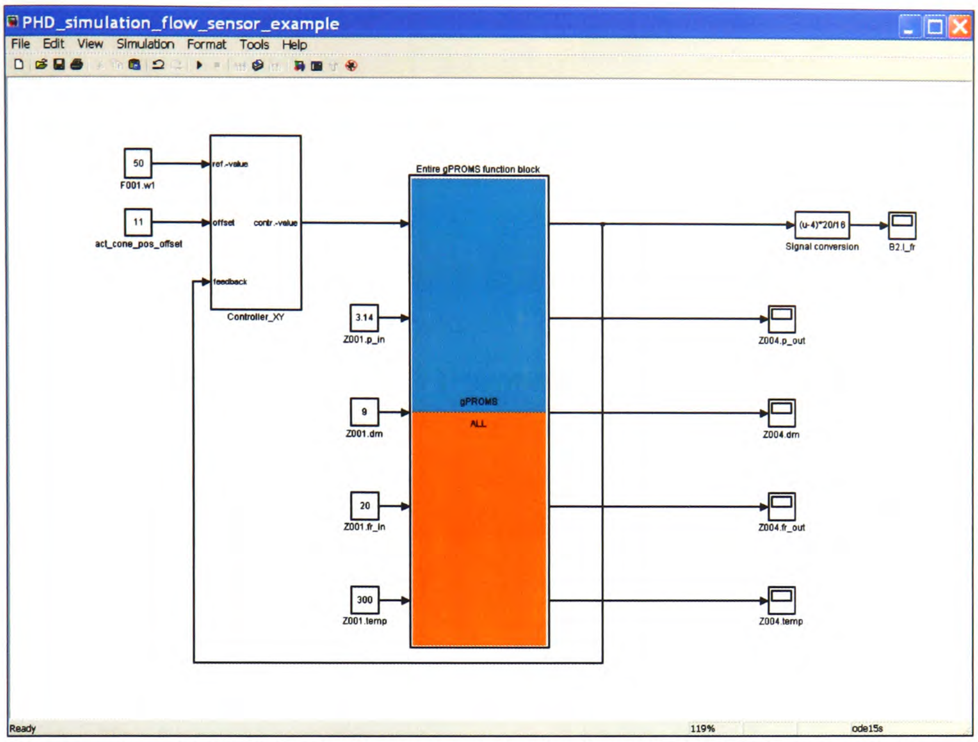


Figure 4-34: Entire Simulink™ scheme including the gPROMS™ function block

4.5.6 Start of Simulation and Presentation of Simulation Results (GUI Tasks)

In the actual prototype, the simulation is started manually by the user and the simulation results are presented within Simulink™. Fig. 4-35 shows an example of an output signal response, in this case, the flow rate signal of the sensor B2.

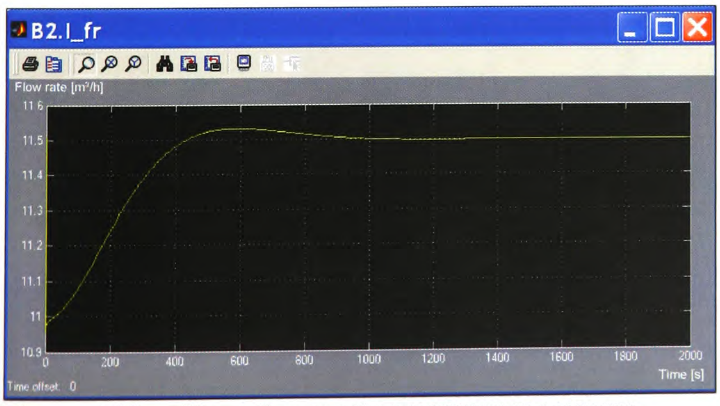


Figure 4-35: Simulation results presented in Simulink

In the next chapter the Model^{CAT} prototype is presented and validated in an industrial plant design scenario.

5 Validation of Model^{CAT} – Fresh Cheese Application Example

In this chapter, the Model^{CAT} approach is validated by establishing virtual start-up scenarios for a realistic application example, namely a fresh cheese production plant. Different process and control design tasks are chosen to demonstrate an automatic model generation based on CAE-plant design with the Model^{CAT} approach. Additionally, the Model^{CAT} approach may also be used in optimization procedures at the final stage of detail engineering. Typical scenarios in optimizing the process and also the control design are outlined for the application example, which were reported in Hoyer *et al.* (2005a). In the next section, the fresh cheese production plant used as application example, is outlined at first.

5.1 Reference Plant

A fresh cheese production line was chosen as reference plant to test the software and procedures presented in this work, see Fig. 5-1. The general functions of the fresh cheese production line are briefly described in the following.

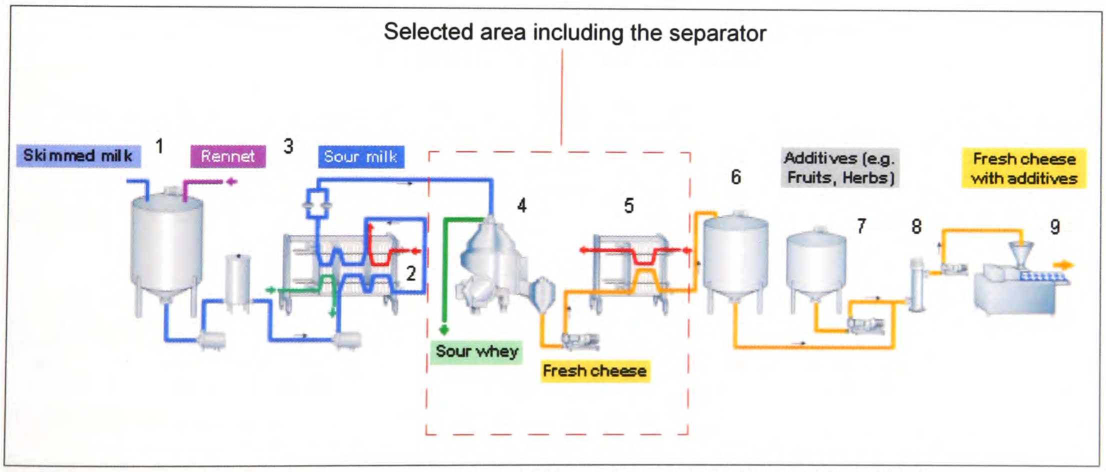


Figure 5-1: Fresh cheese production line (with permission of Tetra Pak (1995))

After pasteurisation, the skimmed milk is pumped into a ripening tank (1) in which culture and rennet are added. After the coagulation process (about 16-20 hours) the "thick milk" (called curd) is stirred and pumped through the first heat exchanger (2) where it is heated. The curd is fed to the separator through filters (3). The separator (4), which is the core component of the fresh cheese production plant, separates the curd flow into fresh cheese and sour whey by centrifugal separation, see also Fig. 5-2. The fresh cheese is pressed through nozzles at the periphery of the separator bowl and delivered to a vat. The sour whey leaves the separator through an outlet at the top. From the vat the fresh cheese is pumped through the second heat exchanger (5) (cooling) to a buffer tank (6) before it is mixed (8) with cream (7) or other ingredients and packed (9).

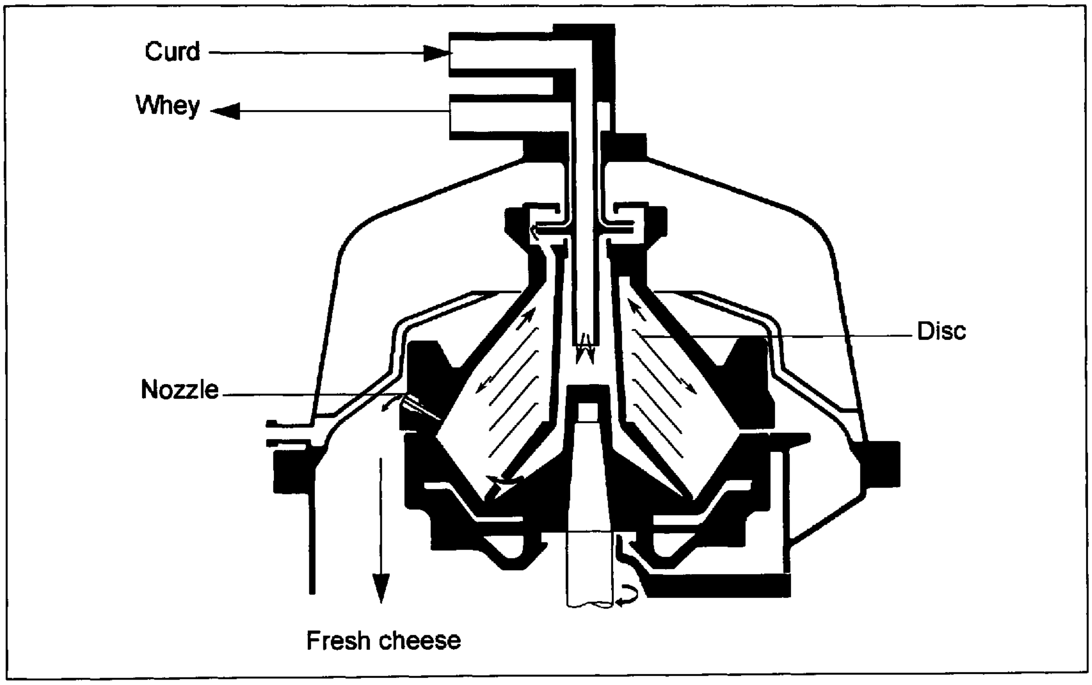


Figure 5-2: Cross section of fresh cheese separator

At the design stage of this fresh cheese production plant, typical virtual start-up scenarios are outlined with Model^{CAT}.

5.2 Virtual Start-Up Scenarios with Model^{CAT}

At the end of detailed engineering, the planning engineer has completed the plant design of the fresh cheese production plant. The fresh cheese production plant is fully specified (including all interconnection relations of process and control domain) and now the Model^{CAT} approach is used for the automatic generation of process and control simulation models from the P&I diagram. By simulation, the interaction between the process components and the process control is tested.

Fig. 5-3 shows the P&I diagram of the entire fresh cheese production plant, including the process components and the corresponding control functions.

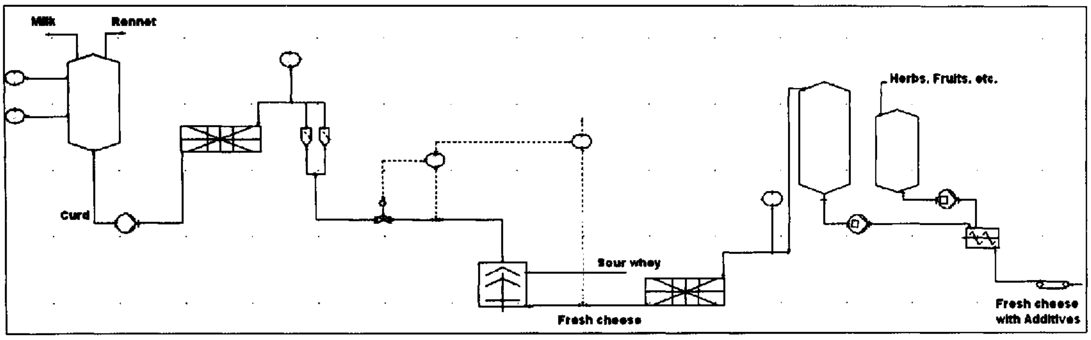


Figure 5-3: P&I diagram of the fresh cheese production plant

In this application example, the fresh cheese separator plays a key role with respect to product quality and process efficiency. The main quality parameters of fresh cheese are dry matter and protein content. In order to monitor the process performance, an inline NIR (Near-InfraRed) sensor is used to measure the quality parameters dry matter and protein content for monitoring and control purposes.

5.2.1 Automatic Simulation Procedure Based on CAE-Plant Design Results

The subsystem associated with the fresh cheese separator was chosen to validate the Model^{CAT} approach. By the selection of this area of the P&I diagram to be simulated, the Model^{CAT} approach is initialised. The selected subsystem is shown in the P&I diagram presented in Fig. 5-4.

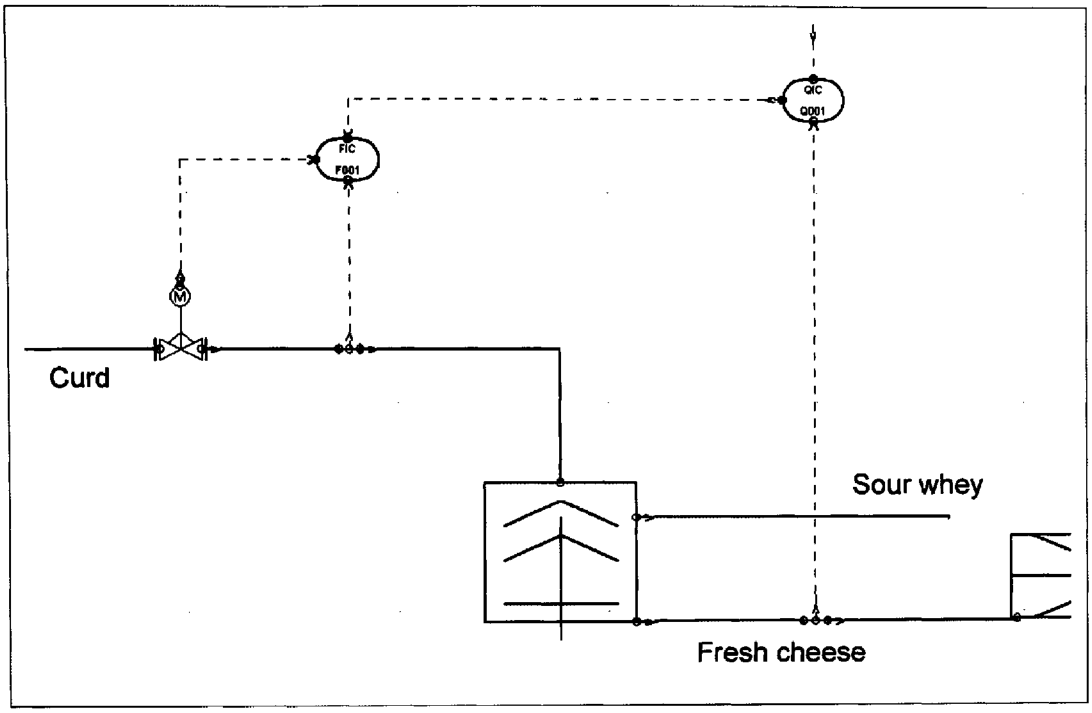
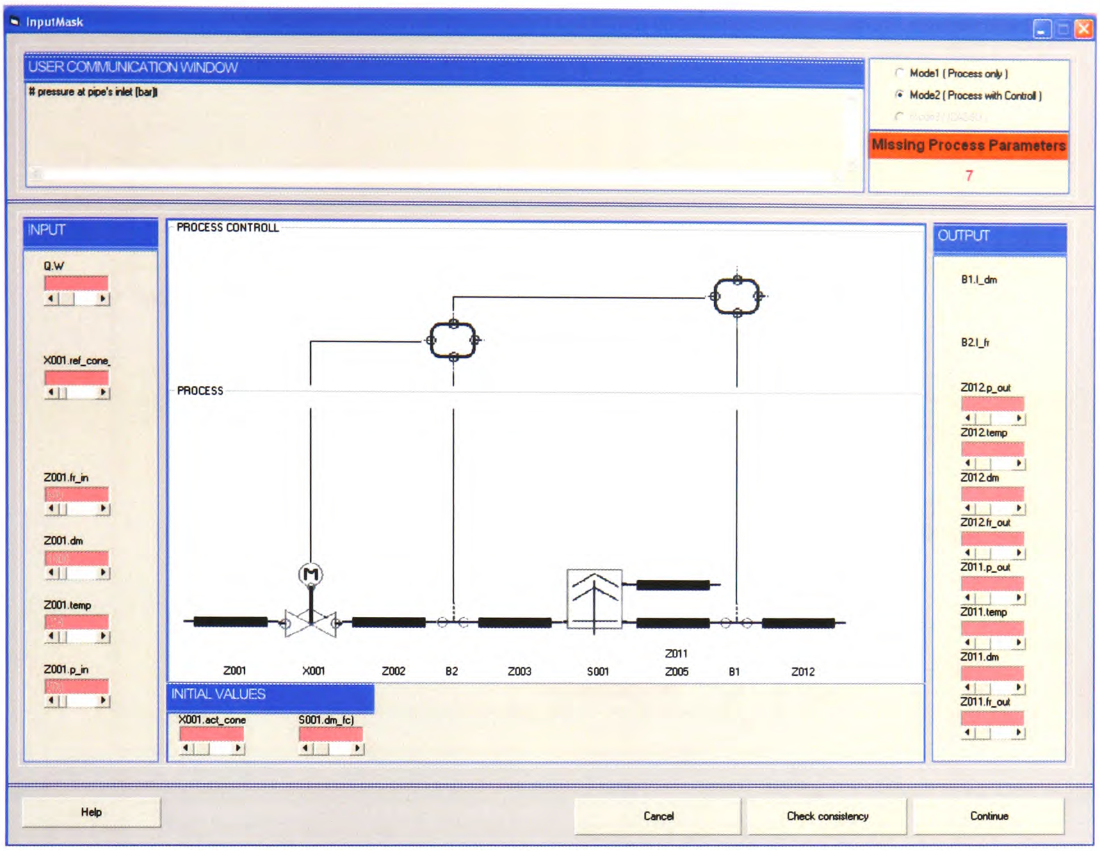


Figure 5-4: P&I diagram with the positioning of the NIR-sensor after the separator

The selected subsystem contains, on the process side, the fresh cheese separator (with input stream "curd", output streams "sour whey" and "fresh cheese"), the valve to control the curd flow, the flow sensor, the dry matter NIR sensor and the respective pipes. The control related components comprise the curd flow control (FIC) and the cascaded dry matter (quality) control (QIC). By simulation with Model^{CAT}, the interaction between the process components and the process control schemes are investigated using a simulated pressure drop at the curd inlet of fresh cheese separator.

With the selection of the plant area in the P&I diagram to be simulated, MAM automatically detects and analyses the components and their interconnections within this area. In this case, the boundary, initial and simulation parameters need to be specified within the relevant GUI window. In Fig. 5-5, the specification of boundary conditions and initial parameters is illustrated.



pressure at pipe's inlet [bar]

○ Mode1 (Process only)
● Mode2 (Process with Controll)

Missing Process Parameters
7

Figure 5-5: Application example: Specification of boundary conditions within the GUI

With the right combination of seven process variables to be specified, the specification of the actuator signal, the specification of one control variable and two initial values, and the definition of an appropriate simulation time, the specification process may be completed. MAM automatically aggregates the process model part into gPROMS™ format and the control model part into Simulink™ format and generates the Matlab™ simulation script. The resulting Simulink™ scheme including the corresponding gPROMS™ derived block is presented in Fig. 5-6. The gPROMS™ block contains the separator, the heat exchanger, the control valve and the flow and quality sensors and is part of the Simulink™ scheme, with the curd flow controller and the cascaded dry matter controller.

