



Systematic Methods and Tools for Computer Aided Modelling

Fedorova, Marina; Gani, Rafiqul; Sin, Gürkan

Publication date:
2015

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Fedorova, M., Gani, R., & Sin, G. (2015). Systematic Methods and Tools for Computer Aided Modelling. Kgs. Lyngby: Technical University of Denmark (DTU).

DTU Library

Technical Information Center of Denmark

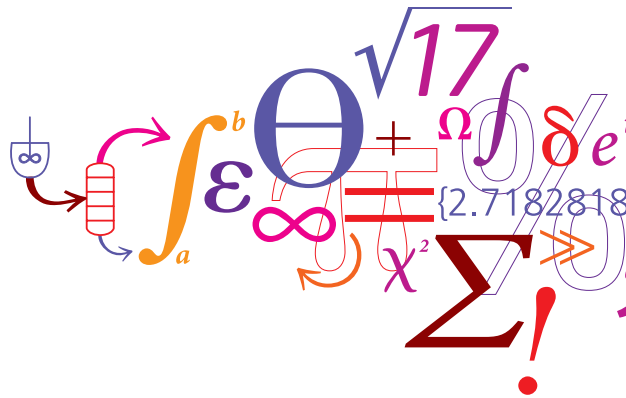
General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Systematic Methods and Tools for Computer Aided Modelling



Marina Fedorova

PhD Thesis

March 2015

Systematic Methods and Tools for Computer Aided Modelling

PhD Thesis

Marina Fedorova

March, 2015

CAPEC-PROCESS Research Center
Department of Chemical & Biochemical Engineering
Technical University of Denmark

PREFACE

This thesis is submitted as partial fulfilment of the requirements for the Doctor of Philosophy (PhD) degree at the Technical University of Denmark (DTU). The PhD project was carried out at the CAPEC-PROCESS center at the Department of Chemical and Biochemical Engineering from April 2012 to March 2015 under the guidance of Professor Rafiqul Gani as the main supervisor and Associate Professor Gürkan Sin as the co-supervisor.

First of all, I would like to express my gratitude to my supervisors Rafiqul Gani and Gürkan Sin for their support, enthusiasm, inspiring ideas and valuable criticism. I am grateful for all amazing opportunities I've been given and for all their time and patience spent on me.

I would like to thank Professor Mauricio Sales-Cruz from Universidad Autonoma Metropolitana (UAM), Mexico, for giving me opportunity to come for the external research stay, for all his hospitality and support.

I thank all my co-workers and friends at CAPEC and PROCESS, especially Hande, Michele, Miguel and my best officemate Peam, for becoming part of my life and part of my family.

Special thanks to my teammates from DTU basketball club, to all girls, guys and my very best coaches Giannis and Matias for numerous hours spent together on and outside the court. Thank you for helping me to forget about my thesis sometimes.

I couldn't do much without endless support from my family: my parents, my grandparents and my sister. They always believed in me, even when I did not believe myself. I am grateful for everything they did for me.

And finally, I would like to thank Manolis for being always on my side, for all love, support and patience

ABSTRACT

Models are playing important roles in design and analysis of chemicals/bio-chemicals based products and the processes that manufacture them. Model-based methods and tools have the potential to decrease the number of experiments, which can be expensive and time consuming, and point to candidates, where the experimental effort could be focused.

In this project a general modelling framework for systematic model building through modelling templates, which supports the reuse of existing models via its new model import and export capabilities, have been developed. The new feature for model transfer has been developed by establishing a connection with an external modelling environment for code generation.

The main contribution of this thesis is a creation of modelling templates and their connection with other modelling tools within a modelling framework. The goal was to create a user-friendly system, which will make the model development process easier and faster and provide the way for unified and consistent model documentation. The modeller can use the template for their specific problem or to extend and/or adopt a model. This is based on the idea of model reuse, which emphasizes the use of a model not only for one specific application but also for future applications involving different needs and levels of detail to match different purposes. As the result the model developer can generate and test models systematically, efficiently and reliably. In this way, development of products and processes can be faster, cheaper and very efficient.

The developed modelling framework involves five main elements: 1) a modelling tool, that includes algorithms for model generation; 2) a template library, which provides building blocks for the templates (generic models previously developed); 3) computer aided methods and tools, that include procedures to perform model translation, model analysis, model verification/validation, model solution and model documentation; 4) model transfer – export/import to/from other application for further extension and application – several types of formats, such as XML-format and COM-objects, are incorporated to allow the export and import of mathematical models; 5) a user interface that provides the work-flow and data-flow to guide the user through the different modelling tasks.

RESUME PÅ DANSK

Modeller spiller vigtige roller til design og analyse af kemi- og biokemibaserede produkter samt til processerne, der fremstille dem. Modelbaserede metoder og værktøjer har potentialet til at formindske antallet af eksperimenter, som kan være dyre og tidskrævende, og til at udvælge kandidater, hvorpå den eksperimentelle indsats bør fokuseres.

I dette projekt blev en generel modelleringsramme udviklet til en systematisk modelopsætning ved hjælp af modelskabeloner. Modelrammen understøtter en ny funktion til genbrug af eksisterende modeller via mulighed for modelimport og -eksport. Denne nye funktion til modeloverførsel var udviklet ved at etablere en forbindelse med et eksternt modelleringsmiljø til kodegenerering.

Denne ph.d.-afhandlings væsentligste bidrag er udvikling modelleringskabeloner og deres forbindelse med andre modelleringsværktøjer inden for en modelleringsramme. Projektets mål var at udvikle et brugervenligt system, som kan lave modeludviklingsprocessen lettere og hurtigere, og at give en fremgangsmåde for ensartet og sammenhængende modeldokumentation. En modelkonstruktør kan anvende en skabelon til sit specifikke problem eller udvide og/eller genbruge en model. Dette koncept er baseret på ideen om modelgenbrug, som understreger at modelbrug ikke kun er til én specifik anvendelse, men også til fremtidige anvendelser, som involverer forskellige behov og detaljeringsgrader til at imødekomme forskellige formål. Resultatet er, at modelkonstruktøren kan generere modeller på en systematisk, effektiv og pålidelig måde. Derfor kan produkt- og procesudviklingen forekomme hurtigere, billigere og meget effektiv.

Den udviklede modelleringsramme involverer fem hovedelementer: 1) et modelleringsværktøj, som inkluderer algoritmer for modeludvikling; 2) et bibliotek af modelskabeloner, som giver byggesten til skabelonerne; 3) Computerunderstøttede modelleringsmetoder og -værktøjer, der omfatter procedurer til at udføre modeloversættelse, modelanalyse, modelverifikation og -validering, modelopløsning og modeldokumentation; 4) modeloverførsel – import/eksport til/fra andre programmer til viderebygning og anvendelse – flere formatter, såsom XML-format og COM-objects, er inkorporeret til at tillade importen og eksporten af matematiske modeller; 5) en brugergrænseflade, som giver workflow og dataflow til at vejlede brugeren gennem de forskellige modelleringsopgaver.

CONTENTS

PREFACE	iii
ABSTRACT	iv
RESUME PÅ DANSK.....	v
CONTENTS	vi
LIST OF FIGURES.....	x
LIST OF TABLES.....	xiv
1 INTRODUCTION	15
1.1 Introduction to computer-aided modelling	16
1.2 Overview of existing methodologies and tools.....	18
1.2.1 Common modelling approaches	18
1.2.2 Novel approaches for model representation	20
1.2.3 Tools integration.....	22
1.3 Work done prior to this project.....	23
1.3.1 ICAS-MoT.....	23
1.3.2 ICAS-ModDev	23
1.3.3 Contribution of this project.....	24
1.4 Objectives of the thesis	24
1.5 Structure of the thesis	24
2 CONCEPT OF TEMPLATE-BASED MODELLING.....	25
2.1 Template-creation	29
2.2 Template-use.....	33
3 COMPUTER AIDED TEMPLATE-BASED MODELLING FRAMEWORK	35
3.1 Stage 1. Modelling objective	38
3.2 Stage 2. Model reuse.....	40
3.2.1 Model library	41
3.2.2 Template application computer-aided tool.....	42
3.3 Stage 3. Model generation	42
3.4 Stage 4. Model construction and analysis.....	45
3.5 Stage 5. Model application	48
3.5.1 Simulation.....	48

3.5.2	Optimization	50
3.6	Stage 6. Model storage.....	52
3.6.1	Add model to the model library.....	53
3.6.2	Template library update.....	53
3.7	Integration of the tools within framework	55
3.8	Import/export of the models	56
4	FRAMEWORK IMPLEMENTATION	58
4.1	MoT – Model construction and analysis tool	59
4.2	ModDev – Model generation tool.....	61
4.3	Template tool – software implementation as ModTem	62
4.3.1	Template-creation framework	62
4.3.2	Template-use framework.....	66
4.4	Model library	67
4.5	XML transfer for tools integration and export/import of the models	68
4.5.1	MOSAIC modelling environment	69
4.5.2	Implementation of the extended features through connection with MOSAIC.....	70
4.6	Other transfer possibilities	71
5	CASE STUDIES	72
5.1	Catalytic membrane fixed bed reactor	73
5.1.1	Template-creation.....	74
5.1.1.1	Step 1. Collecting system information.	74
5.1.1.2	Step 2. Model decomposition.....	76
5.1.1.3	Step 3. Check if template is available	80
5.1.1.4	Step 4B. Create new template	80
5.1.1.5	Step 5B. Derive parts of the model	80
5.1.1.6	Step 6. Provide template description.....	80
5.1.1.7	Step 7. Provide description of the blocks.....	81
5.1.1.8	Step 8. Define connection between blocks	81
5.1.1.9	Step 9. Check template consistency	82
5.1.1.10	Step 10. Finalization of the template.....	88
5.1.2	Template-use	88
5.1.2.1	Scenario 1. Creation of model through template options.....	88
5.1.2.2	Scenario 2. Creation of the model through additional template options	90
5.1.3	Case study conclusion	92
5.2	Oxidation of unsaturated fatty acid with hydrogen peroxide.....	93
5.2.1	Template-creation.....	93
5.2.1.1	Step 1. Collecting system information	93
5.2.1.2	Step 2. Decomposition of the model.....	94

5.2.1.3	Step 3. Check if template is available	95
5.2.1.4	Step 4B. Create new template	96
5.2.1.5	Step 5B. Derive parts of the model	96
5.2.1.6	Step 6. Provide template description.....	96
5.2.1.7	Step 7. Provide description of the blocks.....	96
5.2.1.8	Step 8. Define connection between blocks.	96
5.2.1.9	Step 9. Check template consistency	97
5.2.1.10	Step 10. Finalization of the template.....	99
5.2.2	Template-use	100
5.2.3	Model analysis and verification.....	101
5.2.4	Template updates.....	103
5.2.4.1	Step 1. Collecting system information	103
5.2.4.2	Step 2. Decomposition of the model.....	103
5.2.4.3	Step 3. Check if template is available	103
5.2.4.4	Step 4A. Choose template for updates	103
5.2.4.5	Step 5A. Derive new parts of the model	103
5.2.4.6	Step 6. Update template description.....	103
5.2.4.7	Step 7. Provide description of the blocks.....	104
5.2.4.8	Step 8. Define connection between blocks.	104
5.2.4.9	Step 9. Check template consistency	104
5.2.4.10	Step 10. Finalization of the template.....	105
5.2.5	Case study conclusion	105
5.3	Batch reactor and distillation	106
5.3.1	Case study conclusion	110
5.4	Flowsheet template	111
5.4.1	Modelling objective.....	111
5.4.2	Model generation.....	111
5.4.2.1	System decomposition	111
5.4.2.2	Description of the building blocks	112
5.4.2.3	Aggregation of the building blocks – final model	113
5.4.3	Model analysis of generated model.....	114
5.4.4	Template creation	114
5.4.4.1	Step 1. Collect system information	115
5.4.4.2	Step 2. Decomposition of the model	115
5.4.4.3	Step 3. Check if template is available	116
5.4.4.4	Step 4B. Create new template	116
5.4.4.5	Step 5B. Derive parts of the model	116
5.4.4.6	Step 6. Provide template description.....	117
5.4.4.7	Step 7. Provide description of the blocks.....	117
5.4.4.8	Step 8. Define connections between blocks.....	117
5.4.4.9	Step 9. Check template consistency	117

5.4.4.10	Step 10. Finalization of the template.....	119
5.4.5	Template use for new model development.....	119
5.4.6	Williams-Otto plant model analysis and solution.....	120
5.4.7	Case study conclusion.	121
6	CONCLUSION AND FUTURE WORK.....	123
6.1	Main achievements of the thesis	123
6.2	Overview of the potential future improvements	124
6.2.1	Modelling methodology and computer-aided framework improvements 124	
6.2.2	Software implementation improvements.....	124
7	APPENDIX	126
7.1	Example of the model in XML format	126
7.2	Batch reactor and distillation	128
7.2.1	Model implementation in MoT	128
7.2.2	Generated gProms code.....	129
7.2.3	Generated Matlab code.....	132
7.3	Flowsheet template	136
7.3.1	Flowsheet model transferred from ModDev	136
7.3.2	Flowsheet model implemented in MoT.....	137
7.3.3	Williams-Otto plant model based on the template	137
	NOMENCLATURE	139
	Greek letters.....	141
	Acronyms	141
	REFERENCES	143

LIST OF FIGURES

Figure 2.1. Template options for the flash separator	26
Figure 2.2. Structure of hierarchy in the modelling template	26
Figure 2.3. Main classes of the template structure and examples of the instances.	27
Figure 2.4. Template-creation workflow	31
Figure 2.5. Updated template options for the flash separator with one more option describing phase equilibrium compared to previous version (Figure 2.1)	32
Figure 2.6. Template-use workflow	34
Figure 3.1. Computer-aided modelling framework	35
Figure 3.2. Stages of modelling methodology	37
Figure 3.3. Work-flow (boxes on the left hand side) and data-flow (arrow boxes on the right hand side) for Stage 1: Modelling objective (adopted from Heitzig, 2011)	39
Figure 3.4. Computer-aided workflow for the modelling stage 1: Modelling objective	40
Figure 3.5. Computer-aided modelling framework highlighted parts related to model reuse	41
Figure 3.6. Computer-aided workflow for modelling Stage 2: Model reuse, highlighting automated feature.	42
Figure 3.7. The fundamental modelling objects containing instances of the model equations that define the model	43
Figure 3.8. Computer-aided modelling framework with highlighted part for model generation	44
Figure 3.9. Computer-aided modelling workflow with automated features for the modelling Stage 3: Model generation (based on Jensen (1999))	45
Figure 3.10. Work-flow (boxes on the left hand side) and data-flow (arrow boxes on the right hand side) for Stage 4: Model construction and analysis (based on Heitzig, 2011)	46
Figure 3.11. Computer-aided modelling framework with highlighted part for model construction and numerical analysis	47

Figure 3.12. Computer-aided modelling workflow with automated features for the modelling Stage 4: Model construction and analysis (adopted from Heitzig (2011)).....48

Figure 3.13. Work-flow (boxes on the left hand side) and data-flow (arrow boxes on the right hand side) for Stage 5: Model application - Simulation (adopted from Heitzig, 2011).....49

Figure 3.14. Required features, support and automation potential for the modelling Stage 5A: Model application - Simulation50

Figure 3.15. Work-flow (boxes on the left hand side) and data-flow (arrow boxes on the right hand side) for Stage 5: Model application - Optimization (adopted from Heitzig, 2011).....51

Figure 3.16. Computer-aided modelling workflow with automated features for the modelling Stage 5B: Model application - Optimization52

Figure 3.17. Computer-aided modelling workflow with automated features for modelling Stage 6: Model storage – Template creation55

Figure 3.18. Schematic structure of the model export/import from/to modelling framework57

Figure 4.1. Computer-aided modelling framework with corresponding software tools58

Figure 4.2. MoT screens with constructed and translated model59

Figure 4.3. Architecture of the connection between model-solver-results within ICAS-MoT60

Figure 4.4. Solver options in MoT for dynamic (left) and algebraic (right) systems60

Figure 4.5. Demonstration of the results in ICAS-MoT.....61

Figure 4.6. Main working area of ModDev – described system on the left-hand side and generated equations on the right-hand side.....62

Figure 4.7. Starting screen of the template application tool63

Figure 4.8. Screen for template creation.....63

Figure 4.9. Screen for template update.....64

Figure 4.10. Upper-layer of system information including two sub-layers: number of volumes and number of scales, each of them contains two blocks, representing template options. Green (light-coloured) blocks indicate currently chosen options, blue (dark-coloured) shows the options of the template that are available.64

Figure 4.11. Block information menu in the template-creation tool.65

Figure 4.12. Screen of template use.....	66
Figure 4.13. Schematic architecture of template use part of the software tool	67
Figure 4.14. Screenshot of model library	67
Figure 4.15. ModDev screenshot with highlight of the XML transfer.....	68
Figure 4.16. Template application (ModTem) screenshot with highlight of the XML transfer.....	69
Figure 4.17. MoT screenshot with highlighted XML transfer option	69
Figure 4.18. Model transfer interface and general structure of MOSAIC.....	70
Figure 4.19. Model transfer interface and general structure of MOSAIC.....	71
Figure 5.1. Conceptual schema of the integrated reactor proposed by Abo-Ghander et al. (2012)	74
Figure 5.2. General structure for the catalytic membrane fixed bed reactor template.....	82
Figure 5.3. Model for scenario 1 represented in ICAS-MoT: model equations and plots of temperature on the dehydrogenation side and styrene production.	90
Figure 5.4. Model results for scenario 2 represented in ICAS-MoT: plots of styrene production (left) and temperature profile along the reactor (right).	91
Figure 5.5. Schematic two-phase fed-batch reactor.....	93
Figure 5.6. General structure for the unsaturated fatty acid oxidation template.	97
Figure 5.7. Model prediction compared to experimental data by Papadakis (2013).....	102
Figure 5.8. State-task network for batch reactor and batch distillation.	106
Figure 5.9. Model implementation in MoT	107
Figure 5.10. Schematic architecture of model implementation in ICAS modelling framework and its translation to external tools.....	108
Figure 5.11. Translated model in the MOSAIC modelling environment.....	108
Figure 5.12. Generated code for gPROMS (up) and Matlab (down)	109
Figure 5.13. Resulting plots in Matlab and gProms, showing water flowrate after distillation unit.	110
Figure 5.14. Schematic flowsheet for methyl acetate production.	111
Figure 5.15. Implementation of the flowsheet in ModDev	112
Figure 5.16. Schematic flowsheet for methyl acetate production.	112
Figure 5.17. Equations describing the reactor in the system (screenshot from ModDev).....	113

LIST OF FIGURES

Figure 5.18. Schematic flowsheet for methyl acetate production.	114
Figure 5.19. Flow diagram for the Williams-Otto plant (adapted from Biegler et al., 1997).....	119

LIST OF TABLES

Table 1.1 Overview of some tools and their modelling approaches.....	19
Table 4.1 Mapping of variable types in ICAS-MoT and MOSAIC	71
Table 5.1. Reaction present in the two-phase oxidation system (Köckritz and Martin, 2008)	94
Table 5.2. Addition reactions occurring in the single-phase oxidation system	100
Table 5.3. Simulation results of the Williams-Otto plant model	121

1 INTRODUCTION

Models are playing important roles in design and analysis of chemicals/bio-chemicals based products and the processes that manufacture them. These model-based methods and tools have the potential to decrease the number of experiments, which can be expensive and time consuming, and point to candidates where the experimental effort could be focused.

As the required models may be complex and require multiple time and/or length scales, their development and application for product-process design is not trivial. Therefore, the modelling framework can contribute by reducing the time and resources needed for model development and application, thereby reducing the overall time and cost for product-process development. The idea here is to use the modelling systems to either develop new models for final verification by experiments, or to evaluate and analyse existing model-based systems to identify the possibilities for improvements and/or adoption.

This PhD-project deals with the development of a structured computer-aided modelling framework that integrates modelling tools used for different purposes. This development aims to improve the model development process and make it more efficient.

Some definitions of important terms used in this thesis are given below:

Model – “The representation of a real or virtual physico-chemical, economic, social or human situation, in an alternate mathematical or physical form, for an envisaged purpose” (Cameron and Gani, 2011)

Modelling methodology – Methodology for the process of model development and application, that is, representation of the modelling process in terms of an ordered set of tasks and sub-tasks.

Work-flow – A work-flow summarizes the different steps required to complete a given task.

Computer-aided modelling framework – An architecture, through which the computer-aided methods and tools can be implemented and used according to the work-flow of the methodology.

Modelling tool – Actual implementation of computer-aided modelling framework.

Modelling template – A structure that combines parts of decomposed models and is used to generate various combinations of these parts in order to obtain new modelling objects.

1.1 Introduction to computer-aided modelling

In general, chemical and bio-chemical processes can be characterized by a set of chemical transformations that are connected by various process phenomena. This results in the often complex mathematical models to describe all system balances and take into consideration kinetics, transport, heat transfer etc. mathematical model of a system is a collection of relations among variables representing certain properties of the system measured in particular domains (Yang and Marquardt, 2009). This model can be presented as a number of algebraic equations (AE), ordinary differential equations (ODE), differential-algebraic equations (DAE), partial differential equations (PDE) or partial differential-algebraic equations (PDAE). In general, the model equations can be divided into three main classes of equations (Cameron and Gani, 2011):

- balance equations – mass, energy and/or momentum equations;
- constitutive equations relating intensive variables (temperature, pressure, composition) to constitutive variables (enthalpies, reaction rates etc.);
- connection and conditional equations relating surroundings-system connections, connections between system volumes or phases, summation of mole fractions etc.

By solving a model, one can predict values or distribution of values of some properties of the system (Yang and Marquardt, 2009). As more complex the equation systems are, as more time and resources are needed for their solution. Therefore, the use of software tools in these cases is necessary.

A process model of reliable quality allows predicting process behaviour and based on this, to optimize the process and to improve understanding of the domain system. However, the modelling technology also requires the necessary methods and tools to perform these functions. If the construction of models is mainly manual, it is still not very effective. Manually developed models usually can be applied only for the specific problem for which it was developed, such models are often not well documented and, therefore, difficult to maintain. Therefore, computer aided modelling methodologies are gaining increasing attention in model-based studies of chemical and bio-chemical process-product design, as they can contribute to reducing the time and resources needed for model development and model-based applications by tackling model complexity and, at the same time, providing expert support during model development. In order to provide the most beneficial way for employing these advantages, the computer aided modelling methodologies can be implemented in a specific computer aided modelling framework that provides architecture for using the workflows and dataflows of the methodology through modelling tools (Heitzig, 2011). By using this modelling framework, the user will get a model structure, guidance and support during model development and model application and, therefore, efficiency of the modelling process will increase, modelling errors will be reduced and the quality and reliability of the models will be improved (Mošat et al., 2013; Preisig, 2010; Kuntsche et al., 2011).

Although there are a variety of commercially available modelling tools and software, the effort spent for modelling is still one of the most time-consuming factors in an industrial project involving model-based process engineering techniques (Foss et al, 1998). This is especially true when there is a lack of appropriate models in the model-library of commercially available software to describe a process or unit operation. Hence to support problem specific model development, there is a need for generic model development tools or domain-specific templates. Heitzig (2011) points to the importance of the systematic framework for model development and the need for implementation of templates for modelling cases and creation of the template library.

The model reuse still remains very limited and significant loss of information can occur within organization with people's leave (Cameron and Ingram, 2008). Also, model reuse often occurs simply by copying parts of a modeller's previous work and the exchange of models is still rare (Foss et al., 1998).

Thus, the reuse of existing modelling codes is minimal and every new modelling case requires full model development steps and involvement of expertise at a high level. Therefore, there is a need for a system that would assist in the model development and use/reuse. This system should fulfil certain requirements. For example, it should allow generation of various types of models with corresponding degrees of details; it should provide expert help and guidance in the selection of model building blocks and in the generation of the model equations within a new building block; provide a knowledge representation structure for a model library together with model documentation; allow easy reuse and modification of existing models. This is in agreement with the recommendations of Foss et al. (1998), which emphasizes that a solid understanding of the model development process is required for the development of modelling tools, which in turn aims to improve the efficiency of the modelling process and the quality of the models. However, Foss et al. (1998) also notes the lack of implementation of modelling techniques in existing computer-aided modelling tools. More recently, Klatt and Marquardt (2009) have identified prospective improvements for the present modelling tools, such as multi-scale modelling features together with documentation, maintenance and reuse of models in an efficient and economical way. The benefits of the integration of modelling tools are highlighted in the study of Zhao et al. (2011), which shows the potential for generic, non-case-specific tools, while Kraus et al. (2012) emphasizes the need for unification of modelling documentation. A study by Cameron and Ingram (2008) reports that there is still a lack of possibilities for adequate model documentation including the modelling process and version changes. This confirms the results given by Foss et al. (1998) that the documentation of the final model is often not sufficient.

Often a single modelling tool is unable to provide all the needed features, therefore, integration of modelling tools is a potentially beneficial option (Zhao et al., 2011). The high update-frequency of commercial tools and following risk of incompatibility are reported to be a barrier to modelling (Cameron and Ingram, 2008). Therefore, some

supporting tools that can assist in model implementation in different modelling environments are highly desirable (Foss et al, 1998). Through the tool integration, process models consisting of complex subsystems that cannot be modelled and simulated in a single general-purpose modelling tool can be handled. Also, the necessary support for model re-use can be provided (Schopfer et al, 2004), as well as offering new capabilities for users to write mathematical models in different formats and formalisms in addition to providing options for model transfer to other modelling-simulation software environments.

Better application of the developed models can be achieved by reusing them through a specially developed modelling framework. This framework should increase reuse and exchange of models, provide proper documentations and support a variety of models. One of the most time consuming activities for model developers is the exchange of the information between software systems and tools (Kraus et al, 2014). Modellers are more likely to choose the modelling tool that is easy to learn, has user-friendly interface, requires less effort to set up the problem and provides sufficient and easy documentation for the model and its results (Cameron and Ingram, 2008). Since process modelling tools usually need different implementations for different modelling objectives, model re-usability becomes very important. The models provided by a computer aided modelling tool should, therefore, be accessible via a programming interface to other tools (Bogusch et al, 2001). This can be achieved through a computer aided modelling tool that provides a component interface following certain standards, such as, CAPE-OPEN, COM-object, XML, DLL or TXT. This would allow user to exchange and work with a wide variety of models, increasing thereby models reuse. This integration and transfer of models between different tools will allow the use and analysis of models in the most efficient way, depending on the goal of modelling (i.e. model identification, parameter estimation, process design, system representation/understanding, process control, etc.).

1.2 Overview of existing methodologies and tools

In this section some of the most common approaches to modelling and tools utilizing them are discussed. **Table 1.1** provides an overview of existing tools and approaches that they are based on.

1.2.1 Common modelling approaches

Growing interest and need in modelling systems over the last decades inspired research and development of a various approaches and tools aimed to provide environment and support for the model development. Roughly the modeling approaches can be classified into two groups (Marquardt, 1996): block-oriented and equation-oriented approaches.

In block-oriented (or modular) approaches every process is abstracted by a number of standardized blocks that model the behavior of a process unit or a part of it. The blocks are linked by connections representing the flow of information, material or energy.

Table 1.1 Overview of some tools and their modelling approaches

Modelling systems	Equation-oriented	Block-oriented	Phenomena-oriented	Ontology-based	Natural language	Tool integration	Solvers included	Reference
MOSAIC	*					*		Zerry et al, 2004; Hoang et al, 2013; Kraus et al, 2014; Merchan et al, 2014.
Modeller	*	*		*			*	Westerweele, 2003; Preisig, 2010
Aspen Custom Modeler	*	*				*	*	Aspentech, 2003; 2011
gPROMS	*	*					*	Barton, 1992; Barton and Pantelides, 1994; Oh and Pantelides, 1995
ICAS-MoT	*				*	*	*	Russel and Gani, 2000; Salez-Cruz, 2006; Heitzig, 2011
ICAS-ModDev	*	*	*			*	*	Jensen and Gani, 1996; Jensen, 1998
MODELLA	*	*	*				*	Stephanopoulos et al, 1990
GAMS	*						*	the COMSOL Group
ModKit	*	*				*	*	Bogusch et al, 2001
BatchDesign-Kit		*					*	Limninger and Stephanopoulos, 1998
ABACUSS	*				*		*	Feehery and Barton, 1994
TechTool	*	*	*				*	Limninger et al, 2000; Limninger and Krendl, 1999
OMOLA	*						*	Andersson, 1994
ASCEND	*						*	Piela, 1989; Piela et al, 1991, 1992
CMT (Conceptual Modelling Tool)		*		*				Yang and Marquardt, 2004

Models of process units are precoded by a modeling expert and incorporated in a model library. The end user selects the models from this library and links them to each other and the plant model. The examples of the tools following this approach can be flowsheeting environments, e.g. Aspen (AspenTech, 2011) or ProII (Schneider Electric, 2015).

The modular approach is easy and powerful, however, it has a major drawback that is the lack of precoded models for some unit operations or block with required level of detail.

Equation-oriented approach supports the implementation of unit models by means of modeling languages. Equation-oriented languages allow the development of the models to a large extent. However, the modeling process in this case requires profound knowledge of a specific modeling language, as well as, domain knowledge, as the tools using equation-oriented languages normally do not assist the user in the model development from the engineering perspective.

Large number of research groups has been working on solutions that will improve user experience with modeling tools and, therefore, will allow achieving results of a better quality. These solutions often combine block- and equation-oriented approaches in attempt to incorporate the benefits and to get rid of the drawbacks of both approaches.

Several classifications were provided previously (Marquardt, 1996; Jensen, 1998; Rico-Ramirez, 1998; Hantos and Cameron, 2001; Salez-Cruz, 2006). Some of the recent and commonly used approaches that aim to improve the model development process are given in the next section. These approaches usually include elements of both equation- and block-oriented approaches and provide novel ideas for the model representation.

1.2.2 Novel approaches for model representation

Phenomena-based approach

Phenomena-oriented modeling is an initiative to address model generation from a system theoretical approach and aims at computer-aided support of the problem formulation. The focus lies on the specification of a phenomenological description without directly encoding detailed mathematical expressions. The associated equations can be generated by an automated computer system through the appropriate interpretation of the phenomenological description. Model building by phenomena selection offers several advantages (Linninger, 2000):

The organization and composition of models is easily understood.

Phenomena condense the information of the underlying system in highly compressed form.

Adaptation, reuse and modifications of models can be accomplished through rearrangement and/or aggregation of phenomena rather than re-editing of the numerous equations.

Computer-aided process modeling tools that adopt a phenomena-based approach allow characterization of the systems to be modeled using physicochemical concepts. Some of

them such as Model.la (Bieszczad, 2000) and TechTool (Linninger et al, 2000) do this explicitly by offering a certain phenomena-based modeling language. Drengsig et al (1997) suggests a set of graphical symbols to represent modeling concepts. Other tools, such as Modkit (Bogusch et al, 2001) and ModDev (Jensen, 1998) reflect physicochemical concepts through model building blocks, which accommodate mathematical models as well. The existing tools usually provide a graphical interface for modelers to compose a conceptual model.

Phenomena-based modeling languages have a narrower range of applicability than equation-based systems due to their compressed information content. In general situations, a phenomena-based language might not be able to adequately describe certain elements of the system, though, in might be applied to a specific domain.

To overcome this issue another level of modelling can be introduced that features modification of phenomena-based modelling languages. Meta-modelling concept by Linninger (2000) offers the adaptation of the modelling paradigm of phenomena-based language and supports its evolution and generation of different versions of dialects for phenomena-based modelling language. This feature is implemented in TechTool (Linninger et al., 2000).

ModDev (Jensen, 1998) offers a tool for creation of new building blocks by aggregation of existing building blocks, as well as, a tool for definition of new building blocks for its phenomena-based language. The intention of these tools is to create blocks that cannot be generated by the current knowledge base.

Ontology-based approach

An ontology is an explicit specification of the conceptualization of a certain domain, typically involving concepts, their relations, and axioms that represent complex constraints (Uschold and Gruninger, 1996). For process modeling, an ontology should provide concepts about the behavior of process systems. So, the main motivation for building on ontologies is to structure the modeling process, which encompasses the definition of what and how it is modeled all the way through to solving a specific mathematical problem associated with any activity that involves the use of the model (Preisig and Haug-Warberg, 2012).

OntoCAPE (Morbach et al, 2006; Morbach et al, 2009; Hai et al, 2011) is an example of ontology-based approach for the conceptual modeling. This approach suggests the use of an ontology to provide physicochemical concepts as the basis of an ontology-based conceptual modeling tool. This tool produces conceptual models through loading an ontology, presenting concepts to the human modeler, facilitating the instantiation of concepts, and verifying the resulting conceptual model (Yang and Marquardt, 2004).

The ontology-based approach had been applied to the construction of mechanistic process models in the Modeller project (Westerweele, 2003; Preisig, 2010). The resulting model always satisfies the principles and rules captured in the ontology and

consequently they are structurally accurate, thus, the modeling errors are reduced or avoided (Preisig and Haug-Warberg, 2012).

A distinctive feature of the ontology-based approach is the use of a generic domain ontology expressed in a formal modeling language. The concepts employed for modeling can be processed using common ontology tools that support this ontology language. The use of an explicit domain ontology also holds the potential for promoting information reuse. However, the flexibility of this approach is limited and the model developer has to stay within the options provided by the ontology.