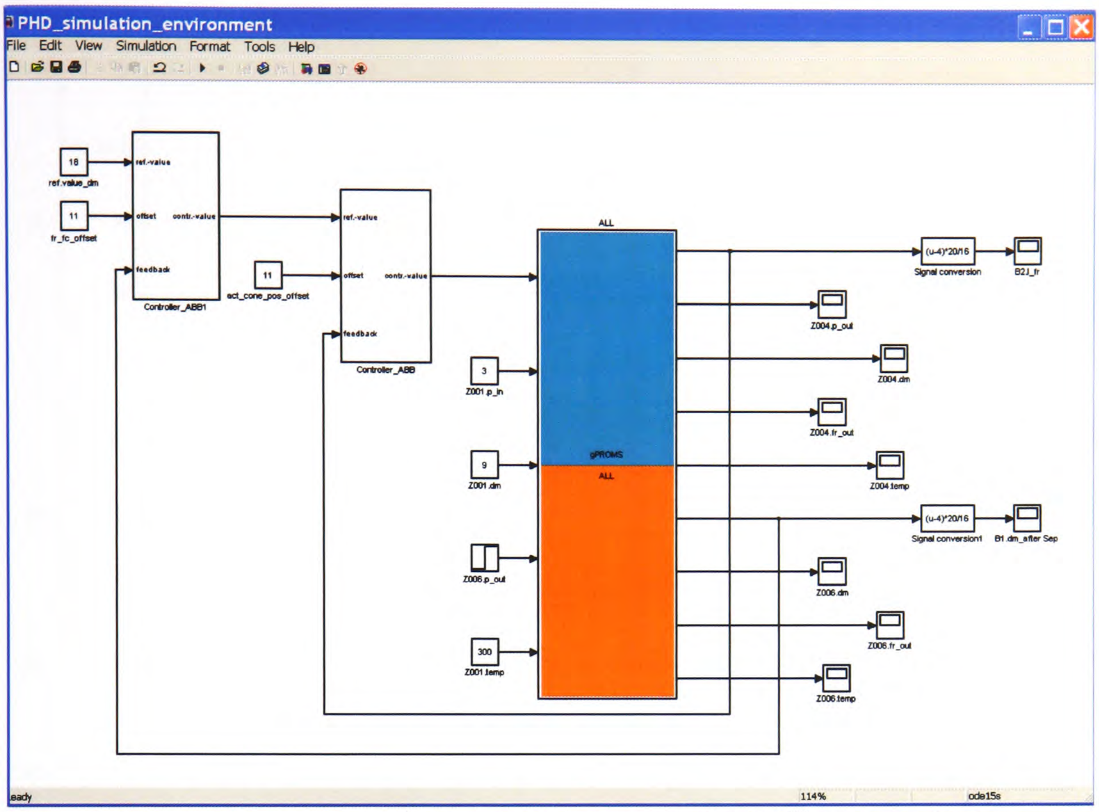


Figure 5-6: Application example: Automatically generated Simulink™ scheme with the gPROMS™ block

5.2.2 Simulation Results

In the current version of Model^{CAT}, the simulation results are presented in the Simulink™ environment, as illustrated in Fig. 5-7, where only the signals from both sensors are presented.

After about *70 s* the disturbance (in this case, a pressure loss of *0.5 bar* at the curd inlet when *t = 500 s*) is compensated and *150 s* are required to reach steady-state again. Therefore, the control action could have been set quite strong (*proportional action coefficient = 0.2* and *integral action coefficient = 0.04*).

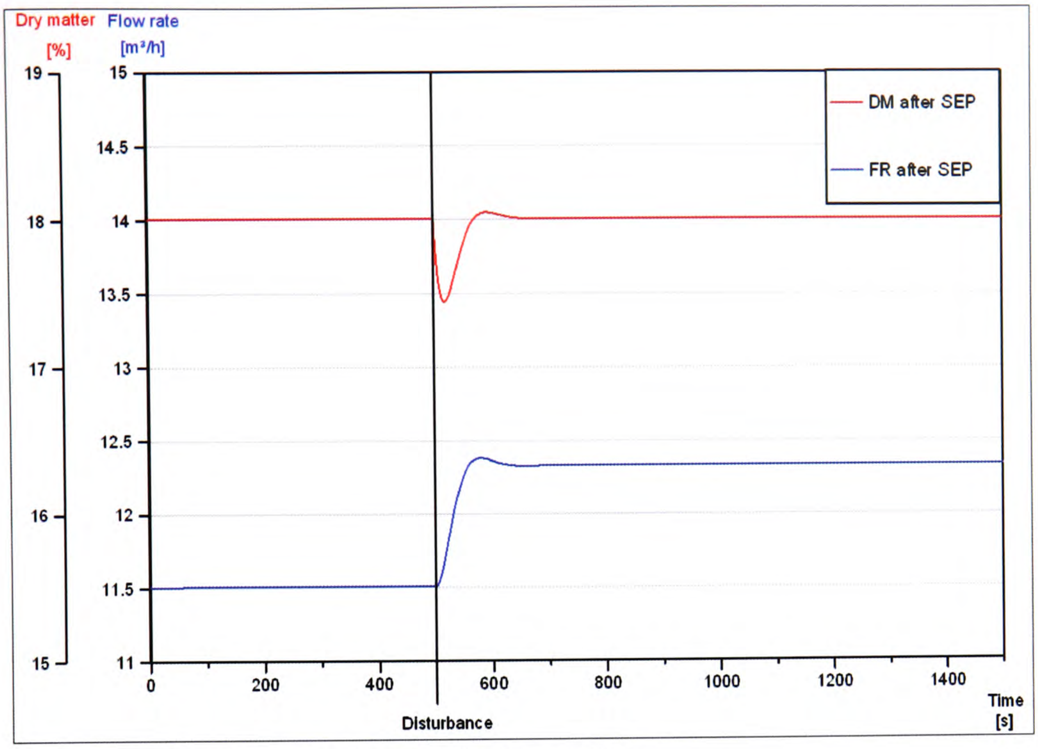


Figure 5-7: Simulation results of application example

In the following section these results are compared to results from plant design on a real industrial fresh cheese production plant.

5.3 Comparison with Real Situation in an Industrial Fresh Cheese Production Plant

The fresh cheese production plant under consideration is operating by the industrial dairy company Humana Milchunion eG in Georgsmarienhütte, Germany. The design of the plant was carried out in 2003 without the use of Model^{CAT}, see Hoyer *et al.* (2003b). Due to space constraints, the position for the NIR-sensor was chosen to be after the heat exchanger when the plant was designed. In this example, the benefits of using Model^{CAT} at the end of detailed engineering are demonstrated retrospectively, as though used before the plant had been built.

Fig. 5-8 illustrates the P&I diagram with the positioning of the NIR-sensor after the heat exchanger.

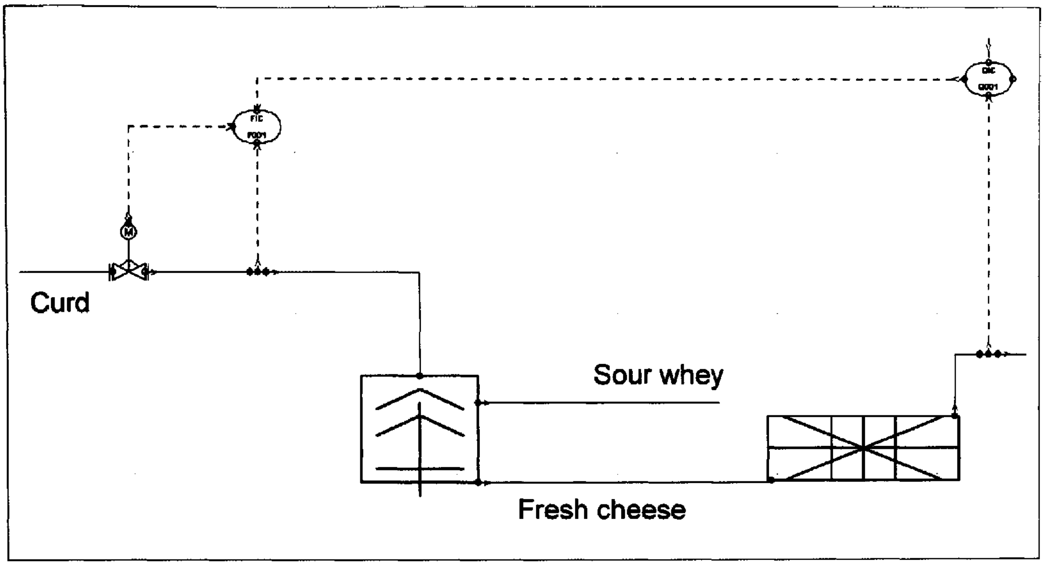


Figure 5-8: P&I diagram with the positioning of the NIR-sensor after the heat exchanger

The Model^{CAT} approach is initialised by the selection of the subsystem, this time with the NIR-sensor after the heat exchanger. After the specification of boundary conditions, initial values and simulation parameters, the entire process and control models are automatically generated as in the first application example, presented in Section 5.2.1.

5.3.1 Simulation Results

The simulation results with the infrared sensor positioned directly after the heat exchanger are shown in Fig. 5-9. They indicate the influence of the transport delay (dead time) of about *120 s* (at t = *500 s*) caused by the position of the NIR-sensor after the heat exchanger. This dead time does not allow the application of high controller gains. Therefore, only weak PI controller settings (*proportional action coefficient = 0.2* and *integral action coefficient = 0.06*) can be used. Compensating the disturbance (the pressure loss of 0.5 bar at the curd inlet) requires in this case about *500 s* (and about *800 s* to achieve a steady-state value) for the system to settle.

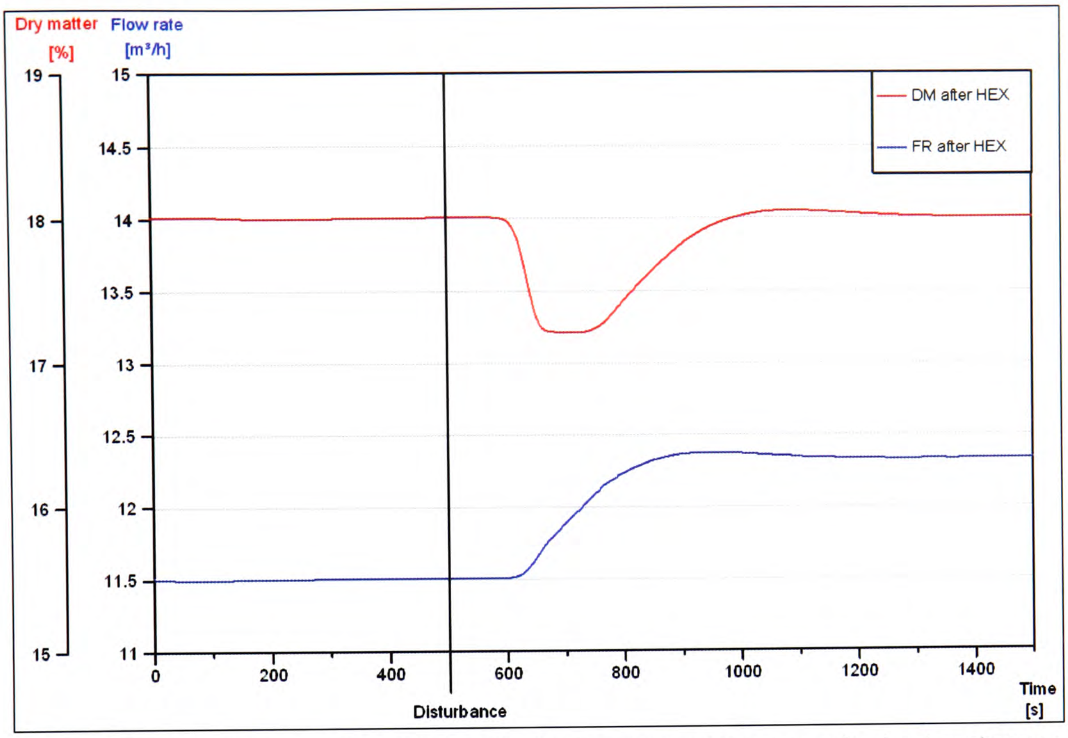


Figure 5-9: Simulation results of the application example with sensor after heat exchanger

Compared with the results from the first scenario with the NIR-sensor directly after the separator (see Fig. 5-7), the process performance is weak due to the dead time resulting from the position after the heat exchanger and the weak controller settings. The set point is reached *430 s* later than in the first scenario (with NIR-sensor directly after the separator).

Within the conventional plant design (= without the support of the Model^{CAT} approach) these relatively large control deviations could not have been detected before the real start-up, but after the plant has already been built. The result would have been that the dry matter control would run with small controller gains allowing inefficient disturbance rejection. This would lead to an inappropriate plant performance, with limitations in terms of functionality and efficiency.

Post commissioning changes to materials and equipment would result as a consequence. Within the real start-up, this backfittings would lead to an enormous effort, especially in terms of time and costs, to change the position of the sensor. Possible consequences and the economic relevance of such backfittings are investigated in more details in Section 6.7. The efforts could have been easily avoided with the Model^{CAT} approach at the end of detailed engineering. At this stage of plant design, these changes could be implemented and tested virtually. Without great effort, the NIR sensor could be placed directly after the fresh cheese separator.

5.3.2 Comparison of Simulation Results with Measurements at the Real Plant

In this section, the simulation results are compared to real data of the fresh cheese production plant, see Fig. 5-10. The red time series (DM_measured) indicates the measured dry matter signal. The pink time series (FR_measured) represents the measured curd flow rate signal. In this example, the set point of the curd flow was changed at $t = 200\ s$ from $11{,}6\ m^3/h$ to $11{,}8\ m^3/h$.

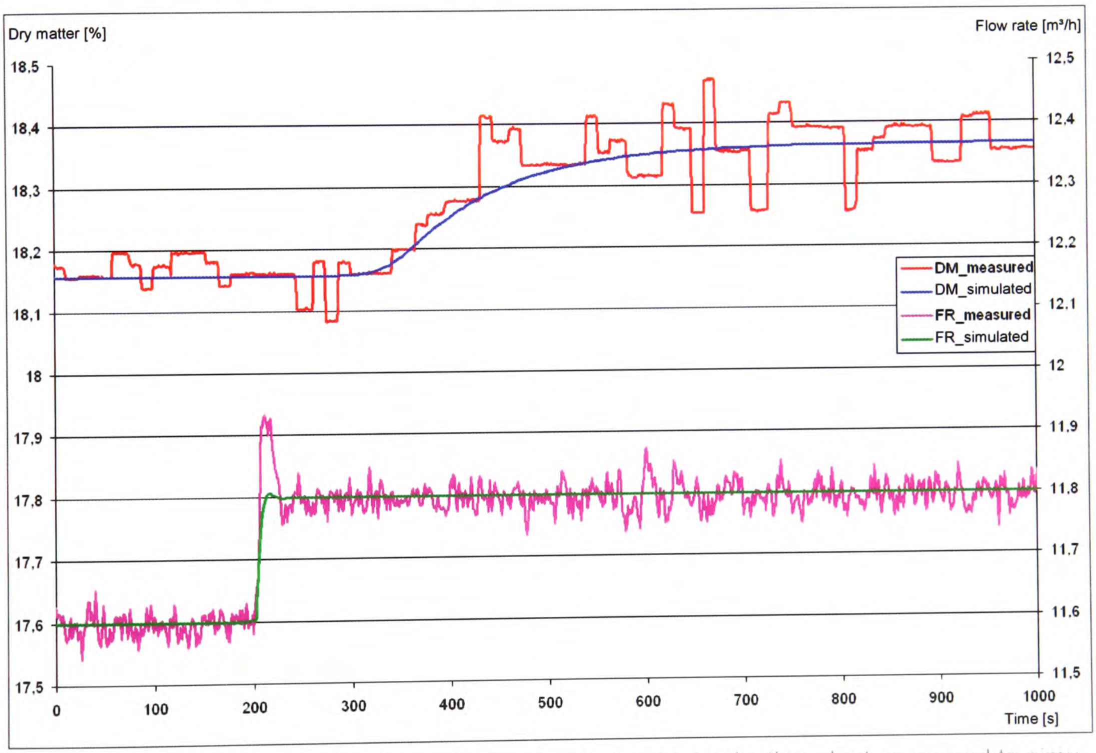


Figure 5-10: Measurements of an industrial fresh cheese production plant compared to simulation results

The results are comparable to the simulation results present in Section 5.3.1: Due to the dead time only weak control performance is carried out. For illustration and also for comparison, the adequate simulation results derived by the Model^{CAT} approach are also presented in Fig. 5-10 for the real plant data. The simulation results are indicated by the blue (dry matter: DM_simulated) and green (flow rate: FR_simulated) time series.

On the industrial fresh cheese production plant, the operator changed the set point of the underlying curd flow control loop, resulting in a delayed transition of the curd flow to the new set point. The underlying curd flow control loop was not modelled, which is the reason for the small overshoot compared to the measured signal. Nevertheless, the static value is reached in both cases in nearly the same time. Looking at the dry matter signals, both signals react due to the set point change of the flow rate after a dead time of about *120 s*. Also the delays in reaching the static value are nearly equal. However, the disturbance variations of the dry matter signal are not reflected by the simulated signal. One reason for these variations of the dry matter signal is the asynchronous recording of the dry matter values as the NIR sensor produced the measurements at arbitrary time intervals. For more details see Hoyer *et al.* (2005d).

Additionally, the use of Model^{CAT} for optimization procedures at the end of detailed engineering is possible and outlined in the following section.

5.4 Optimization of Plant Design with Model^{CAT}

Another application area of the Model^{CAT} approach may be the optimization of plant design at the end of detailed engineering. At this stage, variants of process components and layouts can be investigated very easily by choosing different components from the component database. Additionally, the controller performance could be optimized beforehand on the process model whose behaviour is based on physical (ready-to-buy) models.

The advantage of using the Model^{CAT} approach for these optimization tasks is the quick and easy investigation of the plant's functionality with different process components or new controller settings, before it is built. Possible optimization procedures in the process and control domain are described in the following, starting with the optimization of process design.

5.4.1 Optimization of Process Design

By using the Model^{CAT} approach, the optimization of process components can easily be established at the end of detailed engineering. Different scenarios are conceivable, e.g. the exchange of components with different characteristics or the exchange of components from different component suppliers, as illustrated in Fig. 5-11.