Natural language approach

Natural language approach allows simulation of the chemical process that is programmed by language, which is natural to process engineers. The task of computer-aided environment is to transform incoming information into the form suitable for solution, as well as, to maintain consistency of the model.

A natural language approach has been successfully applied to the design of batch operating procedures by Linninger and Stephanopoulos (1998). There a comprehensive design methodology for the interactive computer-aided development of chemical batch processes for the manufacturing of pharmaceuticals and its implementation in the BatchDesign-Kit (BDK) was presented. BDK offers a high-level man-machine interface through which process engineer can directly implement their processing ideas by using their “natural” language. This language-based design helps chemists and chemical engineers to focus on design decisions and allows generation of more effective recipes without the detailed evaluation of it by a process design group.

This approach combined with ontologies has been utilized in the synthesis of biorefineries (Magioglou et al, 2014) and waste management systems (Magioglou et al, 2015). These works emphasize a benefit of natural language as a common language between experts of modelling and final users.

Natural language approach in the modeling tools allows human developers to generate and evaluate process designs rapidly and efficiently; use of these tools would appear to be simple even for a non-experienced user. However, the tools based on the natural language are limited by the side of their knowledge bases and can be applied only for specific domains.

1.2.3 Tools integration

Computer tools for process engineering cover a wide range of applications and even a small part of the process design might require a large number of tasks with complex relationships. When specific tools are used for each task, the integration of these tools becomes important, though difficult to achieve.

Each tool will typically use a different format for data representation and even if the formats were to be the same, the information embodied in the data set may require interpretation. Moreover, issues that affect the different parts of process engineering

process, as well as, different activities in an enterprise will affect the needs for tool integration (Fraga et al, 2002).

Tool integration requires not only the ability to transfer data from one application to another but also interpretation of the data, management of the applications used and support for multiple users accessing the same data.

The Integrated Computer Aided System, ICAS (Gani et al, 1997), is an example of a fully integrated computer environment for process engineering. ICAS combines computational tools for modelling, simulation, synthesis/design, control and analysis in a single integrated system. These computational tools are present as toolboxes. During the solution of a problem, the user moves from one toolbox to another to solve problems requiring more than one tool. ICAS represents an environment, which is fully integrated, using a common data representation and a common user interface.

Another type of tool integration is a use of special tools – so-called integrators – that maintain information consistency between different modelling and design tools (Körtgen and Nagl, 2011).

MOSAIC (Zerry et al, 2004; Kuntsche et al, 2011) is a tool that is an example of an integrator. It controls the consistency of documentation and modelling information and allows translation of this modelling information to different modelling languages (e.g. gPROMS, Matlab etc). However, this tool offers best translation, if the model is coded in the MOSAIC environment and this requires extra user effort. More benefits can be achieved, if models can be both effectively exported and imported to/from the system.

1.3 Work done prior to this project

Presented work is built on the previous developments in the modelling area by the members of a research group, that are ICAS-MoT (Russel and Gani, 2000; Salez-Cruz, 2006; Heitzig, 2011) and ModDev (Jensen, 1998). Brief overview of the features that have been incorporated in the current work from these tools is given below. More details can be found in the Chapters 3 and 4 of this thesis or in the sources of origin (see above mentioned references).

1.3.1 ICAS-MoT

ICAS-MoT incorporates tools for model construction, analysis and application together with a computer-aided work-flow structure that assists in the model development. The software implementation includes as well solution algorithms and numerical methods.

The idea of the computer-aided work-flow structure by Heitzig (2011) has been adopted for the other model development stages (Sections 3.2, 3.3, 3.6).

1.3.2 ICAS-ModDev

ModDev is the tool for utilization of a model generation feature (see Section 3.3) for the model development process that is based on phenomena-oriented language. The concept and implementation have been done by Jensen (1998).

1.3.3 Contribution of this project

A computer-aided modelling framework has been developed that incorporates ModDev, MoT and a newly developed in this project tool that is ModTem – a tool for modelling templates approach. Three tools have been integrated with each other in a sense that the model equations can be transferred among the tools for reuse.

XML translation of the model equations from MoT and ModDev and transfer of XML files with modelling information among tools in the modelling framework and to the third-party tools has been implemented.

Some updates of the user interfaces of both MoT and ModDev have been made in order to reflect the equation transfer features and to improve user experience.

1.4 Objectives of the thesis

- Development of template-based modelling approach based on the idea of model reuse, which emphasizes the use of a model not only for one specific application but also for future applications involving different needs and levels of details to match different purposes.
- Development of a software architecture and implementation of the developed template-based approach into software tool.
- Development of a computer-aided modelling framework that integrates template-based modelling approach with previously developed modelling methodologies and tools – ICAS-MoT (Russel and Gani, 2000; Salez-Cruz, 2006; Heitzig, 2011) and ModDev (Jensen, 1998). Desired outcome is a user-friendly system, which will make the model development process easier and faster, give opportunity for better model reuse and provide the way for unified and consistent model documentation.
- Highlighting the different aspects of the template-based modelling approach and overall modelling framework through a set of case studies.

1.5 Structure of the thesis

This Ph.D. thesis consists of six chapters and appendix. Chapter 1 includes introduction to the topic and elaborates on the challenges in computer-aided modelling. Detailed description of the developed template-based modelling methodology is given in the Chapter 2. Chapter 3 presents the development of computer-aided framework, which integrates the number of tools with different modelling approaches in one system. Further the computer-aided framework is implemented as a software tool and this implementation and software architecture are described in Chapter 4. Chapter 5 is dedicated to a presentation of the various case studies highlighting different aspects of developed template-based modelling approach. Finally, Chapter 6 summarizes the main achievements of this Ph.D. project and gives overview of the challenges and issues that can be addressed in the future.

2 CONCEPT OF TEMPLATE-BASED MODELLING

A modelling template is a generic model on which other problem-specific models are based – in this way, it is similar to a superstructure of process alternatives – that is, it includes a collection of models for the process it represents. In the template, a model is decomposed into three sets of equation types: balance equations, constitutive relations and connection and/or conditional equations. This decomposition is based on the modelling hierarchy proposed by Cameron and Gani (2011). Within each type of equations, multiple versions of model equations may be stored in the template and retrieved when necessary to generate different problem-specific models. Breaking up a model into conceptually significant pieces with following combination of these pieces among each other in different ways in order to create a new model would greatly simplify model reuse (McGahey and Cameron, 2007).

Template-based modelling is a novel approach, which is based on the idea of the combination of the various parts of the previously developed models into a new set of equations for the specific case.

A number of goals can be achieved by developing models in the template-form. Models are stored in the template library in a format that is easy to maintain and to update. The models are decomposed into equations sets – with the option to select multiple versions within a set. The template helps to generate different versions of the base-model by modifying the description of the phenomena involved and/or models used to describe the phenomena. Furthermore, if more models are available within the model template library, many more models could be generated by employing different combination rules for model use/reuse.

For example, the template for a flash separator can include several options for equations combination (see Figure 2.1). When combining different options in different combinations, four new models can be generated with different detail level. Thus, one template-form can contain a lot of information, which gives opportunity to represent the system in various forms, depending on the current requirement.

Mass balance	$0 = Fz_i - Vy_i - Lx_i$	$dn_i/dt = Fz_i - Vy_i - Lx_i$	
Constitutive equations	$0 = S_i - (Vy_i)/(Fz_i)$	$0 = K_i - p_i(T)/P$ $0 = P_i - f(T)$	
Connection equations	$0 = 1 - \sum S_i$	$0 = y_i - x_i K_i$ $0 = 1 - \sum y_i$	$0 = x_i - n_i / (\sum n_i)$ $0 = L - f(n, T, P)$ $0 = V - f(n, T, P)$

Figure 2.1. Template options for the flash separator

Model-templates are based on the knowledge representation structure (ontology) as highlighted in Figure 2.2. The knowledge in each model template is structured in terms of three main classes of information: upper-layer, sub-layer and block. Here, a class is a pattern with specific parameters or characteristics, which is used for creation of the model objects. These objects are therefore the instances of the class. The collection of instances of the classes represents the total model in the template or the system (see Figure 2.3. Main classes of the template structure and examples of the instances.).

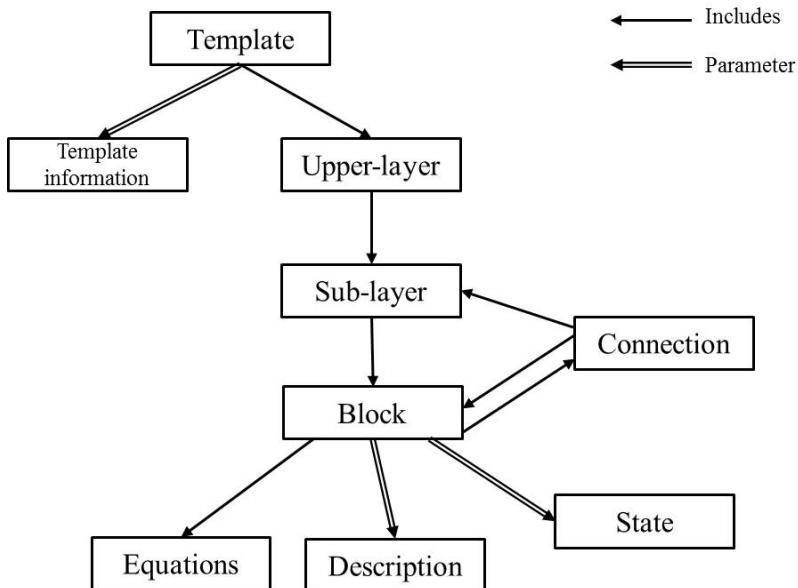


Figure 2.2. Structure of hierarchy in the modelling template.

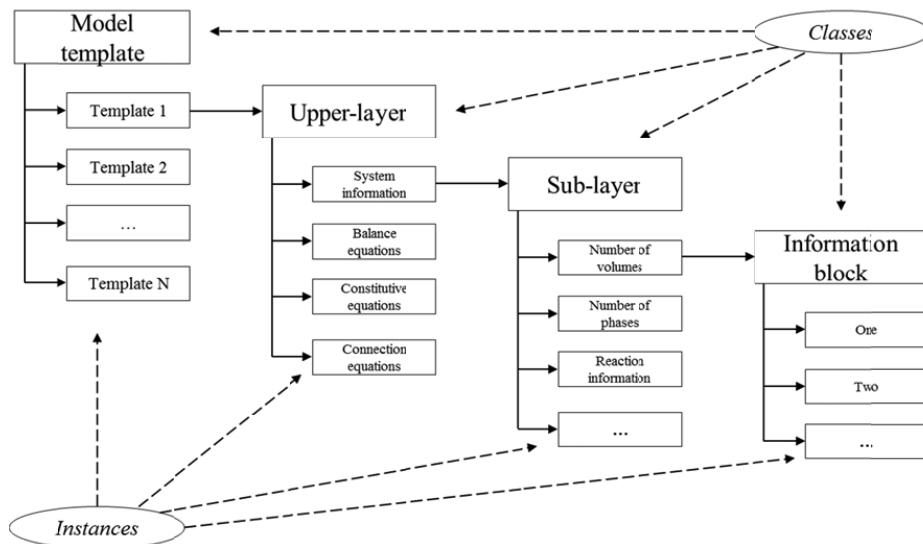


Figure 2.3. Main classes of the template structure and examples of the instances.

Each model template includes four upper-layers, according to the structure of the decomposed model: system information, balance equations, constitutive equations and connection equations. Each upper-layer may include one or more sub-layers, which are related to the description of the system and the phenomena occurring within it. Each sub-layer may include one or more blocks, which contain the data, parameters and/or equations of the corresponding instances of the sub-layer. Each instance of a block is related to one option of the phenomena representing the connected sub-layer.

Also the structure includes certain parameters and minor classes for describing a template, those are equations, block description, connections and block state (see Figure 2.2). Description contains information about phenomena to which the block is related. For most of the sub-layers, a block also includes one or more equations, however, some blocks, especially blocks from the upper-layer of the system information, may not include any equations, but they contain information, which affects equations of the subsequent blocks, for example, number of compounds in the reaction will change the number of balance equations. Therefore, additionally each block includes information about possible connections with blocks related to other sub-layers.

During template creation following information should be provided to define the template blocks - afterwards this information will be employed during template-use:

- *Description*: defines the phenomena to which this block is related. Used to assist the modeller when choosing the template option.
- *Connection* to other blocks: whether the current block affects any other blocks is defined here. This includes information about the block and sub-layer, to which

this block belongs. Information about connections will allow adaptation of the template options reflecting the current choice of the modeller.

Before creating a template user should have a clear understanding of the connections among blocks of the template. Block might depend on the blocks from another sub- and upper-layer, usually it is upper-layer of system information, as there is given information, which can affect balance equations (number of volumes/phases, number of input/output streams), constitutive equations (reactions, assumptions on temperature/pressure relations) and connection equations (number of volumes, nature of the connection). While creating the template the user defines the block dependency and its nature, such as state of the block with respect to the block it is connected to. It means that when affecting block is active and chosen, then the dependent block should change its state, i.e. become available for choice (active) or inactive, or to change the set of equations, which is added to the model by choosing this block. Essentially, the affecting blocks should be from one of the sub-layers located above the sub-layer of the current block and being already described. If, coming from the system description, block depends on another block from certain sub-layer, then the dependencies from all blocks from this sub-layer should be described, as the block state and equations set should be defined for any case. For example, if model includes connection equations for two volumes, then, obviously, the block describing this connection will depend on the choice made at the sub-layer “Number of volumes”. Thus, this block should be active, if there are two volumes in the system, and non-active, if there is only one volume. Therefore, both dependencies on blocks describing one or two system volumes should be defined.

- *Equations* representing the model: the equations corresponding to the equation type from the upper layer and the data-parameters from the corresponding sub-layer are stored in the equations information. If the block is chosen during a template-use, these equations are extracted and added to the final set of modelling equations. If a block is not connected to other blocks, then there will be one set of equations, otherwise several sets may be loaded, depending on the number and the character of connections. It is possible and likely that the blocks that belong to the upper-layers of the system information do not have any equations.
- *State*: a parameter representing the behaviour information of the corresponding block – whether it can be chosen at the moment or not. There are three options here: the block is active by default – that is, information and equations from this block are included automatically in the final model when template is loaded for use; defined – that is, the block can be used as an option for changing the final model; non-active – that is, the block cannot be chosen because of its connections or because it is not fully defined.

2.1 Template-creation

The template application tool consists of two parts: template-creation and template-use. When using the modelling template, obviously a model-template would need to be first created (or must be available), before employing the various options of the template-use tool.

The workflow for template-creation is provided to guide the modeller when creating template. It includes 10 steps and they are shown in Figure 2.4.

Step 1. In order to create a template, all necessary information about the system and the associated phenomena together with model assumptions should be collected and given as data for template documentation. Data-model information includes any system information related to the specific case study, such as number of volumes, number of phases, presence and nature of reactions, any relations involving thermodynamic, kinetic, transport terms, nature of volume and phase connection, types of balance equations etc. The result of the step 1 should incorporate all required knowledge of the system and model equations.

Step 2. The template creation requires the user to decompose the set of model equations into sets of balance, constitutive and connection or/and conditional equations, as it is the base for the template structure.

Step 3. Check, if template for the modelled system is already available in the template library. This will determine whether a new template should be created or the available one will be updated.

Step 4A. If template describing the system exists already in the template library, then it should be taken for updates. Updating the existing templates when possible is more preferable, than creating a new template every time. In this way, template modelling approach benefits can be employed, such as combination of the different parts of template into the new model, and the repetition of the information in the template library will be avoided.

Step 4B. If there is not relevant template in the library, then a new template should be created.

Step 5A. The part of the model that is new for the chosen template and will be used for updates should be defined and derived from the model. This part should be related to the certain phenomena or other parts of the template (balances, connection equations, etc.). If updates are linked to several parts of the template, then each addition to the template should be defined separately.

Step 5B. When the new template is created, all parts of the model should be derived and defined with respect to the phenomena or model parts they represent. The modeller developing the template will need to define the necessary sub-layers inside the four main upper-layers of “System information”, “Balance equations”, “Constitutive equations” and “Connection equations”, based on parts of decomposed model, and assign blocks for each of the sub-layers. Ideally, sub-layers should be as detailed as

possible, as it will allow better reuse of the template parts and ease possible future updates. Otherwise, when update is related only to the part of the sub-layer information, then the identical part will be repeated again or the structure of the template should be changed.

Step 6. Template description is provided giving information about the aspects of the modelled system and type of model. If the existing template was used, then the description should be updated reflecting the changes or extra information added to the template.

Step 7. Parts of the model describing different phenomena in the system and different parts of the system should be identified as information blocks with description of represented model parts. These information blocks form the structure of the template and represent different template options.

Step 8. Definition of connections among template parts is important, as it helps to describe the correct representation of the final model. Dependencies between different information blocks should be defined at this step. For example, template option for use of equation connecting two system volumes depends on whether there are two volumes in the system or only one.

Step 9. At this step the new/updated template should be checked for consistency and information completeness. If the information requires additional revision, then the modeller should go back to step 7.

Step 10. If the template proved to be consistent and complete, the creation process is finalized and the template is added to the template library. After that the template is ready for (re)use.

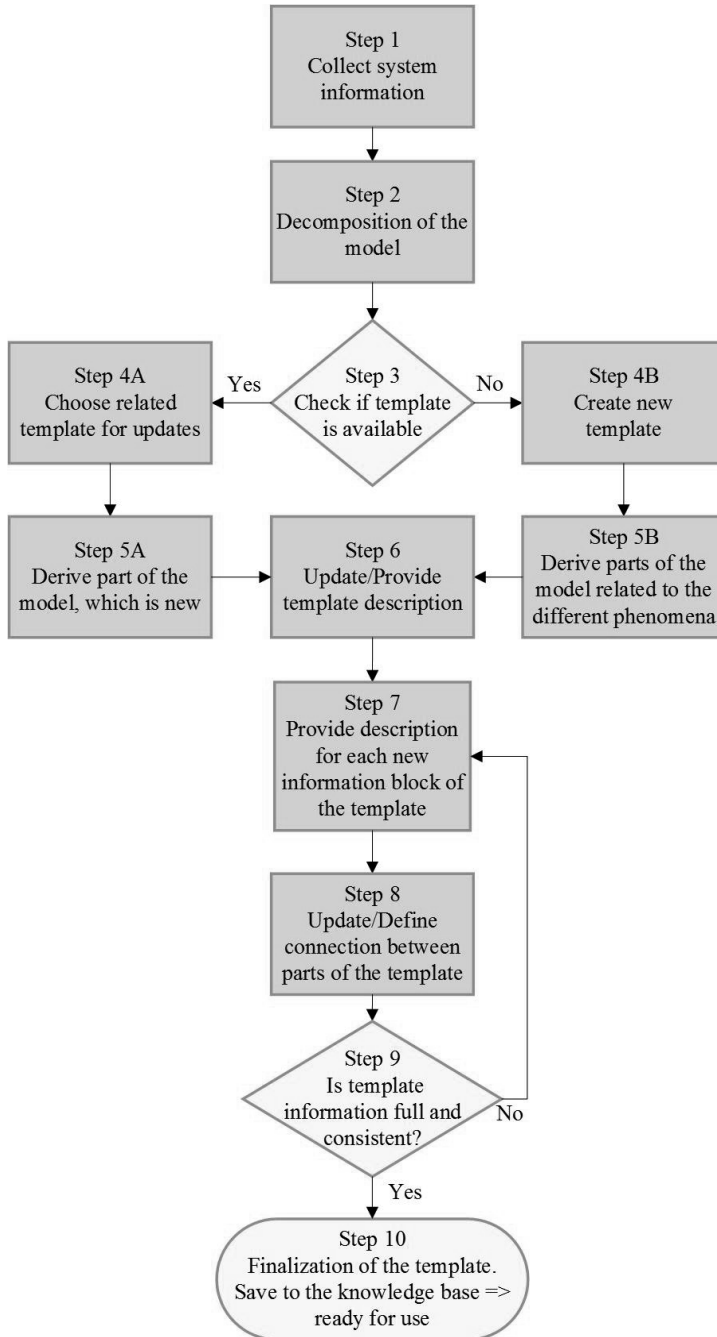


Figure 2.4. Template-creation workflow.

The flash separator model is considered as an example:

$$0 = Fz_i - Vy_i - Lx_i \quad (2.1)$$

$$0 = K_i - (p_i \gamma_i) / (P \varphi_i) \quad (2.2)$$

$$0 = P_i - f(T) \quad (2.3)$$

$$0 = \gamma_i - f(T, x) \quad (2.4)$$

$$0 = \varphi_i - f(T, P, y) \quad (2.5)$$

$$0 = y_i - x_i K_i \quad (2.6)$$

$$0 = 1 - \sum y_i \quad (2.7)$$

The model describes system in a steady state with non-ideal equilibrium, taking into account activity and fugacity coefficients for components in liquid and vapour phases respectively (*step 1 of the workflow*). The model is decomposed in terms of the mass balance (Eqn. (2.1)), constitutive equations (2.2)-(2.5) and connection equations (2.6)-(2.7) (*step 2*).

The template for the flash separator exist already in the system – see Figure 2.1 (*steps 3 and 4A*), however, new model includes different set of constitutive equations describing phase equilibrium, which is not presented in the template (*step 5A*). Therefore, new option should be added to the template, which will allow extracting of the model with non-ideal equilibrium from the template (*step 6*). New block is added to the upper-layer of constitutive equations (*steps 7-8*). Template after updates is shown in Figure 2.5 (*steps 9-10*).

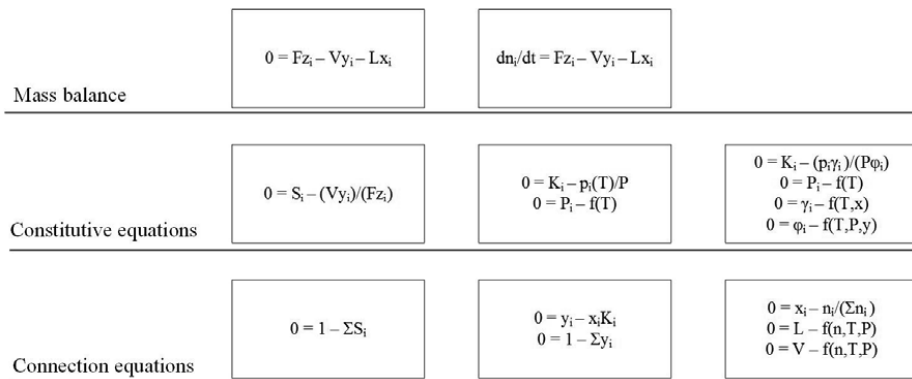


Figure 2.5. Updated template options for the flash separator with one more option describing phase equilibrium compared to previous version (Figure 2.1)

Now, new option can be combined with the one already present in the template to obtain new case-specific models. Thus, dynamic model with non-ideal equilibrium can be extracted from the template without the need to develop it from scratch.

2.2 Template-use

When reusing the model through the template, the modeller is to follow a certain workflow in order to achieve the best result. This is done by answering the questions about a system to be modelled by choosing the most suitable option from the ones available in the template. The workflow for template use is shown in Figure 2.6 and it consists of the following 6 steps:

1. *Select a template* from a library of templates, based on the system information and modelling objective, e.g. the template the flash separator. Even if the template knowledge base does not include the template for the specific case in question, a template for a similar process can be used, and this will provide the starting base for further model development. Therefore, the understanding of the phenomena and the system that should be modelled is necessary.
2. *Select the process system*. At this step the modeller should answer a number of questions related to the system to be modelled. These questions depend on the used template, but the typical questions can be, for example, what is the system volume, number of phases, presence or not of reactions, accumulation, number of components, number of scales etc. Information from this step is used in the subsequent steps. For examples, number of components will affect the balance equations; presence of reaction will define whether the options for kinetics should be available; number of volumes or phase will determine the presence of connection relations.
3. *Select balance volumes*, the model type (steady state, dynamic state), etc., retrieve the balance equations - mass, energy and momentum. Every template already has the corresponding equation type for the selected model-template, however, the user may change the system information and the corresponding equations will be changed accordingly.
4. *Select constitutive equations*. In this step user identifies the constitutive models needed to describe the system volume behaviour, such as thermodynamic relations, transport and reaction rates, diffusion, phase equilibrium, etc., and the template retrieves the corresponding model equations, if available, from the modelling template library.
5. *Select connection and conditional relations*. These are equations describing surroundings and system connections, such as summation of mole fractions.
6. *Generation of the final set of model equations* by aggregating all the model equations retrieved in the previous steps. At the end of this step, the model is ready to be sent to the solver or for further analysis and/or identification.

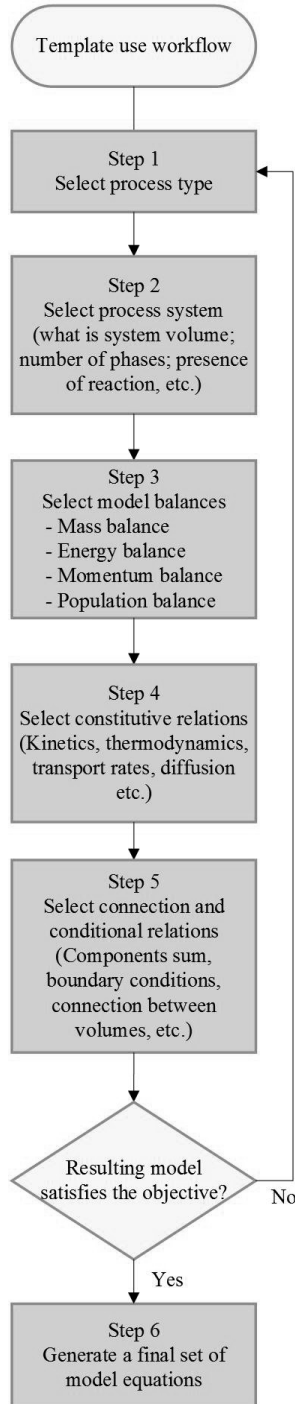


Figure 2.6. Template-use workflow

3 COMPUTER AIDED TEMPLATE-BASED MODELLING FRAMEWORK

Implementation of the modelling methodology into computer-aided modelling framework will allow achieving numerous benefits by using computational tools. The main objectives of the computer-aided modelling framework are to provide structure, guidance and support during model development and application, increase the efficiency of the modelling process, and, improve the quality and reliability of the models. This is achieved by structuring the computer-aided modelling framework on established workflows for different general modelling tasks related to the development and application of models. The modeller is systematically guided through the steps of the different workflows and, at each step, the framework identifies and integrates the required guidance, tools, database and library connections. The framework integrates systematic model derivation tools with generic modelling templates and tools, as shown in Figure.

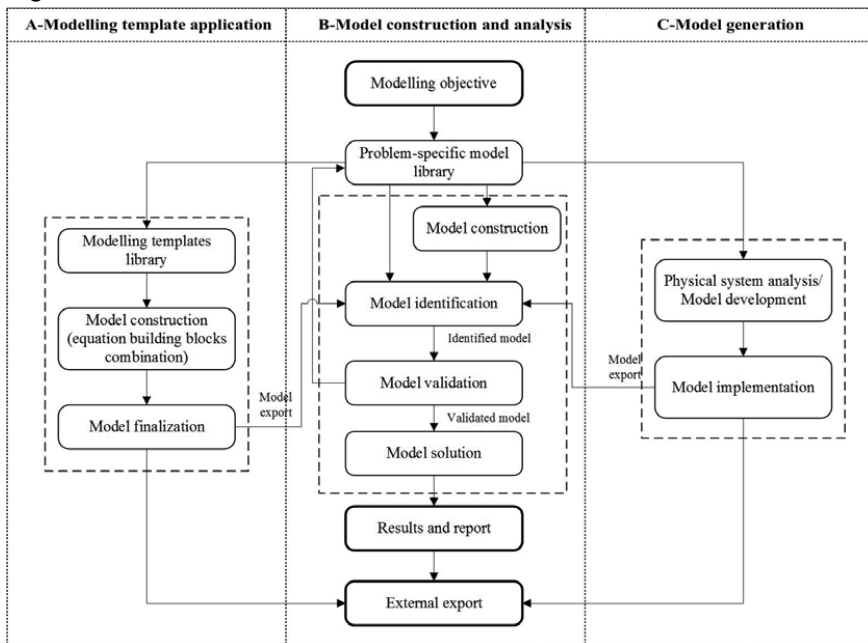


Figure 3.1. Computer-aided modelling framework

Depending on the modelling needs and goals, the modeller has the possibility to create a new model when the needed problem specific model is not available in the library. The modeller has also the possibility to identify, validate and solve the new models or use existing models from the template library. Furthermore, model templates enable the idea of model reuse as the user can employ the template as a base to build different problem-specific models for different applications. The objective here is to provide structured domain-specific knowledge, to speed-up the model development/derivation process and to improve model quality. As the template library includes validated models, the creation of a new template-based model will reduce the modelling time and increase efficiency in the subsequent steps of model analysis, model verification and application. The modelling framework allows the user to develop new, improve existing, solve, validate, and export models for the third party applications. The computer-aided modelling framework consists of three parts, related to the previously described modelling stages: model reuse (part-A), model construction and analysis (part-B) and model generation (part-C). The central part-B defines the main pathway for model construction and analysis, which is based on a generic computer-aided tool by Heitzig et al (2010). Part-C is related to model generation, which purpose is to help the user to generate the model equations with generic model building blocks (Jensen and Gani, 1999). Part-A is concerned the template-based modelling approach and the application of it for model reuse. All parts of the framework are connected with each other allowing easy transfer of modelling information among the tools.

Moreover, the modelling framework includes a problem-specific model library, which contains developed and validated models with associated parameters, initial values and/or experimental data. All these models have been developed by using the modelling workflow (Heitzig et al., 2014) and this guarantees consistency of modelling information and documentation.

Depending on the modelling goals and needs, the modeller has the possibility to create a modelling case based on an existing problem-specific model from the library or an available model template or construct a new model by using the model generation part of the modelling framework. After that the model is analysed and prepared for application by using the central part of the modelling framework.

Additionally, the model transfer capabilities are addressed, as this allows developed models to be shared among different group of modellers using different modelling environments.

Developed computer-aided framework is based on a modelling methodology that includes six modelling stages related to a certain modelling task. Overview of the stages is presented in the Figure 3.2 and further explained in detail.

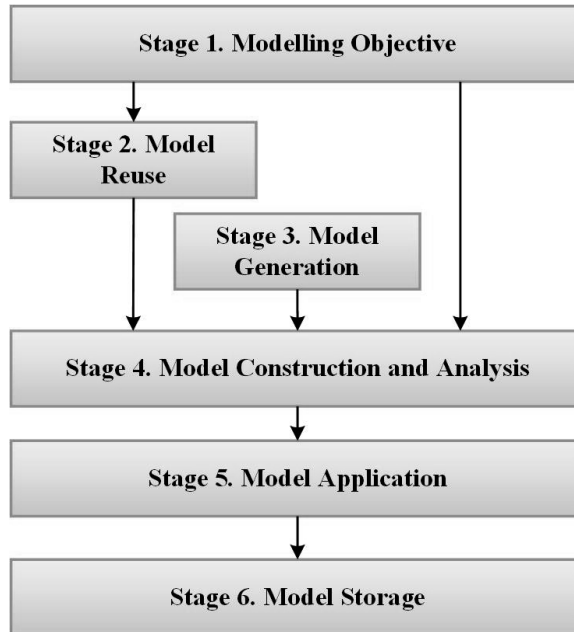


Figure 3.2.Stages of modelling methodology

Stage 1. Modelling objective. Model development starts with definition of the modelling objective, goals that are intended to be achieved and the assumptions that can be done without harming the objective. This stage also includes collection of the system information, which is required for model development and further application.

Stage 2. Model reuse. Here the possibilities for model reuse should be addressed. Based on the modelling objective and collected system information from the previous step any model or model part, which is corresponded to the modelling case, can be considered for reuse. Main goal of this stage is to ease model development process, so the development time will be reduced, if the development starts from the certain initial point or could be avoided by reusing the whole model. It is assumed that the previously developed models, which are available for reuse, have been properly evaluated and verified. Therefore, it is safe to reuse them, as they will not add any additional error to a new model.

Stage 3. Model generation. Model generation means obtaining model equations based on the physical description of the system and the system information. At this stage a certain knowledge base should be used, which can provide the necessary modelling information regarding different parts of the system and connection among them.

Stage 4. Model construction and analysis. The stage of model construction and analysis can follow stages 2 and 3 or can be used as standalone. At this stage a model developer either conducts necessary changes in the model, which has been obtained from the previous stages, or directly constructs a model from scratch based on an available

information about system, personal experience and knowledge and/or expert help. The resulting model obtained after this stage should be consistent with respect to the equation formulation, variables definition and relationship between them, and ready for further analysis and/or solution.

Stage 5. Model application. Here application of the model should be performed, which might include model simulation and optimization.

Stage 6. Model storage. In order to maintain and expand knowledge base and to extend opportunities for model reuse, any updates of the models or new models, which were the result of the stage 5, should be stored in the appropriate format.