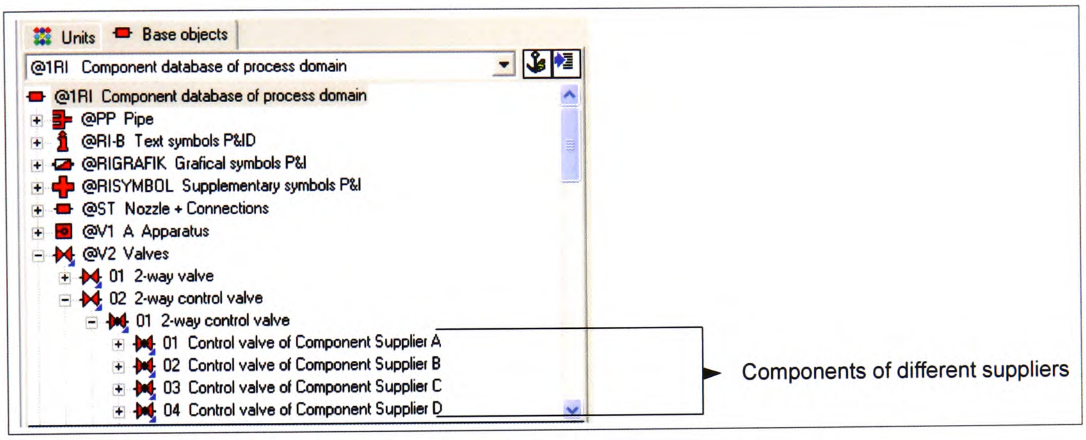


Figure 5-11: Process optimization: Selecting different process components from the component database

With the selected process component, the simulation scripts are easily regenerated. This scenario is illustrated for the flow control example with two different control valves.

Optimization of a Control Valve

In the following, the effects on the flow control example, as selected in Fig. 3-19 are investigated, by choosing two control valves from different component suppliers (with different valve characteristics). As outlined in Section 3.3.1, the valve characteristic is one determining factor of the dynamic characteristics of plant control. Fig. 5-12 illustrates the effects on the entire flow control example: on the top, the simulated step response of the flow control example with the "fast" valve (supplier A) is shown. The rise is completed after about *5 s*, whereas with the valve from supplier B (on the bottom), the rise time is about *150 s*.

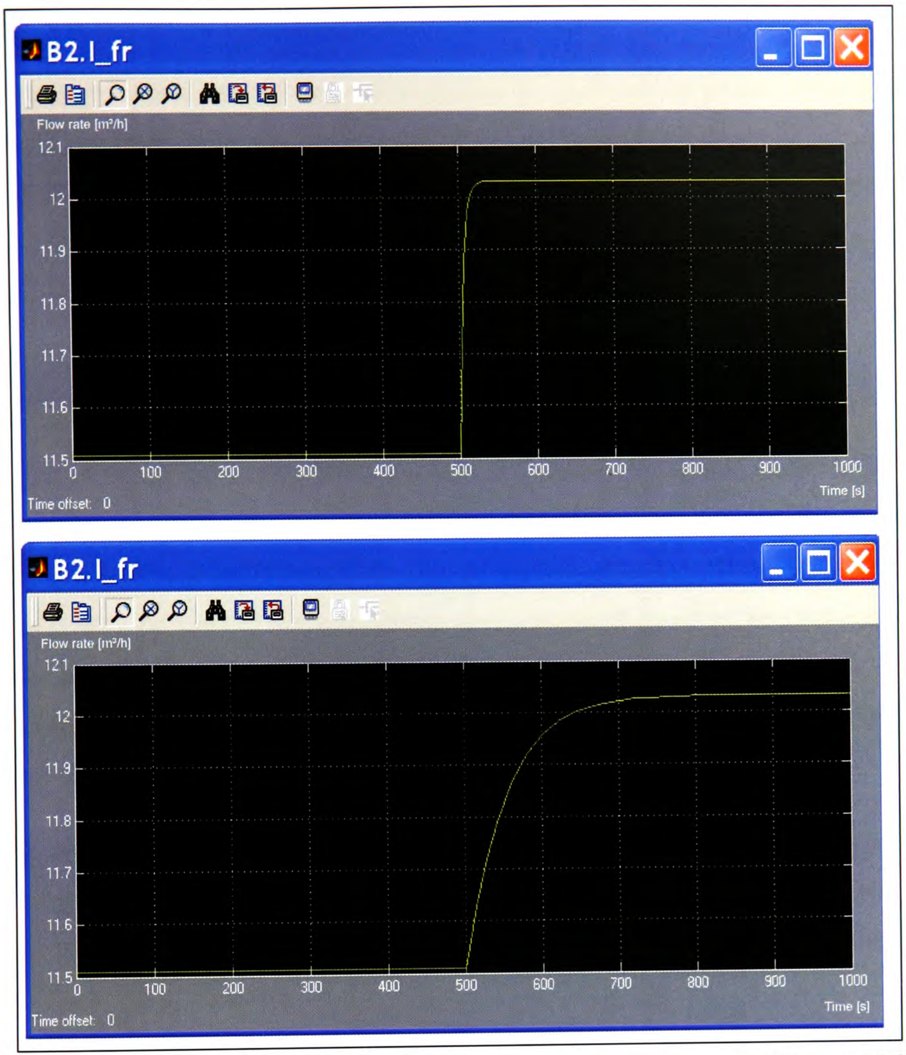


Figure 5-12: Simulation results of flow control example due to different FSCS models (valve): top) supplier A; bottom) supplier B

As a result, the valve from supplier B would cause dramatic drawbacks to the design of the control system with regard to the performance of the flow rate control. With the Model$^{\text{CAT}}$ approach these "weak" points can be detected by simulation before the "incorrect" component is bought, assembled and checked during start-up.

This simple example indicates the enormous usability and benefits of Model$^{\text{CAT}}$. Not only the real start-up time can be reduced, also the quality of the plant design and therefore the plant performance can be improved, before the plant is built.

5.4.2 Optimization of Control Design

Due to time and money constraints during the real start-up phase, many industrial control systems are poorly tuned and switched off controllers are frequently found which results in problematic process behaviour or manual operation, see e.g. Hahn and Nöth (1997) or Willis (1999). As demands increase with respect to process efficiency, product quality and environmental compatibility, the need for better process and control operation raises the question of how the potential of systematic process modelling, model analysis and controller design methods can be made available to industrial control system design, to improve the situation described.

One of the possible answers is the use of CACSD (Computer Aided Control System Design) systems, as outlined by several authors in past decades (see the reviews of Zhen-Yu (1988) and Barker (1991a)). By the establishment of realistic process models with Model^{CAT}, CACSD systems may be used in virtual start-up scenarios, at the end of detailed engineering. This early optimization, in this work used as an heuristic selection of performance, may lead to reduced tuning efforts during the real start-up and in general to better tuned control systems for later operation.

In order to demonstrate such an optimization process within a CACSD system, Model^{CAT} was integrated into the ICACSD (Industrial CACSD) concept, which has been reported in Hoyer *et al.* (2005c) and Hoyer *et al.* (2006a), and is described in the following after a brief introduction in the ICACSD concept.

5.4.2.1 Brief Introduction in the ICACSD Concept

The ICACSD concept as introduced at the 1996 IFAC Congress in San Francisco by Schumann *et al.* (1996) was defined to allow the application of advanced CACSD methods by industrial area engineers and operators with a standard, but limited control education who are not necessarily experts in simulation or control.

ICACSD comprises toolboxes for process identification ICAI (developed by Körner (1999)), qualitative modelling (Model^{ING}, developed by Strickrodt (1997)) and control design (ICAC, developed by Syska (2004)), which have been tailored to the control design and verification tasks and knowledge level of industrial engineers, see Fig. 5-13.

The Model^{CAT} approach extends the ICACSD system, as a further CAE-tool for generating process models with a "close to reality" process representation.

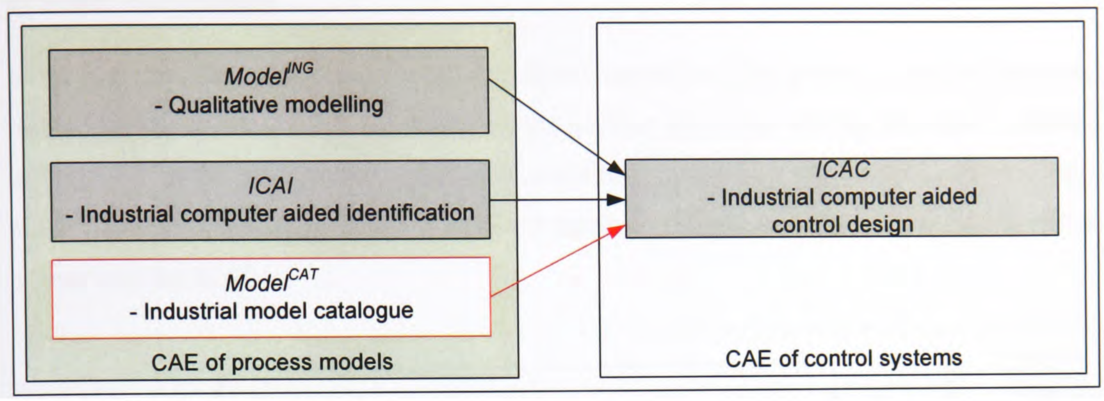


Figure 5-13: ICACSD modules

5.4.2.2 Integration of Model^{CAT} into ICACSD System

In order to integrate the Model^{CAT} approach into the ICACSD concept, the models, as generated by Model^{CAT}, must be pre-processed by applying ICAI to create an appropriately structured linear or nonlinear SISO or MIMO process model. The design path with Model^{CAT}, ICAI and ICAC is shown in more detail in Fig. 5-14. ICAI is used in this case to convert the process simulation models generated by Model^{CAT} into the standardised linear or nonlinear SISO or MIMO format by identification. These standardised process models are used by ICAC to design and optimize standardised linear or nonlinear SISO and MIMO control schemes.

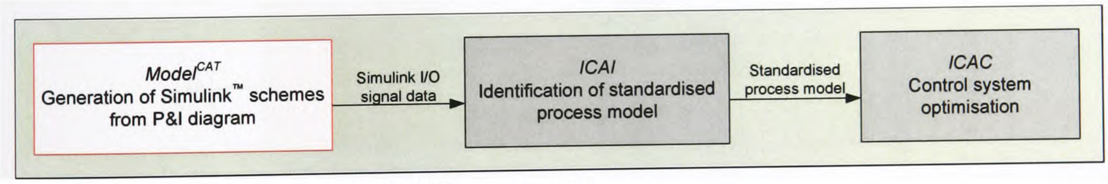


Figure 5-14: Integration of Model^{CAT} into ICACSD system

5.4.2.3 Prototypical Implementation

After completion of plant design, Model^{CAT} automatically generates a Simulink™ scheme from the P&I diagram for the simulation of the selected subsystem including the associated control system. The Model^{CAT} set-up allows direct use of all Matlab™ based CACSD tools to analyse the simulated plant behaviour and to tune and optimize the process control. In the application

example, the subsystem of the fresh cheese production plant, the optimization of the cascaded separator control scheme is demonstrated and executed in two steps, which has been presented by Hoyer *et al.* (2006a).

In the first step, the control loop for the curd flow is optimized. The process model for this control loop is identified with ICAI by using simulation data generated with the Simulink™ scheme in Fig. 5-15: In this scheme the control valve position signal, which will be used as control signal is varied by a test signal which is recorded together with the curd flow output and stored in a "mat"-file for ICAI.

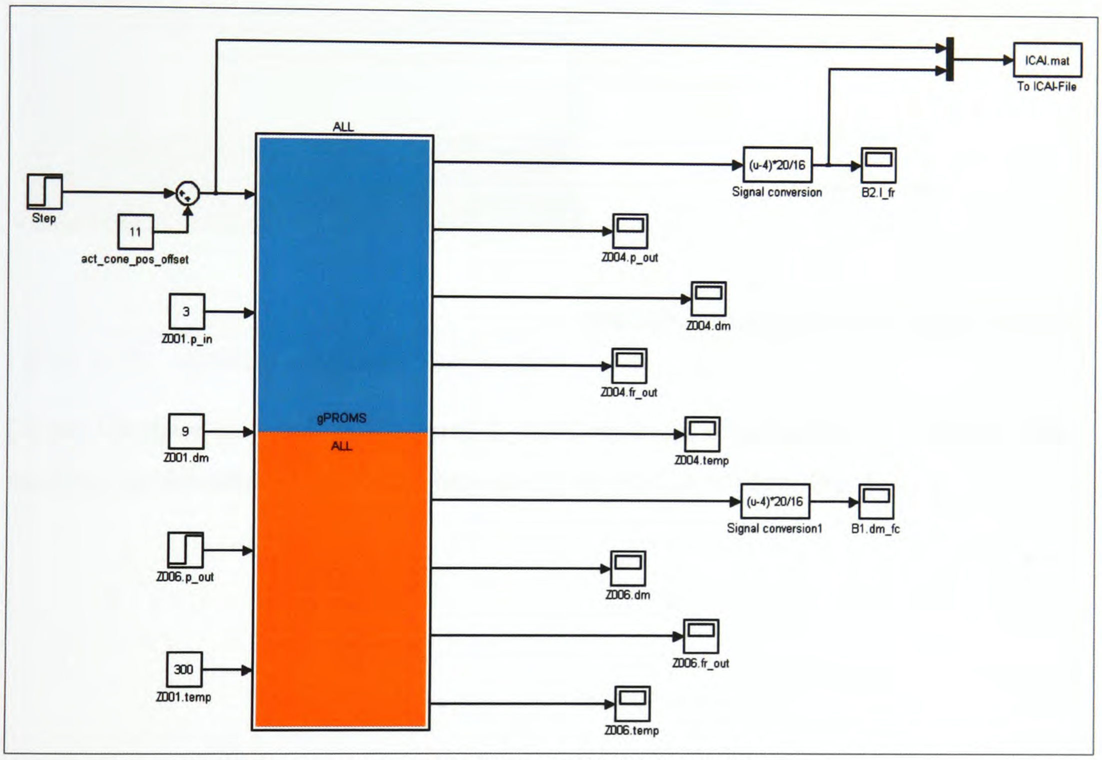
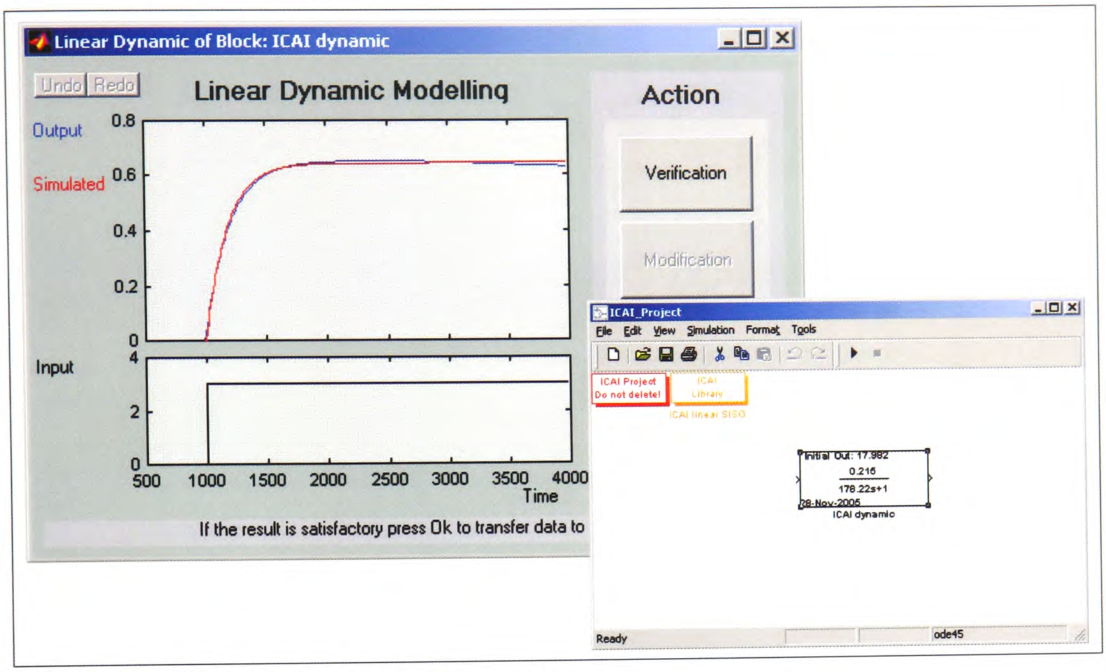


Figure 5-15: ICACSD integration: Extensions for ICACSD system

ICAI identifies from these data a standardised linear (possibly also nonlinear) process model, as shown in Fig. 5-16.



If the result is satisfactory press Ok to transfer data to

Figure 5-16: ICACSD integration: ICAI toolbox

For the identified curd flow process model, ICAC produces an optimal linear controller allowing direct modification of the control behaviour in the design window, Fig. 5-17.

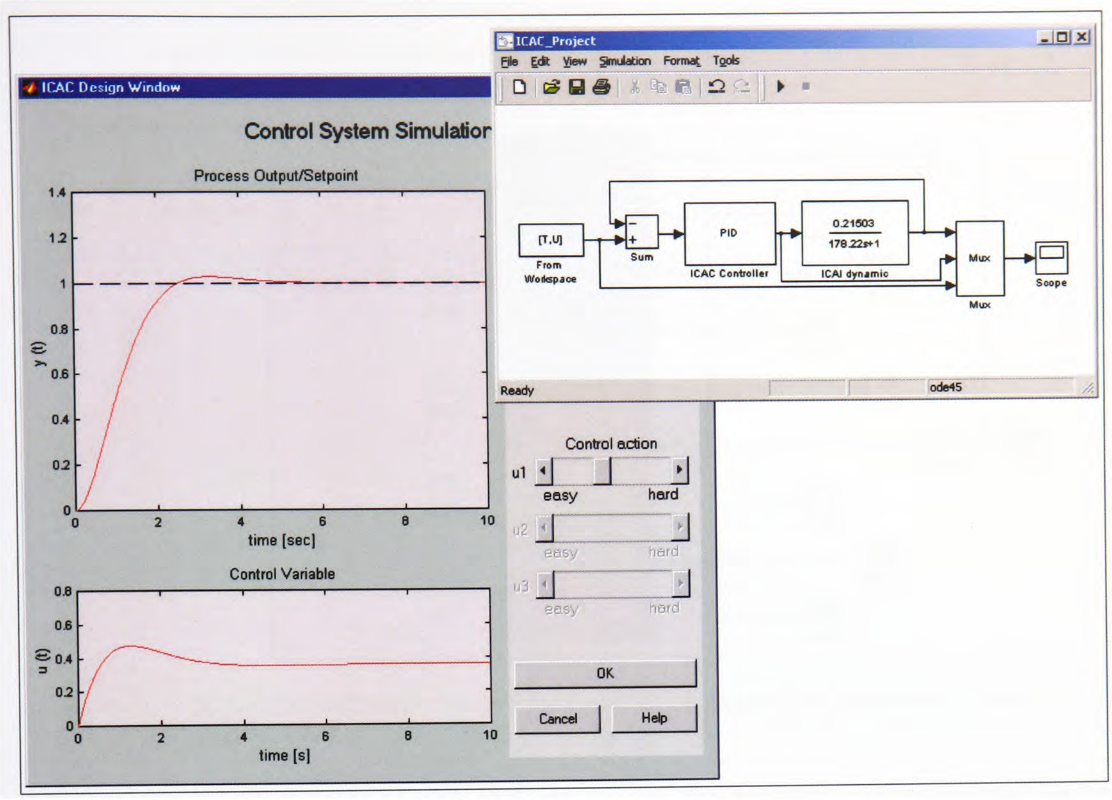


Figure 5-17: ICACSD integration: ICAC toolbox

After the successful design of the curd flow controller, the cascaded dry matter control loop can be designed. Therefore, Fig. 5-15 is extended by the curd flow controller, see Fig. 5-18. This extended scheme is now used for identification of the standardised dry matter process model with ICAI: Therefore a test signal is applied to the reference input of the flow control loop and recorded together with the resulting dry matter signal to a "mat"-file for ICAI. The dry matter control design for the cascaded dry matter controller follows the same path, as described above for curd flow control design using ICAI and ICAC.