Further in this chapter above mentioned modelling stages will be described in more detailed form, providing information about necessary input data, expected output and methodology workflows.

3.1 Stage 1. Modelling objective

This stage has a goal to establish modelling objective and to collect information about system, which is required for model development, application and documentation for possible further reuse. Comprehensive and systematic workflow for this stage has been developed by Heitzig (2011) and it includes two main steps and several sub-steps. Summary of the workflow and is presented in the Figure 3.3.

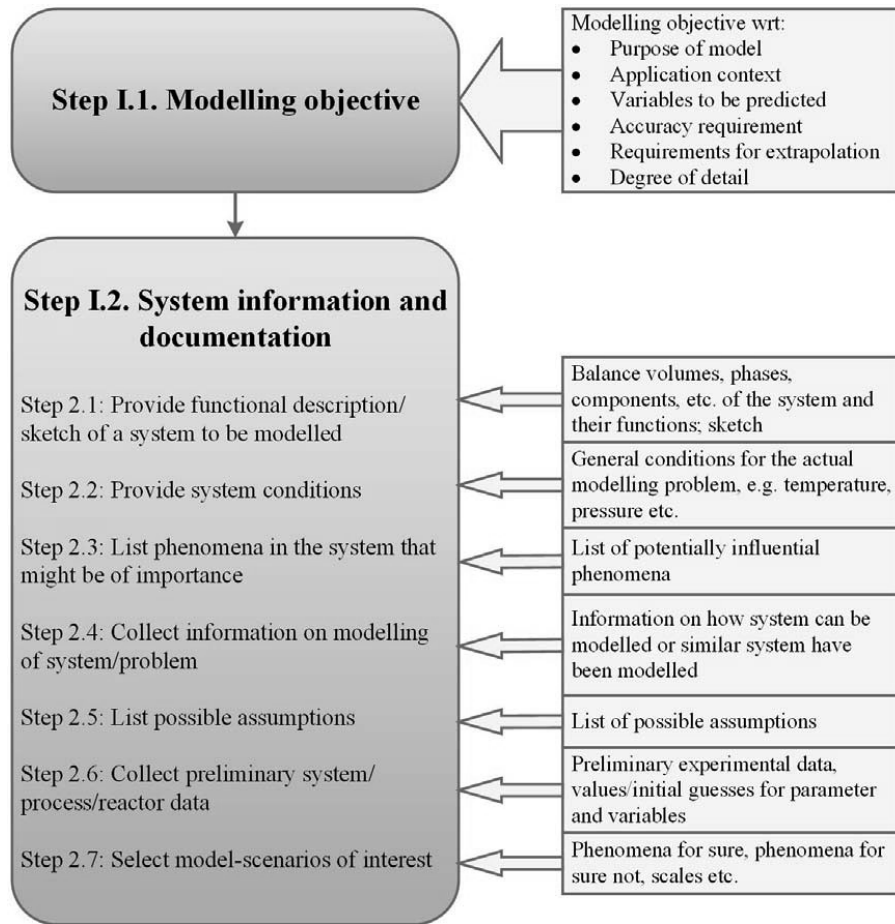


Figure 3.3. Work-flow (boxes on the left hand side) and data-flow (arrow boxes on the right hand side) for Stage 1: Modelling objective (adopted from Heitzig, 2011).

Even though modelling objective is defined manually by the modeller, a computer-aided interface will provide a systematic structure for the description of the system and modelling objective as well as automatic report generation. Modelling workflow with required features and automation potential are identified in the Figure 3.4. This workflow is adopted from Heitzig (2011).

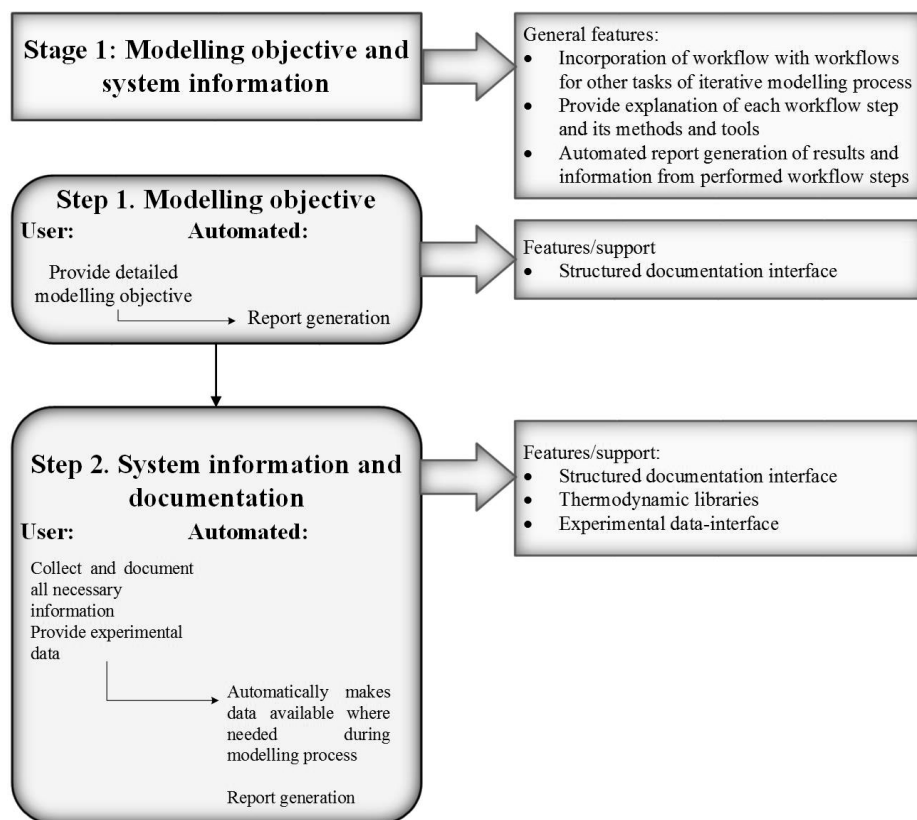


Figure 3.4. Computer-aided workflow for the modelling stage 1: Modelling objective

3.2 Stage 2. Model reuse

Model reuse stage can be considered as the most beneficial in terms of time reducing needed for model development. Naturally, if the modeller can obtain the model for the specific case from the knowledge base and reuse it again or even just a part of it, then the time spent for model development will be reduced significantly.

This modelling stage includes two main concepts: model knowledge base and template-based approach. Model knowledge base consists of the models, which were previously developed and validated and, therefore, can be reused for similar case studies.

The part of the framework describing the use of the model library and generic modelling templates is highlighted in the Figure 3.5. The computer role here is related to the storage and extraction of the information in the libraries, as well as support of the user while following modelling workflows.

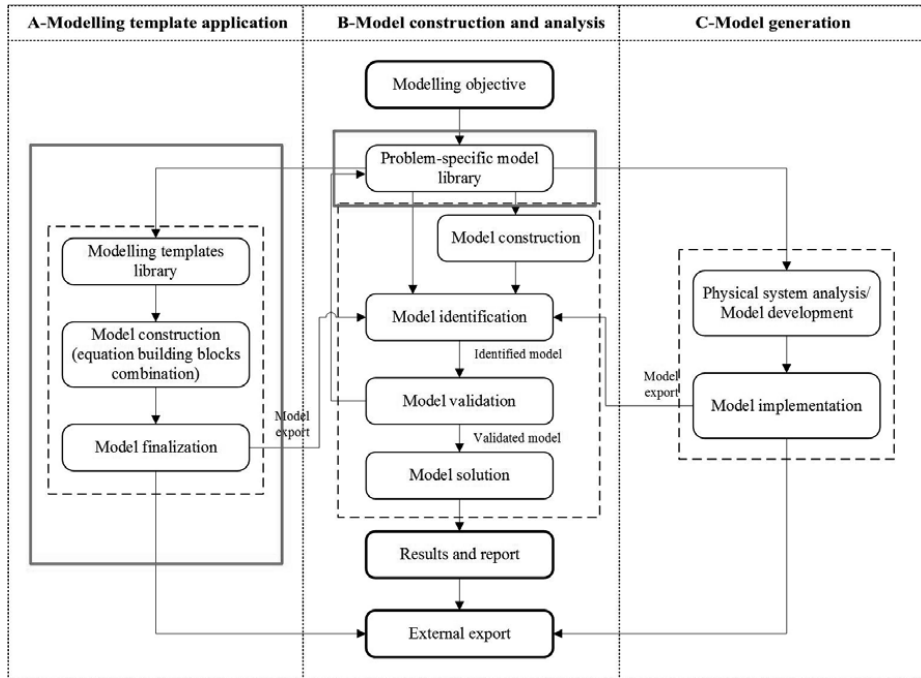


Figure 3.5. Computer-aided modelling framework highlighted parts related to model reuse

3.2.1 Model library

Model library includes the case specific models, which were developed and verified previously. The library is based on a specific knowledge representation ontology, which allows storage the models in the systematic way in order to provide easy access, when the model should be retrieved, and when updating the knowledge base, the new model should be placed according to the ontology.

This modelling ontology includes certain classes of models, which are based on the types of the system. Two big groups are dynamic and static models, which include models for units (reactor, separation, mixing etc.) and processes (flowsheets, case-specific models).

Model library is represented by a special computer database, which includes information about models and the ways to retrieve them. This database includes the list of available models with (i) short name of the model, which defines the purpose of the model, (ii) file path, which determines the way to retrieve the model and (iii) the hierarchy of classes – ontology route, which defines the type of the model, gives additional information for model identification and helps to locate the model in the model library.

If the modeller can obtain the model for their specific case from the knowledge base, then he/she can further proceed to model construction stage.

3.2.2 Template application computer-aided tool

The modelling template includes model equations, model description, assumptions, solution strategy etc., and the framework guides the user through steps for choosing an appropriate template for the specific modelling problem or will allow the modification of an existing template, if necessary.

Figure 3.6 shows computer-aided template-use workflow (Figure 2.6) with emphasis on automated features.

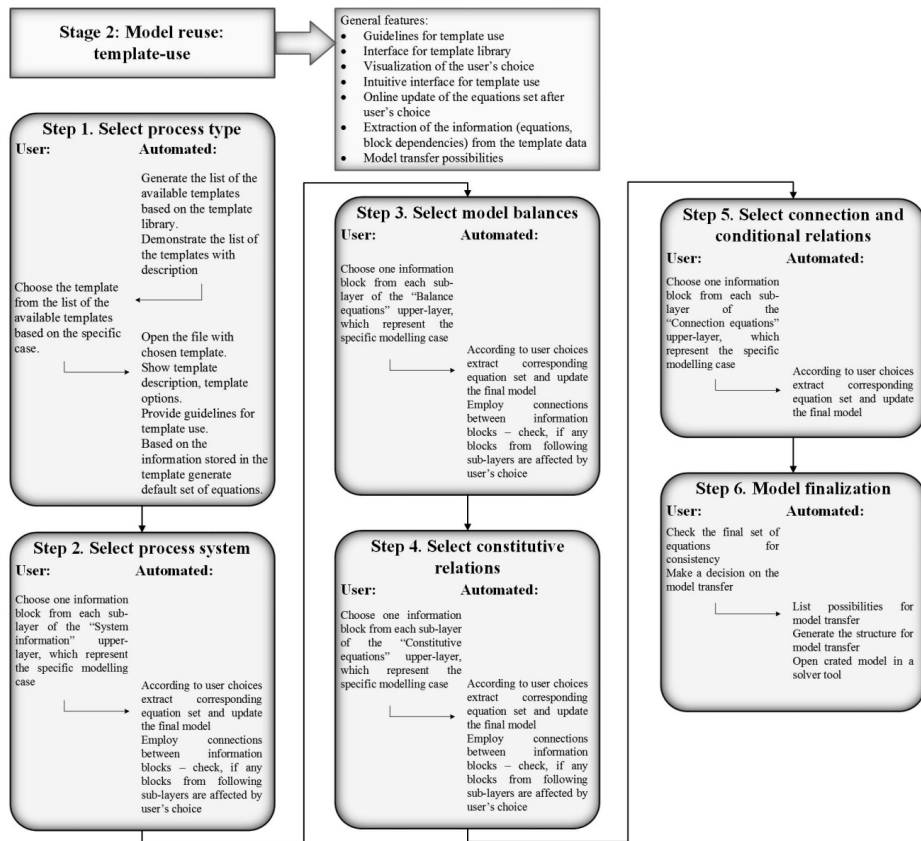


Figure 3.6. Computer-aided workflow for modelling Stage 2: Model reuse, highlighting automated feature.

3.3 Stage 3. Model generation

This stage includes generation of the model based on the physical description of the system. This is based on the model generation methodology by Jensen (1998). The

generation of models with various degree of details is accomplished by multilevel decomposition/aggregation technique. Thus, the system is decomposed into smaller and smaller elements, which are described in detail, and then the described elements are aggregated together.

The overall methodology for this stage is:

1. Decompose the system into fundamental building blocks
2. Describe each of the fundamental building blocks
3. Aggregate the fundamental building blocks

The fundamental building blocks are shells, streams and shell connections. After the system is decomposed into fundamental building blocks, these building blocks are described. The model equations are obtained based on the description that is provided by the modeller and they are taken from the knowledge base, which includes the most common equations to describe the chemical system. The equations are stored in the base according to the hierarchy, based on the equation types. The hierarchy is given in the Figure 3.7.

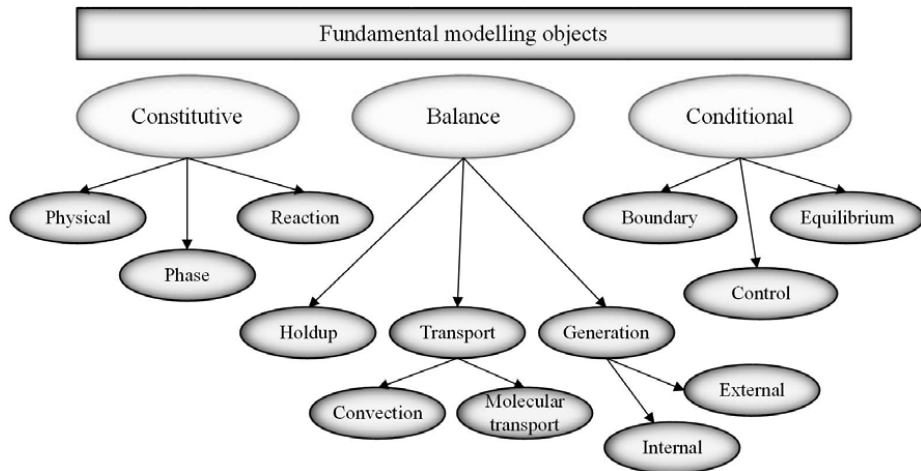


Figure 3.7. The fundamental modelling objects containing instances of the model equations that define the model

Computer-aided tool for model generation should be able to compose model equations for a system described by the modeller. Being part of the generic modelling framework model generation tool will provide opportunity for the modeller to obtain model equations based on the physical system description and then apply this model in the other parts of the framework. Corresponding part of the framework is highlighted in the Figure 3.8.

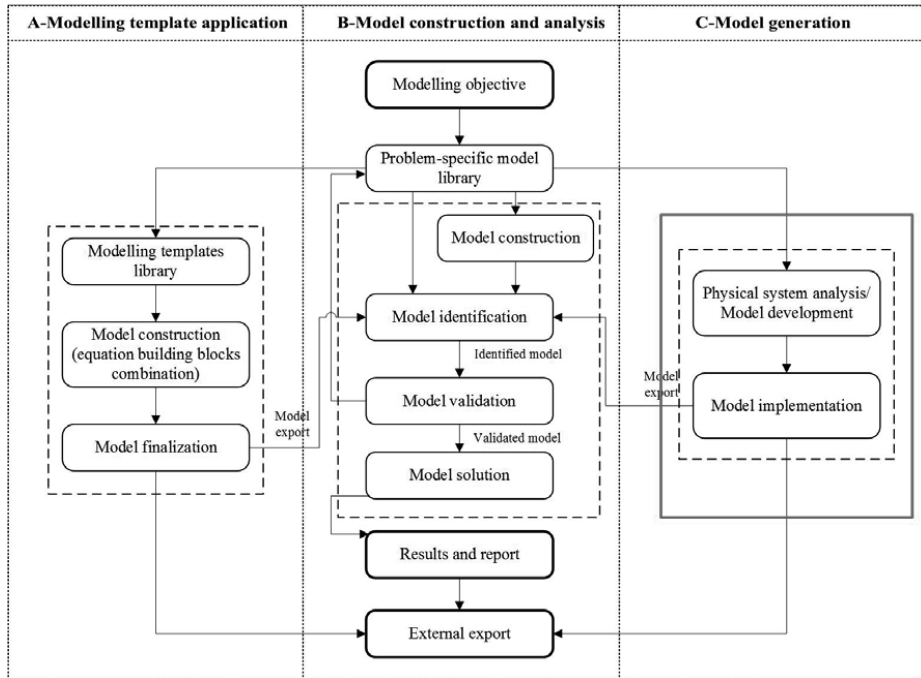


Figure 3.8. Computer-aided modelling framework with highlighted part for model generation

Using model generation tool, the modeller is able to describe a system by decomposing it into fundamental building blocks, which are then combined according to given combination rules. The computer-aided tool automatically generates the set of mathematical equations based on the description provided by modeller. The resulting set of equations is translated to an appropriate computer language for use internally or export to another tool.

The computer-aided tool for model generation had been previously developed by Jensen (1999). Figure 3.9 highlights tool’s workflow with emphases on automated features.

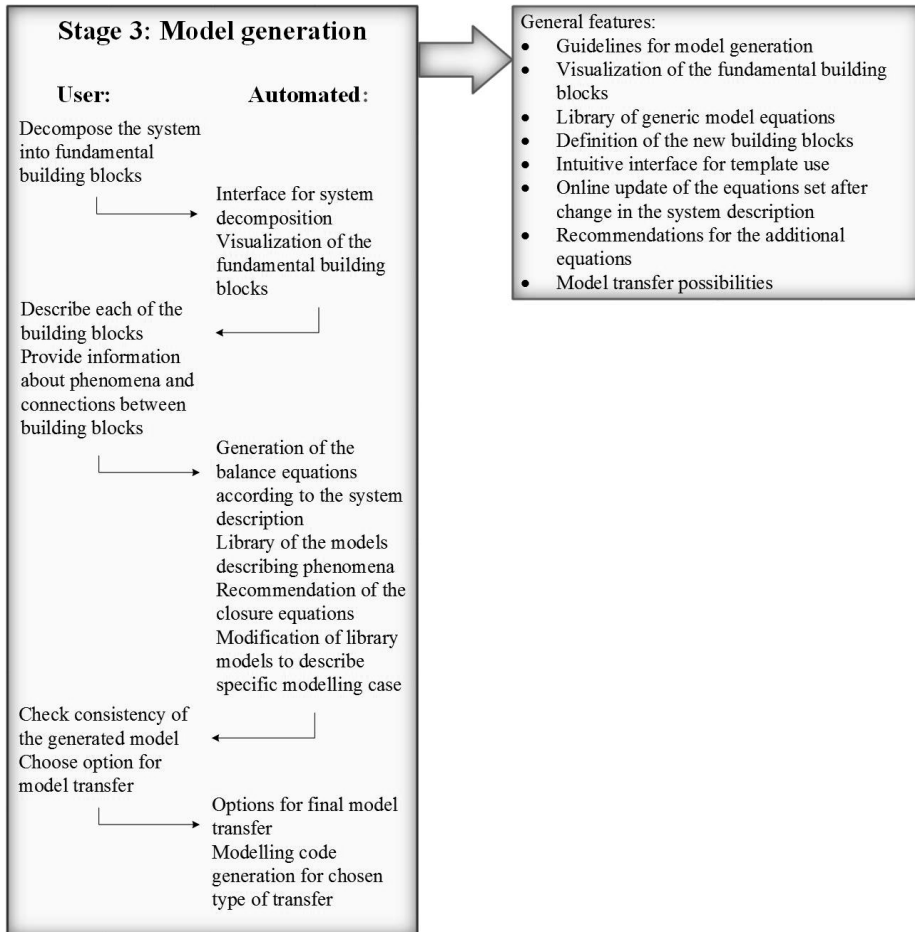


Figure 3.9. Computer-aided modelling workflow with automated features for the modelling Stage 3: Model generation (based on Jensen (1999))

3.4 Stage 4. Model construction and analysis

The objective of the model construction stage is to systematically and efficiently construct a reliable model, which reflects the modelling objective of stage 1, or to adapt and modify, if necessary, the model obtained from stages 2 and 3. This stage includes also construction of possible multi-scale scenarios. The input for the model construction workflow is an initial information about the considered phenomena, assumptions, required scales and/or sub models, integration frameworks and data-flow (Figure 3.10). The output is a linked, translated and analysed (multi-scale) model together with a solution strategy, which is connected to the corresponding solvers – a ready-to-use model object (Heitzig, 2011).

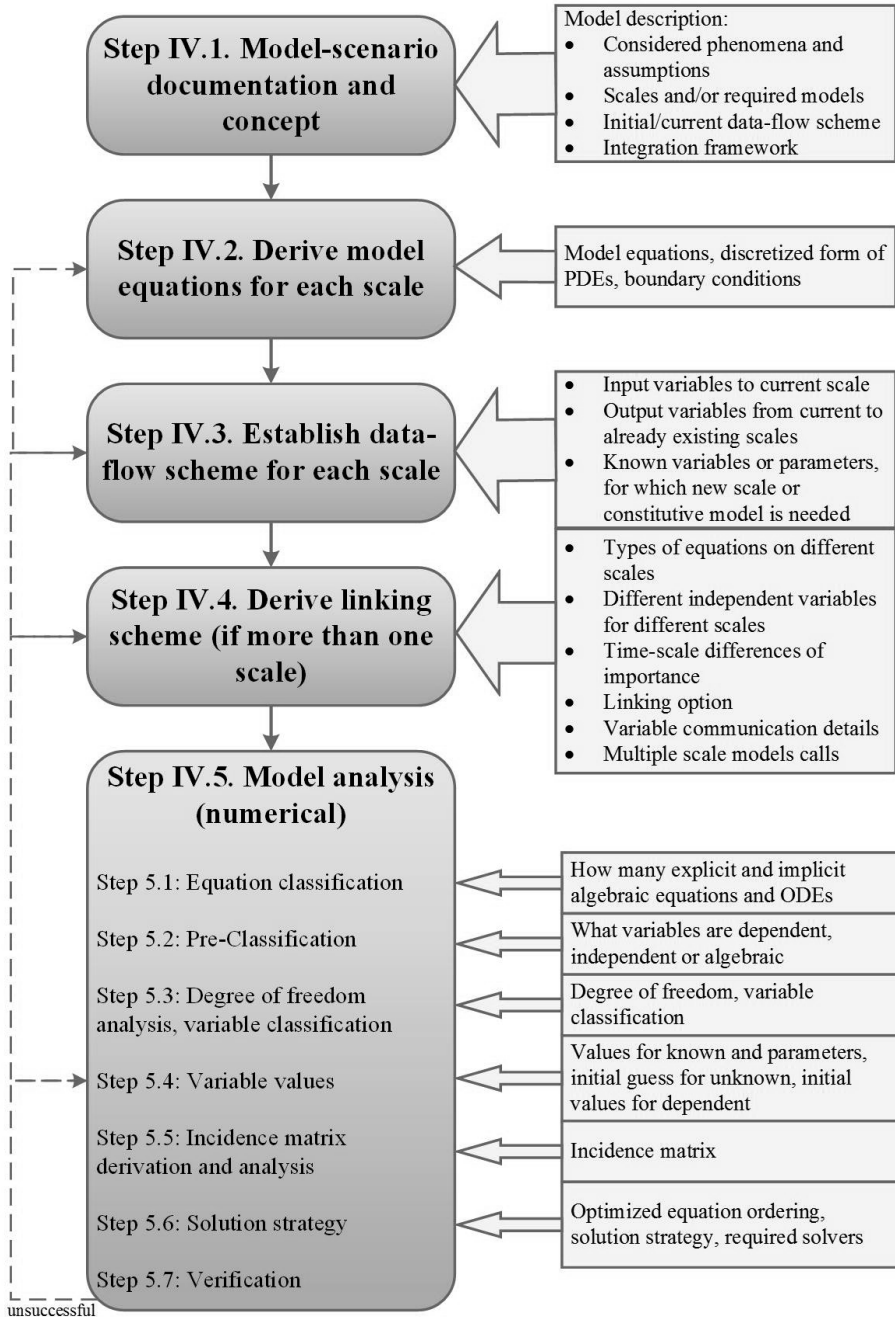


Figure 3.10. Work-flow (boxes on the left hand side) and data-flow (arrow boxes on the right hand side) for Stage 4: Model construction and analysis (based on Heitzig, 2011)

The computer-aided tool for model construction and analysis should provide generic interface for supporting model development at this stage and incorporated required connections to libraries, databases and external tools. Figure 3.11 shows the corresponding part of the modelling framework.

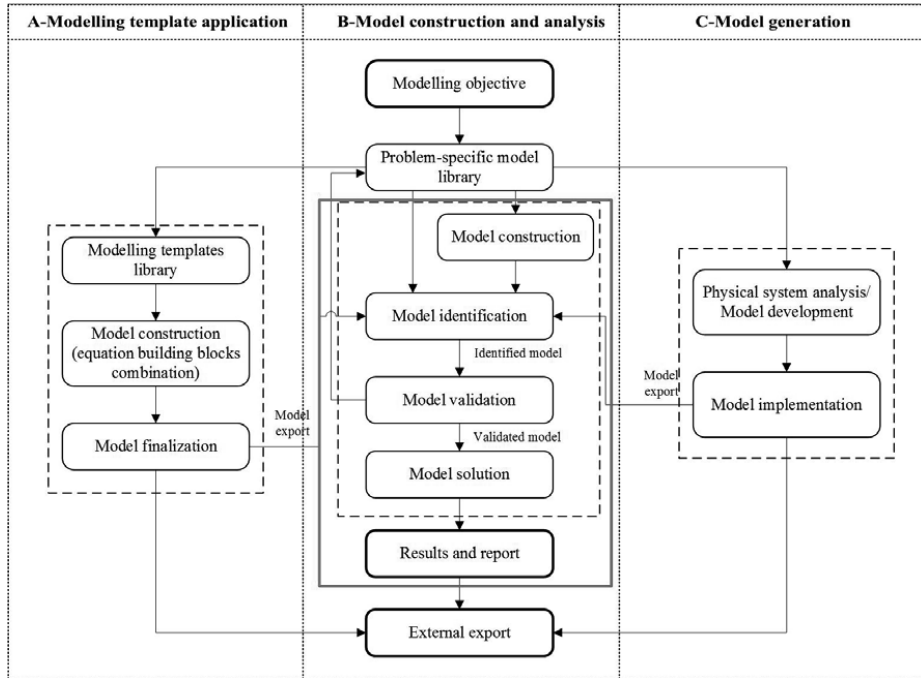


Figure 3.11. Computer-aided modelling framework with highlighted part for model construction and numerical analysis

The computer-aided workflow has been developed based on the generic modelling methodology by Heitzig (2011). It includes the potential for automation and the required features that computer-aided framework should provide.

The workflow presenting automated features of the tool is shown in Figure 3.12.

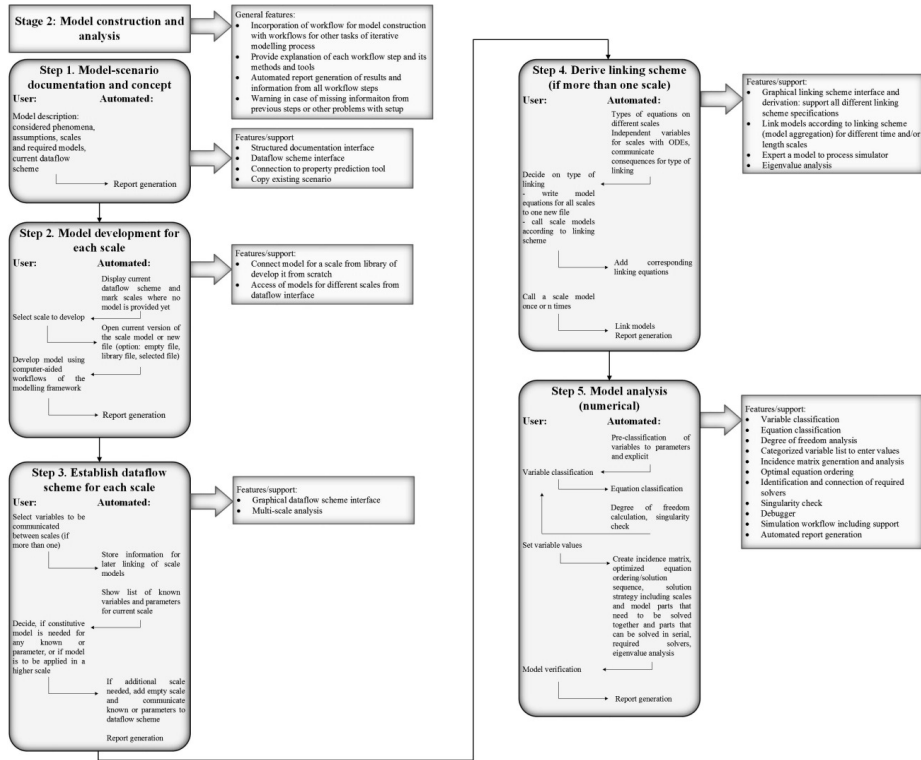


Figure 3.12. Computer-aided modelling workflow with automated features for the modelling Stage 4: Model construction and analysis (adopted from Heitzig (2011))

3.5 Stage 5. Model application

If required, model evaluation and validation should be performed before the application of the final model, in order to ensure reliability and predictability of the model, thus, to provide the best possible results by model application. Here, two workflows for model application developed by Heitzig (2011) are incorporated into overall modelling framework – workflow for simulation and optimization.

3.5.1 Simulation

The model application workflow for simulation has the objective to predict the behaviour of the modelled system under certain conditions. Figure 3.13 shows the proposed workflow.

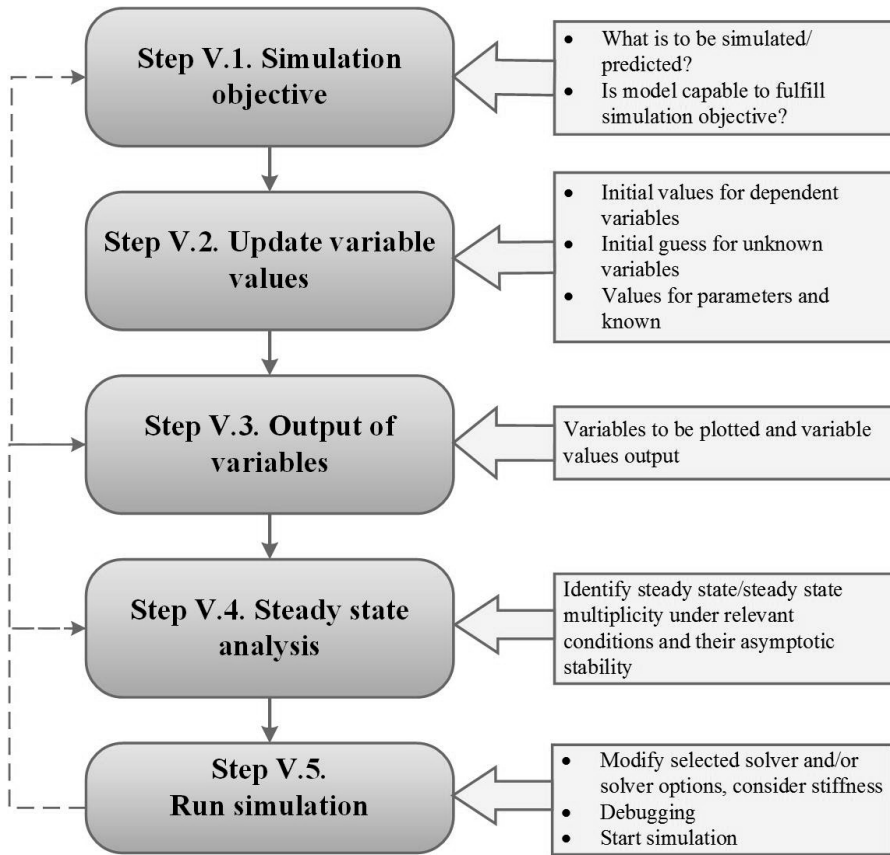


Figure 3.13. Work-flow (boxes on the left hand side) and data-flow (arrow boxes on the right hand side) for Stage 5: Model application - Simulation (adopted from Heitzig, 2011)

The simulation workflow requires the input of the simulation objective as well as an update of the variable values according to the conditions to be simulated.

The main features a computer-aided modelling framework needs to provide to support the modeller during the simulation process are summarized in Figure 3.14.