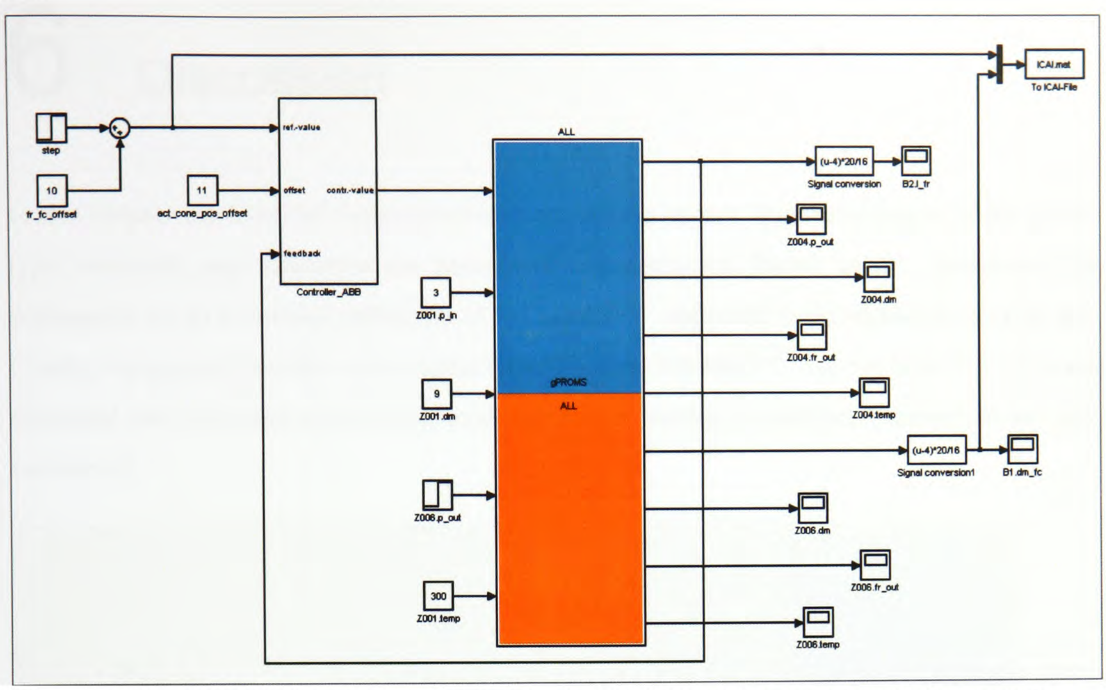


Figure 5-18: ICACSD integration: Extensions for ICACSD system for cascaded controller optimization

After completion of controller design and optimization the complete control scheme can be simulated in the Simulink™ scheme, Fig. 5-6, to test the prospective control performance using the full separator model rather than the standardised ICAI models as used during control design. For more details with regard to the ICAI and ICAC toolbox see the respective literature of Körner (1999) and Syska (2004).

By the integration of the Model^{CAT} approach into CACSD systems, the way towards a broad application of CACSD methods for the optimization of realistic plant models could be opened.

6 Discussion

In this chapter the essential design decisions and results gained from experiences of the prototype realisation and validation are considered and discussed. Based on this discussion, the framework for an industrial realisation of the Model^{CAT} approach is extrapolated, see Fig. 6-1. Finally, the general benefits of an industrial realisation of Model^{CAT} for the plant life cycle are discussed and estimated economic considerations concerning an industrial realisation are also considered.

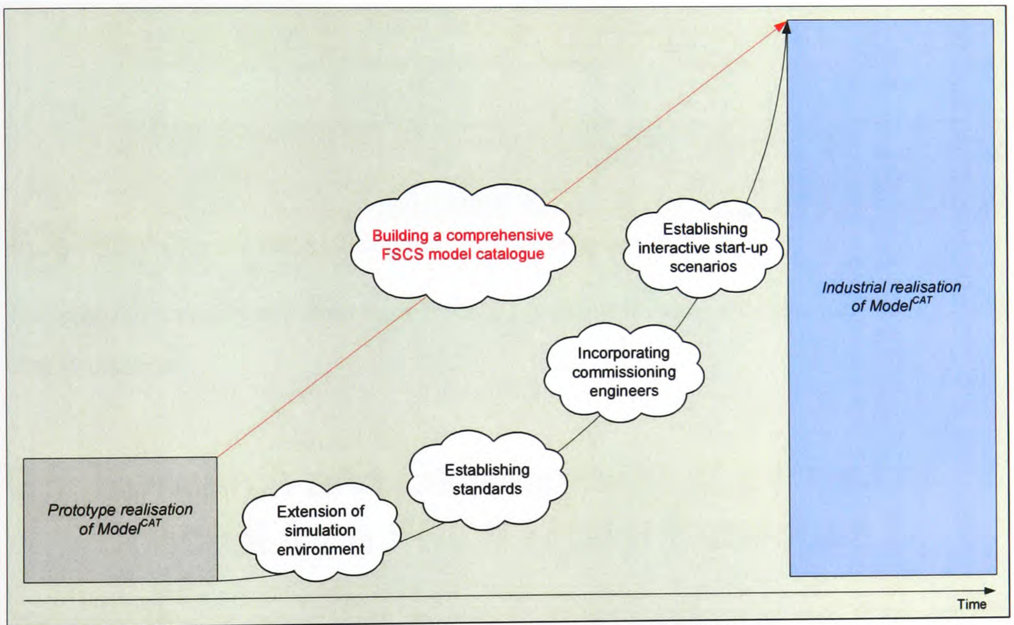


Figure 6-1: Extrapolation of results gained from the Model^{CAT} prototype realisation

The current state of the prototype realisation is shown in Fig. 6-2, including the following issues:

- Entire scheme of the Model^{CAT} approach, including the CAE-plant design tool, MAM, simulation environment and GUI
- Manual model integration of the developed FSCS models into the CAE-plant design tool
- Use of the results of the CAE-plant design (graphical selection of the P&I diagram) as a starting point for an automatic generation of simulation schemes
- Automatic analysis of the graphical object information, separated into process and control domains

- Automatic model aggregation for the control simulation tool Simulink™ and the process simulation tool gPROMS™
- Support for the planning engineer as a non-expert in the selection of the plant subsystem in the P&I diagram, specification of boundary conditions and the start of simulation including the presentation of simulation results.

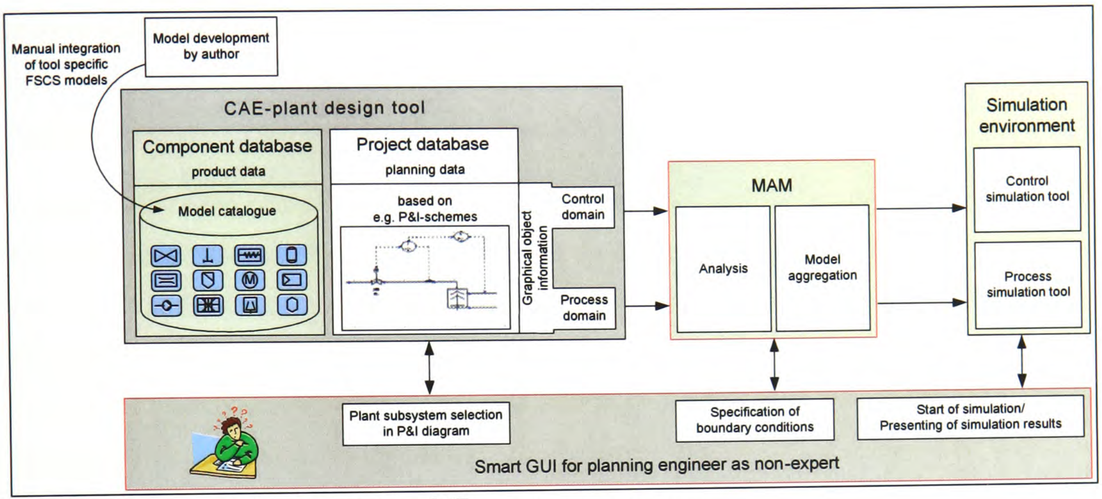


Figure 6-2: Current state of Model^{CAT} prototype realisation

The prototype results are discussed in detail in the following sections with regard to an industrial realisation.

6.1 Research and Development of a Proof of Concept for a FSCS Model Catalogue

The prerequisite for an automatic generation of plant simulation models at the end of detailed engineering is the availability of simulation models of process components (FSCS models). The literature review (Chapter 2) revealed that no catalogue of FSCS models had previously been developed. Orienting towards the requirements of these models, a methodology for the generation of a catalogue of FSCS models has been designed, see Section 3.3.

This methodology was applied in a first prototypical realisation in order to gain knowledge about the applicability of the development of a FSCS model catalogue for industrial realisation. From this reduced scale, essential results were obtained and drawbacks detected, which are considered to indicate a fruitful direction for an industrially relevant development of a FSCS model catalogue. The number of FSCS models developed within the prototype realisation was sufficient to show the principles for the use of FSCS models. In order to develop a comprehensive and therefore useful model catalogue, the actual FSCS model catalogue must be extended by FSCS models of a large set of available industrial components, in order to build up a complete chemical plant. For industrial success, these FSCS models should be provided by the component suppliers as an add-on to the typical technical data for their components. Pressure on the component suppliers could come from the customers, especially big industrial companies. Customers could encourage component suppliers to submit the FSCS models with the technical data for their components by virtue of their spending power. Due to the permanent competition between component suppliers, supply and demand may facilitate this outcome.

Alternatively to the component supplier delivering models, neutral "model developers" could take the role of the model provider. This could be done by engineering companies which are specialised in the development of FSCS models, perhaps in cooperation with universities. The advantage of this approach to model provision could be a neutral and independent view of the component's behaviour. However, this could only be realised in close cooperation with component suppliers, who have the detailed knowledge and expertise of the components they have developed.

Furthermore, the choice of the model type (white or black (secret) box) is essential for the realisation of this concept. The use of white box models, as used in the current Model^{CAT} prototype version, has advantages for later model aggregation and transparency. However, black box models should also be considered, in order to protect the knowledge of component suppliers who are potentially not willing to lay open all internal details. The influences on the model aggregation process of using white and black box models are discussed in Section 6.3.

Model Quality

The model quality is normally a discussion topic in every modelling and simulation task. It is defined by different properties such as robustness, complexity, accuracy, validity and reliability. The robustness and reliability of a model heavily depend on its complexity and accuracy. For the presented modelling approach, where the planning engineer has no influence on the model aggregation procedure, it is very important, that the FSCS models are robust and reliable. This is because the end user, the planning engineer, is not able to go into the details of the FSCS models for making changes.

Model developers would be responsible for the correctness and validation of the supplied FSCS models. For the model validation, every FSCS model should be compared with experimental data from repeatable and carefully executed experiments. Typically in industry, such an approach can only be carried out in the laboratory, which requires a significant effort from many companies. However, due to time and budget constraints this effort is often not spent, see Foss *et al.* (1998).

At this stage, the question how accurate the FSCS models must be, shall be discussed and how much explanation of its limitations will the planning engineer be able to cope with. That the model of a component cannot reach such "one-to-one image", lies in the nature of modelling. However, it is believed, that with a successive increase in complexity, most design mistakes can be tested beforehand in the virtual plant. There will always be specific cases, where the standardised FSCS models are limited in their validity ranges or accuracy. Therefore, the model documentation must contain explicit information, regarding the quality of the validated model. Validity ranges, which are integrated into the model documentation could be compared with the simulation results during runtime. When a model is simulated outside its validity range, the simulation tool could inform the user with an appropriate alarm. Alternatively, the simulation could be aborted. For such cases, simple ways should be established which trigger a kind of "emergency request" to the model developer in order to extend the validity ranges or the accuracy of the FSCS models. By looking "only" at the generation of simulation models for the virtual start-up, aging effects (such as corrosion or leaks resulting from a long life span) can reasonably be neglected.

In order to test and verify FSCS models for an industrial realisation, independent checks may be installed which test the requirements of the FSCS models provided, e.g. by checking the compatibility of the FSCS model's syntax. An example of such independent checks is the compatibility check of PROFIBUS compatibilities in the control domain, established by Bender and co-workers, see e.g. (Römer and Bender, 2002).

Tool Independent Model Syntax

In the prototype realisation, the model syntax was based on the requirements of the chosen simulation tools, gPROMS™ and Simulink™. For a comprehensive industrial realisation of a catalogue of FSCS models, the question of tool independency must be addressed. Due to the variety of simulation tools and preferences, as well as experiences of existing agreements with software tools of different vendors, a neutral model syntax would ease the acceptance of such a comprehensive approach in chemical industries. Therefore, the mathematical representation of FSCS models should be defined using a neutral syntax or a standardised modelling syntax, as established by the Modelica Association (2007). Examples for neutral model descriptions are given by Eggersmann *et al.* (2002) who describe neutral models for conceptual design. They demonstrated how neutral models can be parsed to different simulation tools, such as Modelica, gPROMS™ or to self-developed modelling and simulation tools.

6.2 Research and Development of the Integration of FSCS Models into CAE-Plant Design Tools

With the availability of FSCS models the question arises, as to how FSCS models can be integrated into CAE-plant design tools. From existing literature no evidence was found of such model integration. Therefore, two possibilities for embedding FSCS models into the component database of a CAE-plant design tool have been proposed:

- Separate model catalogue which is linked to the planning objects
- Physically embedding the FSCS models directly in the planning object

The second alternative has been chosen for the prototypical realisation, together with the technical data on the component database level at the planning object. This was done in order to demonstrate the direct connection of the planning object (component) and the FSCS model.

The physical integration may expose an essential drawback: The multitude of FSCS models may lead to update-problems, because every model is referenced in the object-tree at a definitive hierarchical level in the component database. This drawback could be avoided by external storage, as proposed by the first alternative. Possible solutions for such an independent place would be a data warehouse for FSCS models or an Internet based platform. Barker *et al.* (1991b) and Frederick *et al.* (1991) proposed an approach for a widespread use of a model library of block-oriented process models in an UK academic network, called the Joint Academic Network (JANET). Such a detached use of the model catalogue would be most appropriate for this approach and would extend the usability of FSCS models to other applications.

In addition to the model storage place, the method of integrating the FSCS models into the CAE-plant design tools needs to be reconsidered and discussed. In the prototypical realisation, the FSCS models are manually integrated into the component database of the CAE-plant design tool, which is time-consuming and also error prone. The feasibility of an integration into the CAE-plant design tool has been demonstrated by manual integration, but for an industrial realisation a standardised interface which automates the model transfer from component suppliers to the CAE-plant design tool or the external model catalogue, should be used. An example of such a neutral standardised interface syntax is XML (eXtensible Markup Language). In the version of COMOS PT™ used during the development of Model^{CAT}, the XML based interface was realised quite prototypically and with limited functionality. Currently, COMOS PT™ and most other CAE-plant design tools are able to integrate component information by using the XML format satisfactorily. By contrast, some effort has to be expended in order to define agreements and standards (uniform semantics) for the XML based model transfer.

6.3 Research and Development of a Strategy for the Automatic Generation of Plant Models

The idea of automatic generation of plant models based on the results of CAE-plant design arises from the fact that knowledge and data, fully specified and ready for the realisation phases of plant life cycle, can also be used for realising dynamic plant simulations.

The literature review revealed that none of the existing approaches makes use of plant design data for establishing dynamic simulation models. This deficiency has led to the development of a systematic strategy for the automatic generation of plant models for virtual start-ups based on CAE-plant design results. It comprises the analysis of the selected area in the P&I diagram for getting all interconnection relations and open boundaries of the subsystem as well as the aggregation of simulation models of the process subsystem and the control function scheme.

In the prototype realisation, the simulation software tools gPROMS™ and Simulink™ were used. With the use of gO:Simulink™ as a standard interface between gPROMS™ and Simulink™, a clear separation between process and control domains was achieved. gPROMS™ was inserted into Simulink™ as a gPROMS™ function block and represents the process with the strength of a simultaneously solved equation-oriented process simulation tool. In contrast, Simulink™ represents the block based approach to process control systems. By this separation, the process has been adequately modelled by streams and components and the control domain is characterised with signals and control function blocks.

Although the choice for the simulation tools was restricted by budgetary constraints, the main feasibility of a separated simulation environment into process and control domains was demonstrated. For an industrial realisation, Simulink™ could be replaced by a PCS (Process Control System) (DCS/SCADA) emulator, in order to achieve a realistic (one-to-one) image of the real PCS. Even if Simulink™ could not represent the whole range of functionalities of a PCS emulator, the principal idea has been demonstrated successfully. The results from the control domain of the CAE-plant design data at the end of detailed engineering, could be transferred one-to-one to the PCS emulator using a standardised interface, e.g. the currently published CAEX (Computer Aided Engineering eXchange)-standard IEC/PAS 62424, published by the International Electrotechnical Commission (International Electrotechnical Commission, 2005). The CAEX-standard standardises the data exchange between different CAE-plant design tools based on XML, using uniform semantics and could be used for an industrial realisation.

Furthermore, the gPROMS™ block within Simulink™ was inserted as a *quasi* black box, without any information about the model structure within this block. To overcome such deficiencies for an industrial realisation, process simulation tools with graphical and object-oriented model structures should be used. Such tools allow a view on the level at a detailed level by zooming into a subsystem very easily. This could even be broken down into CFD (Computer Fluid Dynamics) examinations. First experiences with a possible integration of a CFD-simulation tool into an equation-oriented process simulation tool were demonstrated by Bezzo *et al.* (2000).

In the prototypical realisation, the information with regard to the selected area results from graphical icons on the P&I diagram. From this graphical information, details of the internal connection were derived in a very pedestrian way *via* external code controlled by MAM, see Section 4.5.3.1. This analysis could be improved if the CAE-plant design tool could automatically provide the object tree information of the selected area. The vendors of CAE-plant design tools could be motivated to provide such feature for an industrial realisation. In the end, supply and demand may determine if this problem will be addressed.

Numerical Problems in Large Scale Plant Simulation

The size and complexity of a chemical plant, which also define the number and quality of component models and their interconnection equations, has an enormous influence on the numerical solution of the super equation. The super equation, as a result from the simultaneous-oriented strategy used, is limited by the number of components and their respective equations.

The main problems of simultaneous simulation strategies are the intensive need for computational power and storage capacities, the application of only one numerical solver for the whole super equation and the high dependency on initial conditions. Due to the relatively small number of components in the prototype realisation example, these problems could be neglected, for an industrial realisation it must be considered. Up to 300,000 equations can be handled within a simultaneously oriented simulation tool to date, see Hommel (2006), and with the increasing power of computers and solver algorithms this number will probably increase, see e.g. (Ponton, 1995).

In order to reduce stability problems and to increase the efficiency of numerical solution, several authors have proposed a combination of the simultaneous and the sequential-modular approach, see e.g. Scharwaechter *et al.* (2002). With the sequential-modular strategy, relatively robust initial conditions can be selected, which can be used as a starting point for the simultaneous strategy. Such a combined approach aims to reduce the problems of each individual simulation approach and exploit the strength of each approach, see e.g. Gilles *et al.* (1986).

Furthermore, a combination of simultaneous and sequential-modular strategies is required if making use of a combination of white and black box (secret) models for the process domain, as proposed in Section 6.1. While white box models may be solved simultaneously, the combination of a white with black box models requires the combination of the simultaneous and sequential-modular strategy with advantages and disadvantages from the corresponding strategy as outlined in Section 2.3.1. A possible realisation of this combined strategy would be that all directly connected white box models could be grouped as a sub-super equation in one subsystem block and solved sequentially with the black box models. Such a combination would also effect the model aggregation and must be addressed in a systematic way for the industrial realisation.

6.4 Research and Development of a Strategy to Support the Planning Engineer with a GUI

The support of the planning engineer to guide him/her from plant design to virtual start-up is an essential task of this work. From a general point of view, it is perhaps too optimistic to assume that a planning engineer is able to cope with the new tasks of modelling and simulation as well as virtual start-up. The prototype realisation has shown that definitive parts of this procedure can be automated from the analysis of the selected area, to the point of the model aggregation, which is generally the preserve of modelling and simulation experts.

For the definition of realistic start-up scenarios (including start-up and shutdown phases, scenarios with load changes, different operating points and tests of the behaviour due to disturbances), boundary conditions and the evaluation of simulation results, the knowledge and especially the experience of a commissioning engineer would be important for a successful virtual start-up of a chemical plant. Therefore, it is proposed that a commissioning engineer should be present within all virtual start-up scenarios.

Selection of Area to Be Simulated

The selection of the area to be simulated is the first part of the user support. In the prototypical realisation, the selection has been carried out on a properly defined plant. Problems will occur, if the plant's complexity increases. In this case, the internal dependencies of a subsystem are generally not known. Therefore, the ongoing specification of these boundary conditions would not necessarily lead to proper initialisation of the super equation. A possible way out of this dilemma would be the selection of definitive subsystems or the entire plant, where the boundary conditions are generally known more intuitively. In general, input and output boundary conditions of the entire plant are known and available because they are automatically set through the environment's conditions (such as the input media energy flows, temperature, etc.). Such definitive input and output states are generally not known from the very beginning in an internal subsystem. Therefore, the entire plant should be simulated with the known boundary conditions first. A kind of zooming functionality could facilitate an insightful view into the selected subsystem afterwards.

Specification of Boundary Conditions

Only from the correct number and values of boundary conditions, can the system of equations be initialised appropriately. The literature revealed insufficient support for non-expert users in such task, which led to the proposed GUI specification. In the methods of this work, an indirect way of checking boundary conditions has been proposed, using the consistency check feature of the respective process simulation tool. This indirect feature should demonstrate the procedure which is required for reducing the degrees of freedom and the start of the simulation by a non-expert. For industrial realisation, the algorithms for such consistency checks must be implemented in the GUI, avoiding a pedestrian approach within the simulation tool.

The GUI specification window for the boundary conditions has shown, in principle, the solution to this task. For the application example the specification task was sufficiently elaborated to demonstrate this functionality within the Model^{CAT} approach. However, the actual GUI specification window is limited with respect to the size of the plant. Even with component icons, which are adjustable to the size of the plant (as realised in the prototype realisation), the overview of the current prototype is becoming unclear with higher numbers of components within the GUI specification window. For an industrial realisation, a clear representation could be achieved by scalable graphical component icons, as realised in the current prototype and to preserve an overview; scroll bars for horizontal and vertical adjustment can be used.

Furthermore, at this point, the possibility of integrating the specification tasks directly into the P&I diagram of the CAE-plant design tool may be thought about. For this purpose, the CAE-plant design tool must be extended, with appropriate functionality, and directly integrated into the P&I diagram. The advantage would be the direct use of the scaling features of the CAE-plant design tool. However, a clear separation between the plant design and the simulation task should be achieved, in order to establish easy use.

Alternatively, in considering a process simulation tool with graphical representation, as discussed in Section 6.3, the specification task could also be integrated into the scheme of the process simulation tool. Having the industrial realisation in mind, the integration of this GUI task into the process simulation tool would also be a practicable solution, provided that the user interface of the simulation tool is extendable. A close cooperation with the vendor of the process simulation tool may be required.

Presentation of Simulation Results

In the prototype realisation the simulation results are presented in the simulation environment, using Simulink™ scopes. This has been considered to be sufficient to illustrate the feasibility of the Model^{CAT} approach. For an industrial realisation the simulation results should be presented in the context of the GUI-window or directly in the simulation environment, including graphical interactive functionalities, e.g. to illustrate the current level of a tank or the actual flow direction and so forth.

The help system contains several topics, ranging from general information about the Model^{CAT} approach, via the selection of the simulation area to the specification of boundary conditions and initial values, to the specification of simulation parameters, to the start of simulation and to simulation results. In an industrial realisation these topics must be supplemented, especially from a cognitive-sensitive point of view, see Section 3.6.2. Access to the help system should also be possible from any part of the software environment that is accessible by the planning engineer.

6.5 Projecting the Prototypical Realisation of Model^{CAT} to an Industrial Realisation

This section summarises the experiences gained from the prototype realisation with respect to an industrial realisation of the Model^{CAT} approach as illustrated in Fig. 6-3 (based on Fig. 6-2) and including the following issues:

- Establishing an entire scheme of the Model^{CAT} approach with further developments of process and control simulation tools, such as a PCS emulator for a "one-to-one image" of the CAE-plant design data of the control domain and a graphically object-oriented process simulation tool, with the ability to look into further details of the process
- Automatic integration of FSCS models provided by component suppliers respectively neutral model developers
- Facilitating the direct use of an object-oriented tree structure for the CAE-plant design tool as a starting point for an automatic generation of simulation schemes
- Establishing a standardised interface (e.g. CAEX standard) to transfer control design data to the PCS emulator
- Using a standardised interface from the MAM to the process simulation
- Supporting the planning engineer as a non-expert and additionally the commissioning engineer, by using extended help functions: Selection of plant subsystems, by zooming into details, specification of boundary conditions and starting of simulations in order to execute interactive virtual start-up simulations

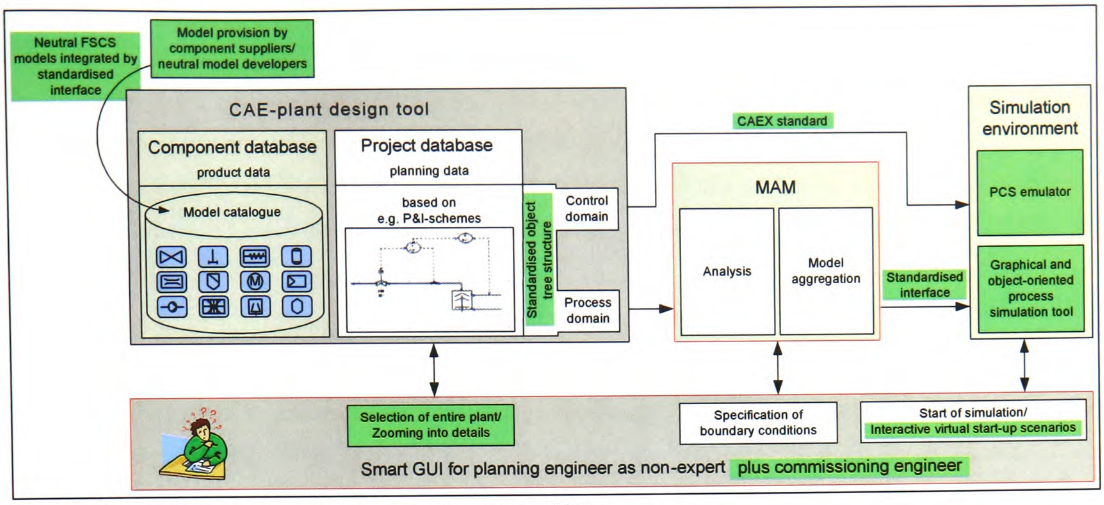


Figure 6-3: Industrial realisation of the Model^{CAT} approach

6.6 Perceived Benefits for the Plant Life Cycle

Assuming that the prototype's limitations will have been removed in future, this section gives a general discussion on the perceived benefits to an industrial realisation of the Model^{CAT} approach.

As outlined in the Introduction (Chapter 1), up to 20 % of the total investment costs are spent in order to correct failures and malfunctions during the start-up and commissioning phases (Weber, 2002). Taking the data of Holroyd (1967), Finneran *et al.* (1968) and Bernecker (2001), 30 % of these fault correction costs are due to design and engineering mistakes. By using of Model^{CAT} these costs may be reduced or possibly avoided.

Several sources of design mistakes (which could be detected and corrected with an industrial realisation of the Model^{CAT} approach at the end of detailed engineering) are presented in the following:

1. Cooperation between Process and Control Domains
   This category contains mistakes, which result from insufficient cooperation between these domains, see e.g. Aström and Häggelund (1995), Nöth (1998), Luyben *et al.* (1998) or Erickson and Hedrick (1999). Examples for such mistakes are:
   - Incorrect positioning of sensors, see prototype validation example (Section 5.3)
   - Incorrect choice of actuators
   - Inappropriate settings of controller parameters (or entire control scheme)
   - None complementary process and control design (see general statement by Cooper and Tracey (2005): "A bad process, though automated, is still a bad process".

With Model^{CAT} such errors can easily be detected at the end of detailed engineering. Changes in the position of sensors or in the choice of components (e.g. actuators) can easily be done without spending much time and money compared to real start-up. At this design stage and with the use of FSCS models, a relatively detailed and accurate (depends on the quality of the FSCS models) representation of the entire process can be achieved, which can be used to very easily adjust and test the control scheme and especially the controller settings using simulation.

2. Over- and Underdimensioning of Components

   Generally, over- and underdimensions of components could have negative effects on the plant's function, see e.g. Gans (1976), Hirschberg (1999), Anderson (2000) or Sattler and Kasper (2000b). When e.g. a valve is oversized, the valve is not sufficiently open to allow the valve to accurately control the flow rate. That is, when the valve plug is very close to the valve seat, large shear forces act on the plug, which tend to completely close the valve. In addition, oversized components are more expensive.

   When the valve is undersized, the valve may be almost fully open so that accurate control is not possible or in certain cases the required flow cannot be met even when the valve is fully open. As a result, underdimensioned components may lead to malfunctions or suboptimal operation of the plant. Corrections of such faults during real start-up would require significant efforts.

   Over- and underdimensioning may be easily detected using Model^{CAT} within an industrial realisation. Suboptimal performance may be detected on the virtual plant, thus avoiding wasted resources at the real start-up phase.

3. Incorrect Choice of Components

   Nearly the same consequences as with over- and underdimensioned components occur from mistakes in the choice of components. An example from the fresh cheese production plant would be an incorrect choice of pump. If e.g a fluid pump is chosen which is unable to pump fresh cheese, which is a highly viscous medium, breakdowns of the pump and corrections during real start-up may result. That could be avoided with the use of Model^{CAT}.

4. Missing Components

   Considerable time delays will occur during real start-up, if single components are not considered within plant design or if they have been downsized (perhaps due to missing knowledge). To insert a missing pump because of e.g. pressure drops in the piping, is a relative simple operation and easy to solve. An even bigger mistake would be a missing buffer tank, such as outlined by Bernecker (2001).

   These missing components could be detected in a virtual start-up scenario making use of Model^{CAT}.

5. Material and Energy Feedback
   Inappropriate arrangements of components, could cause undesired feedback of fluids. Investigations would be time-consuming and are often not undertaken due to time pressures, despite knowing that there may be feedback problems (Weber, 2006).
   This kind of malfunction could be easily detected using Model^{CAT}.

Additionally, consistency checks, which detect compatibility problems should be an integral part of the Model^{CAT} approach. Examples of compatibility problems are the incompatibilities between connectors (e.g. the diameter of the pipe does not fit with the diameter of the valve) or the components material property does not allow the medium used. Relatively costly malfunctions during real start-up arise due to material incompatibilities, which may lead to corrosion (Weber, 2006). In order to put things right, plausibility checks can easily be integrated into the CAE-plant design and mistakes easily detected.

Summarised, it can be said that the functional testing of the virtual plant with Model^{CAT} leads to the following benefits for the plant life cycle:

- Testing the virtual plant in virtual start-up scenarios before plant is built (at the end of detailed engineering)
- Early testing of the interrelationships between process and control domains
- Detecting of incorrect, missing or incorrectly dimensioned components
- Optimization of the plant's functionality and product quality at a very early stage
- Savings in time and money by avoiding fault correction procedures during real start-up (incl. logistic and personnel costs)

In order to illustrate the changes within the plant's life cycle qualitatively, Fig. 6-4 compares a plant life cycle with and without the use of the Model^{CAT} approach.

The upper plant life cycle phase depicts the general procedure without the use of Model^{CAT}. The timing of every phase, compared relatively with each other, are based on average numbers, published by Sattler and Kasper (2000a). With the extension of the detailed engineering phase by a virtual start-up using the Model^{CAT} approach, a considerable number of mistakes from plant design may be detected on the virtual plant. Considerable reduction is expected in the duration of the real start-up phase and also small reductions during assembly and construction phases.

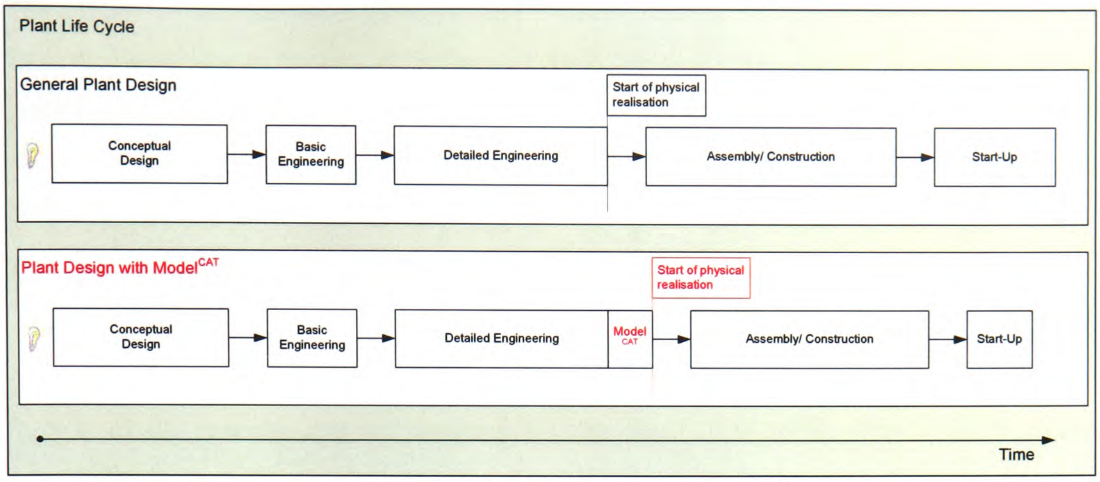


Figure 6-4: Qualitative overview of time savings with the Model^{CAT} approach

The benefits of a time reduction on the total plant life cycle has considerable effects on the economics of a project. An earlier start in production leads to earlier gains for the industrial company. The return on investment (ROI) can be achieved sooner by this earlier start of production.

At this stage, it must be considered that this qualitative overview is based on the assumption that the industrial realisation of the Model^{CAT} approach will be fulfilled. However, especially in the initial stage of an industrial realisation, a transition stage may occur, where not all components will be accompanied by Model^{CAT} compatible models, which implies that the time and costs of using Model^{CAT} will be increased and the benefits from the methods reduced. Such a transition stage could last several years, and heavily depends on the collaboration between industries (engineering companies, component suppliers and software vendors) and universities.

Some of the qualitative investigations are verified within the following quantitative analysis in the following section.

6.7 Economic Aspects

In this section, the economic benefits due to the use of Model^{CAT} are investigated with regard to time and money in a manner which is partly heuristic and partly based on published data. In all ongoing evaluations, the Model^{CAT} approach is compared with the usual design procedure (i.e. without the use of Model^{CAT}). This comparison is based on the following considerations:

The analysis is based on the total investment effort (= *100 %*), which comprises the total effort from the initial idea to the end of the start-up phase. Up to *20 %* of the total investment efforts (costs) arise from fault correction procedures, see Bernecker (2001) or Weber (2002). Bernecker (2001) distinguishes three kinds of plant characteristics/processes, namely fault corrections for:

- Approved characteristics/processes which cause *5* to *10 %* of the total investment effort
- Relatively new characteristics/processes which cause *10* to *15 %* of the total investment effort
- Radically new characteristics/processes which cause *15* to *20 %* of the total investment effort

For the following calculations the average value of *12.5* % of the total investment effort for fault correction efforts, is assumed.

As outlined in the Introduction (Chapter 1), about *30 %* of these fault correction efforts result from plant design mistakes (Bernecker, 2001), which implies that *30 %* of the *12.5 %* (= *3.75 %*) of the fault correction efforts are reasonable for the ongoing calculations. It is assumed that possibly these plant design faults, which have been described in Section 6.6, may be reduced or even eliminated by the Model^{CAT} approach. For the ongoing calculations, it is estimated by the author that *50 %* of the plant design errors may be detected and corrected by simulation using the Model^{CAT} approach. In this case, the total effort reduction due to the use of the Model^{CAT} approach is *1.88 %* (= *50 %* of *3.75 %*).

This percentage will be reduced by the additional efforts which are required to establish the virtual start-up scenarios using the Model^{CAT} approach. These additional efforts include the following issues, sorted into descending order of significance:

- Extra expenses due to the extension of the plant design phase due to establishing start-up scenarios (i.e. personnel costs)
- Increase in component costs from the component supplier, due to the additional provision of FSCS models
- Costs for the integration and development of MAM and GUI to be used with CAE-plant design tools and the process and control simulation tools, including the training of planning engineers
- Costs for establishing the simulation software and additional hardware (historical costs), and also start-up, operating and upgrading costs for licences plus maintenance services

In total, these additional costs are estimated to be *0.75 %* of the total investment effort. This percentage results from a comparison with figures from establishing operator training system. Schumann (2007) stated that up to *0.5* to *1 %* of the total investment is used in establishing operator training systems, where the generation of process models is mainly done manually. The mean value of the numbers of Schumann (2007) has been taken for this estimation in order to take into consideration, not only time savings due to an automatic model simulation on the one hand, but also the greater efforts in developing more accurate models (FSCS models) on the other hand. Compared to relatively simple models for use in operator training systems, the efforts to establish FSCS models, is considerably higher. With such conservative estimation, the effect of greater efforts for the establishment of FSCS models is compensated by a fully automated generation of plant simulation models using Model^{CAT}.