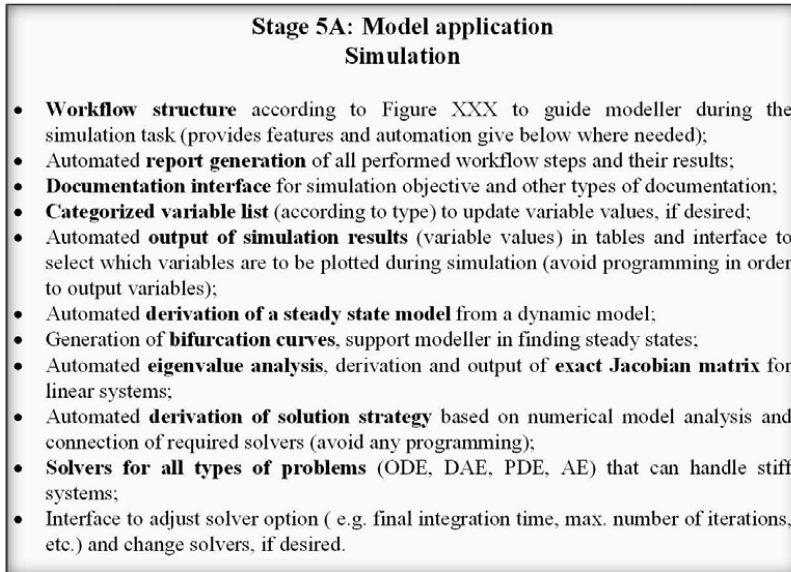


Figure 3.14. Required features, support and automation potential for the modelling
Stage 5A: Model application - Simulation

3.5.2 Optimization

The model application workflow for optimization has the goal to optimize the design of the actual system by fixing the design parameters. Figure 3.15 demonstrates the workflow to be followed, if a developed model is to be applied for optimization.

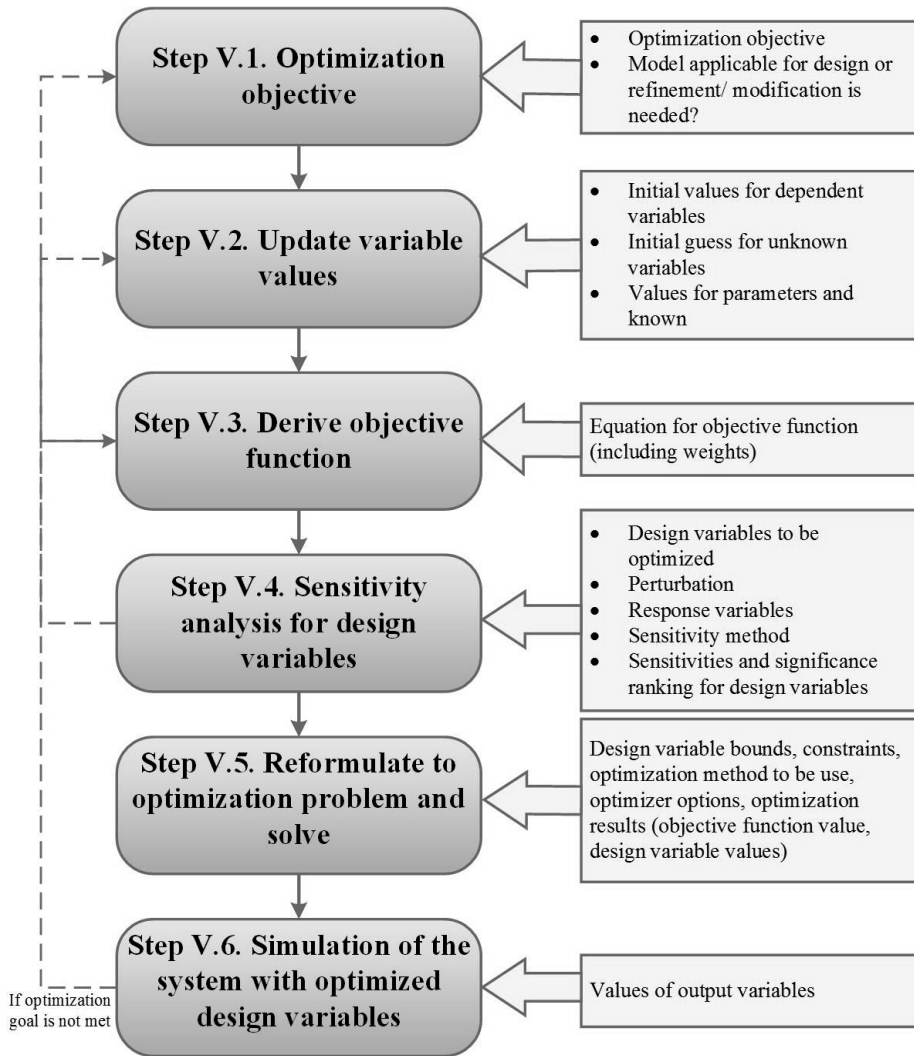


Figure 3.15. Work-flow (boxes on the left hand side) and data-flow (arrow boxes on the right hand side) for Stage 5: Model application - Optimization (adopted from Heitzig, 2011)

The input for an optimization problem is the design objective, the design variables to be optimized and their initial guesses, the objective function and the applied optimization algorithm.

The obtained output is the sensitivity of the objective function with respect to the design variables, the optimized values of the design variables, the values of the objective

function and the results of the verification simulation applying optimized design variable values.

Computer-aided modelling workflow that provides automated support for modeller during optimization stage is shown in Figure 3.16. In addition to the features described in the workflows at the previous stages, the optimization workflow should allow the automated transformation of the model implementation into an optimization problem and the reverse transformation back to a simulation problem.

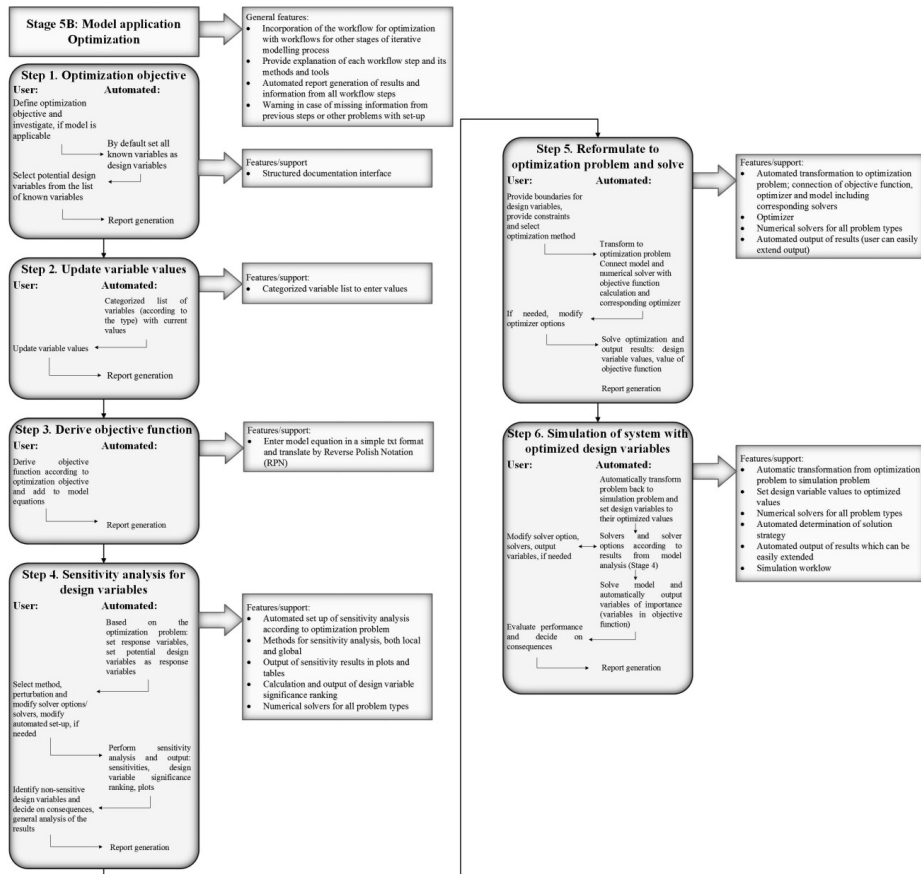


Figure 3.16. Computer-aided modelling workflow with automated features for the modelling Stage 5B: Model application - Optimization

3.6 Stage 6. Model storage

This stage is important for the model reuse idea, as in order to be able to reuse the model, it should be stored in the library or knowledge base with proper documentation available. Therefore, even though the goal of modelling had been achieved at the

previous stage, some more steps should be taken in order to update the knowledge bases and to ensure the reuse possibilities for developed model. The methodologies and systems for model reuse, which can provide the unified structure for model storage, will allow an expansion of the knowledge and sharing it among different groups of modellers.

3.6.1 Add model to the model library

First, the newly developed model should be added to the model library, as it is the easiest and most obvious way for knowledge storage for further reuse.

The model library is represented as a computer database or a computer-aided list with special structure. Therefore, in order to add a newly developed model to the library this structure should be satisfied. In order to add the new model to the model library the ontology route should be established and the name and model description should be provided. Generally, addition of the new model to the library is a manual procedure, where the modeller should provide necessary information in order to store the model. However, the model storage is automated, thus the information in the library will be secure and the extraction of it during the stage 2 of modelling methodology will be faster and easier. Additionally, the interface for adding the new model in the library can be provided.

3.6.2 Template library update

Additionally, the newly developed model can be converted to a modelling template. This is especially important, if there are possibilities and/or needs for further extension of the developed model with different options for system representation or model refinement. In this way, the template-model will incorporate all varieties and options and will make the use of different combinations easier and less time-consuming.

The use of computer-aided features for model storage for further reuse will impact beneficially to model development process, as it will provide fast access to the models and template stored in the computer libraries or databases. Also, it will give possibilities for better knowledge organization and easier updates and exchange among different modellers.

In order to create a template, the modeller should have a model, which is verified and consistent, after that the model should be prepared for template creation. If the model has been developed throughout all stages of the modelling framework, then the modeller would have all required information at this point.

Figure 3.17 demonstrates workflow for template-creation (Figure 2.4) updated with respect to the automated features.

Steps 1 and 2 of the workflow including system information collection and model decomposition are manual and should be done by the modeller in order to prepare the model for template creation.

Steps 3 and 4 involve the use of the template library. Computer-aided tool provides the interface for the list of the available templates, where user can choose, depending on the goal, between two options: to choose the template for update from the list of the available templates or to create a new template.

Further, at step 5, parts of the model should be derived according to the template ontology and the template structure classes (see Figure 2.2 and Figure 2.3). The computer-aided tool for template creation provides interface and visualization of the template classes, thereby providing assistance to the modeller.

In steps 6-8 of the workflow, the modeller provides the template description and information about blocks and connections between them. The computer-aided tool automatically stores data related to the block description, information about block dependencies and equations sets, which afterwards is extracted when the template is used.

Following step 9 for checking template consistency is also manual. Further, the template is finalized and saved and the computer-aided tool makes it available in the template library for reuse.

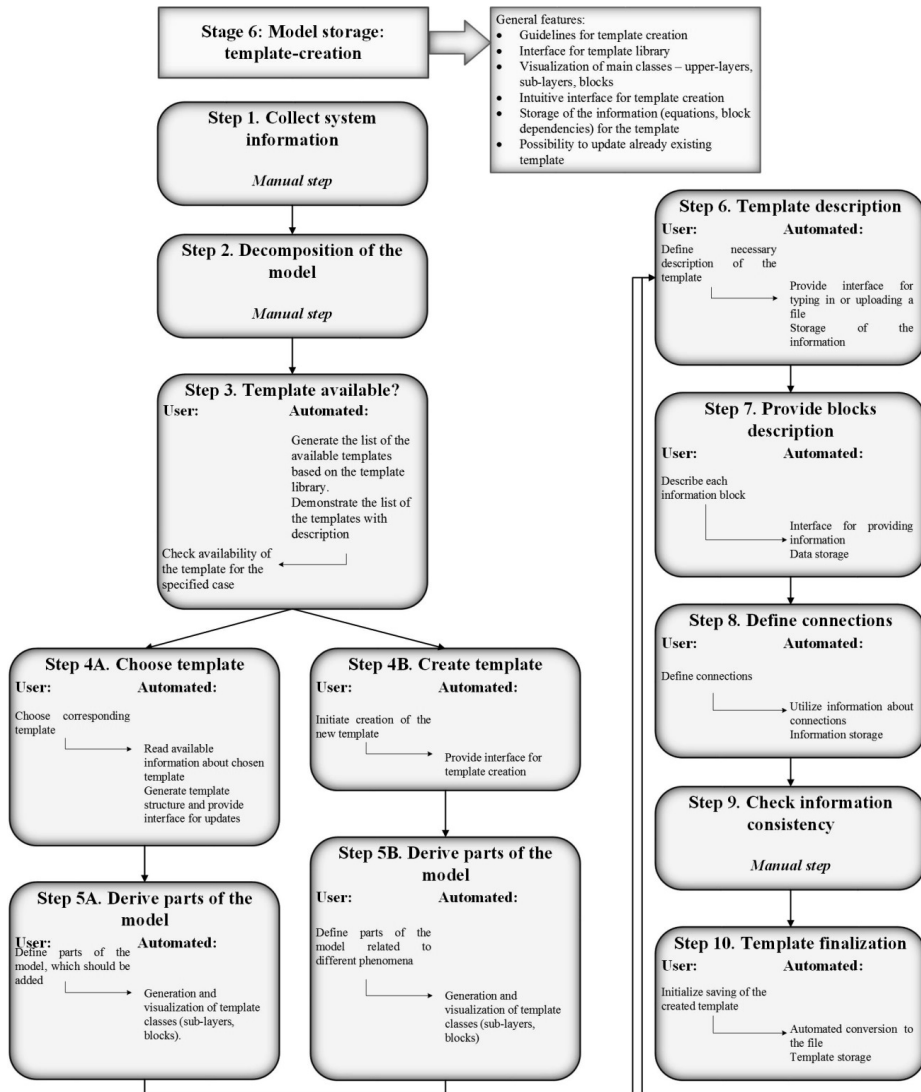


Figure 3.17. Computer-aided modelling workflow with automated features for modelling Stage 6: Model storage – Template creation

3.7 Integration of the tools within framework

One of the greatest benefits of the developed modelling framework is integration of the various tools that assist at the different stages of model development process. This gives possibility to handle more complex systems, which, otherwise, would require the use of several tools that often will mean translation of the available information into various languages and structures, repetition on the same modelling steps for different tools and

possible errors in the final solution due to incompatibilities of the tools. Therefore, the use of the integrated modelling framework will reduce the time and energy spent on model development, as well as provide more consistency and reliability to the final model.

Presented in the framework modelling tools are connected to each other through the transfer of the developed models. The description of the models is done within the same structure for all tools that are involved in the modelling framework. Thus, from the model reuse and model generation tools information can be directly transferred to the model construction and analysis tool. Also, final models from modelling stages 3 and 4 (model generation and model construction) can be utilized in the model reuse tool as a template or a part of the model library.

In order to use the model transfer among the different tools certain structure and transfer language should be introduced. Computer-aided framework should provide automatic translation of the model to and from this transfer language and ensure the correct structure of the translated information. Implementation of the model transfer with structured form and language can as well provide the basis for export/import of the models to/from external tools.

3.8 Import/export of the models

One of the most time consuming activities for model developers is the exchange of the information between software systems and tools (Kraus et al, 2014). Since process modelling tools usually need different implementations for different modelling objectives, model re-usability and model transfer become very important. This can be achieved through a computer aided modelling tool that provides a component interface following certain standards that would allow the modeller to exchange and work with a wide variety of models, increasing thereby models reuse. This integration and transfer of models between different tools will allow the use and analysis of models in the most efficient way, depending on the goal of modelling (i.e. model identification, parameter estimation, process design, system representation/understanding, process control, etc.).

The starting point is the mathematical model created in one of the tools of the modelling framework. Otherwise, it can be a model that is saved in text format and then pre-screened and pre-translated through the modelling framework to make sure that it is valid, confirming that all model equations are consistent and syntactically correct. Next, the model is translated and stored in the structured file. This structured modelling file is then processed by the special engine, which reads the modelling information, translates it and generates a new code representation of the model that can be interpreted by another tool. Figure 3.18 demonstrates the structure of the model export/import.

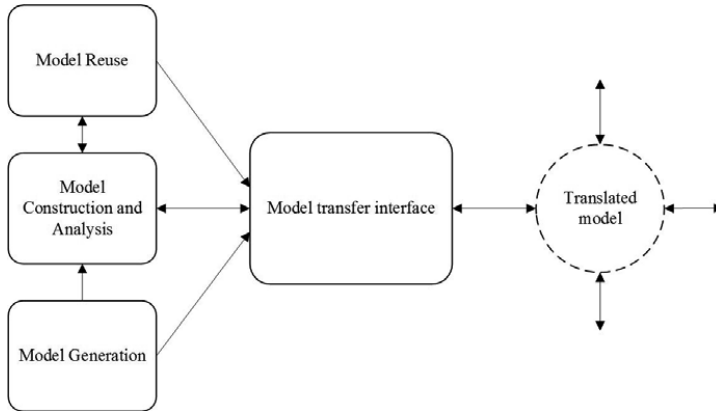


Figure 3.18. Schematic structure of the model export/import from/to modelling framework

4 FRAMEWORK IMPLEMENTATION

For easy and beneficial application, parts of the generic modelling framework have been implemented as software tools. Each part of the modelling framework has corresponding software tool, which can communicate with each other and the modelling information can be transferred among these tools. The modelling framework with corresponding software tools is shown in Figure 4.1

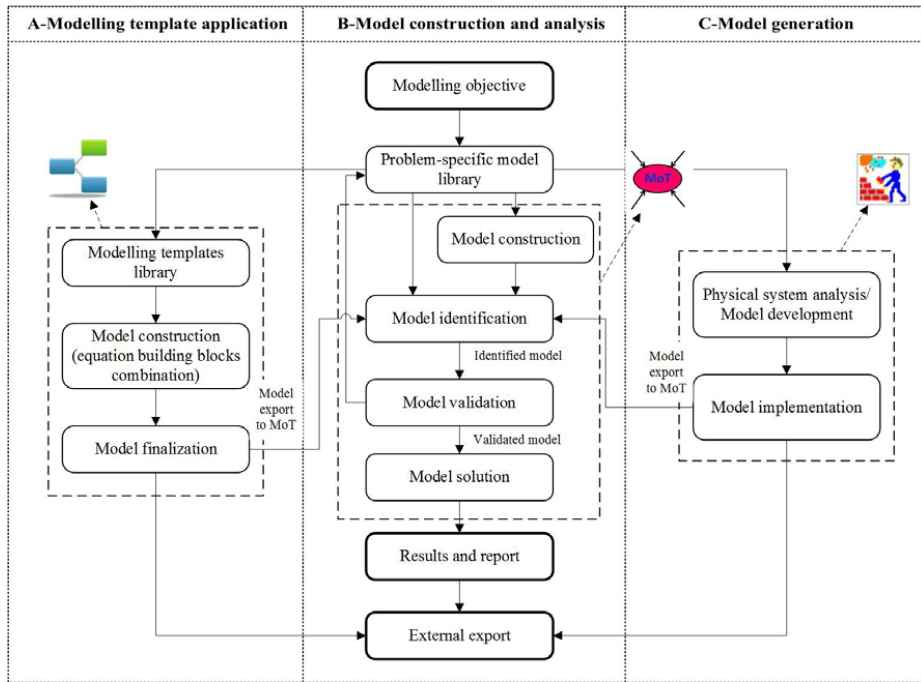


Figure 4.1. Computer-aided modelling framework with corresponding software tools

The tool representing model reuse stage (see Part-A in Figure 4.1) – ModTem – includes implementation of the model library and the modelling templates. A software tool that includes algorithms for model generation is ModDev (Jensen and Gani, 1999). Computer aided methods and tools, that include procedures to perform model translation, model analysis, model verification/validation, model solution and model

documentation are implemented in software tool called MoT (Russel and Gani (2000), Sales-Cruz & Gani (2003), Sales-Cruz (2006), Heitzig (2011), Heitzig et al (2014)). Following sections of this chapter will provide a detailed overview of the software tools and implementation of the modelling framework features.

4.1 MoT – Model construction and analysis tool

MoT (Modelling Testbed) tool is the part of the integrated computer-aided system ICAS (Gani et al., 1997) developed prior to this project. General idea of MoT is to introduce modelling environment, which allows the modeller to construct the desired model and perform its analysis and application using embedded tools and methods. The model is represented as a set of mathematical equations written in a natural language, e.g. similar to the way equations are written in books and scientific articles. Therefore, no specific knowledge of additional programming language is required. Once the model is constructed or transferred to ICAS-MoT, the equations are analysed (ordering of equations, degree of freedom analysis, etc.) and automatically processed and translated to the machine language. Figure 4.2 shows the program screens with constructed and translated model in ICAS-MoT.

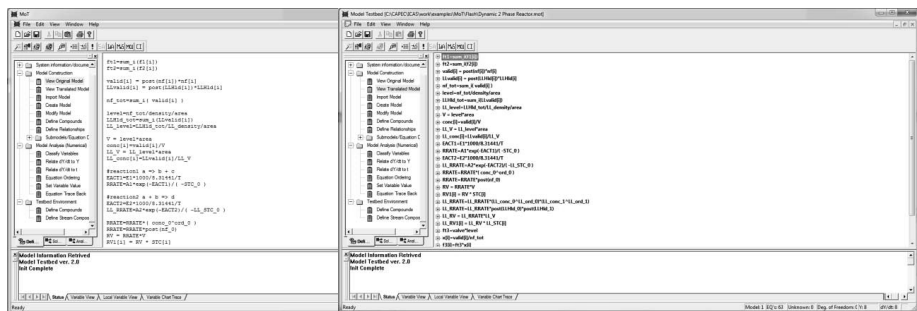


Figure 4.2. MoT screens with constructed and translated model

ICAS-MoT provides several options for solving the modelling equations (see Figure 4.3) – for algebraic equations (AEs): Broyden, Newton-Powell, Wenstein, single line, Gauss; for differential-algebraic equations (DAEs): BDF and Runge-Kutta; and also numerical optimization methods. The main numerical code in ICAS-MoT handles the solution of large sets of mixed systems of ordinary differential and algebraic equations with facilities for initialization and automatic detection and handling of discontinuities. Partial differential equations (PDEs) are converted automatically to DAEs by discretization using method of lines or orthogonal collocation. Figure 4.4 demonstrates screenshots illustrating choice of solver options in MoT.

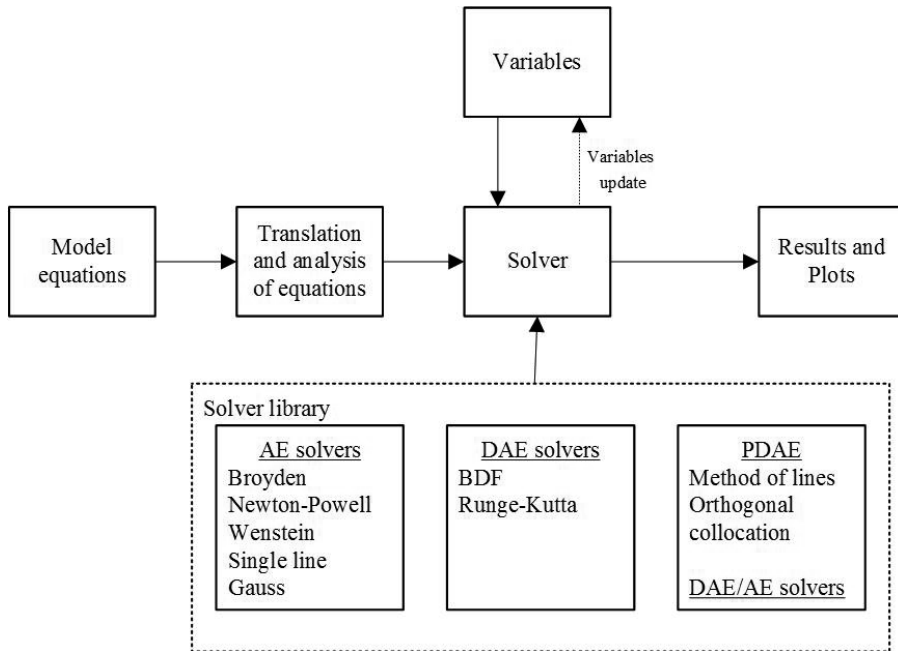


Figure 4.3. Architecture of the connection between model-solver-results within ICAS-MoT

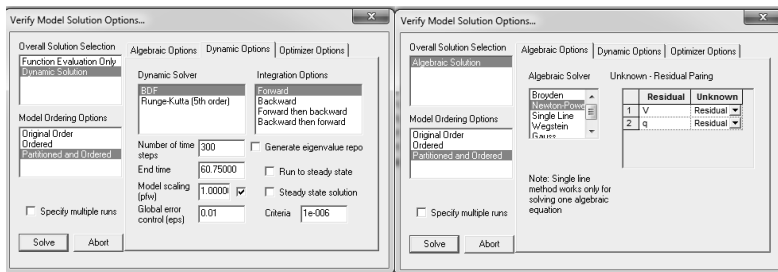


Figure 4.4. Solver options in MoT for dynamic (left) and algebraic (right) systems

Results of the model solution are presented in the table format and, in case of iterative solution procedure, the plots are illustrating results in the real time (see Figure 4.5).

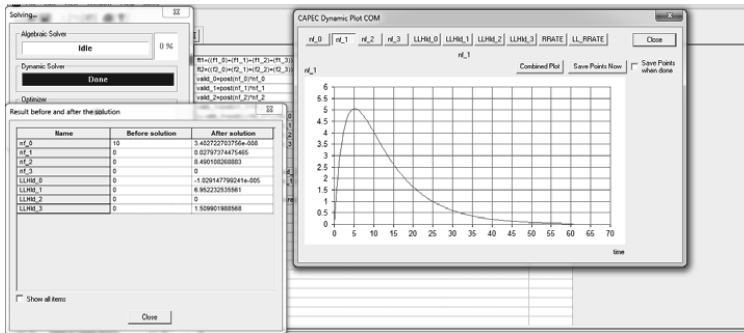


Figure 4.5. Demonstration of the results in ICAS-MoT

Additionally, MoT provides options for model analysis (sensitivity, identifiability, uncertainty analysis, parameter estimation) and optimization. For more detailed overview of these features the reader is addressed to Sales-Cruz & Gani (2003), Heitzig et al. (2011a&b), Heitzig et al. (2014).

4.2 ModDev – Model generation tool

The part of the modelling framework related to model generation is using the previously developed software tool – ModDev (Jensen, 1999). ModDev is a knowledge-based system that is able to generate the model equations for a system described by the modeller following a given structure for describing the modelling needs. In addition, it provides options for manual definition of new model building blocks and/or new variable types as well as a set of fundamental building blocks that may be combined to create the new model building blocks.

The software tool provides features for visualization of the building blocks, i.e. the modeller draws the required building blocks and describes them in the corresponding forms (see Figure 4.6). The system automatically generates the mathematical equations, based on the description provided by the modeller.

The resulting set of equations can be directly transferred to ICAS-MoT or translated to the appropriate modelling language and exported to another tool. After creation and validation, the model can be as well added to the problem-specific model library, giving, thereby, opportunities for its use as a template in other model-based applications.

More details on the implementation of the methodology can be found in Jensen (1999).

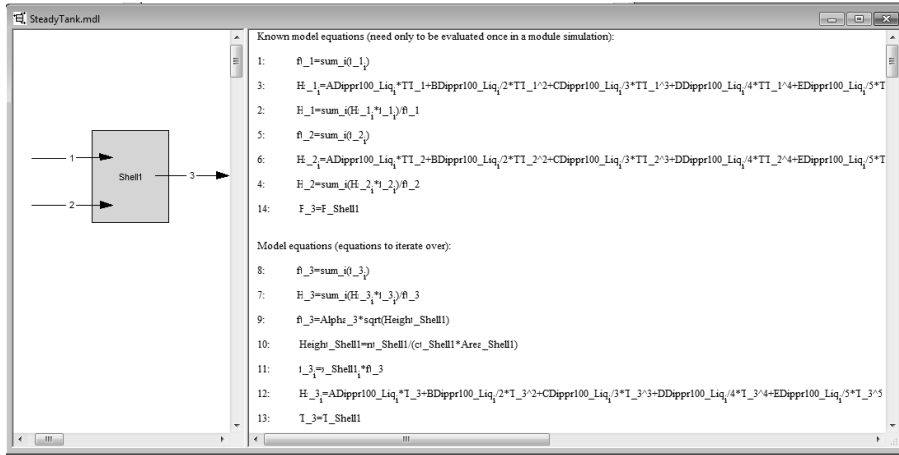


Figure 4.6. Main working area of ModDev – described system on the left-hand side and generated equations on the right-hand side

4.3 Template tool – software implementation as ModTem

The template-based modelling approach is implemented as a software tool – ModTem (modelling templates), providing easy and user-friendly interface for rapid and more efficient development and use of models. It provides an environment for creation of new templates or addition of the new building blocks for existing templates and an environment for template use. The use of the software doesn't require additional knowledge or trainings, as the interface is meant to be intuitively understandable. The tool guides the user through the steps of the workflow for template-creation or template-use. It allows the user to impose changes to an existing template, to create new templates and/or prepare an existing model for use in a specific application. The final model equations are translated as an MoT-object and can be solved and identified through the ICAS-MoT modelling platform or can be transferred to a text or XML-file in order to use it in external simulation environment. After model development and analysis is complete, the newly created model can be added to the model library and used as well for template creation and/or updates.

Frameworks for template-creation and template-use are described in the following Sections 4.3.1 and 4.3.2 respectively.

4.3.1 Template-creation framework

This part of the framework is based on the computer-aided workflow for template-creation (Figure 3.17). The framework guides the user to select options to open previously saved models, modify/update an existing template or create a new template. After model preparation according to workflow steps 1 and 2 modeller is required to choose the template from the library or to create a new template. Figure

4.7 demonstrates the starting screen of the template application software, where implementation of these two options is presented: to choose the template for the updates or to request creation of a new template.

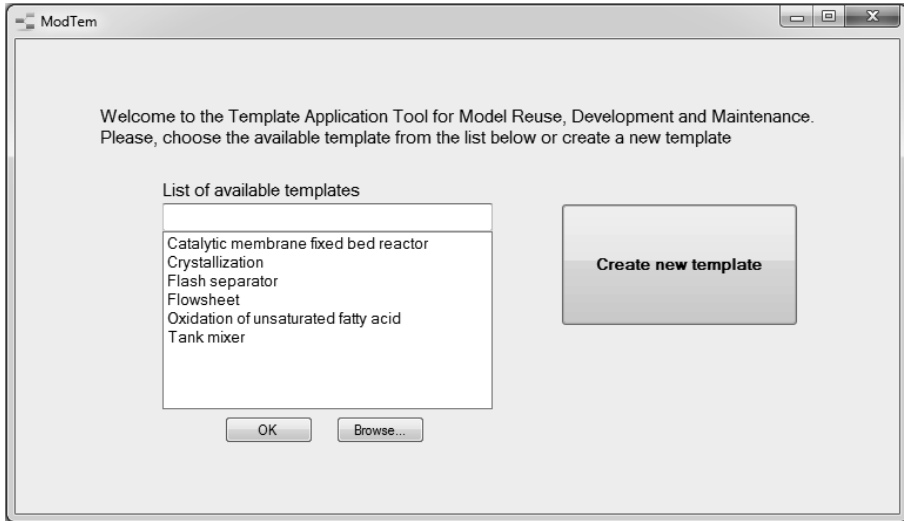


Figure 4.7. Starting screen of the template application tool

Both options will lead to the screen for template creation/update, where the parts of the template can be created (Figure 4.8 and Figure 4.9).



Figure 4.8. Screen for template creation

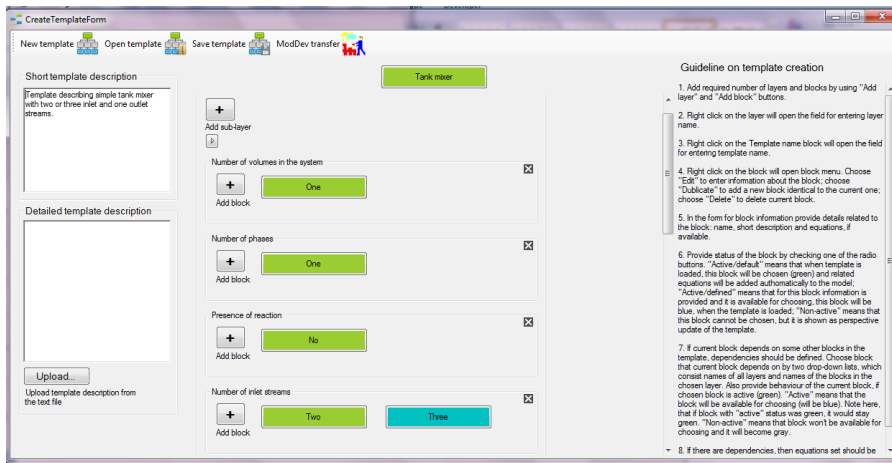


Figure 4.9. Screen for template update

Four upper-layers of “System information”, “Balance equations”, “Constitutive equations” and “Connection equations” are already present in the template by default. The modeller (to be called the “user”) developing the template will need to create the necessary sub-layers inside the four main upper-layers, based on parts of decomposed model, and assign blocks for each of the sub-layers (steps 3-6 of the template creation workflow from Figure 3.17). Software tool provides options to add/delete new sub-layers, add, delete or duplicate information blocks.

Figure 4.10 shows partially “System Information” upper-layer, the instances of sub-layer class (number of volumes, number of scales), and the data for the blocks inside these sub-layers.

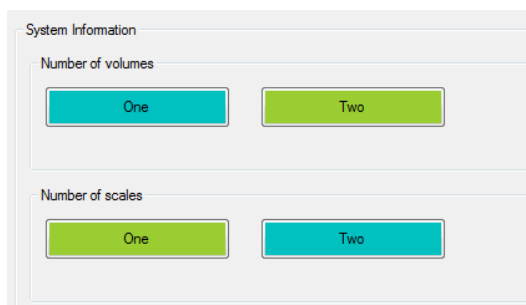


Figure 4.10. Upper-layer of system information including two sub-layers: number of volumes and number of scales, each of them contains two blocks, representing template options. Green (light-coloured) blocks indicate currently chosen options, blue (dark-coloured) shows the options of the template that are available.

In steps 7 and 8 of the workflow, the user utilizes the block information menu to provide information about blocks and connections among them (see Figure 4.11). Software tool stores required information about block description, dependencies and equations sets (for explanation, see Chapter 2).

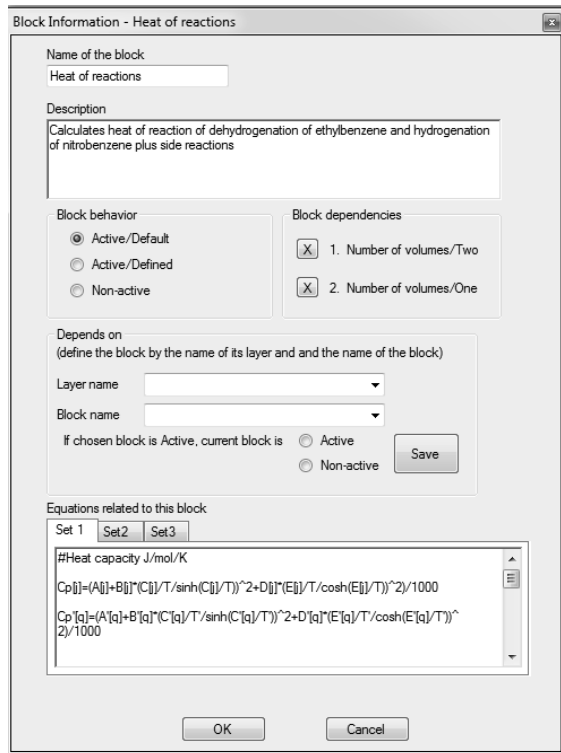


Figure 4.11. Block information menu in the template-creation tool.

Software tool can handle dependence of the block on one sub-layer. If block should depend on more than one sub-layer choice, then user should break down current sub-layer into two or more sub-layers, which will provide more detailed part of the described phenomena, or to create hierarchical dependencies for sub-layers, which are involved.

In order to finalize template-creation process the given template information should be check for consistency. Currently, this check can be done only manually, by observation, otherwise, by transferring model from the template to MoT and revising the variables and equations translation. Afterwards, the template can be saved. When newly created template is saved to the library, it will be available for choosing from the list of templates at the start of the program (Figure 4.7).

4.3.2 Template-use framework

After template is created and saved, it becomes available for utilization within template-use framework. In this framework the user is not able to make changes in the template, but can obtain new versions of the model by changing available options allowed by the template, based on the modelling goals for the corresponding model-object.

The starting point is the selection of an appropriate template from the library of created templates (Figure 4.7).

In the tool, the template is presented as a combination of upper- and sub-layers, related to the identified phenomena in the system and containing block data. Following the workflow steps (see Figure 2.6), the user is able to choose one of the blocks (if there is more than one) from the each sub-layer, and the choice depends on the system to be modelled and the assumptions that have been made in the retrieved template.

Depending on the user choice, the program applies connections between blocks that had been defined while creating template. Also, information from the chosen blocks is added to the final model. Thus, the equations from the chosen blocks are collected to represent the final set of equations. Figure 4.12 demonstrates the screenshot of the template use procedure with highlighted blocks showing user's choice and the final mathematical model on the right-hand side of the window.

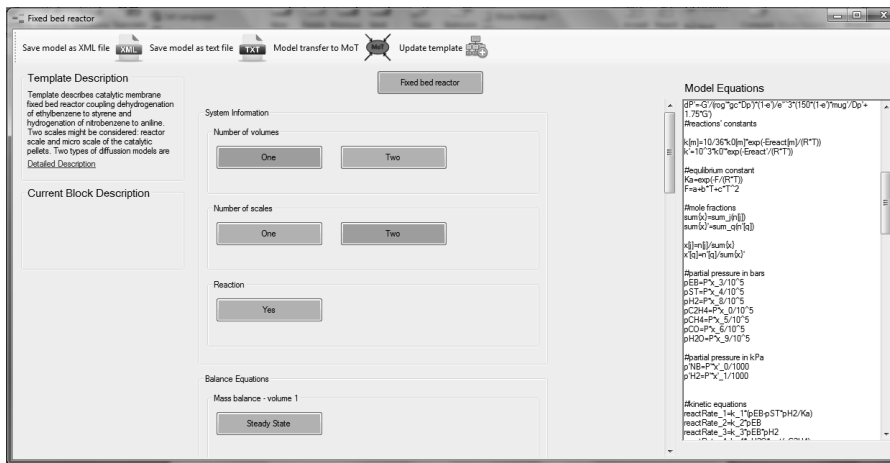


Figure 4.12. Screen of template use

Modelling equations obtained from the template can be directly transferred to ICAS-MoT for solution and analysis. The model equations together with available documentation can be also saved as a test or XML file in order to transfer it to the third-party applications. Moreover, the newly created model can be added to the model library and can be as well used for template creation and/or updates (see Figure 4.13).

As pointed in the template-use workflow (see Figure 2.6), all new models (adopted from a library model) must be checked for consistency.

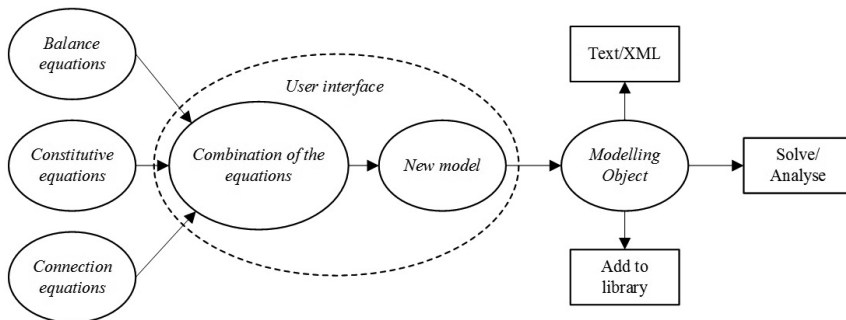


Figure 4.13. Schematic architecture of template use part of the software tool

4.4 Model library

Implementation of the model library includes MS Excel file with the list of available model, which are structured according to the model library ontology (see Section 3.2.1), and the user interface as a part of the modelling framework. The program automatically reads database from Excel file and interprets it to the user interface view.

When using model library, the modeller should choose the required model according to the ontology route. The software provides visualization of the library (see

Figure 4.14) and after the model is chosen, it automatically opens corresponding modelling file in ICAS-MoT.

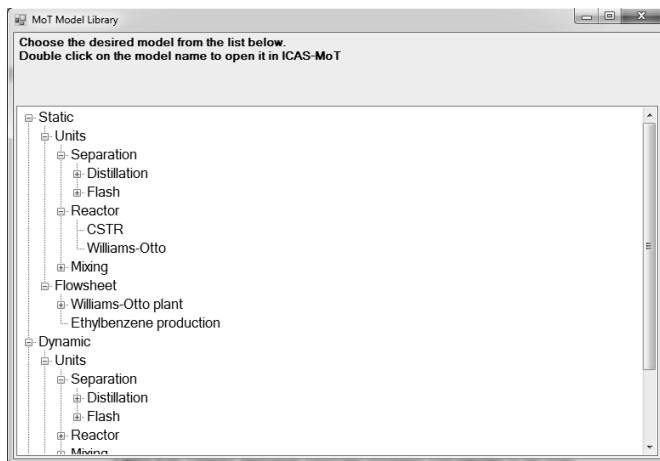


Figure 4.14. Screenshot of model library

Apart from the modelling files, the library includes also tutorial and example files for ICAS-MoT, which can be used for learning advanced features of the tool, for example, call of sub-models, then-if, backward integration etc.

4.5 XML transfer for tools integration and export/import of the models

Additional features of the generic modelling framework include integration of the computer tools within the framework as well as possibility to transfer modelling information to the external tools or, in opposite way, to import the models to the developed modelling environment. This is implemented through XML transfer (W3C, 2008). The specific XML schema has been established and it contains a formal description of the structure and elements in the XML files representing the model. Such XML file includes model equations, information about variable types and their values and indexes. Example of the XML-file with the model can be found in Appendix 7.1.

This XML schema has been implemented in all software tools involved in the developed modelling framework, allowing translation of the model information to the transfer file in XML-format and exchange of these files between tools. Modelling template tool (ModTem) and model generation tool (ModDev) can automatically translate obtained modelling equations to the XML form (see Figure 4.15 and Figure 4.16 for software screenshots). In this form information can either be directly transferred to MoT or can be saved to a XML file. Once in MoT, the XML file is interpreted and the equations are translated into MoT modelling object for further utilization within the tool. Apart from reading XML information, MoT can create XML files with constructed model as well (see Figure 4.17).

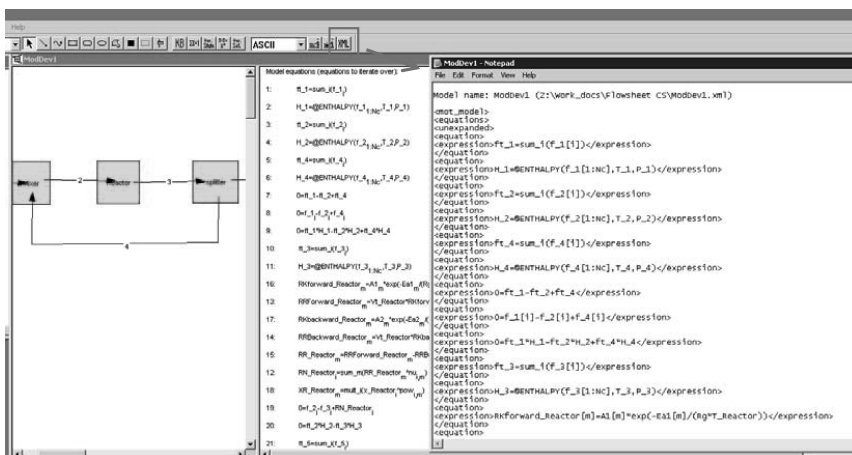


Figure 4.15. ModDev screenshot with highlight of the XML transfer

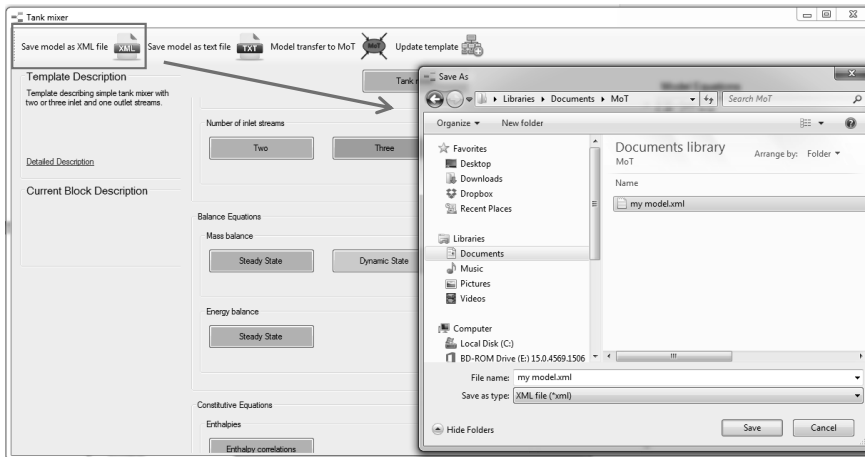


Figure 4.16. Template application (ModTem) screenshot with highlight of the XML transfer

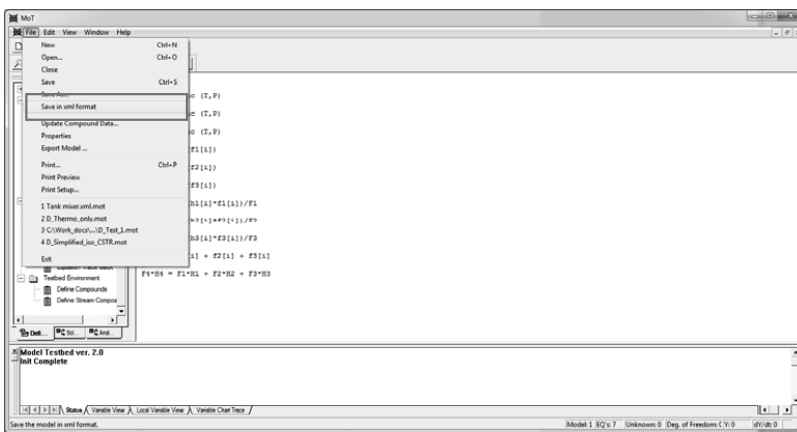


Figure 4.17. MoT screenshot with highlighted XML transfer option

In addition to internal use, the model transfer via XML-format is used for export and import of the models. This is achieved through the connection of the developed modelling framework with the MOSAIC modelling environment.

4.5.1 MOSAIC modelling environment

MOSAIC is a non-commercial, web based modelling and code generation tool developed at TU Berlin, Germany (Zerry et al., 2004). The MOSAIC models are stored in a platform independent way by using XML/MathML (Ausbrooks et al, 2001)

standards and SQL-Databases (Codd, 1970). All elements of a model (e.g. notations, equations) are exchangeable and reusable. Detailed descriptions of additional features of MOSAIC have been published, focusing on experimental design (Hoang et al. 2013), data management aspects (Kraus et al. 2014) and model integration activities (Merchan et al. 2014). The general structure of MOSAIC is shown in Figure 4.18.

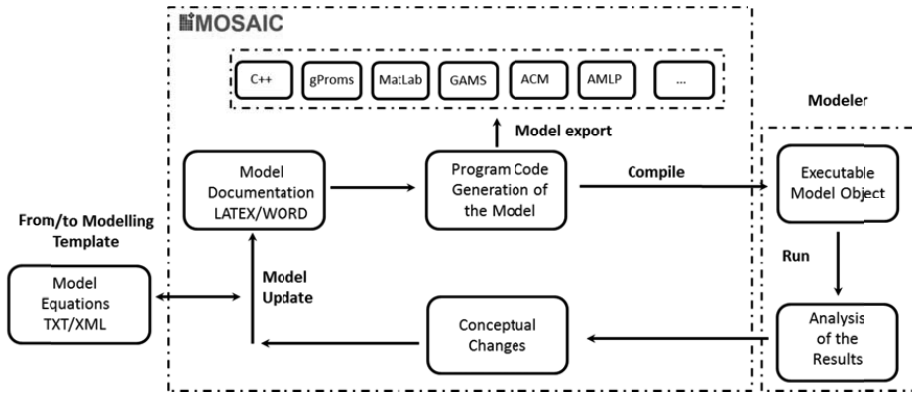


Figure 4.18. Model transfer interface and general structure of MOSAIC

4.5.2 Implementation of the extended features through connection with MOSAIC

The link between the ICAS modelling framework and the MOSAIC modelling environment is established through a XML schema, which has been implemented in both ICAS modelling framework and MOSAIC, allowing translation of the model information to the transfer file in XML-format and exchange of these files between tools. Through this link it is possible to translate models into various languages together with full model documentation. This transfer is possible through connection to the MOSAIC modelling environment, which translates XML-file with MoT model and based on it generates code for external software tools. In the same way, XML-file of the correct format can be opened in ICAS-MoT, and in this case only information about model equations is required, as MoT will extract variable information from the equations automatically.

Both modelling environments store equations in a generic unexpanded way, e.g.

$$Fz_i = Vy_i + Lx_i \quad (4.1)$$

Then based on the information about indices, e.g. $i \in \{1,2,3\}$, equations can be expanded to:

$$Fz_1 = Vy_1 + Lx_1 \quad (4.2)$$

$$Fz_2 = Vy_2 + Lx_2 \quad (4.3)$$

$$Fz_3 = Vy_3 + Lx_3 \quad (4.4)$$

According to this common structure the XML transfer file includes three main parts: “equations”, “indexing” and “variables”.

Variables can be of different types, which are also included in the transfer file. However, MOSAIC and ICAS-MoT have different indications for variable types. Therefore, the mapping of variable types has been done and a numerical type-id was introduced for the indication in the transfer file (see Table 4.1). Variables describing Dependent prime (type 6) are not explicitly expressed in MOSAIC, but for the model transfer it can be determined by analysing the equations’ differential operators.

Table 4.1 Mapping of variable types in ICAS-MoT and MOSAIC

Variable Type	ICAS-MoT	MOSAIC
1	Parameter	Parameter
2	Known	Design variable
3	Unknown	Iteration variable
4	Explicit variable	Calculated variable
5	Dependent variable	Iteration variable (state variable)
6	Dependent prime	---

Figure 4.19 illustrates the model taken from model library in ICAS-MoT format and transferred to the MOSAIC modelling environment.

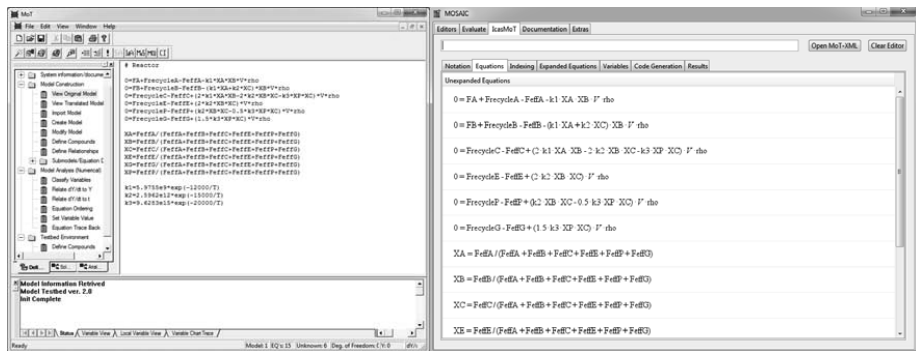


Figure 4.19. Model transfer interface and general structure of MOSAIC

4.6 Other transfer possibilities

Additionally, transfer of the model from ICAS-MoT can be done through the generation of COM-Objects. These COM-Objects can be used in MS Excel through a general Excel-COM macro interface. This interface permits loading, manipulation and evaluation of ICAS-MoT models (Sales-Cruz, 2006).

5 CASE STUDIES

This chapter presents several case studies, which are used to highlight different features of the developed template-based modelling methodology and the benefits of the tool integration and model transfer.

Case study 1. Catalytic membrane fixed bed reactor - Section 5.1.

Highlights a template-creation and a template-use for the multi-scale system with two system volumes and various options for constitutive equations. Extraction of the model based on the template options using template application tool (ModTem) and its solution through the tool for model analysis and application (MoT).

Case study 2. Oxidation of an unsaturated fatty acid with hydrogen peroxide – Section 5.2.

Highlights template-creation for the two-phase system with connection between phases. Demonstrates the development of a new case-specific model based on the template and template update as a result of the new model analysis.

Case study 3. Batch reactor and distillation – Section 5.3.

A short case study demonstrating external model transfer by connection of the developed modelling framework with MOSAIC modelling environment and describing the benefits of such integration.

Case study 4. Flowsheet template – Section 5.4.

Highlights template-creation and template-use for the development of a flowsheet model. Points out the benefits of the generic template against the case-specific model.

5.1 Catalytic membrane fixed bed reactor

Catalytic membrane fixed bed reactors are typically used in oil refinery for production of chemicals such as styrene. In this case study the reactor coupling dehydrogenation of ethylbenzene with hydrogenation of nitrobenzene is considered. The reactor is composed of two compartments within a shell containing a bundle of hydrogenation tubes. On the shell side, dehydrogenation of ethylbenzene takes place producing styrene and hydrogen. Five side reactions also occur, producing benzene, toluene, and non-condensable gases like ethylene, methane, carbon monoxide, and carbon dioxide. Inside the membrane tubes, concurrently flowing nitrobenzene reacts with hydrogen to produce aniline as a second major useful product from the integrated membrane reactor. Both reactor parts are packed with catalyst pellets (Abo-Ghander et al, 2012).

Coupling of dehydrogenation of ethylbenzene with hydrogenation of nitrobenzene in a catalytic membrane reactor can lead to significant improvements in the conversion of ethylbenzene and the production of styrene (Abo-Ghander et al., 2008). The conceptual schema of the process is shown in Figure 5.1.

The mathematical model describing process is obtained from Abo-Ghander et al (2012). The original model includes several variations, such as different modelling scales, different characterizations of diffusion inside catalytic pellet and temperature dependence.

The purpose of the case study is to present the process of template creation for the case, when numerous volumes and scales, as well as, different constitutive equations are involved. Further, the created template will be used for modelling case involving some of the options, thus highlighting Stages 2, 4 and 5 of the modelling methodology (see Figure 3.2).

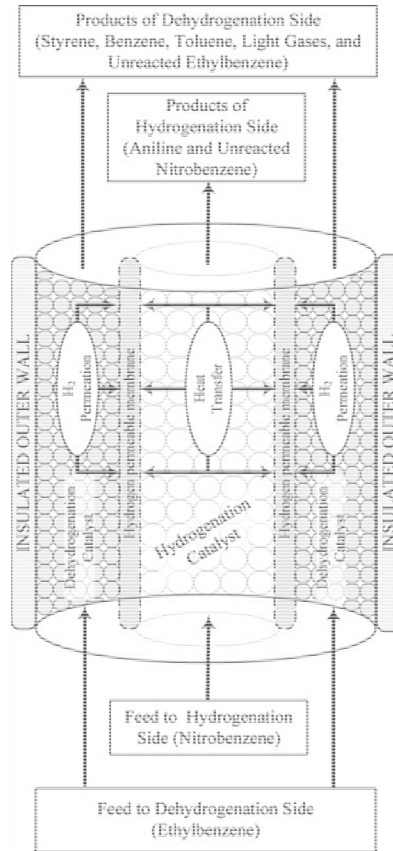


Figure 5.1. Conceptual schema of the integrated reactor proposed by Abo-Ghander et al. (2012)

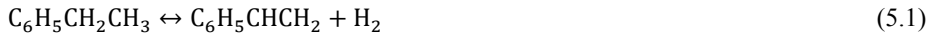
5.1.1 Template-creation

As the mathematical model is already obtained and the main goal is to create a template, Stage 6: Model storage of the modelling methodology (see **Figure 3.2**) is used directly. In order to create a template, the process model equations should be decomposed by following the steps of the template-creation workflow (**Figure 3.17**).

5.1.1.1 Step 1. Collecting system information.

The system has two volumes, separated by a membrane – hydrogenation and dehydrogenation sides. Each volume may be considered to represent a two-scale system with catalyst particles as micro-scale and the reactor as macro-scale. Reactions occur in the vapour phase. There are 6 reactions on the dehydrogenation side and one reaction on the hydrogenation side.

Dehydrogenation side



Hydrogenation side



A number of assumptions have been adopted for both the reactor model and the particle scale model (Abo-Ghander et al., 2012).

Reactor model assumptions:

1. Steady state operation.
2. Ideal gas behaviour on both the shell and the tube compartments.
3. Plug flow for the fixed beds on both the shell and the tube sides.
4. Heterogeneous model, i.e., there are concentration and temperature gradients within the catalyst particles.
5. The flow rates on both sides are large enough to minimize external mass and heat transfer resistances. Only intra-particle diffusion is considered.
6. The reactor external wall is adiabatic.
7. Catalyst deactivation is neglected.
8. Pressure gradients along both compartments are estimated based on Ergun's equation.

Catalyst pellet model assumptions:

1. Steady-state molar and energy flow.
2. Particles are spherical and their porous structure is isentropic.
3. Ideal gas law.
4. Concentration and temperature profiles within the catalyst particles are spherically symmetrical.
5. External mass and heat transfer resistances are negligible due to the high gas flow rates.
6. Negligible viscous flow inside the catalyst particles; the pellets are isobaric.
7. Convective diffusion is neglected, only ordinary molecular and Knudsen diffusion are significant for gas motion within the solid particles.
8. Convective energy transfer is negligible.
9. Thermal conductivities of the catalyst pellets are constant on both sides of the reactor.

5.1.1.2 Step 2. Model decomposition

At this step the model should be decomposed according to model decomposition technique (Cameron and Gani, 2011), in order to prepare the information for presenting it in template form. Thus, the balance, constitutive and connection equations should be identified.

Balance equations

The set of balance equations consists of mass, energy and momentum balances for both of the reactor sides, all of them are in steady state (the independent variable is the reactor length).

Dehydrogenation side:

Mass balance

$$\frac{dn_i}{dz} = \sum_{j=1}^6 \sigma_{ij} \Psi_j (1 - \varepsilon) A_{cs} \rho_s - 2\pi r_3 N a_i j_i \quad (5.8)$$

Energy balance

$$\frac{dT}{dz} = \frac{\sum_{j=1}^6 \sigma_{ij} [-\Delta H(T)]_j r_j (1 - \varepsilon) A_{cs} \rho_s + NQ}{\sum_{i=1}^{10} n_i C p_i} \quad (5.9)$$

Momentum balance

$$\frac{dP}{dz} = -\frac{G}{\rho_g g_c D_p} \left(\frac{1 - \varepsilon}{\varepsilon^3} \right) \left[\frac{150(1 - \varepsilon) \mu_g}{D_p} + 1.75G \right] \quad (5.10)$$

Hydrogenation side:

Mass balance

$$\frac{dn'_i}{dz} = \sigma_i \Psi' (1 - \varepsilon') A'_{cs} \rho'_s + 2\pi r_3 a_i j_i \quad (5.11)$$

Energy balance

$$\frac{dT'}{dz} = \frac{(2\pi r_3) \sum_{i=1}^4 a_i j_i \int_{T'}^T C p_i dT + [-\Delta H(T)]' r' (1 - \varepsilon') A'_{cs} \rho'_s - Q}{\sum_{i=1}^4 n'_i C p'_i} \quad (5.12)$$

Momentum balance

$$\frac{dP'}{dz} = -\frac{G'}{\rho'_g g_c D'_p} \left(\frac{1 - \varepsilon'}{\varepsilon'^3} \right) \left[\frac{150(1 - \varepsilon') \mu'_g}{D'_p} + 1.75G' \right] \quad (5.13)$$

Constitutive equations

Constitutive equations include reaction rates, diffusion models for catalytic pellets, including molecular diffusion and Knudsen diffusion, equations for calculating viscosity, density, thermal conductivity, mass velocity and partial pressure.

Reaction rates

Reaction rates expressions, as well as the values of kinetic and equilibrium constants and activation energies for reactions are based on studies of Elnashie et al. (1993) and Amon et al. (1999). These expressions (Eqns. (5.14)-(5.24)) characterize kinetics of chemical reactions in this case study (Eqns. (5.1)-(5.7)).

$$r_1 = k_1 \left(p_{EB} - p_{ST} \frac{p_{H_2}}{K_A} \right) \quad (5.14)$$

$$r_2 = k_2 p_{EB} \quad (5.15)$$

$$r_3 = k_3 p_{EB} p_{H_2} \quad (5.16)$$

$$r_4 = k_4 p_{H_2} p_{C_2H_4}^{1/2} \quad (5.17)$$

$$r_5 = k_5 p_{H_2} p_{CH_4} \quad (5.18)$$

$$r_6 = k_6 \left(\frac{P}{T^3} \right) p_{H_2} p_{CO} \quad (5.19)$$

$$r' = \frac{k' K_{NB} K_{H_2} p'_{NB} \sqrt{p'_{H_2}}}{\left(1 + K_{NB} p'_{NB} + K_{H_2} \sqrt{p'_{H_2}} \right)^2} \quad (5.20)$$

$$k_i = \frac{10}{36} k_{i0} \exp \left(- \frac{E_i}{RT} \right) \quad (5.21)$$

$$k' = 10^3 k'_0 \exp \left(- \frac{E'}{RT} \right) \quad (5.22)$$

$$K_A = \exp \left(- \frac{\Delta F}{RT} \right) \quad (5.23)$$

$$\Delta F = a + bT + cT^2 \quad (5.24)$$

Diffusion

Diffusion inside catalyst pellets is represented by two models: the dusty gas model (Veldsink et al, 1995) and the Fickian diffusion model (Suwanwarangkul et al, 2003). The dusty gas model is more rigorous than the Fickian diffusion model, though the latter is more widely used due to its simplicity (Abo-Ghander et al., 2012).

The dusty gas model is represented by the following equations, describing the molar flux inside the catalyst pellet. Equations are analogues for dehydrogenation and hydrogenation sides, the difference between them depend on the number of reactions and number of compounds on each side.

$$\frac{dN_i}{dy} + \frac{2}{y} N_i = \sum_{j=1}^6 \sigma_{ij} r_j \rho \quad (5.25)$$

$$\frac{P}{RT} \frac{dx_i}{dy} = \sum_{\substack{j=1 \\ j \neq i}}^{10} \frac{x_i N_j - x_j N_i}{D_{ij}^e} - \frac{N_i}{D_{ik}^e} \quad (5.26)$$

$$\frac{dN_i'}{dy'} + \frac{2}{y'} N_i' = \sigma_i' r' \rho' \quad (5.27)$$

$$\frac{P'}{RT'} \frac{dx_i'}{dy'} = \sum_{\substack{j=1 \\ j \neq i}}^4 \frac{x_i' N_j' - x_j' N_i'}{D_{ij}^{e'}} - \frac{N_i'}{D_{ik}^{e'}} \quad (5.28)$$

Alternatively, the diffusion inside the catalyst pellets is described by the Fickian diffusion model:

$$\frac{d^2 C_i}{dy^2} + \frac{2}{y} \frac{dC_i}{dy} = -\frac{1}{D_{ie}} \sum_{j=1}^6 \sigma_{ij} r_i \rho \quad (5.29)$$

$$\frac{d^2 C_i'}{dy'^2} + \frac{2}{y'} \frac{dC_i'}{dy'} = -\frac{\sigma_i' r' \rho}{D_{ie}'} \quad (5.30)$$

The energy flux inside the catalyst pellet is described by the following equations for both sides of the reactor:

$$\frac{d^2 T}{dy^2} + \frac{2}{y} \frac{dT}{dy} = \frac{1}{k_e} \sum_{j=1}^6 [\Delta H(T)]_j r_j \rho \quad (5.31)$$

$$\frac{d^2 T'}{dy'^2} + \frac{2}{y'} \frac{dT'}{dy'} = \frac{1}{k_e'} [\Delta H(T)]' r' \rho \quad (5.32)$$

Both diffusion models include also equations for calculating Knudsen and molecular diffusion, as well as effective diffusivities:

$$D_{ij} = \frac{0.00143 T^{1.75}}{P \sqrt{\frac{2}{\frac{1}{M_i} + \frac{1}{M_j}} \left[\sum_i^{1/3} + \sum_j^{1/3} \right]^2}} \quad (5.33)$$

$$D_{ki} = \frac{2}{3} r_{\text{pore}} \sqrt{\frac{8RT}{\pi M_i}} \quad (5.34)$$

$$D_{ij}^e = \frac{\varepsilon}{\tau} D_{ij} \quad (5.35)$$

$$D_{ki}^e = \frac{\varepsilon}{\tau} D_{ki} \quad (5.36)$$

Other constitutive equations

Other constitutive equations are correlations for viscosity, gas density, heat capacity and thermal conductivity, which are retrieved from the ICAS database and connected to the modelling framework.

$$\mu_i = \frac{A_i T^{B_i}}{1 + \frac{C_i}{T} + \frac{D_i}{T^2}} \quad (5.37)$$

$$\rho_g = \frac{PM_{\text{mix}}}{RT} \quad (5.38)$$

$$C_{p_i} = A_i + B_i \left(\frac{C_i}{T \sinh\left(\frac{C_i}{T}\right)} \right)^2 + D_i \left(\frac{E_i}{T \cosh\left(\frac{E_i}{T}\right)} \right)^2 \quad (5.39)$$

$$k_{t_i} = \frac{A_i T^{B_i}}{1 + \frac{C_i}{T} + \frac{D_i}{T^2}} \quad (5.40)$$

Mass velocity is calculated as:

$$G = \frac{\dot{m}}{Acs} \quad (5.41)$$

Connection equations

Connection equations consist of connection between the volumes and connection between the two scales.