The result from these economical considerations is a total effort saving of about *1.13 %* of the total effort (= *1.88 % - 0.75 %*). These calculations are illustrated graphically in Fig. 6-5. Each step of Table 6-1, starting with the fault correction effort (*12.5 %*) is assigned.

Table 6-1: Effort savings resulting from Model^{CAT}

Step	Type of Effort	Effort in %
0)	Total effort (Total investment costs)	100.0
1)	Fault correction effort	12.5
2)	Fault correction costs due to plant design mistakes (30 % of Fault correction effort)	3.75
3)	Effort reduction with Model^{CAT} (Estimation 50 % of plant design mistakes)	1.88
4)	Additional Effort due to the use of Model^{CAT}	-0.75
5)	***Total effort savings***	***1.13***

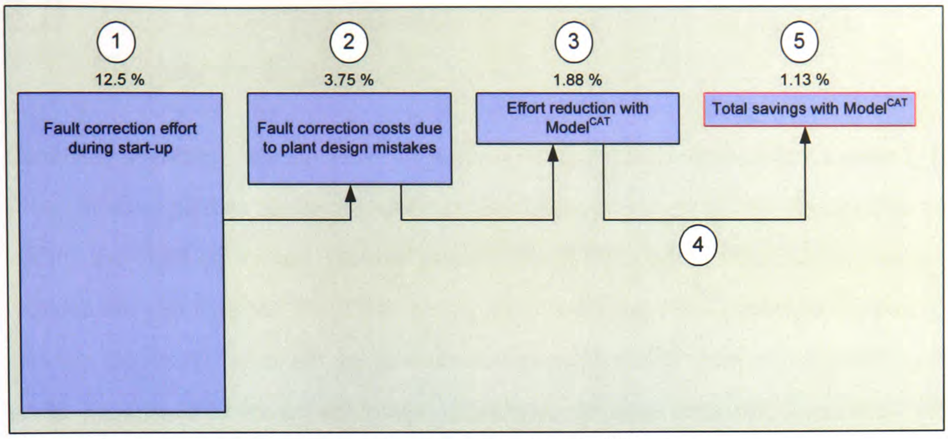


Figure 6-5: Economic savings by using the Model^{CAT} approach

Related to the total investment effort of a fresh cheese production plant of about *£500,000*, the total effort savings due to the use of Model^{CAT} would be *£5,650*. However, related to the total investment costs of an average plant size in the chemical process industry of about *£50M*, according to Jokinen (1996), the total savings due to the use of Model^{CAT} may add up to *£565,000*.

Even knowing that the numbers, presented in this section, are based on relatively rough but nevertheless conservative estimations, this economic evaluation of the Model^{CAT} approach demonstrates the considerable potential of automatic generation of plant simulation models. On the other hand, at this stage it should be pointed out that there is still a lot of work to do in future, in order to establish this concept in the chemical industry. It is estimated that with a close and intensive cooperation between universities, leading chemical companies, software vendors and component suppliers, a considerable and reliable set of FSCS models and an improved software environment can be achieved within the next 5 to 10 years. The first workshop on the topic "Virtual commissioning“ took place in June 2006, established by the GMA-VDI/VDE Society for Measurement and Automatic Control, and showed the enormous interest in this topic from industry (software vendors and engineering companies) and university. Also at the last industrial fair in Hanover 2007, a special plenary session was held with virtual commissioning in mind. This interest, especially from industry, demonstrates the enormous demand on a rapid realisation of the industrial realisation of the Model^{CAT} approach. The relatively cheap modelling and simulation (due to the nature of computers) and increasing processor speeds may enhance this process. In comparison to a period of time resulted from shut-downs during the real start-up phase, the Model^{CAT} approach is considered economical.

6.8 The Overall Results Matched against Expectations

Generally speaking, the aims and objectives of this work, outlined in Chapter 1, have been fulfilled. With regard to the broad scope of the thesis, it was necessary to consider various aspects within the field of virtual start-up simulation at the end of detailed engineering. This work reduces the gap between the plant design and modelling and simulation. Planning and commissioning engineers, who are in general non-experts in the field of modelling and simulation, could be supported by an automated generation of plant simulation models. However, at this point it should be stated, that only by the provision of robust and reliable FSCS models, will such an overall modelling task be made possible.

Bearing in mind that the implementation of the overall approach presented in this thesis is intentionally a prototype, all test resulted have indicated benefits from its use. The experiences gained with the implementation were valuable to the prototype development and to propose further improvements for an industrial realisation. Also frequent discussions on the prototype, not only with colleagues but also at international conferences, have improved the quality of the prototype realisation.

Pushing the prototype to industrial usability, the Model^{CAT} approach still requires extensions and refinements which could only be achieved by the cooperative work of chemical industry, component suppliers and vendors of simulation and CAE-plant design tools.

The aim of such a comprehensive approach was to start simply and increase the complexity successively. This increase can start from relatively basic dynamics to highly sophisticated and complex non-linear equations, where the parameterisation of the model varies with the change in other states, or even different AI (Artificial Intelligence) techniques, such as genetic algorithms (Michalewicz, 1996), fuzzy logic (Passino and Yurkovich, 1998), neural networks (Bishop, 1995) or as an example of a particular application (Chong *et al.*, 2000). Also an increase in the level of details with regard to distributed models, such as CFD simulation could be included. However, every increase in complexity also requires an additional effort from the model provider to deliver a reliable and robust model. An historical look at the situation in the domain of electrical PCBs (Printed Circuit Board) shows that a step-by-step increase in complexity can be successful, see e.g. Thomas (2005).

The relatively good match between the goals of the project and its results show that the original expectations were justified. It has been shown that further work to take the prototype to industrial realisation is promising.

7 Conclusions and Recommendations

The catalogue based model generation of plant models has been developed as a concept to establish dynamic plant simulations after completion of plant design, such that the plant's functions can be tested and planning faults can be corrected before it is built, yielding considerable savings of time and money. The importance and value of the easy generation of plant models has been outlined by many authors previously; see e.g. Cameron (2005). This has been achieved only by an enormous effort and expertise in mathematic modelling and simulation and generally by the use of generic models. By the catalogue based generation of plant models, where every model reflects the behaviour of fully specified physical component (ready-to-buy), this situation can be improved. The work presented in this thesis was aimed to be a step in this direction and a study into the feasibility of facilitating this ambition.

In order to reach the aim of establishing dynamic plant simulations, the concept was subdivided into four objectives, as described in Section 1.1, namely:

- Development of a Proof of Concept for a Simulation Model Catalogue Required for Virtual Start-Up Procedures Scheduled at the End of Detailed Engineering
- Integration of FSCS Models into CAE-Plant Design Tools
- Development of a Strategy for the Automatic Generation of Plant Models Based on the Results of Plant Design Making Use of FSCS Models
- Proposal of a Strategy for the Support of the Planning Engineer with a Suitable GUI (Graphical User Interface) to Guide Her/Him from Plant Design to Virtual Simulation

The development of methodologies for these four objectives led to the successful establishment of a prototype realisation. The experiences and knowledge that were derived from the prototypical realisation and its validation have been discussed in Chapter 6 and were instructive in drawing the conclusions on an industrial realisation in future, see Section 7.1.

The following recommendations, see Section 7.2, are made with the same objectives in view and give possible directions for future work.

7.1 Conclusions

Objective 1:
Development of a Proof of Concept for a Simulation Model Catalogue Required for Virtual Start-Up Procedures Scheduled at the End of Detailed Engineering

In order to establish meaningful simulation scenarios at the end of detailed engineering, the methodology for the design of FSCS models has been proposed. It comprises the requirements of the FSCS model itself, separated into process and control domains, and the requirements for the automatic model aggregation. The prototypical realisation/validation of FSCS models has proved the feasibility of this methodology.

For an industrial realisation, this catalogue of FSCS models must be extended to include a range of models of available physical components (ready-to-buy). The most promising way for the provision of FSCS models is to encourage their provision by the component suppliers in order to distribute the efforts required for model generation. The FSCS models should be either tool independent or a standardised presentation should be considered (such as the Modelica standardisation), in order to ease the acceptance of the FSCS models. The future will show which of these or other approaches will prevail.

Objective 2:
Integration of FSCS Models into CAE-Plant Design Tools

The integration of FSCS models into CAE-plant design tools is a prerequisite for the automatic generation of plant models based on CAE-plant design. The physical embedding of the FSCS models directly into the component database on the planning object level can be and has been manually realised, proving that this concept is feasible.

For an industrial realisation, the FSCS models have not necessarily to be physically integrated into the component database, i.e. due to referencing problems. The external storage of the FSCS model catalogue separated from the component database can be established for an industrial realisation. Furthermore, the manual integration of FSCS models is time-consuming and error-prone. This can be avoided by an automatic integration procedure using standardised templates based on XML.

Objective 3:
Strategy for the Automatic Generation of Plant Models Based on the Results of Plant Design Making Use of FSCS Models

For the automatic generation of plant simulation models based on the results of CAE-plant design tools and the use of FSCS models, a methodology has been proposed, which analyses a selected area of the P&I diagram (within a CAE-plant design tool), aggregates the simulation models of the process and the control domain and converts (parses) them to the respective simulation code. This methodology was successfully implemented and validated in the prototype realisation example.

For an industrial realisation, the simulation in the control domain should be realised using a PCS emulator, which reflects the real functions of a PCS much more closely than Simulink™. The functions of the process simulation tool (gPROMS™) should be extended by graphical visualisation of the models which allows zooming into details. The prototype realisation revealed that the interfaces between the CAE-plant design tool and MAM and further to the process and control simulation tools are not sufficiently standardised. A standard interface definition, such as the CAEX-standard, would be advisable for a simplified and more convenient model aggregation in an industrial realisation.

Objective 4:
Strategy for the Support of the Planning Engineer with a Suitable GUI (Graphical User Interface) to Guide Her/Him from Plant Design to Virtual Simulation

In order to support the planning engineer, considered as a non-expert in the field of modelling and simulation, a methodology for supporting the planning engineer has been proposed. A smart graphical user interface (GUI) was developed to guide the planning engineer through the simulation task with three subtasks, from defining the area to be simulated (1), via specification of boundary conditions and initial values (2) to the specification of simulation parameters (3).

The prototype realisation has demonstrated that the user support strategy was fruitful and effective for establishing dynamic plant simulations by non-experts. While the model analysis and aggregation could be automatised by MAM without the presence of modelling and simulation experts, the planning engineer should work more closely together with the commissioning engineer to establish dynamic plant simulations, especially if the plant's complexity increases. The specification of boundary conditions can also be simplified if the plant is always simulated in full and the subsystem of interest is investigated as a part of the entire plant.

The actual prototype realisation does not fulfil the possibility for executing realistic start-up scenarios including start-up and shut down procedures or load changes. However, these scenarios are the most critical ones of the start-up phase and should also be considered as part of an industrial implementation. Therefore, possible ways should be investigated for commissioning engineers to establish virtual start-up scenarios within a software environment. Finally, the presentation of simulation results should be extended by more interactive functionalities and graphical visualisation, in order to improve the understanding of the simulation results.

7.2 Recommendations for Future Work

Based on the discussion of the prototype realisation and the conclusions of this work, possible extensions of Model^{CAT} for an industrial realisation are summarised in the following:

- **Extension to the Catalogue of FSCS Models Provided by Component Suppliers**
  The number and complexity of FSCS models must be increased. It is recommended that the component supplier, the chemical industry and universities should work closely together in order to establish a framework for defining a standardised model provision procedure.
  The advantages of simulation support for plant design should be communicated widely to the chemical industry and the component suppliers. It is believed that the commercial pressures will stimulate rapid acceptance of the use of the Model^{CAT} approach, if agreements on model structures and a critical number of component models can be achieved.
- **Extension to a Neutral Exchange Format**
  A XML based standard for the exchange of plant design data (CAEX) has been defined by the IEC. These XML based techniques should be considered for the exchange of FSCS models between component suppliers and chemical industry and for the integration into the CAE-plant design tool. This is proposed to ease the model integration into the model catalogue and the use within industry.
- **Extension to Simulation Independent Model Description and Development of Appropriate Parsers**
  Simulation independent model descriptions have been considered by a number of researchers. These techniques should be considered for the neutral description of the models in order to allow the use of different simulation tools by converting the neutral description to the respective simulation tool. The development of appropriate parsers (or wrapper) should be part of these considerations.

- **Extension to the Simulation Environment**
  It is recommended to replace the control simulation tool with a process control system emulator in order to achieve a one-to-one representation of the real process control system. Additionally, it should be investigated how a process simulation tool with a graphical frontend could be used for presenting simulation results for zooming into different level of process details.
- **Extension to the GUI**
  The intuitive handling of the GUI should be increased for the future work. A direct integration into a CAE-plant design tool should be investigated in more detail.

  Furthermore, it is recommended to set the focus on the execution of realistic start-up scenarios. Real start-up scenarios (including start-up and shut down procedure or load changes) should be developed and implemented for the realisation of virtual start-up scenarios.

Recommendations for the Scope of Applications

Through an industrial realisation of the Model$^{\mathrm{CAT}}$approach, as proposed in this work, always assuming that the technical and logistic problems will be resolved in future, a new level of aggregated plant simulation models may be achieved. The automatic and easy generation of simulation models could be employed for use during conceptual design, making use of catalogues of generic simulation models. Furthermore, a possible application area of the Model$^{\mathrm{CAT}}$ approach may be in integration with CACSD systems, using Model$^{\mathrm{CAT}}$ as "Process model supplier". By the development of reliable and meaningful process models, the results of CACSD systems could also be lifted to another level. Finally, the industrial realisation of Model$^{\mathrm{CAT}}$ may be used during operational phases for optimization procedures to increase plant's performance or to implement modifications.

References

AEGIS (2004). *acslXtreme*. http://www.aegistg.com/

ANDERSON, N.G. (2000). *Practical process research & development.* Academic Press, San Diego.

ANDERSSON, M. (1990). *Omola - an object-oriented language for model representation.* PhD Thesis, Lund Institute of Technology, Lund University, Lund.

ASPENTECH (2002). *Aspen Plus*. http://www.aspentech.com/

ASPENTECH (2004). *Aspen Customer Dynamics*. http://www.aspentech.com/

ASTRÖM, K.J. (2003). *Modeling of complex systems.* In Modeling, Control and Optimization of Complex Systems - in Honor of Professor Yu-Chi Ho; Kluwer International Series on Discrete Event Dynamic Systems, Kluwer Academic, Norwell, vol. 14, pp. 19-45.

ASTRÖM, K.J. and T. HÄGGLUND (1995). *PID-controller: Theory, design and tuning.* Instrument Society of America, NC.

AUREL (2002). *CADSIM Plus*. http://www.aurelsystems.com/

AUTODESK (2003). *AutoCad.* http://usa.autodesk.com/

BALLINGER, G.H., E.S. FRAGA and R. BANARESALCANTARA (1995). *Case-study in-process engineering tool integration.* Computers & Chemical Engineering, vol. 19, Suppl., pp. 759-764.

BALLINGER, G.H., R. BANARESALCANTARA, D. COSTELLO, E.S. FRAGA, J. KING, J. KRABBE, D.M. LAING, R.C. MCKINNEL, J.W. PONTON, N. SKILLING and M.W. SPENCELEY (1994). *Developing an environment for creative process design.* Chemical Engineering Research & Design, vol. 72, no. A3, pp. 316-324.

BALLSTAEDT, S.-P. (1997). *Wissensvermittlung - Die Gestaltung von Lernmaterial.* Beltz Psychologie Verlags Union, Weinheim.

BANARES-ALCANTARA, R. (1995). *Design support systems for process engineering-I. Requirements and proposed solutions for a design process representation.* Computers & Chemical Engineering, vol. 19, no. 3, pp. 267-277.

BARKER, H.A., M. CHEN, P.W. GRANT, C.P. JOBLING and P. TOWNSEND (1989). *A man-machine interface for computer-aided design and simulation of control systems.* AUTOMATICA, vol. 25, no. 2, pp. 311-316.

BARKER, H.A. (1991a). *Computer aided design in control systems: Selected papers from the IFAC symposium.* IFAC Symposium on Computer Aided Design and Control Systems, Pergamon Press, Oxford, UK.

BARKER, H.A., P.H. HAMMOND, T. HUYNH-QUOC, C.P. JOBLING and D.K. FREDERICK (1991b). *A model library for the control engineering community.* International IEE Conference Publication 322, Edinburgh, vol. 1, pp. 300-305.

BARKER, H.A., M. CHEN, P.W. GRANT, C.P. JOBLING and P. TOWNSEND (1993a). *Graphical user interfaces in computer-aided control system design.* In ed. LINKENS, D. A., CAD for Control systems, Marcel Dekker, New York, pp. 561-578.

BARKER, H.A., M. CHEN, P.W. GRANT, C.P. JOBLING and P. TOWNSEND (1993b). *Open architecture for computer-aided control engineering.* IEEE Control Systems Magazine, vol. 13, pp. 17-27.

BARTON, P.I. (2000). *The equation oriented strategy for process flowsheeting.* Department of Chemical Engineering. Massachusetts Institute of Technology, Cambridge.

BARTON, P.I. and C.C. PANTELIDES (1993). *gPROMS - A combined discrete/continuous modeling environment for chemical processing systems.* Sim. Ser., vol. 25, no. 3, pp. 25-34.

BAYER (2003). *Operator Training Simulation.* http://www.bayertechnology.com/index2.htm/

BAYER, B. and W. MARQUARDT (2003). *A comparison of data models in chemical engineering.* Concurrent engineering research and applications, vol. 11, no. 2, pp. 129-138.

BAYER, B., R. SCHNEIDER and W. MARQUARDT (2000). *Integration of data models for process design - first steps and experiences.* Computers & Chemical Engineering, vol. 24, no. 2-7, pp. 599-605.

BAYER, B., W. MARQUARDT, K. WEIDENHAUPT and M. JARKE (2001). *A flowsheet centered architecture for conceptual design.* In eds. GANI, R. and S. JORGENSEN, European symposium on computer aided process engineering, Elsevier, vol. 11, pp. 345-350.

BECKER, S. and B. WESTFECHTEL (2003). *UML - Based definition of integration models for incremental development processes in chemical engineering.* Integrated Design and Process Technology, pp. 1-10.

BECKER, S., T. HAASE, B. WESTFECHTEL and J. WILHELMS (2002). *Integration tools supporting cooperative development processes in chemical engineering.* Proceedings of the sixth biennial world conference on integrated design and process technology (IDPT2002). Society for Design and Process Science, Pasadena, California, pp. 1-10.

BERNECKER, G. (2001). *Planung und Bau verfahrenstechnischer Anlagen.* Springer-Verlag, Heidelberg.

BEZZO, F., S. MACCHIETTO and C.C. PANTELIDES (2000). *A general framework for the integration of computational fluid dynamics and process simulation.* Computers & Chemical Engineering, vol. 24, no. 2-6, pp. 653-658.

BIALKOWSKI, W.L. (1994). *Dreams versus reality: A view from both sides of the GAP.* Pulp and Paper, vol. 94, no. 11, pp. 19-27.