Connection between volumes

Two volumes are connected by hydrogen transfer across membrane from volume 1 to volume 2 and by heat transfer from volume 2 to volume 1. These equations including any additional support expressions are:

$$J_{H_2} = \frac{Q_0 \exp\left(-\frac{E_{H_2,P}}{RT}\right)}{\delta_{H_2}} \left(\sqrt{P_{H_2}} - \sqrt{P'_{H_2}} \right) \quad (5.42)$$

$$Q = \frac{2\pi r_1 (T' - T)}{\left[\frac{1}{h'} + \frac{r_1}{k_{ss}} \ln\left(\frac{r_2}{r_1}\right) + \frac{r_1}{k_{pd}} \ln\left(\frac{r_3}{r_2}\right) + \frac{r_1}{r_2 h} \right]} \quad (5.43)$$

$$\frac{hD_t}{k_g} = 0.813 \left(\frac{D_p G}{\mu_g} \right)^{0.9} \exp\left(-\frac{6D_p}{D_t}\right) \quad (5.44)$$

$$\frac{h'D'_t}{k'_g} = 3.50 \left(\frac{D'_p G'}{\mu'_g} \right)^{0.7} \exp\left(-\frac{4.6D'_p}{D'_t}\right) \quad (5.45)$$

Connection between scales

Two scales – catalytic pellet and reactor – are connected through the effectiveness factors, which allow the evaluation of the actual reaction rates and the actual heat released/absorbed. These factors are represented by:

$$\eta_j = \frac{\int_0^{R_p} r_j \rho (4\pi y^2) dy}{r_j \rho \left(\frac{4}{3} \pi R_p^3 \right)} \quad (5.46)$$

$$\eta_j r_j(T_s, C_{is}) = 3 \int_0^{1.0} r_j \omega^2 d\omega = \Psi_j \quad (5.47)$$

$$\eta' = \frac{\int_0^{R_p} r' \rho' (4\pi y'^2) dy'}{r' \rho' (\frac{4}{3} \pi R_p^3)} \quad (5.48)$$

$$\eta' r'(T_s, C_{is}) = 3 \int_0^{1.0} r' \omega'^2 d\omega' = \Psi' \quad (5.49)$$

Calculated terms (5.47) and (5.49) are used to update the reactor model – equations (5.8) and (5.11).

Diffusion and temperature dependence of catalytic pellets depend on the choice of the model scale. Thus, if only reactor scale is chosen the template options related to catalytic pellets should become redundant (equations (5.25)-(5.39)).

5.1.1.3 Step 3. Check if template is available

This case study is highlighting the creation of a template for a new model. So, the template does not exist.

5.1.1.4 Step 4B. Create new template

Since there is no available template, a new one should be created.

5.1.1.5 Step 5B. Derive parts of the model

In this step equations introduced in step 2 are allocated to the upper-layers of the template and the required number of sub-layers and blocks are created. Upper-layer of system information includes sub-layers reflecting the number of volumes in the system, number of scales, as catalytic pellets can be considered as micro-scale, and the presence of the reactions. Upper-layer of balance equations includes sub-layers for mass, energy and momentum balances for both volumes and mass balance for multi-scale option. Upper-layer of constitutive equations includes sub-layers for diffusion, temperature dependence, viscosity, gas density, heat capacity and thermal conductivity. Upper-layer of connection equations includes sub-layers for connection between volumes and connection between scales.

5.1.1.6 Step 6. Provide template description

The following description is added – the template represents the catalytic membrane fixed bed reactor coupling dehydrogenation of ethylbenzene to produce styrene with hydrogenation of nitrobenzene to produce aniline. Two scales might be considered: reactor scale and micro scale of the catalytic pellets. Two types of diffusion models are described: dusty gas model and Fickian diffusion model.

5.1.1.7 Step 7. Provide description of the blocks

Each sub-layer created in the step 5 contains one or more blocks, which represent available template options. The following information for the blocks should be provided: equations related to the phenomena, which are represented by the current block, informative description and block behaviour. At this step equations from Section 5.1.1.2 are distributed to the blocks. Full description of the template blocks will be given below in Section 5.1.1.9.

5.1.1.8 Step 8. Define connection between blocks

Connections between blocks are defined by user. For each block this includes information about the conditioning block and the sub-layer to which it belongs. Figure 5.2 shows the created template structure for the catalytic membrane fixed bed reactor reflecting the main options of the template and connections between them. Different patterns of the blocks reflect how the template options from upper-layers II, III and IV depend on the choices from level I (if the block has a certain pattern, then it depends on the choice from the layer with the same pattern; white blocks do not have dependencies).

Catalytic Membrane Fixed Bed Reactor			
I - System Information			
Number of Volumes	<input checked="" type="checkbox"/> One	<input checked="" type="checkbox"/> Two	
Number of Scales	<input checked="" type="checkbox"/> One	<input checked="" type="checkbox"/> Two	
II - Balance equations			
Mass	<input checked="" type="checkbox"/> Volume 1	<input checked="" type="checkbox"/> Volume 2	<input checked="" type="checkbox"/> Micro-scale
Energy	<input checked="" type="checkbox"/> Volume 1	<input checked="" type="checkbox"/> Volume 2	
Momentum	<input checked="" type="checkbox"/> Volume 1	<input checked="" type="checkbox"/> Volume 2	
III - Constitutive equations			
Diffusion	<input checked="" type="checkbox"/> Dusty gas model	<input checked="" type="checkbox"/> Fickian model	
Isothermal pellets	<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> No	
IV - Connection equations			
Between volumes	<input checked="" type="checkbox"/> Hydrogen and heat transfer		
Between scales	<input checked="" type="checkbox"/> Effectiveness factor		

Figure 5.2. General structure for the catalytic membrane fixed bed reactor template.

5.1.1.9 Step 9. Check template consistency

The overview of the developed template is presented below and this will give the opportunity to check the template consistency. The template information includes description of each block in each sub-layer. Status of the block is defined in the default situation (i.e. when default choices provided by template are utilized).

5.1.1.9.1 Upper-layer I – System information

Sub-layer – Number of volumes

Block 1

Name: One

Description: Only one volume is present in the system

Status: Available to choose

Connections: -

Equations: -

Block 2

Name: Two

Description: Two system volumes are considered

Status: Chosen by default

Connections: -

Equations: -

Sub-layer – Number of scales

Block 1

Name: One

Description: One reactor scale is considered

Status: Chosen by default

Connections: -

Equations: -

Block 2

Name: Two

Description: Two scales are considered – reactor and catalytic pellet inside reactor

Status: Available to choose

Connections: -

Equations: -

Sub-layer – Reaction I

Block 1

Name: Yes

Description: Reactions take place in the system

Status: Chosen by default

Connections: -

Equations: -

Sub-layer – Reaction II

Block 1

Name: Styrene

Description: Reaction set includes equations representing production of styrene

Status: Not available

Connections: Available, if option “Number of volumes/One” is selected

Equations: -

Block 2

Name: Aniline

Description: Reaction set includes equations representing production of aniline

Status: Not available

Connections: Available, if option “Number of volumes/One” is selected

Equations: -

5.1.1.9.2 Upper-layer II – Balance equations

Sub-layer – Mass balance – Volume 1

Block 1

Name: Steady state

Description: Mass balance of the one volume in the system

Status: Chosen by default

Connections: Depends on the choice from sub-layers “System information/Reaction II” and “System information/Number of volumes”

Equations: If option “System information/Number of volumes/Two” is selected – Eqn. (5.8)

If option “System information/Reaction II/Styrene” is selected:

$$\frac{dn_i}{dz} = \sum_{j=1}^6 \sigma_{ij} (1 - \varepsilon) A_{cs} \rho_s \quad (5.50)$$

If option “System information/Reaction II/Aniline” is selected:

$$\frac{dn'_i}{dz} = \sigma_i (1 - \varepsilon') A'_{cs} \rho'_s \quad (5.51)$$

Sub-layer – Mass balance – Volume 2

Block 1

Name: Steady state

Description: Mass balance of the second volume in the system

Status: Chosen by default

Connections: Depends on the choice from sub-layer “System information/Number of volumes”

Equations: If option “System information/Number of volumes/Two” is selected – Eqn. (5.11)

Sub-layer – Energy balance – Volume 1

Block 1

Name: Steady state

Description: Energy balance of the one volume in the system

Status: Chosen by default

Connections: Depends on the choice from sub-layers “System information/Reaction II” and “System information/Number of volumes”

Equations: If option “System information/Number of volumes/Two” is selected – Eqn. (5.9)

If option “System information/Reaction II/Styrene” is selected:

$$\frac{dT}{dz} = \frac{\sum_{j=1}^6 \sigma_{ij} [-\Delta H(T)]_j r_j (1 - \varepsilon) A_{cs} \rho_s}{\sum_{i=1}^{10} n_i C p_i} \quad (5.52)$$

If option “System information/Reaction II/Aniline” is selected:

$$\frac{dT'}{dz} = \frac{[-\Delta H(T)]' r' (1 - \varepsilon') A'_{cs} \rho'_s}{\sum_{i=1}^4 n'_i C p'_i} \quad (5.53)$$

Sub-layer – Energy balance – Volume 2

Block 1

Name: Steady state

Description: Energy balance of the second volume in the system

Status: Chosen by default

Connections: Depends on the choice from sub-layers “System information/Reaction II” and “System information/Number of volumes”

Equations: If option “System information/Number of volumes/Two” is selected – Eqn. (5.12)

Sub-layer – Momentum balance – Volume 1

Block 1

Name: Steady state

Description: Momentum balance of the one volume in the system

Status: Chosen by default

Connections: Depends on the choice from sub-layers “System information/Reaction II” and “System information/Number of volumes”

Equations: If option “System information/Number of volumes/Two” is selected – Eqn. (5.10)

If option “System information/Reaction II/Styrene” is selected – Eqn. (5.10)

If option “System information/Reaction II/Aniline” is selected – Eqn. (5.13)

Sub-layer – Momentum balance – Volume 2

Block 1

Name: Steady state

Description: Momentum balance of the second volume in the system

Status: Chosen by default

Connections: Depends on the choice from sub-layers “System information/Reaction II” and “System information/Number of volumes”

Equations: If option “System information/Number of volumes/Two” is selected – Eqn. (5.13)

5.1.1.9.3 Upper-layer III – Constitutive equations

Sub-layer – Diffusion

Block 1

Name: Dusty gas

Description: Diffusion inside catalyst pellet is described by dusty gas model

Status: Not active

Connections: Depends on the choice from sub-layers “System information/Number of scales”

Equations: If option “System information/Number of scales/Two” is selected – Eqns. (5.25)-(5.28), (5.33)-(5.36).

Block 2

Name: Fickian model

Description: Diffusion inside catalyst pellet is described by Fickian model

Status: Not active

Connections: Depends on the choice from sub-layers “System information/Number of scales”

Equations: If option “System information/Number of scales/Two” is selected – Eqns. (5.29)-(5.30), (5.33)-(5.36).

Sub-layer – Isothermal catalyst

Block 1

Name: Yes

Description: Catalytic pellets are considered to be isothermal

Status: Not active

Connections: Depends on the choice from sub-layers “System information/Number of scales”

Equations: If option “System information/Number of scales/Two” is selected – Eqn. (5.31)

Block 2

Name: No

Description: Catalytic pellets are considered to be non-isothermal

Status: Not active

Connections: Depends on the choice from sub-layers “System information/Number of scales”

Equations: If option “System information/Number of scales/Two” is selected – Eqn. (5.32)

Sub-layer – Reaction rates

Block 1

Name: Reaction rates

Description: Reaction rates of dehydrogenation of ethylbenzene and hydrogenation of nitrobenzene plus side reactions

Status: Chosen by default

Connections: Depends on the choice from sub-layers “System information/Reaction II” and “System information/Number of volumes”

Equations: If option “System information/Number of volumes/Two” is selected – Eqns. (5.14)-(5.24).

If option “System information/Reaction II/Styrene” is selected – Eqns. (5.14)-(5.19), (5.21), (5.23)-(5.24).

If option “System information/Reaction II/Aniline” is selected – Eqns. (5.20), (5.22).

Sub-layer – Thermal conductivity

Block 1

Name: Gas mixture thermal conductivity

Description: Calculates thermal conductivity of gas mixture in order to calculate heat transfer across membrane. Valid only if two volumes.

Status: Chosen by default

Connections: Depends on the choice from sub-layer “System information/Number of volumes”

Equations: If option “System information/Number of volumes/Two” is selected – Eqn. (5.40)

Sub-layer – Density

Block 1

Name: Gas mixture density

Description: Calculates density of the gas mixture in one or both volumes of the system.

Status: Chosen by default

Connections: Depends on the choice from sub-layers “System information/Reaction II” and “System information/Number of volumes”

Equations: (5.38)

5.1.1.9.4 Upper-layer IV – Connection equations

Sub-layer – Volumes connection

Block 1

Name: Hydrogen and heat transfer across membrane

Description: Describes hydrogen and heat transfer across membrane.

Status: Chosen by default

Connections: Depends on the choice from sub-layer “System information/Number of volumes”

Equations: If option “System information/Number of volumes/Two” is selected – Eqns. (5.41)-(5.45).

Sub-layer – Scales connection

Block 1

Name: Connection between scales

Description: Connection between scales through the correction factor

Status: Chosen by default

Connections: Depends on the choice from sub-layer “System information/Number of scales”.

Equations: If option “System information/Number of scale/Two” is selected – Eqns. (5.46)-(5.49).

5.1.1.10 Step 10. Finalization of the template

After template is checked it is saved to the template library that makes it available for use at the modelling stage 2: Model reuse.

5.1.2 Template-use

In this section the model reuse feature of the modelling framework will be highlighted throughout the template-use. The template created through the template-creation tool can be used in different applications through the template-use tool. In this case study the objective is to obtain the model for the catalytic membrane fixed bed reactor.

Two different scenarios are considered. Objective of the first is to obtain the model, which takes into account diffusion inside catalyst pellets and uses the combination of the options in the template. Another scenario will consider one-volume reactor, thereby, using modification of the initial model throughout the template. While working with the template in both case studies, the user extracts model information based on the modelling objective. The user should choose one information block among all the options from each of the sub-layers. Further, the obtained models will be transferred directly to ICAS-MoT for model analysis and simulation.

5.1.2.1 Scenario 1. Creation of model through template options

In this scenario the model will be obtained by model reuse through the template. The modelling objective is to get the model for catalytic membrane fixed bed reactor taking into account diffusion inside catalytic pellets. The Stage 2 (model reuse) of the framework will include the template-use procedure. The workflow for the template to be employed in this case study is the following (see workflow in Figure 2.6).

Step 1. Template for the catalytic membrane fixed bed reactor is chosen.

Step 2. System has two volumes, separated by membrane; one vapour phase; 6 reactions on the dehydrogenation side and one reaction on the hydrogenation side. Every volume is considered as 2-scale system with catalyst particles as micro-scale and reactor as macro-scale.

The following choices are made in the upper-layer “System information”:

- Sub-layer “Number of volumes” – Block “Two”
- Sub-layer “Number of scales” – Block “Two”
- Sub-layer “Reaction I” – Block “Yes”

Step 3. For both volumes models are in steady state, distributed in angular direction, for mass, energy and momentum balances (the independent variable is the reactor length).

The following choices are made in the upper-layer “Balance equations”:

- Sub-layer “Mass balance – Volume 1” – Block “Steady state”
- Sub-layer “Mass balance – Volume 2” – Block “Steady state”
- Sub-layer “Energy balance – Volume 1” – Block “Steady state”
- Sub-layer “Energy balance – Volume 2” – Block “Steady state”
- Sub-layer “Momentum balance – Volume 1” – Block “Steady state”
- Sub-layer “Momentum balance – Volume 2” – Block “Steady state”

Step 4. Reaction rate model is selected based on availability in the template. Catalyst pellets are chosen to be isothermal. Diffusion inside catalyst pellets is represented by dusty gas model.

The following choices are made in the upper-layer “Constitutive equations”:

- Sub-layer “Diffusion” – Block “Dusty gas”
- Sub-layer “Isothermal pellets” – Block “Yes”
- Sub-layer “Reaction rates” – Block “Reaction rates”
- Sub-layer “Thermal conductivity” – Block “Gas mixture thermal conductivity”
- Sub-layer “Density” – Block “Gas mixture density”

Step 5. As the system is considered to have two volumes and two scales, options for connection equations should be chosen from the template. Two volumes are connected by hydrogen transfer across membrane from volume 1 to volume 2 and by heat transfer from volume 2 to volume 1. Micro-scale is connected to reactor scale through effectiveness factor.

The following choices are made in the upper-layer “Connection equations”:

- Sub-layer “Volumes connection” – Block “Hydrogen and heat transfer across membrane”
- Sub-layer “Scales connection” – Block “Connection between scales”

Step 6. Final model is constructed based on answers from steps 1-5. Employed equations are Eqns. (5.8)-(5.28), (5.33)-(5.49). The system automatically includes required changes in the equations according to the user’s choice. Resulting model

contains of 46 ODEs and 582 AEs, including among others 14 ODEs for mass balance, 4 ODEs for energy and momentum balance, 25 AEs obtained from reaction rates sub-layer, 28 ODEs and 260 AEs describing diffusion inside pellets.

After generating the final model, it has been translated into a model object and solved by using ICAS-MoT. Figure 5.3 shows the results of the simulation for styrene production and temperature profile on the dehydrogenation side.

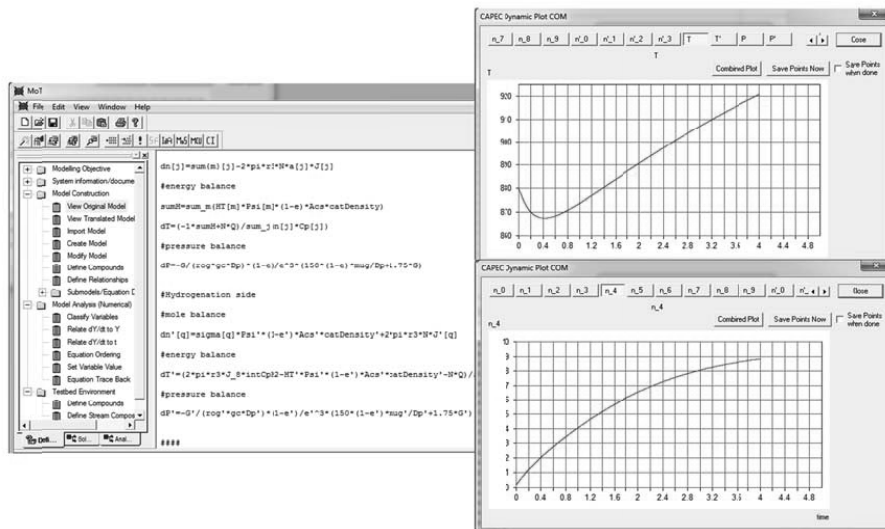


Figure 5.3. Model for scenario 1 represented in ICAS-MoT: model equations and plots of temperature on the dehydrogenation side and styrene production.

5.1.2.2 Scenario 2. Creation of the model through additional template options

This scenario also considers reuse of the model through the template. However, in this case the options of the template that were generated based on the original model will be used.

The modelling objective is to get the model for catalytic fixed bed reactor for production of styrene. The workflow for the template to be employed in this case study is the following.

Step 1. Template for the catalytic membrane fixed bed reactor is chosen.

Step 2. System has one volume; one vapour phase; 6 reactions for production of styrene. Diffusion inside the catalytic pellets is not considered in this scenario.

The following choices are made in the upper-layer “System information”:

- Sub-layer “Number of volumes” – Block “One”
- Sub-layer “Number of scales” – Block “One”
- Sub-layer “Reaction I” – Block “Yes”
- Sub-layer “Reaction II” – Block “Styrene”

Step 3. Balance models are in steady state, distributed in angular direction, for mass, energy and momentum (the independent variable is the reactor length).

The following choices are made in the upper-layer “Balance equations”:

- Sub-layer “Mass balance – Volume 1” – Block “Steady state”
- Sub-layer “Energy balance – Volume 1” – Block “Steady state”
- Sub-layer “Momentum balance – Volume 1” – Block “Steady state”

Step 4. Reaction rate model is selected based on availability in the template. Catalyst pellets are chosen to be isothermal. Diffusion inside catalyst pellets is represented by dusty gas model.

The following choices are made in the upper-layer “Constitutive equations”:

- Sub-layer “Reaction rates” – Block “Reaction rates”
- Sub-layer “Density” – Block “Gas mixture density”

Step 5. As only one volume and one scale are considered in the system, no connection equations are needed. Only closure equations are added to the model at this step.

Step 6. Final model is constructed based on answers from steps 1-5. Employed equations are Eqns. (5.8)-(5.10), (5.14)-(5.19), (5.21), (5.23)-(5.24), (5.37)-(5.39), (5.41). The system automatically includes required changes in the equations according to the user’s choice. Resulting model contains of 12 ODEs and 291 AEs, including among others 10 ODEs for mass balance, 2 ODEs for energy and momentum balance, 21 AEs obtained from reaction rates sub-layer.

Further, the obtained model is directly transferred to the tool for analysis and solution (ICAS-MoT). Figure 5.4 demonstrates results of the model solution for scenario 2.

From the plots one can see the drop of the styrene amount along the reactor due to the drop in temperature and therefore favour of the reverse reaction (5.1). In scenario 1 this has been avoided by coupling the reactions.

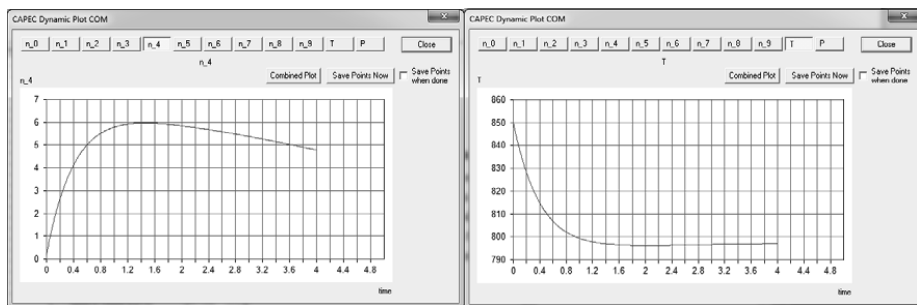


Figure 5.4. Model results for scenario 2 represented in ICAS-MoT: plots of styrene production (left) and temperature profile along the reactor (right).

5.1.3 Case study conclusion

The newly created template allows generation of several problem-specific models by using different combinations of the template options, such as one- or multi-scale option or different diffusion model option (see Figure 5.2 for all options). Moreover, by generating new options based on the available choice a new general model could also be obtained for use as a starting point for further model development and analysis.

Further, the different parts of the template can be updated giving the opportunity to develop new options and new combinations for model reuse. This updates may include, for example, another set of reactions taken place in the reactor. Further solution, analysis and possible optimization of the obtained model can be done by using other parts of the modelling framework (see Figure 4.1).

5.2 Oxidation of unsaturated fatty acid with hydrogen peroxide

The second case study is related to oxidation of unsaturated fatty acids with hydrogen peroxide. The originally considered system includes two phases – oil and water, with phase transfer agent delivering oxidizer from the water phase to the oil phase, where reactions take place. The desired products of the process are short-chained acids, which are the result of cleavage of the unsaturated fatty acid.

This case study highlights the use of the template application for multiphase system with various reactions and mass transfer between phases. Also, it shows the benefits of getting the preliminary model of the system from the template for further model development.

5.2.1 Template-creation

The used model has been developed by Papadakis (2013). In his work the main reaction path for the two phase model has been described and conceptual model has been developed and qualitatively verified.

5.2.1.1 Step 1. Collecting system information

Figure 5.5 illustrates the fed-batch reactor to be studied, the reactions happen in the reactor with slow addition of hydrogen peroxide solution, in defined temperature (usually below 100°C) and certain reaction time (4hr-24hr) (Venturello et al., 1998). The initial reaction mixture consists of unsaturated fatty acid (organic phase), and the catalyst is based on peroxopolyoxo-tungstphosphate combined with phase transfer catalysts. The addition of hydrogen peroxide solution (aqueous solution of oxidant) is slowly added in the reactor.

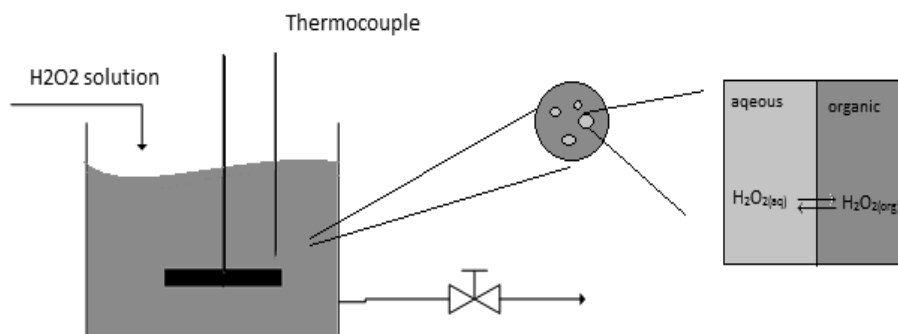


Figure 5.5. Schematic two-phase fed-batch reactor.

The reaction pathway is the following: the epoxidation of unsaturated fatty acid by hydrogen peroxide, the hydroxylation of epoxide by water to diol, and finally the bond cleavage of diol by hydrogen peroxide to carboxylic acids. Except the main reaction pathway, side reactions may occur, such as ketonization and formation of high molecular weight compounds (see Table 5.1).

Table 5.1. Reaction present in the two-phase oxidation system (Köckritz and Martin, 2008)

Chemical reaction	Comments
$\text{UFA} + \text{H}_2\text{O}_2 \xrightarrow[+\text{cat}]{k_1} \text{EP} + \text{H}_2\text{O}$	Epoxidation of unsaturated fatty acid (UFA)
$\text{EP} + \text{H}_2\text{O} \xrightarrow[\text{H}^+]{k_2} \text{D}$	Hydrolysis of epoxide (EP)
$\text{D} + \text{H}_2\text{O}_2 \xrightarrow[+\text{cat}]{k_3} \text{AD}_1 + \text{AD}_2 + 2\text{H}_2\text{O}$	Cleavage of diol (D) to aldehydes (AD)
$\text{D} + 3\text{H}_2\text{O}_2 \xrightarrow[+\text{cat}]{k_4} \text{AC}_1 + \text{AC}_2 + 4\text{H}_2\text{O}$	Direct cleavage of diol (D) to carboxylic acids (AC)
$\text{AD}_1 + \text{H}_2\text{O}_2 \xrightarrow[\text{cat}]{k_5} \text{AC}_1 + \text{H}_2\text{O}$	Oxidation of the aldehyde to acid
$\text{AD}_2 + \text{H}_2\text{O}_2 \xrightarrow[\text{cat}]{k_6} \text{AC}_2 + \text{H}_2\text{O}$	Oxidation of the aldehyde to acid
$\text{UFA} + \text{H}_2\text{O}_2 \xrightarrow[+\text{cat}]{k_7} \text{KE} + \text{H}_2\text{O}$	Ketonization (KE)
$\text{UFA} + \text{EP} \xrightarrow{k_8} \text{HWM}$	Polymerization reactions (High weight molecules, HWM)
$2\text{D} \xrightarrow{k_9} \text{HWM} + \text{H}_2\text{O}$	
$\text{EP} + \text{AC}_1 + \text{AC}_2 \xrightarrow{k_{10}} \text{HWM}$	
$\text{D} + \text{AC}_1 + \text{AC}_2 \xrightarrow{k_{11}} \text{HWM}$	
$2\text{H}_2\text{O}_{2(\text{aq})} \xrightarrow{k_d} 2\text{H}_2\text{O}_{(\text{aq})} + \text{O}_2$	Decomposition of hydrogen peroxide

The following assumptions have been made for the developed model:

1. The aqueous phase can be represented as very small water droplets; it can be assumed that there is no change in size of droplets during the reaction. The number of the droplets changes by time.
2. There are no mass transfer limitations of hydrogen peroxide while it is migrating from the aqueous phase to the organic phase.
3. The acids will not dissociate because the pH-value is always lower than the pks-value of acids.
4. No reaction products in the aqueous phase.
5. All the reactions occur in the organic phase. The hydrolysis of the epoxide occurs at the interphase of the droplet.

5.2.1.2 Step 2. Decomposition of the model

Balance equations

Mass balances are based on the different species in the system and the occurring reactions.

$$\frac{dn_i}{dt} = \left(\sum_{j=1}^{NR} v_j r_j \right) V^k \quad (5.54)$$

$$\frac{dn_{H_2O}}{dt} = \left(\sum_{j=1}^{NR} v_j r_{H_2O,j} \right) V^k + F_o c_{H_2O,o} \quad (5.55)$$

$$\frac{dn_{H_2O_2}^{aq}}{dt} = \left(\sum_{j=1}^{NR} v_j r_{H_2O_2,j} - J_{H_2O_2} \right) V^k + F_o c_{H_2O_2,o} \quad (5.56)$$

$$\frac{d(n_{H_2O_2}^{org})}{dt} = \left(\sum_{j=1}^{NR} v_j r_{H_2O_2,j} + J_{H_2O_2} \right) V^k \quad (5.57)$$

Where $i = UFA, EP, D, AC1, AC2, AD1, AD2, HWM$.

Constitutive equations

Reaction rates are based on elementary kinetics:

$$r_j = k_j \cdot c_A \cdot c_B \quad (5.58)$$

Other constitutive equations include the ideal mixture rule for the calculation of the volume of the phase k :

$$V_k = \sum_i \frac{n_i}{\rho_i^n} \quad (5.59)$$

Reparameterized Arrhenius equation for the temperature dependency of kinetic model parameters:

$$k_i = k_i^{ref} \exp \left[-\frac{E_{a,i}}{RT_K} \right] \quad (5.60)$$

Correlation for water density calculations:

$$\rho_{H_2O} = 2 \times 10^{-05} \times T^3 - 0.006 \times T^2 + 0.0227 \times T + 999.94 \quad (5.61)$$

Correlation for other compounds density can be obtained by equation (58) and the correlation parameters are to be found in the modelling framework database.

$$\rho = \left(\frac{A}{B} \right) \left(1 + \left(1 - \frac{T}{C} \right)^D \right) \quad (5.62)$$

Connection equations

Connection between volumes is expressed by the mass transfer of hydrogen peroxide:

$$J_{H_2O_2} = a_{dr} k_L \left(c_{H_2O_2}^{aq} - H_{H_2O_2} c_{H_2O_2}^{org,*} \right) \quad (5.63)$$

5.2.1.3 Step 3. Check if template is available

A template is not available in the template library.

5.2.1.4 Step 4B. Create new template

A new template is to be created.

5.2.1.5 Step 5B. Derive parts of the model

Upper-layer of system information includes sub-layers reflecting the number of phases in the system and the presence of the reactions. Upper-layer of balance equations includes sub-layers for mass balances for both phases. Upper-layer of constitutive equations includes sub-layers describing system's kinetics. Upper-layer of connection equations includes sub-layer for connection between phases.

5.2.1.6 Step 6. Provide template description

Template presents the two phase system of unsaturated fatty acid oxidation by hydrogen peroxide. Template can also include option for one phase system by excluding phase connection equations.

5.2.1.7 Step 7. Provide description of the blocks

Each sub-layer created in the step 5 contains one or more blocks, which represent available template options. In the current step the parameters of the blocks is provided. This includes equations related to the phenomena, which is represented by the current block, informative description and block behaviour. At this step equations from Section 5.2.1.1 are distributed to the blocks. Full description of the template blocks will be given below in Section 5.2.1.9.

5.2.1.8 Step 8. Define connection between blocks.

Schematic view of the main template blocks and connections are shown in Figure 5.6.

CASE STUDIES

Oxidation of unsaturated fatty acids (UFA) with hydrogen peroxide

I - System Information		
Number of Phases	One	Two
Reactions	Only main	Plus side reactions
Decomposition of H_2O_2	Yes	No
II - Balance equations		
Mass	Dynamic mass balance	
III - Constitutive equations		
Kinetics	Elementary kinetics	
IV - Connection equations		
Between phases	Transfer of hydrogen peroxide	

Figure 5.6. General structure for the unsaturated fatty acid oxidation template.

5.2.1.9 Step 9. Check template consistency

5.2.1.9.1 Upper-layer I. System information

Sub-layer – Number of phases

Block 1

Name: Two

Description: Two liquid phases are present in the system – water and oil phase

Status: Chosen by default

Connections: -

Equations: -

Sub-layer – Final product

Block 1

Name: Epoxide

Description: Desirable final product is epoxide

Status: Available to choose

Connections: -

Equations: -

Block 2

Name: Short acids

Description: Desirable final products are shorter carboxylic acids

Status: Chosen by default

Connections: -

Equations: -

Sub-layer – Reactions

Block 1

Name: Only main

Description: Only main reactions (first six reactions in Table 5.1) are considered in the model

Status: Available to choose

Connections: -

Equations: -

Block 2

Name: Includes side reactions

Description: Main and side reactions are considered in the model

Status: Chosen by default

Connections: -

Equations: -

Sub-layer – Hydrogen peroxide decomposition

Block 1

Name: Yes

Description: Hydrogen peroxide decomposition is taken into account

Status: Available to choose

Connections: -

Equations: -

Block 2

Name: No

Description: Hydrogen peroxide decomposition is not considered

Status: Chosen by default

Connections: -

Equations: -

5.2.1.9.2 Upper-layer II. Balance equations

Sub-layer – Mass balance

Block 1

Name: Dynamic state

Description: Variables describing mass balance depend on time

Status: Chosen by default

Connections: Depends on the choice from sub-layers “System information/Reactions” and “System information/Hydrogen peroxide decomposition”

Equations: Eqns. (5.54)-(5.57)

5.2.1.9.3 Upper-layer III. Constitutive equations

Sub-layer – Reactions kinetics

Block 1

Name: Elementary

Description: Elementary representation of reaction rates $r=k*c_1*c_2$

Status: Chosen by default

Connections: -

Equations: (5.58), (5.60)

5.2.1.9.4 Upper-layer IV. Connection equations

Sub-layer – Connection between phases

Block 1

Name: Hydrogen peroxide transfer

Description: Phases connected by the transfer of hydrogen peroxide from water to oil phase

Status: Chosen by default

Connections: Depends on the choice from sub-layer “System information/Number of phases”

Equations: If option “System information/Number of phases/Two” is selected – Eqn. (5.63).

5.2.1.10 Step 10. Finalization of the template

After template is checked it is saved to the template library that makes it available for use at the modelling stage 2: Model reuse.

5.2.2 Template-use

The created template is used here for developing the model for one phase system using a solvent. As the system will have extra species, more reactions will occur. However, the terms related to phase connection are not be used as there is only one phase now.

In this scenario the reaction occurs in a fed-batch reactor where a hydrogen peroxide solution in t-BuOH is added drop-wise for a certain period of time. The unsaturated fatty acid (UFA) is mixed with t-BuOH. As t-BuOH is a solvent which can dissolve both polar and non-polar compounds, the resulting mixture is a single phase system (Behr et al., 2012). Table 5.2 includes reactions occurring in single phase system in addition to the ones described in Table 5.1

Table 5.2. Addition reactions occurring in the single-phase oxidation system

Chemical reaction	Comments
$D + H_2O_2 \xrightarrow{k_7} AL + 2H_2O$	Production of alcyloins
$AL + H_2O_2 \xrightarrow{k_8} AC_1 + AD_1 + H_2O$	Alcyloins to acids and aldehydes
$AL + H_2O_2 \xrightarrow{k_9} AC_2 + AD_2 + H_2O$	
$EP + 2H_2O_2 \xrightarrow{k_{14}} 2HPAE$	Hydroperoxystearic acid methyl ester
$HPAE \xrightarrow{k_{15}} AD_1 + AD_2 + H_2O$	

In the following template use workflow the choices made by user are described.

Step 1. Select template

Previously created template for the oxidation of unsaturated fatty acids with hydrogen peroxide is selected for use.

Step 2. Select process system

The system is chosen to have a single phase, because of the use of solvent. The catalytic cleavage of UFA has main and side reactions. Desired products are shorter length carboxylic acids.

Step 3. Definition of balance equations

Mass balances are based on the different species in the system and occurring reactions. Equation (5.54) is used by template to represent balance equations. Energy and momentum balances are not considered in this case.

Step 4. Definition of constitutive equations

Template allows the use of simple kinetic equations for the reactions (Eqns. (5.58), (5.60)).

Step 5. Definition of connection equations

Studied system is in one phase, therefore, there are no connection equations as the created model is for a two-phase system.

Step 6. Finalization of the equation set

The final set of equations, obtained from the template, includes equations (5.54)-(5.55), (5.58)-(5.62).

The resulting model does not satisfy the modelling needs completely, therefore, some updates should be done. These will be discussed in the next section.

5.2.3 Model analysis and verification

Based on the equations obtained from 6.2, a final model has been developed. As one phase system includes more compounds, mass balance should be updated with equations related to additional species: acyloins and hydroperoxystearic acid methyl ester. Also model is updated with kinetic equations for additional reactions.

Further analysis of the one-phase system model by Papadakis (2013) led to improvement of the reaction rates equations. Several options had been considered and tested against experimental data. These options are following.

1. The Langmuir-Hinshelwood mechanism (Zhan et al, 2013; Gallot et al, 1998).

$$r_1 = \frac{k_1 c_{UFA} c_{H_2O_2}}{\left(1 + k_3 c_D + k_1 c_{AC_1} + k_2 c_{AC_2}\right)^2} \quad (5.64)$$

$$r_2 = \frac{k_2 c_{EP} c_{H_2O}}{\left(1 + k_3 c_D + k_1 c_{AC_1} + k_2 c_{AC_2}\right)^2} \quad (5.65)$$

2. Non-elementary kinetics (Levenspiel, 1998)

$$r_2 = \bar{k}_2 c_{EP} c_{H_2O} \frac{1}{1 + \frac{c_D}{c_{EP}}} \quad (5.66)$$

$$r_3 = \bar{k}_3 c_D c_{H_2O_2} \frac{1}{1 + \frac{c_{AC_1}}{c_D}} \quad (5.67)$$

$$r_4 = \bar{k}_4 c_D c_{H_2O_2} \frac{1}{1 + \frac{c_{AC_1}}{c_D}} \quad (5.68)$$

$$r_7 = k_7 c_D c_{H_2O_2} \frac{1}{1 + \frac{c_{AC_1}}{c_D}} \quad (5.69)$$

3. Non-elementary kinetics with higher inhibition order

$$r_2 = \bar{k}_2 c_{EP} c_{H_2O} \frac{1}{\left(1 + \frac{c_D}{c_{EP}}\right)^2} \quad (5.70)$$

$$r_3 = \bar{k}_3 c_D c_{H_2O_2} \frac{1}{\left(1 + \frac{c_{AC1}}{c_D}\right)^2} \quad (5.71)$$

$$r_4 = \bar{k}_4 c_D c_{H_2O_2} \frac{1}{\left(1 + \frac{c_{AC1}}{c_D}\right)^2} \quad (5.72)$$

$$r_7 = k_7 c_D c_{H_2O_2} \frac{1}{\left(1 + \frac{c_{AC1}}{c_D}\right)^2} \quad (5.73)$$

4. Non-elementary kinetics with higher inhibition order for epoxide

$$r_2 = \bar{k}_2 c_{EP}^2 c_{H_2O} \frac{1}{\left(1 + \frac{c_D}{c_{EP}}\right)^2} \quad (5.74)$$

Best result was shown by option number 3 – non-elementary kinetics with higher inhibition order.

Final model prediction results were verified using data by Behr et al. (2013) (Figure 5.7).

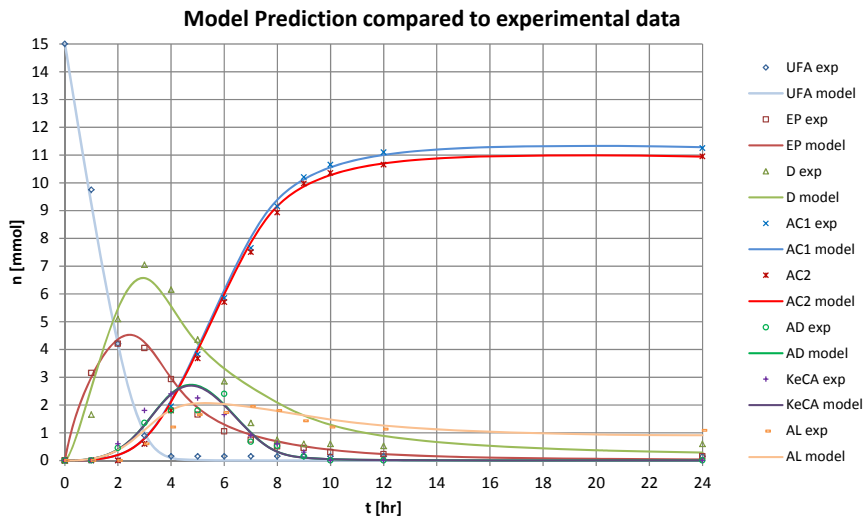


Figure 5.7. Model prediction compared to experimental data by Papadakis (2013)

Verification shows good fit of the model with SSE value of $9,68 \cdot 10^{-7}$ (Papadakis, 2013). Therefore, model development can be considered successful.

5.2.4 Template updates

In order to increase knowledge of the system and to employ the reuse of the models the developments described in the previous section 5.2.3 should be reflected in the template. Therefore, stage 6: Model storage of modelling framework and workflow for template update (Figure 2.4) are used.

5.2.4.1 Step 1. Collecting system information

Oxidation of unsaturated fatty acid with hydrogen peroxide is happening in the one phase system created by using a solvent t-BuOH. Reactions from Table 5.1 and Table 5.2 are present in the system.

5.2.4.2 Step 2. Decomposition of the model

Balance equations

Mass balances are based on the different species in the system and the occurring reactions. These equations are (5.54)-(5.55).

Constitutive equations

Reaction rates are based on non-elementary kinetics with higher inhibition order – Eqns. (5.70)-(5.73).

Other constitutive equations are identical to the ones described in section 5.2.1.2.

Connection equations

As the system includes only one phase no connection equations are present in the model.

5.2.4.3 Step 3. Check if template is available

Template for oxidation of unsaturated fatty acid with hydrogen peroxide is available in the template library.

5.2.4.4 Step 4A. Choose template for updates

Template found at Step 3 will be used for updates.

5.2.4.5 Step 5A. Derive new parts of the model

Upper-layer of system information includes option for one phase in the corresponding sub-layer. Upper-layer of constitutive equations includes sub-layers describing several kinetics options for one-phase system.

5.2.4.6 Step 6. Update template description

Template presents the two phase system of unsaturated fatty acid oxidation by hydrogen peroxide. Template can also include option for one phase system by excluding phase connection equations.

5.2.4.7 Step 7. Provide description of the blocks

At this step additional equations describing one-phase system from Section 5.2.4.2 are distributed to the blocks. Full description of the template blocks will be given below in Section 5.2.4.9.

5.2.4.8 Step 8. Define connection between blocks.

General connection scheme is identical to the one presented in Figure 5.6.

5.2.4.9 Step 9. Check template consistency

Only blocks that have been added or changed to the system during the update are described.

5.2.4.9.1 Upper-layer I. System information*Sub-layer – Number of phases*Block 2

Name: One

Description: System is present in one phase

Status: Available to choose

Connections: -

Equations: -

5.2.4.9.2 Upper-layer II. Balance equations*Sub-layer – Mass balance*Block 1

Name: Dynamic state

Description: Variables describing mass balance depend on time

Status: Chosen by default

Connections: Depends on the choice from sub-layers “System information/Reactions” and “System information/Hydrogen peroxide decomposition”

Equations: If option “System information/Number of phases/One” is selected – Eqns. (5.54)-(5.55).

5.2.4.9.3 Upper-layer III. Constitutive equations*Sub-layer – Reactions kinetics*Block 2

Name: Langmuir-Hinshelwood

Description: Use of Langmuir-Hinshelwood mechanism for reaction rates

Status: Available to choose

Connections: -

Equations: (5.64)-(5.65)

Block 3

Name: Non-elementary

Description: Use of non-elementary kinetics for reaction rates expressions

Status: Available to choose

Connections: -

Equations: (5.66)-(5.69)

Block 4

Name: Non-elementary inhibition order

Description: Use of non-elementary kinetics with higher inhibition order

Status: Available to choose

Connections: -

Equations: (5.70)-(5.73)

Block 5

Name: Non-elementary epoxide order

Description: Use of non-elementary kinetics with higher order for epoxide

Status: Available to choose

Connections: -

Equations: (5.74)

5.2.4.10 Step 10. Finalization of the template

Updated template is saved to the template library.

5.2.5 Case study conclusion

Model development speed for one-phase system was increased significantly by use of the modelling template. Since the starting model has been obtained from the template, further improvements and updates were done much easier, than in the case, when the model should have been developed from scratch. Instead, developer could allocate more time for model analysis and evaluation.

Moreover, the initial template has been updated based of the result of model development for one phase system and an option for non-elementary kinetics has been added to the template. Further use of the template for modelling of the specific cases may lead to the additional improvements of the initial template, such as options for different types of unsaturated fatty acids with one or more double bonds and also further verification of the two-phase model with validated reaction kinetics. Other parts of the modelling framework (Figure 4.1) can be used for model optimization with respect to initial conditions.

5.3 Batch reactor and distillation

In industrial process modelling practice, the average number of modelling and simulation tools used by a work group is approximately five (Kraus et al, 2014). Although the diversity of tools should potentially increase the probability of finding a solution in an efficient manner, it can also lead to difficulties when the models need to be re-used by or shared within different work groups. For this reason an important feature of the tools integration is the capability to import and export the generated mathematical models from/to different modelling tools.

The aim of this case study is to highlight model reuse through the connection with other computer-aided tools. In particular, the established connection between ICAS-MoT and MOSAIC modelling environment will be tested in order to illustrate the benefits of such tool integration.

The application of the modelling framework and its model transfer capabilities are highlighted through a case study involving the modelling of a batch reactor followed by a batch distillation operation. Objective here is to obtain the model describing the reaction and separation operations in the production of methyl acetate from acetic acid and methanol (Figure 5.8).

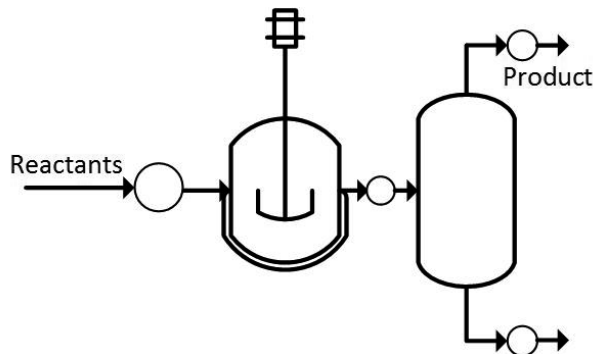


Figure 5.8. State-task network for batch reactor and batch distillation.

In this case the model library is employed for model reuse. By following ontology route: Dynamic->Flowsheet->Batch->Reactor-Distillation - required model is obtained and opened in modelling environment of ICAS-MoT. The model solution is obtained by using dynamic solver in MoT and the results are illustrated in Figure 5.9. Full model as implemented in MoT is given in Appendix 7.2.1.

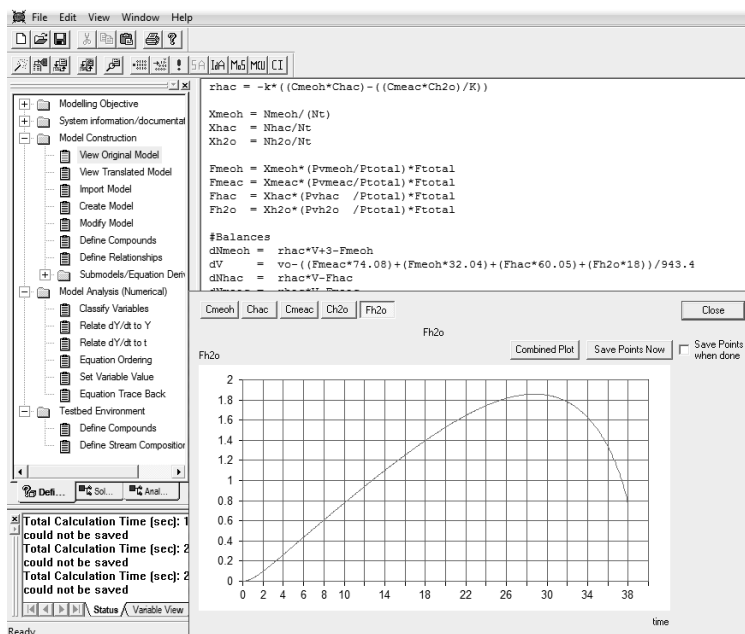


Figure 5.9. Model implementation in MoT

In order to verify obtained result, the model is transferred to MOSAIC modelling environment via established XML translation. This translation is done without additional effort from the user. All equations, variables and relations among them, as well as documentation that has been provided, are automatically transferred and implemented. In MOSAIC, the model can be further translated into various modelling languages, as described in section 4.5.

Figure 5.10 shows schematic implementation of the transfer process starting from obtaining model from the model library of the ICAS modelling framework, then translating it to MOSAIC modelling environment, where code representation of the model for gPROMS® (Process Systems Enterprise, 2014) and Matlab® (the MathWorks Inc., 2015) takes place. Further, Figure 5.11 demonstrates translated model in MOSAIC.

Generated code representations were employed in the corresponding software tools in order to solve the model and to compare results with the ones given by MoT. Generated code is shown in Figure 5.12. Full code translations can be found in Appendices 7.2.2 and 7.2.3 respectively for gPROMS® and Matlab®.

Results obtained from external tools are found to be identical to the ones given by ICAS-MoT (see plots in Figure 5.13) and this confirms the successful transfer and use of the new transfer feature.

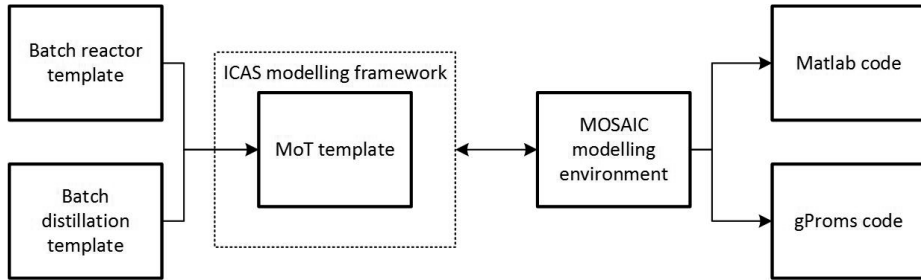


Figure 5.10. Schematic architecture of model implementation in ICAS modelling framework and its translation to external tools.

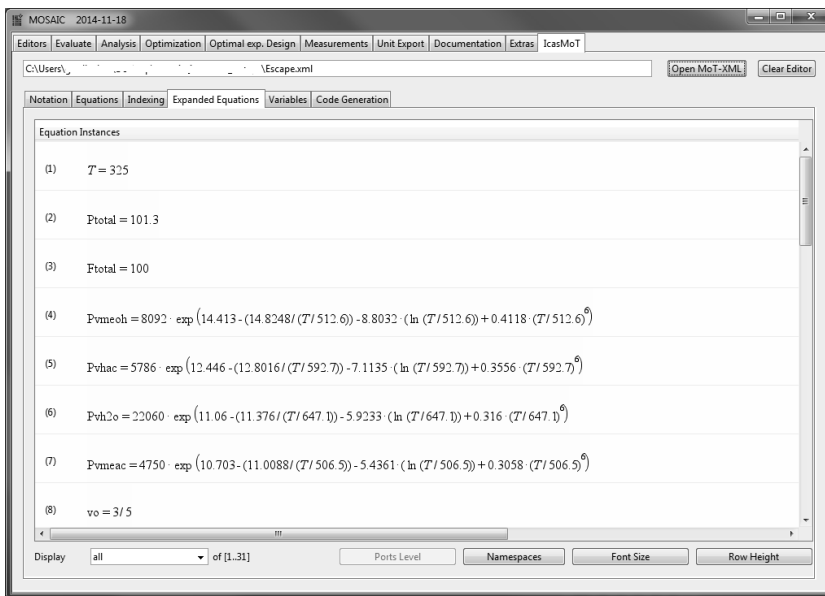


Figure 5.11. Translated model in the MOSAIC modelling environment

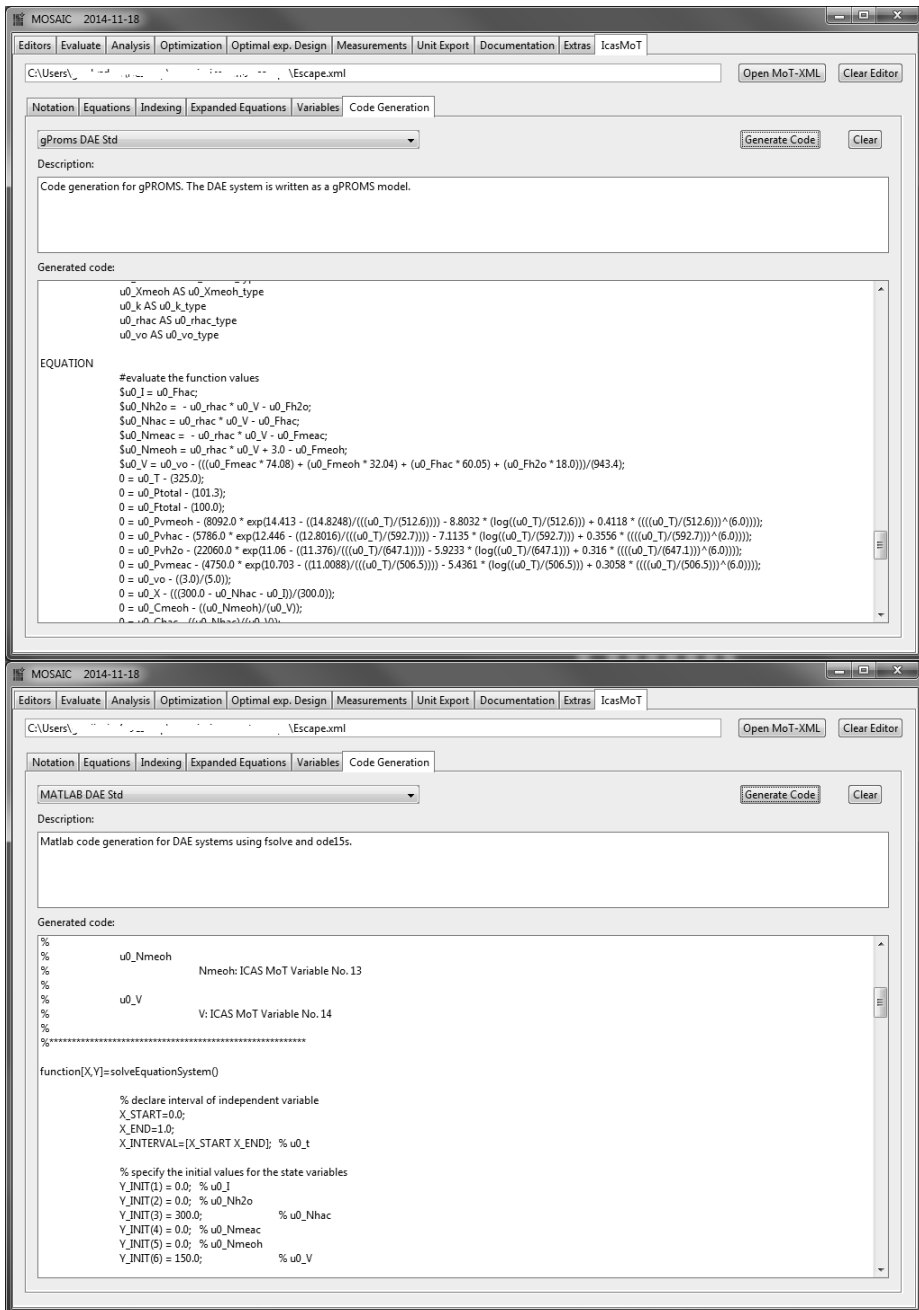


Figure 5.12. Generated code for gPROMS (up) and Matlab (down)

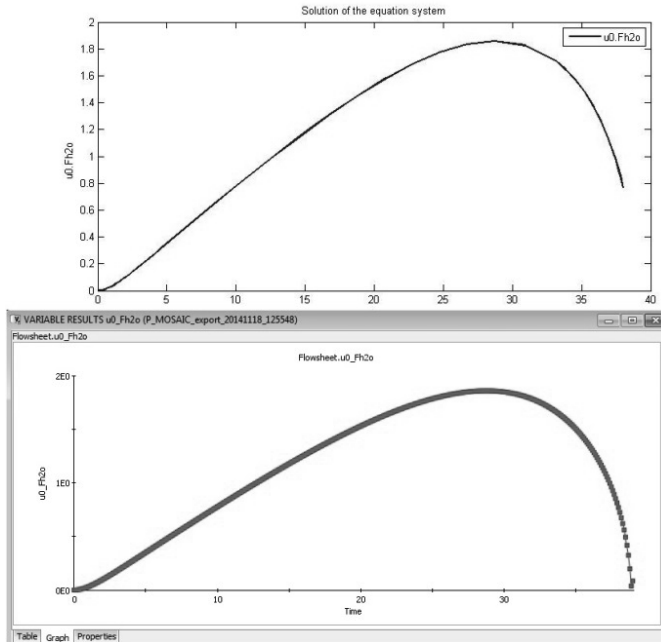


Figure 5.13. Resulting plots in Matlab and gProms, showing water flowrate after distillation unit.

5.3.1 Case study conclusion

The possibility to translate the model, created in one modelling environment, to a variety of other languages in order to use it in the different modelling tools gives the flexibility to the model development and application process. This allows the model to be shared among modellers, who may be using different tools. Thus, a range of tools can be used for diverse purposes (e.g. model identification, analysis, control strategy etc.) without the need for rewriting the model. Also, by using another tool, a better solution strategy could be employed.

5.4 Flowsheet template

In this case study the integrated use of all three tools in the modelling framework is highlighted through a development of a simple flowsheet model. Initially the model equations are obtained from the tool for model generation (ModDev) and then transferred to model analysis tool (MoT). Further, based on this model, a modelling template is created, which is used later for the modelling of Williams-Otto plant (Williams and Otto, 1960). The main goal of the case study is a demonstration of the template reuse for the system with similar repetitive objects.

5.4.1 Modelling objective

Objective of the modelling is a development of a simple flowsheet model for simulation of methylacetate production from methanol and acetic acid. The goal is to describe mass balances and reaction kinetics. Separator in the model will be presented as a component splitter. Schematic flowsheet of the system is presented in Figure 5.14

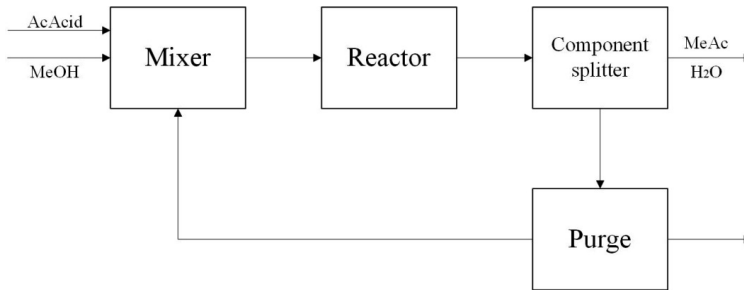


Figure 5.14. Schematic flowsheet for methyl acetate production.

5.4.2 Model generation

Currently template library doesn't include a suitable template for this case. Therefore, the tool for model generation – ModDev (Jensen and Gani, 1999), will be used in order to speed up the model development.

5.4.2.1 System decomposition

The system to be modeled is decomposed into building blocks including shells and streams. The shells include representation of a mixer, a reactor, a distillation column as a component splitter and a purge (divider). Software tool provides necessary support in system decomposition via drawing shells and streams connecting them. Figure 5.15 demonstrates the implementation of the flowsheet system in ModDev.

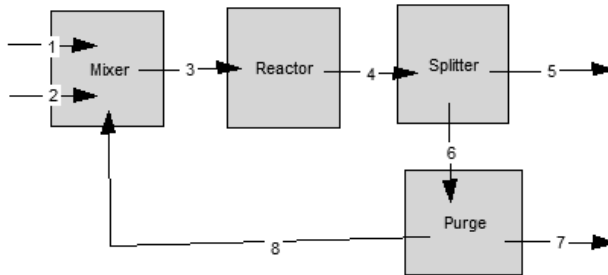


Figure 5.15. Implementation of the flowsheet in ModDev

5.4.2.2 Description of the building blocks

Each of the building blocks should be described according to the system to be modelled and modelling objective.

Mixer. Equations for the mixer include only mass balance and they were generated automatically by the tool.

Reactor. The reaction occurring in the reactor is as following:



This reaction should be added to reactor, by choosing it from the reaction library provided by the tool (see Figure 5.16). After defining the reaction the mass balance equation for the reactor is automatically updated with reaction generation of components.

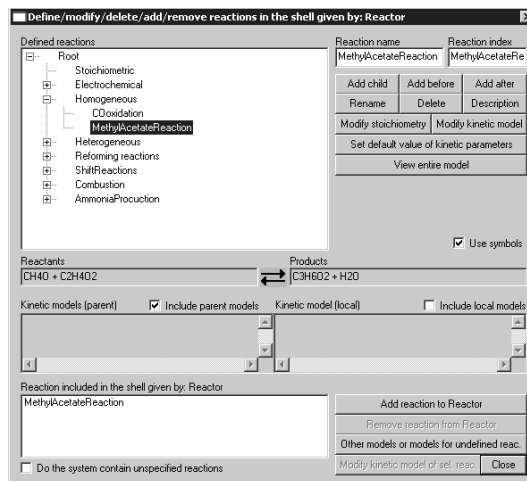


Figure 5.16. Schematic flowsheet for methyl acetate production.

The kinetics of the reversible reaction should be defined and the tool is assisting with that providing the knowledge database of the equations. Required equations are obtained by the ModDev database route: *Reaction* → *KineticModel* → *HomogeneousReactions* → *Reversible* → *Forward and backward reaction rates* → *PowerRateLawForwardAndBackward* → *Arrenius*. To include stoichiometry of the reaction the following equation is taken from the library: *Reaction* → *Closure relations* → *Relation between component and reaction based rates* → *Mole basis*. Additionally, the relation between reaction rate and component concentration is defined by: *Reaction* → *ReactionConcentrationFactor* → *Molefraction* → *2.order*. To finalize the reactor description a closure equation for mole fraction calculation is added: *Quantity* → *Component* → *MoleFraction* → *MoleFlow*.

Resulting equations generated by ModDev describing the reactor are shown in Figure 5.17.

```

RKforward_Reactor_m=A1_m*exp(-Ea1_m/(Rg*T_Reactor))
RRForward_Reactor_m=Vt_Reactor*RKforward_Reactor_m*XR_Forward_Reactor_m
RKbackward_Reactor_m=A2_m*exp(-Ea2_m/(Rg*T_Reactor))
RRBackward_Reactor_m=Vt_Reactor*RKbackward_Reactor_m*XR_Backward_Reactor_m
RR_Reactor_m=RRForward_Reactor_m-RRBackward_Reactor_m
RN_Reactor_i=sum_m(nu_i_m*RR_Reactor_m)
x_Reactor_i=f_Reactor_i/ft_Reactor
XR_Reactor_m=x_Reactor_1:Nc*x_Reactor_1:Nc
0=f_3_i-f_4_i+RN_Reactor_i

```

Figure 5.17. Equations describing the reactor in the system (screenshot from ModDev)

Component splitter. Relations for the component splitter are obtained by the following library route: *Transport* → *InterPhase* → *Component* → *SplitterRelation* → *SplitterFactor* and *(1-SplitterFactor)*.

Divider. Relations for the divider are obtained by: *Quantity* → *Mass* → *DividerRelations* → *SingleStage*

5.4.2.3 Aggregation of the building blocks – final model

The software tool automatically defines connections among shells and streams and generates model equations accordingly with respect to indices and balances across the units. Automatic generation of equations reduces amounts of errors, which can occur during manual model construction. Resulting model can be directly transferred to MoT via XML file for further analysis and evaluation.

Figure 5.18 demonstrates the final generated equations after describing all building blocks.

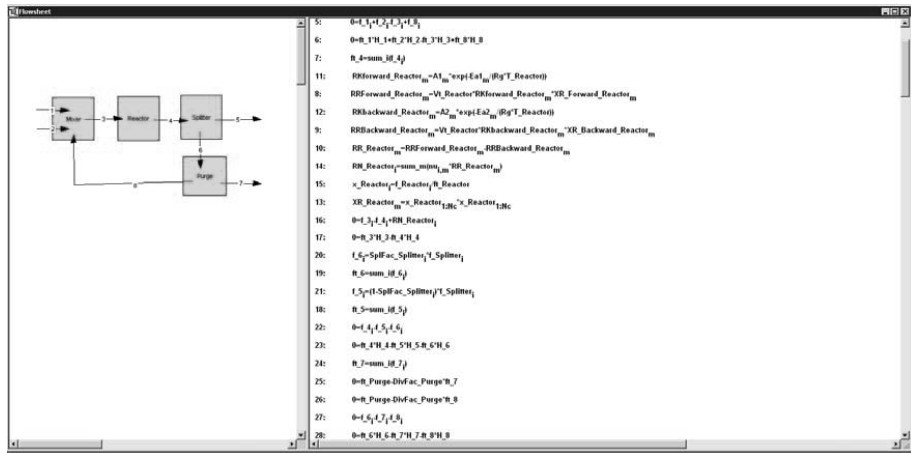


Figure 5.18. Schematic flowsheet for methyl acetate production.

5.4.3 Model analysis of generated model

After transfer of model to MoT some modification of the model should be done due to the following reasons:

- Redundancy of some equations, such as automatically generated mass balances across splitter and divider.
- Currently the connection between ModDev and MoT does not support transfer of variable and relationship information. Therefore, relationships (indices) should be defined as well as variable classification for model analysis.
- Some errors can occur while translating the model into MoT modelling object due to possible language incompatibilities.

However, despite these drawbacks the model development speed can be increased by using ModDev-MoT connection compared to the model development from scratch, as the system keeps equations consistent and provides the support for the modeller.

Full models, both the one transferred from ModDev and the one, which has been corrected, are given in the Appendix 7.3.

Resulting model was analysed and solved by using MoT in order to check that it is correct. Further, developed model is used for creation of the flowsheet template.

5.4.4 Template creation

This part of the case study reflects modeling stage 6: Model storage (see Figure 3.2). Based on the developed model, a flowsheet template will be created at this stage by following template-creation workflow (Figure 2.4)

5.4.4.1 Step 1. Collect system information

The system consists of mixer, reactor, component splitter and purge (divider) connected into a process flowsheet (Figure 5.14).