BIEGLER, L.T. (1989). *Chemical process simulation.* Chemical Engineering Progress, vol. 85, no. 10, pp. 50-61.

BIEGLER, L.T., I.E. GROSSMANN and A.A. WESTERBERG (1997). *Systematic methods of chemical process design.* Prentice Hall PTR, Upper Saddle River.

BIEGLER, L.T., D. ALKAYA and K. ANSELMO (1999). *Multi - solver modeling for process simulation and optimization.* Carnegie Mellon University, Pittsburgh.

BISHOP, C.M. (1995). *Neural networks for pattern recognition.* Oxford University Press, New York.

BOGUSCH, R. (2001). *Eine Softwareumgebung für die rechnergestützte Modellierung verfahrenstechnischer Prozesse.* VDI-Verlag, Düsseldorf.

BOGUSCH, R., B. LOHMANN and W. MARQUARDT (2001). *Computer-aided process modeling with MODKIT.* Computers & Chemical Engineering, vol. 25, no. 7-33, pp. 963-995.

BRAUNSCHWEIG, B.L., C.C. PANTELIDES, H.I. BRITT and S. SAMA (2000). *Process modeling: The promise of open software architectures.* On the Horizon: Chemical Engineering Progress, vol. 9, pp. 65-76.

BRENAN, K., S. CAMPELL and L.A. PETZOLD (1989). *Numerical solution of initial-value problems in differential-algebraic equations.* Elsevier, New York.

BRÜCKNER, M. (2004). *Plant engineering benefits from efficient IT tools - Anlagenbau profitiert von effizienten IT-Werkzeugen.* 3R international-Zeitschrift für die Rohrleitungspraxis, vol. 6, pp. 1-4.

BURR, P.S. (1993). *The design of optimal air separation and liquefaction processes with the OPTISIM equation-oriented simulator and its application to on-line and off-line plant simulation.* AIChE Symp. Series 89, vol. 294, pp. 1-7.

CAMERON, D., C. CLAUSEN and W. MORTON (2002). *Dynamic simulators for operator training.* In eds. BRAUNSCHWEIG, B. and R. GANI, Software Architectures and Tools for Computer Aided Process Engineering, Elsevier Science, Amsterdam, vol. 11, pp. 393-431.

CAMERON, I.T. (1983). *Large scale transient analysis of processes - The state of the art.* Lat.am.j.chem.eng.appl.chem., vol. 13, pp. 215-228.

CAMERON, I.T. (2005). *Modelling across the process life cycle: A risk management perspective.* In eds. PUIGJANER, L. and A. ESPUNA, European Symposium on Computer Aided Process Engineering (ESCAPE) -15, Elsevier Science, Barcelona, vol. 20, pp. 3-19.

CAPEC (2004). *ICAS & Toolboxes.* http://www.capec.kt.dtu.dk/Software/ICAS-and-its-Tools/

CARPANZANO, E., R. FABBRI and L. FERRARINI (1998). *A structured methodology for the design and implementation of hybrid robot controllers.* Proceedings of the 1998 IEEE International Conference on Control Applications, Trieste, Italy, vol. 1, pp. 572-577.

CELLIER, F.E. (1993). *Integrated continuous-system modeling and simulation enironments.* In ed. LINKENS, D. A., CAD for Control systems, Marcel Dekker, New York, pp. 1-29.

CHARWAT, H.J. (1996). *Farbkonzept für die Prozeßführung mit Bildschirmen R. Oldenbourg Verlag.* atp, vol. 38, no. 5, pp. 50-55.

CHEMSTATIONS (2002). *CHEMCAD.* http://www.chemstations.net/

CHEN, H.-S. and M.A. STADTHERR (1985). *A simultaneous-modular approach to process flowsheeting and optimization - Part I: Theory and implementation.* AIChE Journal, vol. 31, no. 11, pp. 1843-1856.

CHONG, A.Z.S., S.J. WILCOX and J. WARD (2000). *Application of a neural-network-based controller on an industrial chain grate stoker fired boiler.* Journal of the Institute of Energy, vol. 73, no. 497, pp. 208-214.

CO-LAN (2006). *The CAPE-OPEN Laboratories Network (CO-LaN).* http://www.colan.org/

CONTRERAS, J. and J.M. FERRER (2005). *Dynamic simulation: A case study.* Hydrocarbon Engineering, vol. 10, no. 5, pp. 103-107.

COOPER, D.P. and M. TRACEY (2005). *Supply chain integration via information technology: strategic implications and future trends.* Int. J. Integrated Supply Management, vol. 1, no. 3, pp. 237-257.

COSTELLO, D.J., E.S. FRAGA, N. SKILLING, G.H. BALLINGER, R. BANARES-ALCANTARA, J. KRABBE, D.M. LAING, R.C. MCKINNEL, J.W. PONTON and M.W. SPENCELEY (1996). *épée: A support environment for process engineering software*. Computers & Chemical Engineering, vol. 20, no. 12, pp. 1399-1412.

COUPER, J.R., W.R. PENNEY, J.R. FAIR and S.M. WALAS (2005). *Chemical process equipment*. Elsevier, Burlington.

DARNELL, P.A. and R.A. KOLK (1990). *An interactive simulation and control design environment*. Proceedings of the 1990 European Simulation Symposium, SCS, pp. 55-60.

DAVID, R. and H. ALLA (1992). *Petri nets and grafcet*. Prentice Hall, New York.

DÖMGES, R., K. POHL and M. JARKE (1996). *PRO-ART/CE - An environment for managing the evolution of chemical process simulation models*. In Modelling and Simulation 1996, Soc. Computer Simulation, San Diego, pp. 1012-1017.

DRENGSTIG, T., S.O. WASBO and B.A. FOSS (1997). *A formal graphical based process modeling methodology*. Computers & Chemical Engineering, vol. 21, Suppl., pp. 835-840.

DYNASIM (2003). *Dymola*. http://www.dynasim.se/

EGGERSMANN, M., R. SCHNEIDER and W. MARQUARDT (2002). *Modeling work processes in chemical engineering - from recording to supporting*. In European Symposium on Computer Aided Process Engineering - 12, Elsevier Science B.V., Amsterdam, vol. 10, pp. 871-876.

ELMQVIST, H. (1977). *SIMNON - An interactive simulation language for non-linear systems*. Proc. Simulation '77, Montreux, France.

ELMQVIST, H. (1978). *A structured model language for large continuous systems*. PhD Thesis, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

ELMQVIST, H., S.E. MATTSSON and M. OTTER (1998). *Modelica - The new object-oriented modeling language*. In Simulation: Past, Present and Future, Soc. Computer Simulation, San Diego, pp. 127-131.

ENDER, D.B. (1993). *Process control performance not as good as you think*. Control Engineering, vol. 40, pp. 180-190.

ERICKSON, K.T. and J.L. HEDRICK (1999). *Plantwide process control*. John Wiley & Sons, Inc., New York.

FELLEISEN, M. (2001). *Prozeßleittechnik für die Verfahrensindustrie*. Oldenbourg-Industrieverlag, München.

FINNERAN, J.A., N.J. SWEENEY and T.G. HUTCHINSON (1968). *Startup performance of large ammonia plants.* Chem.Eng. Prog., vol. 64, no. 8, pp. 72-77.

FISCHER, C. (2007). *Konzeption eines Styleguides für Bedienoberflächen bei der R&S Mobilfunk GmbH.* Diploma Thesis, University of Applied Sciences and Arts Hanover, Hanover.

FISCHER, H. (2005). *Sicherheit und Transparenz vom Design bis zum Betrieb - Mit durchgängiger Simulation zum Null-Fehler-Engineering.* In VDI-Berichte, VDI-Verlag, Düsseldorf, vol. 1883, pp. 449-456.

FOSS, B.A., B. LOHMANN and W. MARQUARDT (1998). *A field study of the industrial modeling process.* Journal of Process Control, vol. 8, no. 5-6, pp. 325-338.

FRAGA, E.S. and K.I.M. MCKINNON (1994). *CHiPS: A process synthesis package.* Trans IChemE, vol. 72, no. A, pp. 389-394.

FRANZKOWIAK, M. (2004). *Eine berufspraktische Ratgeber-Reihe für das Berufsfeld der Technischen Redaktion - Konzeption und prototypische Umsetzung.* Diploma Thesis, University of Applied Sciences and Arts Hanover, Hanover.

FREDERICK, D.K., H.A. BARKER, P.H. HAMMOND, C.P. JOBLING and F.M. SEILER (1991). *Plant models for use with computer-aided control-system analysis and design packages.* IFAC Computer Aided Design in Control Systems, Swansea, pp. 239-246.

FRITZ, M. and S. ENGELL (1997). *An open software architecture for batch process simulation.* Computers & Chemical Engineering, vol. 21, Suppl., pp. 769-773.

FROESE, T. (1995). *Übergeordnete Regelung chemischer Verfahren mit Fuzzy Control und Künstlichen Neuronalen Netzen.* In ed. ZIMMERMANN, H.-J., Neuro + Fuzzy, VDI-Verlag, pp. 102-118.

FRÜH, K.F. and W. AHRENS (2004). *Handbuch der Prozessautomatisierung: Prozessleittechnik für verfahrenstechnische Anlagen.* Oldenbourg, München.

GANI, R., G. HYTOFT, C. JAKSLAND and A.K. JENSEN (1997). *An integrated computer aided system for integrated design of chemical processes.* Computers & Chemical Engineering, vol. 21, no. 10, pp. 1135-1146.

GANS, M. (1976). *The A to Z of plant startup.* CE, vol. 3, no. 15, pp. 72-82.

GEBHARDT, B. (1990). *Die blockorientierte Simulationssprache FSIMUL.* Springer Verlag, Berlin.

GEITNER, G.H. (2001). *Blockbibliothek Funktionsplan V.3.1. A&D Softwareguide.* Automation & Drives, vol. 2001/2002, pp. 38.

GEOFFRION, A.M. (1989). *Computer-based modeling environments.* European Journal of Operational Research, vol. 41, pp. 33-43.

GILLES, E.D., P. HOLL and W. MARQUARDT (1986). *Dynamic simulation of complex chemical processes - Dynamische Simulation komplexer chemischer Prozesse.* Chemie Ingenieur Technik, vol. 58, no. 4, pp. 268-278.

GILLES, E.D., P. HOLL, W. MARQUARDT, H. SCHNEIDER, R. MAHLER, K. BRINKMANN and K.H. WILL (1990). *Ein Trainingssimulator zur Ausbildung von Betriebspersonal in der chemischen Industrie.* In Automatisierungstechnische Praxis, vol. 32, pp. 343-350.

GOLDFARB, S. and T. BRADLEY (1995). *Process modelling for design, commissioning and operational analysis.* Computers & Chemical Engineering, vol. 19, Suppl., pp. 433-439.

GONG, J., G. HYTOFT and R. GANI (1995). *An integrated computer aided system for integrated design and control of chemical processes.* Computers & Chemical Engineering, vol. 19, Suppl., pp. 489-494.

GRACE, A.C.W. (1991). *SIMULAB, an integrated environment for simulation and control.* Proceedings of the 1991 American Control Conference, American Autom. Control Council, pp. 1015-1020.

HAHN, V. and G. NÖTH (1997). *Anforderungen an regelungstechnisches CAE in der chemischen Verfahrenstechnik.* In VDI/VDE-Jahrbuch, VDI-Verlag, Düsseldorf, pp. 260-280.

HEGNER, B. and H. SCHOENMAKERS (1985). *Chemasim- experience with BASF's simultaneous process simulator.* Inst. Chem. Eng. Symp. Ser., vol. 92, pp. 365-375.

HELLER, M., D. JAGER, M. SCHLUTER, R. SCHNEIDER and B. WESTFECHTEL (2004). *A management system for dynamic and interorganizational design processes in chemical engineering.* Computers & Chemical Engineering, vol. 29, no. 1, pp. 93-111.

HIRSCHBERG, H.G. (1999). *Handbuch Verfahrenstechnik und Anlagenbau - Chemie, Technik, Wirtschaftlichkeit.* Springer, Berlin.

HOLL, P. (1994). *Entwicklung und Einsatz eines dynamischen Simulators für verfahrenstechnische Prozesse.* VDI-Verlag, Düsseldorf.

HOLROYD, R. (1967). *Ultra large single stream chemical plants: Their advantages and disadvantages.* Chem. & Ind., vol. 5, pp. 1310-1315.

HOMMEL, M. (2006). *Parallelisierte Simulationsprozesse für virtuelles Prototyping in der Automobilindustrie.* Thesis, Department of Electrical Engineering and Information Technology, Brunswick, Technical University at Brunswick.

HOYER, M., C.S. HORN, R. SCHUMANN and G.C. PREMIER (2003a). *Integration of process and control simulation into the engineering process.* In eds. VERBRAECK, A. and V. HLUPIC, Proceedings of the 15th European Simulation Symposium, pp. 355-360.

HOYER, M., R. SCHUMANN and E. WÜST (2003b). *Design of a fresh-cheese separator control system based on an identified dynamic separator model.* IDF (International Dairy Federation) Bulletin: Pre-Summit Symposium on innovative research in dairy science and technology.

HOYER, M., R. SCHUMANN and G.C. PREMIER (2004). *Plant engineering process with integrated simulation.* Proceedings of the 1st ICI (International Conference on Informatics), Cesme, Izmir, pp. 1-6.

HOYER, M., R. SCHUMANN and G.C. PREMIER (2005a). *ModelCAT - a concept for model aggregation, simulation and control system design of chemical plant processes.* Proceedings of the GMA-Congress, VDI-Verlag, Baden-Baden, vol. 1883, pp. 893-900.

HOYER, M., R. SCHUMANN and G.C. PREMIER (2005b). *An approach for integrating process and control simulation into the plant engineering process.* In eds. PUIGJANER, L. and A. ESPUNA, European Symposium on Computer Aided Process Engineering (ESCAPE) -15, Elsevier Science, Barcelona, vol. 20, pp. 1603-1608.

HOYER, M., R. SCHUMANN and G.C. PREMIER (2005c). *ModelCAT - a model catalogue based approach to process modelling.* Proceedings of the 16th IFAC World Congress, Prague, Elsevier, pp. 285-290.

HOYER, M., R. SCHUMANN, E. WÜST and G.C. PREMIER (2005d). *Feedback control system design for a fresh cheese separator.* Proceedings of the 16th IFAC World Congress, Prague, Elsevier, pp.79-84.

HOYER, M., R. SCHUMANN and G.C. PREMIER (2006a). *ModelCAT: A model generation and simulation module for the ICACSD system.* Proceedings of the 14th IFAC Symposium on System Identification, Newcastle, Australia, Elsevier.

HOYER, M., R. SCHUMANN and G.C. PREMIER (2006b). *Industrial CACSD for the plant design process.* Proceedings of the IEEE Conference on Computer Aided Control System Design, Munich, pp. 1843-1848.

HYPERION (2003). *Hyperion - Dynamic analysis and training simulation business unit.* http://www.hyperion.com.cy/EN/index.html/

HYPROTECH (2002). *Hysys.* http://www.hyprotech.com/

INNOTEC (2002, 2005). *COMOS PT.* http://www.innotec.de/en/index.php/

INRIA (2007a). *Scicos - A dynamic system builder and simulator user's guide.* http://www.scilab.org/

INRIA (2007b). *Introduction to Scilab - User's guide.* http://www.scilab.org/

INTERGRAPH (2000). *Effiziente Integrationsmodelle.* In Interprocess, vol. 1, pp.1-7.

INTERGRAPH (2005). *SmartPlant.* http://www.intergraph.com/smartplant/default.asp/

INTERNATIONAL ELECTROTECHNICAL COMMISSION (2003). *IEC 61131-3,* Programmable controllers - guidelines for the application and implementation of programming languages.

INTERNATIONAL ELECTROTECHNICAL COMMISSION (2005). *CAEX IEC/PAS 62424,* Representation of process control engineering requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (1984). *ISO 3511-2:1984,* Process measurement control functions and instrumentation - Symbolic representation - Part 2: Extension of basic requirements.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (1994). *ISO 10303-1:1994.* Industrial automation systems and integration - Product data representation and exchange - Part 1: Overview and fundamental principles.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (2000). *Flow diagrams for process plants - General rules (ISO 10628:1997)*; German version EN ISO 10628:2000.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (2005). *ISO/DIS 10303-221.* Industrial automation systems and integration -- Product data representation and exchange -- Part 221: Application protocol: Functional data and their schematic representation for process plants.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (2006). *ISO 9241-110:2006.* Ergonomics of human-system interaction - Part 110: Dialogue principles.

INVENSYS-SIMSCI-ESSCOR (2002). *Pro II.* www.simsci-esscor.com/

JAAKO, J. (1998). *Aspects of process modeling.* Control Engineering Laboratory and Department of Process Engineering, University of Oulu, Oulu, Finland.

JARKE, M., T. LIST and J. KÖLLER (2000). *The challenge of process data warehousing.* Proceedings of the 26th Conference on Very Large Databases, Cairo, Aachen, pp. 473-483.

JARKE, M., M.A. JEUSFELD, C. QUIX and P. VASSILIADIS (1999). *Architecture and quality in data warehouses: An extended repository approach.* Information Systems, vol. 24, no. 3, pp. 229-253.

JENSEN, A.K. and R. GANI (1996). *A computer aided system for generation of problem specific process models.* Computers & Chemical Engineering, vol. 20, Suppl., pp. 145-150.

JOKINEN, P.A. (1996). *A life-cycle approach to automation information management in the process industries.* ISA Transactions, vol. 35, no. 4, pp. 285-296.

KAHLERT, J. (2006). *BORIS.* http://www.kahlert.com/web/english/e_set_home.php/

KARHELA, T. (2002). *A software architecture for configuration and usage of process simulation models software component technology and XML-based approach.* Thesis, Helsinki University of Technology, Helsinki.

KHEIR, N.A. (1988). *System modeling and computer simulation.* Marcel Dekker, Inc., New York, USA.

KIENDL, H. and J. KAHLERT (1991). *Dora-PC - Dortmunder Regelungstechnische Anwenderprogramme.* 7. Symposium Hagen, Vieweg-Verlag, Braunschweig, vol. 4, pp. 376-380.

KIM, W. (1990). *Introduction to object-oriented databases.* The MIT Press, Cambridge.

KÖRNER, S. (1999). *A structured approach to identification techniques for the analysis of industrial processes.* PhD Thesis, School of Technology, University of Glamorgan, Pontypridd.

KOTHE, W., P. KASSLER and D. SCHWARZ (1995). *Praxiserfahrung mit der dynamischen Anlagensimulation am Beispiel SPEEDUP.* In ed. SCHULER, H., Prozeßsimulation, VCH, Weinheim, pp. 259-290.

KRAUSE, H. (2003). *Emulator for the digital control system "Symphony Melody" of ABB.* Proceedings of the GMA-Congress, VDI Verlag GmbH, Düsseldorf, pp. 747-754.

KRÖNER, A. (2003). *Algorithmen zur Analyse und Lösung kontinuierlich-diskreter differentiell-algebraischer Prozessmodelle in der Simulationsumgebung DIVA.* VDI-Verlag, Düsseldorf.

KRÖNER, A., P. HOLL, M. W. and E.D. GILLES (1990). *DIVA - an open architecture for dynamic simulation.* Computers & Chemical Engineering, vol. 14, pp. 1289-1295.

LEE, S., I. JEONG and I. MOON (2000). *Development of evaluation algorithms for operator training system.* Computers & Chemical Engineering, vol. 24, no. 2-7, pp. 1517-1522.

LEVY, S., E. SUBRAHMANIAN, S.L. KONDA, R. COYNE, A. WESTERBERG and Y. REICH (1993). *An overview of the n-dim environment.* Technical Report EDRC-05-65-93.

LI, P.H.M. and J.O. GRAY (1993). *Graphical system input for design.* In ed. LINKENS, D. A., CAD for Control Systems, Marcel Dekker, New York, pp. 532-559.

LIEN, K. and T. PERRIS (1996). *Future directions for CAPE research perceptions of industrial needs and opportunities.* Computers & Chemical Engineering, vol. 20, Suppl., pp. 1551-1557.

LINNINGER, A.A. (2001). *Recent advances in process systems engineering.* Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference, Rediscovering Measurement in the Age of Informatics, IEEE, New York, pp. 1665-1670.

LINNINGER, A.A. and H. KRENDL (1999). *TechTool - Computer-aided generation of process models (Part 1 - A generic mathematical language).* Computers & Chemical Engineering, vol. 23, Suppl., pp. 703-706.

LJUNG, L. (1987). *System identification: Theory for the user.* Prentice Hall, Englewood Cliffs, NJ.

LUYBEN, L.L., B.D. TYRÉUS and M.L. LUYBEN (1998). *Plantwide process control.* McGraw-Hill, New York.

MAFFEZZONI, C. and R. GIRELLI (1998). *MOSES: Modular modelling of physical systems in an object-oriented database.* Mathematical and Computer Modelling of Dynamical Systems, vol. 4, no. 2, pp. 121-147.

MAFFEZZONI, C., L. FERRARINI and E. CARPANZANO (1999). *Object-oriented models for advanced automation engineering.* Control Engineering Practice, vol. 7, no. 8, pp. 957-968.

MANDLER, J.A. (2000). *Modelling for control analysis and design in complex industrial separation and liquefaction processes.* Journal of Process Control, vol. 10, no. 2-3, pp. 167-175.

MARQUARDT, W. (1992). *An object-oriented representation of structured process models.* Proceedings of the 8th European Symposium on Computer Aided Process Engineering (ESCAPE), vol. 1, Suppl., pp. 329-336.

MARQUARDT, W. (1994). *Numerical methods for the simulation of differential-algebraic process models.* In Process Engineering, Aachen, pp. 1-39.

MARQUARDT, W. (1996). *Trends in computer-aided process modeling.* Computers & Chemical Engineering, vol. 20, no. 6, pp. 591-609.

MARQUARDT, W. and M. NAGL (2004). *Workflow and information centered support of design processes--the IMPROVE perspective.* Computers & Chemical Engineering, vol. 29, no. 1, pp. 65-82.

MARQUARDT, W., L. VON WEDEL and B. BAYER (2000). *Perspectives on lifecycle process modeling*. Proceedings of the Fifth International Conference on Foundations of Computer-Aided Process Design, Amer Inst Chemical Engineers, New York, vol. 96, pp. 192-214.

MATHWORKS (2003a). *System identification toolbox/ Control system toolbox.* http://www.mathworks.com/

MATHWORKS (2003b). *Matlab/Simulink.* http://www.mathworks.com/

MATLEY, J. (1969). *Keys to successful plant startups.* Chemical Engineering, vol. 8, pp. 110-130.

MCGUIRE, M.L. and K. JONES (1989). *Maximizing the potential of process engineering databases.* Chemical Engineering Progress, vol. 85, no. 11, pp. 78-83.

MICHALEWICZ, Z. (1996). *Genetic algorithms + data structures = evolution programs.* Springer, Berlin.

MICROSOFT (2000). *MSDN Library Visual Studio 6.0.*

MICROSOFT (2003). *C++.* http://www.microsoft.com/

MITCHELL, E.E.L. and J.S. GAUTHIER (1976). *Advanced continuous simulation language (ACSL).* Simulation, pp. 72-78.

MODELICA ASSOCIATION (2007). *Modelica.* http://www.modelica.org/

NAJAFI, M., S. FURIC and R. NIKOUKHAH (2005). *SCICOS: A general purpose modeling and simulation environment.* Proceedings of the 4th International Modelica Conference, Hamburg, pp. 367-374. http://www.Modelica.org/events/Conference2005/

NATIONAL INSTRUMENTS (2003). *LabVIEW.* http://www.ni.com/labview/

NIELSEN, J. (1993). *Usability engineering.* Academic Press, Inc., London.

NILSSON, B. (1989). *Structured modelling of chemical processes - an object-oriented approach.* Thesis, Department of Automatic Control, Lund Institute of Technology, Lund.

NÖTH, G. (1998). *Randbedingungen für den Einsatz regelungstechnischer Methoden.* In VDI Berichte, no. 1397, pp. 87-94.

OLLERO, P. and C. AMSELEM (1983). *Decomposition algorithm for chemical process simulation.* Chemical Engineering Research, vol. 61, pp. 303-307.

OTTER, M. and H. ELMQVIST (2001). *Modelica: Language, libraries, tools, workshop and EU-project RealSim.* German Aerospace Center, pp. 1-8.

PASSINO, K.M. and S. YURKOVICH (1998). *Fuzzy control.* Addison-Wesley, Menlo Park.

PERKINS, J.D. and R.W.H. SARGENT (1982). *SPEEDUP: A computer program for steady-state and dynamic simulation and design of chemical processes.* Selected Topics in Computer-Aided Process Design and Analysis, AIChE Symposium, vol. 78, no. 214, pp. 1-11.

PERKINS, J.D., R.W.H. SARGENT, R. VÁZQUEZ-ROMAN and J.H. CHO (1996). *Computer generation of process models.* Computers & Chemical Engineering, vol. 20, no. 6, pp. 635-639.

PETERS, M.S., K.D. TIMMERHAUS and R.E. WEST (1991). *Plant design and economics for chemical engineers.* McGraw-Hill.

PIELA, P., T. EPPERLY, K. WESTERBERG and A. WESTERBERG (1991). ASCEND. *An object-oriented computer environment for modeling and analysis: The modeling language.* Computers & Chemical Engineering, vol. 15, no. 1, pp. 53-72.

PLASS, L. (2001). *Schlüsselfertig: Global engineering: Herausforderung für den Anlagenbau.* CHEMIE TECHNIK, vol. 30, no. 1, pp. 24-28.

POHL, K., R. DOMGES and M. JARKE (1996). *Tool integration support for modeling in open-CAPE environments.* Computers & Chemical Engineering, vol. 20, Suppl. 2, pp. 1613-1618.

PONTON, J. (1995). *Process systems engineering: Halfway through the first century.* Chemical Engineering Science, vol. 50, no. 24, pp. 4045-4059.

PREISIG, H. (1996). *Computer-aided modelling: Two paradigms on control.* Computers & Chemical Engineering, vol. 20, Suppl., pp. 981-986.

PROSIM (2002). *PROSIM Plus.* www.prosim.net/

PSE (2003a). *gO:Simulink: User's guide for gProms v2.2, v1.1.* Process Systems Enterprise Ltd., London. http://www.psenterprise.com/gPROMS/

PSE (2003b). *gPROMS v2.3.1 Introductory user guide.* PSE.

PSE (2006). *gPROMS v2.3.6.* http://www.psenterprise.com/

RAUPRICH, G., C. HAUS and W. AHRENS (2002). *CAE for process control - integration of different trades and plant maintenance.* PLT-CAE - Integration in gewerkeübergreifendes Engineering und Plant- Maintenance. atp, vol. 44, no. 2, pp. 50-62.

RÖMER, M. and K. BENDER (2002). *The certification test of PROFIBUS-PA-devices.* atp, vol. 8, pp. 38-44.

SALES-CRUZ, M. and R. GANI (2003). *A modelling tool for different stages of the process life.* In Dynamic Model Development: Methods, Theory and Applications, Elsevier Science B.V., Amsterdam, vol. 16, pp. 209-249.

SARGENT, R.W.H. and A.W. WESTERBERG (1964). *SPEED-UP in chemical engineering design*. Trans. Inst. Chem. Eng., vol. 42, pp. 190-197.

SATTLER, K. and W. KASPER (2000a). *Verfahrenstechnische Anlagen - Planung, Bau und Betrieb*. Wiley-VCH, Weinheim.

SATTLER, K. and W. KASPER (2000b). *Verfahrenstechnische Anlagen*, Band 2. Wiley-VCH, Weinheim.

SCHARWAECHTER, H., L. VON WEDEL, A. YANG and W. MARQUARDT (2002). *A tool integration framework for dynamic simulation in process engineering*. Proceedings of the 15th Int. Fed. Aut. Control (IFAC) World Congress, Barcelona.

SCHLÜTER, M. and E. HABERSTROH (2002). *Design of twin screw-extruders with the MOREX-simulation-software*. Proceedings of the 18th Polymer Processing Society Conference, Guimaraes, Portugal.

SCHNEIDER, R. (2003). *Grundlagen der Simulationstechnik*. Manuscript. Aachen, pp. 1-136.

SCHULER, H. (2004). *Prozessleittechnik im Informationsverbund des Unternehmens*. In eds. FRÜH, K. F. and U. MAIER, Handbuch der Prozessautomatisierung-Prozessleittechnik für verfahrenstechnische Anlagen, Oldenbourg Industrieverlag, München, pp. 12-28.

SCHUMANN, R. (2007). *Virtual commissioning - concept and realisation approaches*. Proceedings of the VDI/VDE GMA Congress, VDI, Baden-Baden, to be published.

SCHUMANN, R., S. KÖRNER, K.J. BAKER and M. STRICKRODT (1996). *Shaping CACSD for Practical Use in Process Industry*. Proceedings of the 13th World Congress of IFAC, San Francisco, vol. L, pp. 223-228.

SEIDER, W.D., J.D. SEADER and D.R. LEWIN (1999). *Process design principles*. John Wiley & Sons, New York.

SEYBOLD, H.J. (2000). *Simulation reduces costs*. IT & Production, vol. 1, pp. 14-16.

SHNEIDERMAN, B. (2001). *User interface design*. mitp-Verlag, Bonn.

SINNOTT, R.K., C.J. M. and R.J. F. (2005). *Chemical engineering design*. Elsevier Butterworth-Heinemann, Amsterdam.

SOFTANY (2006). *WinCHM*. www.softany.com/winchm/

STAWARZ, A.C. and B. SOWERBY (1995). *Cost effective operator training*. Computers & Chemical Engineering, vol. 19, Suppl. 1, pp. 459-464.

STEPHANOPOULOS, G., G. HENNING and H. LEONE (1990). *MODEL.LA. A modeling language for process engineering, I.* The formal framework. Computers & Chemical Engineering, vol. 14, no. 8, pp. 813-846.

STEPHANOPOULOS, G., J. JOHNSTEN, T. KRITICOS, R. LAKSHMANAN, M. MAVROVOUNIOTIS and C. SILETTI (1987). *Design-Kit: An object-oriented environment for process engineering.* Computers & Chemical Engineering, vol. 11, Suppl., pp. 655-674.

STRAUSS, J.C. (1967). *The SCi continuous system simulation language (CSSL).* Simulation, vol. 9, pp. 281-303.

STRICKRODT, M. (1997). *An integrated knowledge engineering approach to process modelling.* PhD Thesis, University of Glamorgan, Pontypridd.

SULC, B. (2004). *Some tools for simplified modelling for power plant operator training simulators.* In ed. HAMZA, M. H., Proceedings of the 23rd IASTED International Conference on Modelling, Identification and Control MIC 2004, ACTA Press, Grindelwald, Switzerland, pp. 294-299.

SUNDQUIST, S., J. ILME, MAATTA and I. TURUNEN (2000). *Using models in different stages of process life-cycle.* Computers & Chemical Engineering, vol. 24, no. 2-7, pp. 1253-1259.

SYAMLAL, M. (2003). *Development of technologies and analytical capabilities for vision 21 energy plants.* Cooperative agreement no de-fc26-00nt40954. Fluent Inc., Morgantown.

SYSKA, B. (2004). *Structured approach to industrial control system design.* Thesis, School of Technology, University of Glamorgan, Pontypridd.

TETRA PAK (1995). *Dairy processing handbook.* Tetra Pak Processing Systems AB, Lund.

THOMAS, F. (2005). *Wie alles begann. Geschichte der Elektronic Design Automation.* Design & Elektronik, vol. 9, pp. 100-102.

TOLSMA, J.E., J.A. CLABAUGH and P.I. BARTON (2002). *Symbolic incorporation of external procedures into process modeling environments.* Ind. Eng. Chem. Res., vol. 41, pp. 3867-3876.

TRÄNKLE, F., M. ZEITZ, M. GINKEL and E.D. GILLES (2000). *PROMOT: A modeling tool for chemical processes.* Mathematical and Computer Modelling of Dynamical Systems, vol. 6, no. 3, pp. 283-307.

UPADHYE, R.S. and E.A. GRENS (1972). *An efficient algorithm for optimum decomposition of recycle systems.* AIChE Journal, vol. 18, no. 3, pp. 533-539.

VAN DEN BOSCH, P.P.J. (1993). *PSI: An established block-oriented simulation program.* In ed. LINKENS, D. A., CAD for Control Systems, Marcel Dekkar Inc., New York, pp. 195-212.

VAN DER MEER, M. (2002). *Auswahl von Simulationssoftware als Hilfsmittel zur Planung, Auslegung und Automatisierung verfahrenstechnischer Prozesse.* Diploma Thesis, Department of Mechanical Engineering, University of Applied Sciences and Arts, Hanover.

VAN SCHIJNDEL, J. and S. PISTIKOPOULOS (2000). *Towards the integration of process design, process control and process operability: Current status and future trends.* Proceedings of the AIChE Symp. Series 323, vol. 96, pp. 99-112.

VDI/VDE (1995). *VDI/VDE 3968 - Herstellerneutrale Konfigurierung von Prozessleitsystemen.* Beuth Verlag, Düsseldorf.

VON WEDEL, L. and W. MARQUARDT (2000). *ROME: A repository to support the integration of models over the lifecycle of model-based engineering process.* In Computer Aided Process Engineering, vol. 10, pp. 535-540.

VON WEDEL, L. (2003). *An environment for heterogeneous model management in chemical process engineering.* Thesis, Faculty of Mechanical Engineering, RWTH Aachen, Aachen.

WEBER, K.H. (2002). *Inbetriebnahme verfahrenstechnischer Anlagen - Praxishandbuch mit Checklisten und Beispielen.* Springer-Verlag, Heidelberg.

WEBER, K.H. (2006). *Inbetriebnahme verfahrenstechnischer Anlagen - Praxishandbuch mit Checklisten und Beispielen.* Springer-Verlag, Heidelberg.

WESTERBERG, A.W. (1979). *Process flowsheeting.* Univ. Press, Cambridge.

WESTERBERG, A.W., E. SUBRAHMANIAN, Y. REICH and S. KONDA (1997). *Designing the process design process.* Computers & Chemical Engineering, vol. 21, Suppl., pp. 1-9.

WESTERWEELE, M.R., H.A. PREISIG and M. WEISS (1999). *Concept and design of Modeller, a computer-aided modelling tool.* Computers & Chemical Engineering, vol. 23, Suppl., pp. 751-754.

WILLIS, M.J. (1999). *Proportional-Integral-Derivative control.* Dept. of Chemical and Process Engineering, University of Newcastle, Newcastle.

YE, N., S. VALLURI, M. BARKER and P.-Y. YU (2000). *Integration of advanced process control and full-scale dynamic simulation.* ISA Transactions, vol. 39, no. 2, pp. 273-280.

YLÉN, J.-P., M. PALJAKKA, T. KARHELA, S. J. and J. K. (2005). *Experiences on utilising plant scale dynamic simulation in process industry.* Proceedings of the 19th European Conference on Modelling and Simulation (ECMS), Riga, Latvia, pp. 685-690.

ZHEN-YU, C. (1998). *Low cost automation.* IFAC Symposium on Computer Aided Design and Control Systems, Pergamon Press, Shenyang.

Appendix of Thesis

This Appendix contains additional information about selected topics of this PhD-Thesis:

- "Appendix A - Published Papers" (page 187)
- "Appendix B - Source Code of FSCS Models" (see enclosed compact disc)
- "Appendix C - Entire Simulations Scripts" (see enclosed compact disc), including the entire gPROMS™ script, the gSF script and the entire Simulink™ script

Appendix A - Published Papers

All published papers (2003 to 2006) are listed within this Appendix. This Appendix is structured as follows:

1. HOYER, M., C.S. HORN, R. SCHUMANN and G.C. PREMIER (2003a). *Integration of process and control simulation into the engineering process.* In eds. VERBRAECK, A. and V. HLUPIC, Proceedings of the 15th European Simulation Symposium, pp. 355-360.
2. HOYER, M., R. SCHUMANN and E. WÜST (2003b). *Design of a fresh-cheese separator control system based on an identified dynamic separator model.* IDF (International Dairy Federation) Bulletin: Pre-Summit Symposium on innovative research in dairy science and technology.
3. HOYER, M., R. SCHUMANN and G.C. PREMIER (2004). *Plant engineering process with integrated simulation.* Proceedings of the 1st ICI (International Conference on Informatics), Cesme, Izmir, pp. 1-6.
4. HOYER, M., R. SCHUMANN and G.C. PREMIER (2005a). *Model^{CAT} - a concept for model aggregation, simulation and control system design of chemical plant processes.* Proceedings of the GMA-Congress, VDI-Verlag, Baden-Baden, vol. 1883, pp. 893-900.
5. HOYER, M., R. SCHUMANN and G.C. PREMIER (2005b). *An approach for integrating process and control simulation into the plant engineering process.* In eds. PUIGJANER, L. and A. ESPUNA, European Symposium on Computer Aided Process Engineering (ESCAPE) -15, Elsevier Science, Barcelona, vol. 20, pp. 1603-1608.
6. HOYER, M., R. SCHUMANN and G.C. PREMIER (2005c). *Model^{CAT} - a model catalogue based approach to process modelling.* Proceedings of the 16th IFAC World Congress, Prague, Elsevier, pp. 285-290.
7. HOYER, M., R. SCHUMANN, E. WÜST and G.C. PREMIER (2005d). *Feedback control system design for a fresh cheese separator.* Proceedings of the 16th IFAC World Congress, Prague, Elsevier, pp.79-84.
8. HOYER, M., R. SCHUMANN and G.C. PREMIER (2006a). *Model^{CAT}: A model generation and simulation module for the ICACSD system.* Proceedings of the 14th IFAC Symposium on System Identification, Newcastle, Australia, Elsevier.
9. HOYER, M., R. SCHUMANN and G.C. PREMIER (2006b). *Industrial CACSD for the plant design process.* Proceedings of the IEEE Conference on Computer Aided Control System Design, Munich, pp. 1843-1848.