5.4.4.2 Step 2. Decomposition of the modelBalance equations

Balance equations include mass balances for each process unit. Since the model was obtained from ModDev and then adapted for MoT, the equations will be presented in MoT format.

Mixer:

$$0=f_{in1}[i]+f_{in2}[i]-f_{mixer}[i]+f_{recycle}[i] \quad (5.76)$$

Reactor:

$$0=f_{mixer}[i]-f_{reactor}[i]+RN_{Reactor}[i] \quad (5.77)$$

Component splitter:

$$f_{splitter}[i]=SplFac_Splitter[i]*f_{reactor}[i] \quad (5.78)$$

$$f_{product}[i]=(1-SplFac_Splitter[i])*f_{reactor}[i] \quad (5.79)$$

Divider:

$$0=f_{purge}[i]-DivFac_Purge*f_{splitter}[i] \quad (5.80)$$

$$0=f_{recycle}[i]-(1-DivFac_Purge)*f_{splitter}[i] \quad (5.81)$$

Constitutive equations

Constitutive equations include reaction rates for reversible reaction and calculation of kinetic constants.

$$RKforward_Reactor[m]=A1[m]*exp(-Ea1[m]/(Rg*T_Reactor)) \quad (5.82)$$

$$RRForward_Reactor[m]=Vt_Reactor*RKforward_Reactor[m] \\ *XR_Forward_Reactor[m] \quad (5.83)$$

$$RKbackward_Reactor[m]=A2[m]*exp(-Ea2[m]/(Rg*T_Reactor)) \quad (5.84)$$

$$RRBackward_Reactor[m]=Vt_Reactor* \\ RKbackward_Reactor[m]*XR_Backward_Reactor[m] \quad (5.85)$$

$$RR_Reactor[m]=RRForward_Reactor[m]-RRBackward_Reactor[m] \quad (5.86)$$

$$RN_Reactor[i]=sum_{m=1}^M(nu[i][m]*RR_Reactor[m]) \quad (5.87)$$

$$XR_Forward_Reactor[m]=x_Reactor_0*x_Reactor_1 \quad (5.88)$$

$$XR_Backward_Reactor[m]=x_Reactor_2*x_Reactor_3 \quad (5.89)$$

Connection equations

Connection equations include calculation of mole fractions.

$$x_Reactor[i]=f_4[i]/ft_4 \quad (5.90)$$

$$ft_4=sum_i(f_4[i]) \quad (5.91)$$

For all equations, i – number of components, m – number of reactions.

5.4.4.3 Step 3. Check if template is available

Currently no similar template is available in the template library.

5.4.4.4 Step 4B. Create new template

Since there is no available template, a new one will be created.

5.4.4.5 Step 5B. Derive parts of the model

In this step equations introduced in step 2 are allocated to the upper-layers of the template and the required number of sub-layers and blocks are created. In order to keep template simple at the moment, no sub-layers are created in the upper-layer of system information. Later, if this template will be extended, this upper-layer might include information about flowsheet organization or construction of specific units.

Upper-layer of balance equations includes sub-layers for mass balances for each of the process units – mixer, reactor, splitter and divider. Upper-layer of constitutive equations includes sub-layers for reaction rates and kinetics. Upper-layer of connection equations includes sub-layer for closure equations.

5.4.4.6 Step 6. Provide template description

Template provides building blocks for a flowsheet model. It includes mixer, reactor, component splitter and divider.

5.4.4.7 Step 7. Provide description of the blocks

Eqns. (5.76)-(5.81) are distributed among the blocks from the corresponding sub-layers of the upper-layer of balance equations.

Two options for reaction rates are provided – for reversible (Eqns. (5.82)-(5.87)) and irreversible (Eqns. (5.82)-(5.83), modified (5.86), (5.87)-(5.88)) reactions.

Closure relations (Eqns. (5.90)-(5.91)) are included in a block from the upper-layer of connection equations.

5.4.4.8 Step 8. Define connections between blocks

Blocks describing reaction rates and kinetics depend on whether the reaction is chosen to be reversible or not.

5.4.4.9 Step 9. Check template consistency**5.4.4.9.1 Upper-layer I. System information**

No blocks are created in this upper-layer.

5.4.4.9.2 Upper-layer II. Balance equations*Sub-layer – Mass balance - Mixer*Block 1

Name: Steady state

Description: Provides mass balance across the mixer in a steady state

Status: Chosen by default

Connections: -

Equations: (5.76)

*Sub-layer – Mass balance - Reactor*Block 1

Name: Steady state

Description: Provides mass balance across the reactor in a steady state

Status: Chosen by default

Connections: -

Equations: (5.77)

Sub-layer – Mass balance – Component splitter

Block 1

Name: Steady state

Description: Provides mass balance across the splitter in a steady state

Status: Chosen by default

Connections: -

Equations: (5.78)-(5.79)

Sub-layer – Mass balance - Divider

Block 1

Name: Steady state

Description: Provides mass balance across the divider in a steady state

Status: Chosen by default

Connections: -

Equations: (5.76)

5.4.4.9.3 Upper-layer III. Constitutive equations

Sub-layer – Reaction rates

Block 1

Name: Reversible

Description: Provides rates for reversible reaction

Status: Available to choose

Connections: -

Equations: (5.83), (5.85)-(5.89)

Block 2

Name: Irreversible

Description: Provides rates for irreversible reaction

Status: Chosen by default

Connections: -

Equations: (5.83), (5.86)-(5.88)

Sub-layer – Kinetic constant

Block 1

Name: Arrhenius

Description: Provides rates for reversible reaction

Status: Chosen by default

Connections: Depends on the choice from “Constitutive equations/Reaction rates” sub-layer

Equations: if “Constitutive equations/Reaction rates/Reversible” is chosen, then Eqns. (5.82), (5.84);

if “Constitutive equations/Reaction rates/Irreversible” is chosen, then Eqn. (5.82)

5.4.4.9.4 Upper-layer IV. Connection equations

Sub-layer – Closure relations

Block 1

Name: Mole fractions

Description: Calculation of mole fraction of the components in the reactor

Status: Chosen by default

Connections: -

Equations: (5.90)-(5.91)

5.4.4.10 Step 10. Finalization of the template

Template is saved and can be extracted from the template library and used for model development.

5.4.5 Template use for new model development

To highlight the use of a newly created template a modeling case of the Williams-Otto plant (Biegler et al., 1997) will be considered. A schematic flow diagram for this case is shown in Figure 5.19

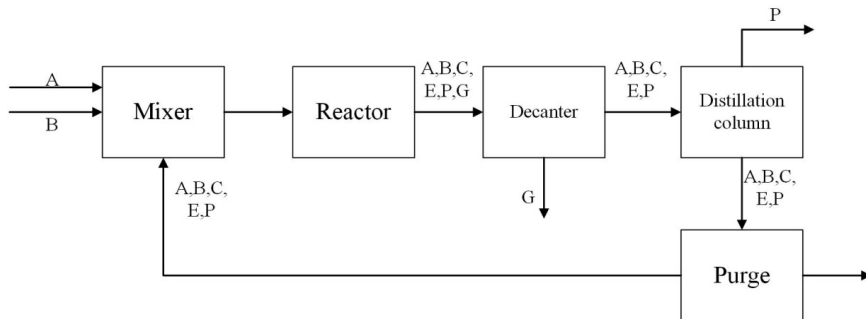
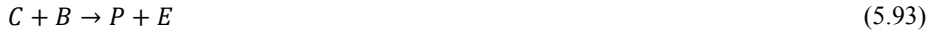


Figure 5.19. Flow diagram for the Williams-Otto plant (adapted from Biegler et al., 1997)

In the reactor, the following reactions take place:



P is the main product, C is intermediate and E is a by-product. After the reaction by-product G is removed through a decanter. In the following distillation column the product P is removed as the top product and the unreacted A and B together with C is recycled back to the reactor. A purge is also used (Cameron and Gani, 2011).

Distillation column can be modeled as a component splitter, however it is assumed that product P cannot be completely separated and its amount in the bottom product depends on the amount of the component E .

Newly created template for the flowsheet model can be used in this case. The steps of workflow for template-use (see Figure 2.6) are followed.

Step 1. Template the flowsheet modelling is chosen.

Step 2. System includes mixer, reactor, two component splitters and divider.

Step 3. For all units models are in steady state.

The following choices are made in the upper-layer “Balance equations”:

- Sub-layer “Mass balance – Mixer” – Block “Steady state”
- Sub-layer “Mass balance – Reactor” – Block “Steady state”
- Sub-layer “Mass balance – Splitter” – Block “Steady state”
- Sub-layer “Mass balance – Divider” – Block “Steady state”

Note, that the system has two splitters, therefore, equations provided by this block are doubled.

Step 4. All equations are irreversible. Kinetic constants are calculated by Arrhenius equation.

The following choices are made in the upper-layer “Constitutive equations”:

- Sub-layer “Reaction rates” – Block “Irreversible”
- Sub-layer “Kinetic constant” – Block “Arrhenius”

Step 5.

The following choices are made in the upper-layer “Connection equations”:

- Sub-layer “Closure relations” – Block “Mole fraction”

Step 6. Final model is constructed based on answers from steps 1-5. Employed equations are Eqns. (5.76)-(5.77), (5.78)-(5.79) doubled, (5.80)-(5.83), (5.86)-(5.91).

After last step the model is directly transferred to MoT for further analysis.

5.4.6 Williams-Otto plant model analysis and solution

Before the model can be solved some case specific changes need to be done.

1. Update variable names for the component splitters, so the program engine can distinguish them.
2. Equation (5.88) should be adapted for the three reactions occurring in the system reactor.

3. Take into account separation of product P in the column (dependency on E).

Full model implementation in MoT is given in Appendix 7.3.3.

The variables to be specified are following (Cameron and Gani, 2011).

$$A1[1] = 5.9755 * 10^9; A1[2] = 2.5962 * 10^{12}; A1[3] = 9.6283 * 10^{15};$$

$$E1[1]/R_g = 12000; E1[2]/R_g = 15000; E1[3]/R_g = 20000;$$

$$SplFac_{decanter[i]} = 1 \text{ for } i = A, B, C, E, P; SplFac_{decanter[G]} = 0;$$

$$SplFac_{column[i]} = 1 \text{ for } i = A, B, C, E, G; SplFac_P = 0.1; DivFac_{purge} = 0.1$$

$$T_{Reactor} = 674.43; V = 1500; f_{in1_A} = 13179.6; f_{in2_B} = 30038.3;$$

After applying required changes the model was solved using Wegstein algebraic solver available in MoT. Simulation results are shown in Table 5.3.

Table 5.3. Simulation results of the Williams-Otto plant model

Compound	f_recycle	f_waste	f_purge	f_product
A	41743.8	0	4638.2	0
B	129423.2	0	14380.3	0
C	6842.2	0	760.2	0
E	128097.6	0	14233.0	0
P	12809.7	0	1423.3	4648.5
G	0	3134.2	0	0

These results are identical to the ones obtained by another approach (Cameron and Gani, 2011). Therefore, model development can be considered successful and the model can be used for further application, e.g. optimization.

5.4.7 Case study conclusion.

This case study highlights the use of the template approach for the models with repetitive part, e.g. flowsheets.

In this case study a template for a generic flowsheet model has been implemented and applied for the development of the Williams-Otto plant model. The results of the model simulation were compared to the results of another modeling approach implemented in ICAS-MoT (Sales-Cruz, 2006) and were found identical. However, the model implemented by Sales-Cruz (2006) is case specific, and the template developed in this work allows obtaining a generic model, which can be reused for any flowsheet modeling case.

Further improvement of the developed template can be done in the direction of adding new process units or applying more rigorous models, for example, distillation column

instead of component splitter, as well as updating the template with energy balances and dynamic balance equations.

6 CONCLUSION AND FUTURE WORK

Template-based modelling approach has been developed and implemented as a computer-aided software tool, which guides the user through workflow steps in order to generate a problem-specific model. The application of the tool has been highlighted with several case studies, in which models are generated for different objectives.

The modelling template is intended to be a model repository for storing, retrieving, modifying, updating models in knowledge based systems for easy maintenance and connection to model solution and analysis software. The models constructed by the template can then be exported to different simulators for various model application purposes including simulation, parameter estimation, optimization and many more. Recommendation for the future work can be extending the application range for the template-based modelling, as well as improving the different aspects of workflow and its software implementation.

6.1 Main achievements of the thesis

- Development of template-based modelling approach based on the idea of model reuse, which emphasizes the use of a model not only for one specific application but also for future applications involving different needs and levels of details to match different purposes.
 - Development of the modelling template based on the idea of model equations decomposition into sub-groups of balance, constitutive and connection equations.
 - Identification of modelling tasks for template creation and template use as well as connections among them.
 - Development of the workflows for template creation and template use and overall modelling template structure.
- Development of a software architecture and implementation of the developed template-based approach into software tool.
 - Development of a new software tool for template application.
 - Implementation of the template library.
- Development of a computer-aided modelling framework that integrates template-based modelling approach with previously developed modelling methodologies and tools – ICAS-MoT (Salez-Cruz, 2006; Heitzig, 2011) and ModDev (Jensen, 1998).

- Identification of the required tools and features in order to provide the required support to a modeller.
- Realisation of the integration among three tools – ModDev, MoT and ModTem, to provide to a modeller the possibility to use features of different tools without the need to redo model construction.
- Realization of an integration with the MOSAIC modelling environment (Kraus et al., 2014) in order to transfer models and modelling information to external tools.
- Highlighting the different aspects of the template-based modelling approach and overall modelling framework through a set of case studies.
 - Demonstration of the template workflow use for different purposes.
 - Demonstration of the model transfer among the framework tools and to external tools.

6.2 Overview of the potential future improvements

Overview of the potential improvements includes the ones related to modelling methodology and modelling framework, as well as the ones reflecting software implementation possible updates.

6.2.1 Modelling methodology and computer-aided framework improvements

Some of the potential improvement for the MoT modelling workflows had been identified by Heitzig (2011). Possibilities for additional improvements of the template-based methodology can be evaluated during template application for various specific cases, which can reveal potential drawbacks or missing parts of the methodology.

Current challenges can be identified as following:

- Proposed way of the model decomposition and the ontology for a template creation are not strictly formal. This leaves the room for each user own interpretation of the methodology. It can have benefits and drawbacks and should be further investigated.
- Following up the previous point – assistance during model decomposition should be offers (Step 2 of the template creation workflow).
- Better automatization of the template tool is required. This is discussed in the next section.
- Ways to implement a version control system for templates should be investigated.

6.2.2 Software implementation improvements

The closest attention will be paid to potential improvements of the template application tool – ModTem, model library, integration of the modelling tool and transfer of the

models. Improvements of other software tools – MoT and ModDev – will be considered only with respect to above mentioned features. For the potential improvements of these tools the reader is referred to Heitzig (2011) and Jensen (1999).

- Direct translation of the model generated in ModDev to MoT.
- Transfer of additional model information from ModDev: variable information, model analysis information, information about indices.
- Possibility to transfer only certain equations from ModDev, not the whole model.
- Connection between ModDev and ModTem for model templates creation.
- Further testing of different model types transfer to external tools.
- Improve compatibility of models from different tools, so that no additional verification and model code tuning is needed, for example, transfer of ordinary differential equations from ModDev to MoT.
- ModTem interface improvement.
- Development of more “smart” engine for template tool, which can automatically generate new template options from existing ones, when possible, for example, steady/dynamic state for the balances.
- Reducing the need for manual equations modifications during template creation.
- Introduction of a version control system for the template tool.
- Development of consistency check tool.
- Introduction of the more formal documentation for templates in the template tool, e.g. variable information, description of each equation/block in a structured way. Generation of template information based of information provided in template documentation.

7 APPENDIX

7.1 Example of the model in XML format

In this appendix an example the XML format is illustrated with a short example of dynamic tank model. The information in XML file contains model equations - also expanded, when one or more variables have indexes, indexing relationship and variable types and values.

```

<mot_model>
  <equations>
    <unexpanded>
      <equation>
        <expression>P2 = Patm + ro*g*h/101325</expression>
      </equation>
      <equation>
        <expression>F1 = Cv*sqrt(P1 - P2)</expression>
      </equation>
      <equation>
        <expression>F2 = Cv*sqrt(P2 - P3)</expression>
      </equation>
      <equation>
        <expression>dh = (F1 - F2)/A</expression>
      </equation>
    </unexpanded>
    <expanded>
      <equation>
        <expression>P2=Patm+ro*g*h/101325</expression>
      </equation>
      <equation>
        <expression>F1=Cv*sqrt(P1-P2)</expression>
      </equation>
      <equation>
        <expression>F2=Cv*sqrt(P2-P3)</expression>
      </equation>
      <equation>
        <expression>dh=(F1-F2)/A</expression>
      </equation>
    </expanded>
  </equations>
  <indexing/>
  <variables>
    <variable>
      <name>P2</name>
      <value>0.0000000000</value>
      <type>4:Explicit_calc</type>
    </variable>
    <variable>
      <name>Patm</name>

```



```
<value>1.000000000</value>
<type>1:Parameter</type>
</variable>
<variable>
  <name>ro</name>
  <value>1000.000000000</value>
  <type>2:Known</type>
</variable>
<variable>
  <name>g</name>
  <value>9.810000000</value>
  <type>1:Parameter</type>
</variable>
<variable>
  <name>h</name>
  <value>0.000000000</value>
  <type>5:State_dependent</type>
  <pair_with>dh</pair_with>
</variable>
<variable>
  <name>F1</name>
  <value>0.000000000</value>
  <type>4:Explicit_calc</type>
</variable>
<variable>
  <name>Cv</name>
  <value>34.325000000</value>
  <type>2:Known</type>
</variable>
<variable>
  <name>P1</name>
  <value>2.000000000</value>
  <type>2:Known</type>
</variable>
<variable>
  <name>F2</name>
  <value>0.000000000</value>
  <type>4:Explicit_calc</type>
</variable>
<variable>
  <name>P3</name>
  <value>1.000000000</value>
  <type>2:Known</type>
</variable>
<variable>
  <name>dh</name>
  <value>0.000000000</value>
  <type>6:Derivation_dependent_prime</type>
  <pair_with>h</pair_with>
</variable>
<variable>
  <name>A</name>
  <value>12.500000000</value>
  <type>2:Known</type>
</variable>
</variables>
</mot_model>
```

7.2 Batch reactor and distillation

7.2.1 Model implementation in MoT

```

#Initial Conditions
;t(0)=1e-08
;Nmeoh(0)= 0
;V(0)      = 150
;Nhac(0)   = 300
;Nmeac(0)  = 0
;Nh2o(0)   = 0
;I(0)      = 0
;t(f)      = 38

#System Conditions
T          = 325
Ptotal    = 101.3
Ftotal    = 100

#Vapor pressure
Pvmeoh = 8092*exp(14.413-(14.8248/(T/512.6)) - 8.8032*(ln(T/512.6)) +
0.4118*(T/512.6)^6)
Pvhac  = 5786*exp(12.446-(12.8016/(T/592.7)) - 7.1135*(ln(T/592.7)) +
0.3556*(T/592.7)^6)
Pvh2o  = 22060*exp(11.06-(11.376/(T/647.1)) - 5.9233*(ln(T/647.1)) +
0.316*(T/647.1)^6)
Pvmeac = 4750*exp(10.703-(11.0088/(T/506.5)) - 5.4361*(ln(T/506.5)) +
0.3058*(T/506.5)^6)

vo      = 3/5
X        = (300 - Nhac - I)/300

Cmeoh = Nmeoh/V
Chac  = Nhac/V
Cmeac = Nmeac/V
Ch2o  = Nh2o/V
Nt     = Nmeoh + Nhac + Nmeac + Nh2o
Xmeac = Nmeac/Nt

k      = (8.88e8)*exp(-7032.1/T)
K      = 5.2*exp((-8000/1.978)*((1/298) - (1/T)))
rhac   = -k*((Cmeoh*Chac)-((Cmeac*Ch2o)/K))

Xmeoh = Nmeoh/(Nt)
Xhac  = Nhac/Nt
Xh2o  = Nh2o/Nt

Fmeoh = Xmeoh*(Pvmeoh/Ptotal)*Ftotal
Fmeac = Xmeac*(Pvmeac/Ptotal)*Ftotal
Fhac  = Xhac*(Pvhac/Ptotal)*Ftotal
Fh2o  = Xh2o*(Pvh2o/Ptotal)*Ftotal

#Balances
dNmeoh = rhac*V+3-Fmeoh
dV      = vo-(Fmeac*74.08)+(Fmeoh*32.04)+(Fhac*60.05)+(Fh2o*18))/943.4
dNhac  = rhac*V-Fhac
dNmeac = -rhac*V-Fmeac
dNh2o  = -rhac*V-Fh2o
dI      = Fhac

```

7.2.2 Generated gProms code

```
DECLARE TYPE
  u0_I_type = 0.0 : -1.0E9 : 1.0E9
  u0_Nh2o_type = 0.0 : -1.0E9 : 1.0E9
  u0_Nhac_type = 300.0 : -1.0E9 : 1.0E9
  u0_Nmeac_type = 0.0 : -1.0E9 : 1.0E9
  u0_Nmeoh_type = 0.0 : -1.0E9 : 1.0E9
  u0_V_type = 150.0 : -1.0E9 : 1.0E9
  u0_Ch2o_type = 0.0 : -1.0E9 : 1.0E9
  u0_Chac_type = 0.0 : -1.0E9 : 1.0E9
  u0_Cmeac_type = 0.0 : -1.0E9 : 1.0E9
  u0_Cmeoh_type = 0.0 : -1.0E9 : 1.0E9
  u0_Fh2o_type = 0.0 : -1.0E9 : 1.0E9
  u0_Fhac_type = 0.0 : -1.0E9 : 1.0E9
  u0_Fmeac_type = 0.0 : -1.0E9 : 1.0E9
  u0_Fmeoh_type = 0.0 : -1.0E9 : 1.0E9
  u0_Ftotal_type = 0.0 : -1.0E9 : 1.0E9
  u0_K_1_type = 0.0 : -1.0E9 : 1.0E9
  u0_Nt_type = 0.0 : -1.0E9 : 1.0E9
  u0_Ptotal_type = 0.0 : -1.0E9 : 1.0E9
  u0_Pvh2o_type = 0.0 : -1.0E9 : 1.0E9
  u0_Pvhac_type = 0.0 : -1.0E9 : 1.0E9
  u0_Pvmeac_type = 0.0 : -1.0E9 : 1.0E9
  u0_Pvmeoh_type = 0.0 : -1.0E9 : 1.0E9
  u0_T_type = 0.0 : -1.0E9 : 1.0E9
  u0_X_type = 0.0 : -1.0E9 : 1.0E9
  u0_Xh2o_type = 0.0 : -1.0E9 : 1.0E9
  u0_Xhac_type = 0.0 : -1.0E9 : 1.0E9
  u0_Xmeac_type = 0.0 : -1.0E9 : 1.0E9
  u0_Xmeoh_type = 0.0 : -1.0E9 : 1.0E9
  u0_k_type = 0.0 : -1.0E9 : 1.0E9
  u0_rhac_type = 0.0 : -1.0E9 : 1.0E9
  u0_vo_type = 0.0 : -1.0E9 : 1.0E9
```

END

MODEL MOSAIC_export

VARIABLE

```
#read out parameters

#read out variables
#Differential:
u0_I AS u0_I_type
u0_Nh2o AS u0_Nh2o_type
u0_Nhac AS u0_Nhac_type
u0_Nmeac AS u0_Nmeac_type
u0_Nmeoh AS u0_Nmeoh_type
```

APPENDIX

```

u0_V AS u0_V_type
#Algebraic:
u0_Ch2o AS u0_Ch2o_type
u0_Chac AS u0_Chac_type
u0_Cmeac AS u0_Cmeac_type
u0_Cmeoh AS u0_Cmeoh_type
u0_Fh2o AS u0_Fh2o_type
u0_Fhac AS u0_Fhac_type
u0_Fmeac AS u0_Fmeac_type
u0_Fmeoh AS u0_Fmeoh_type
u0_Ftotal AS u0_Ftotal_type
u0_K_1 AS u0_K_1_type
u0_Nt AS u0_Nt_type
u0_Ptotal AS u0_Ptotal_type
u0_Pvh2o AS u0_Pvh2o_type
u0_Pvhac AS u0_Pvhac_type
u0_Pvmeac AS u0_Pvmeac_type
u0_Pvmeoh AS u0_Pvmeoh_type
u0_T AS u0_T_type
u0_X AS u0_X_type
u0_Xh2o AS u0_Xh2o_type
u0_Xhac AS u0_Xhac_type
u0_Xmeac AS u0_Xmeac_type
u0_Xmeoh AS u0_Xmeoh_type
u0_k AS u0_k_type
u0_rhac AS u0_rhac_type
u0_vo AS u0_vo_type

```

EQUATION

```

#evaluate the function values
$u0_I = u0_Fhac;
$u0_Nh2o = - u0_rhac * u0_V - u0_Fh2o;
$u0_Nhac = u0_rhac * u0_V - u0_Fhac;
$u0_Nmeac = - u0_rhac * u0_V - u0_Fmeac;
$u0_Nmeoh = u0_rhac * u0_V + 3.0 - u0_Fmeoh;
$u0_V = u0_vo - (((u0_Fmeac * 74.08) + (u0_Fmeoh * 32.04) + (u0_Fhac * 60.05) +
(u0_Fh2o * 18.0)))/(943.4);
0 = u0_T - (325.0);
0 = u0_Ptotal - (101.3);
0 = u0_Ftotal - (100.0);
0 = u0_Pvmeoh - (8092.0 * exp(14.413 - ((14.8248)/((u0_T)/(512.6)))) - 8.8032 *
(log((u0_T)/(512.6))) + 0.4118 * (((u0_T)/(512.6))^(6.0)));
0 = u0_Pvhac - (5786.0 * exp(12.446 - ((12.8016)/((u0_T)/(592.7)))) - 7.1135 *
(log((u0_T)/(592.7))) + 0.3556 * (((u0_T)/(592.7))^(6.0)));
0 = u0_Pvh2o - (22060.0 * exp(11.06 - ((11.376)/((u0_T)/(647.1)))) - 5.9233 *
(log((u0_T)/(647.1))) + 0.316 * (((u0_T)/(647.1))^(6.0)));
0 = u0_Pvmeac - (4750.0 * exp(10.703 - ((11.0088)/((u0_T)/(506.5)))) - 5.4361 *
(log((u0_T)/(506.5))) + 0.3058 * (((u0_T)/(506.5))^(6.0)));
0 = u0_vo - ((3.0)/(5.0));

```

APPENDIX

```

0 = u0_X - (((300.0 - u0_Nhac - u0_I)/(300.0));
0 = u0_Cmeoh - ((u0_Nmeoh)/(u0_V));
0 = u0_Chac - ((u0_Nhac)/(u0_V));
0 = u0_Cmeac - ((u0_Nmeac)/(u0_V));
0 = u0_Ch2o - ((u0_Nh2o)/(u0_V));
0 = u0_Nt - (u0_Nmeoh + u0_Nhac + u0_Nmeac + u0_Nh2o);
0 = u0_Xmeac - ((u0_Nmeac)/(u0_Nt));
0 = u0_k - ((8.88E8) * exp(- (7032.1)/(u0_T)));
0 = u0_K_1 - (5.2 * exp((- (8000.0)/(1.978)) * (((1.0)/(298.0)) - ((1.0)/(u0_T))));
0 = u0_rhac - ( - u0_k * ((u0_Cmeoh * u0_Chac) - (((u0_Cmeac *
u0_Ch2o))/(u0_K_1))));
0 = u0_Xmeoh - ((u0_Nmeoh)/(u0_Nt));
0 = u0_Xhac - ((u0_Nhac)/(u0_Nt));
0 = u0_Xh2o - ((u0_Nh2o)/(u0_Nt));
0 = u0_Fmeoh - (u0_Xmeoh * ((u0_Pvmeoh)/(u0_Ptotal)) * u0_Ftotal);
0 = u0_Fmeac - (u0_Xmeac * ((u0_Pvmeac)/(u0_Ptotal)) * u0_Ftotal);
0 = u0_Fhac - (u0_Xhac * ((u0_Pvhac)/(u0_Ptotal)) * u0_Ftotal);
0 = u0_Fh2o - (u0_Xh2o * ((u0_Pvh2o)/(u0_Ptotal)) * u0_Ftotal);

ASSIGN
#set parameters

INITIAL
#define initial values
u0_I = 0.0;
u0_Nh2o = 0.0;
u0_Nhac = 300.0;
u0_Nmeac = 0.0;
u0_Nmeoh = 0.0;
u0_V = 150.0;

END

PROCESS P_MOSAIC_export

UNIT Flowsheet AS MOSAIC_export

SCHEDULE
CONTINUE FOR 39.0

END

```

7.2.3 Generated Matlab code

Function for results display is not shown.

```
function[X,Y]=solveEquationSystem()

% declare interval of independent variable
X_START=0.0;
X_END=38.0;
X_INTERVAL=[X_START X_END];           % u0_t

% specify the initial values for the state variables
Y_INIT(1) = 0.0;           % u0_I
Y_INIT(2) = 0.0;           % u0_Nh2o
Y_INIT(3) = 300.0;         % u0_Nhac
Y_INIT(4) = 0.0;           % u0_Nmeac
Y_INIT(5) = 0.0;           % u0_Nmeoh
Y_INIT(6) = 150.0;         % u0_V
Y_INIT(7) = 0.0;           % u0_Ch2o
Y_INIT(8) = 0.0;           % u0_Chac
Y_INIT(9) = 0.0;           % u0_Cmeac
Y_INIT(10) = 0.0;          % u0_Cmeoh
Y_INIT(11) = 0.0;          % u0_Fh2o
Y_INIT(12) = 0.0;          % u0_Fhac
Y_INIT(13) = 0.0;          % u0_Fmeac
Y_INIT(14) = 0.0;          % u0_Fmeoh
Y_INIT(15) = 100.0;        % u0_Ftotal
Y_INIT(16) = 1.0;          % u0_K
Y_INIT(17) = 300.0;        % u0_Nt
Y_INIT(18) = 101.3;        % u0_Ptotal
Y_INIT(19) = 11.0;         % u0_Pvh2o
Y_INIT(20) = 7.0;          % u0_Pvhac
Y_INIT(21) = 82.0;         % u0_Pvmeac
Y_INIT(22) = 56.0;         % u0_Pvmeoh
Y_INIT(23) = 325.0;        % u0_T
Y_INIT(24) = 0.0;          % u0_X
Y_INIT(25) = 0.0;          % u0_Xh2o
Y_INIT(26) = 0.0;          % u0_Xhac
Y_INIT(27) = 0.0;          % u0_Xmeac
Y_INIT(28) = 0.0;          % u0_Xmeoh
Y_INIT(29) = 1.0;          % u0_k
Y_INIT(30) = 0.0;          % u0_rhac
Y_INIT(31) = 0.6;          % u0_vo

% declare parameters
PARAMS = zeros(0);

NLE_OPTIONS = optimset('MaxIter',1000,'TolFun',1e-10,'Display','Iter');
[Y_INIT,FVAL]=fsolve(@(Y)daeSystemLHS(X_START,Y,PARAMS),Y_INIT,NLE_OPTIONS);
```

APPENDIX

```
FVAL_OF_INIT_NLE = FVAL
INIT_STATES = Y_INIT'

M = daeSystemMM();

OPTIONS = odeset('Mass',M);

[X,Y] = ode15s(@(X,Y)daeSystemLHS(X,Y,PARAMS),X_INTERVAL,Y_INIT',OPTIONS);

displayResults(X,Y);

end

function MASS = daeSystemMM()

    MASS = zeros(31, 31); %total number of equations
    MASS(1:6, 1:6)=eye(6,6); % number of odes

end

% evaluate the differential function.
function[DYDX] = daeSystemLHS(X,Y,PARAMS)

    % read out variables
    u0_I = Y(1);
    u0_Nh2o = Y(2);
    u0_Nhac = Y(3);
    u0_Nmeac = Y(4);
    u0_Nmeoh = Y(5);
    u0_V = Y(6);
    u0_Ch2o = Y(7);
    u0_Chac = Y(8);
    u0_Cmeac = Y(9);
    u0_Cmeoh = Y(10);
    u0_Fh2o = Y(11);
    u0_Fhac = Y(12);
    u0_Fmeac = Y(13);
    u0_Fmeoh = Y(14);
    u0_Ftotal = Y(15);
    u0_K = Y(16);
    u0_Nt = Y(17);
    u0_Ptotal = Y(18);
    u0_Pvh2o = Y(19);
    u0_Pvhac = Y(20);
    u0_Pvmeac = Y(21);
    u0_Pvmeoh = Y(22);
    u0_T = Y(23);
    u0_X = Y(24);
```

APPENDIX

```

u0_Xh2o = Y(25);
u0_Xhac = Y(26);
u0_Xmeac = Y(27);
u0_Xmeoh = Y(28);
u0_k = Y(29);
u0_rhac = Y(30);
u0_vo = Y(31);

% read out differential variable
u0_t = X;

% evaluate the function values
DYDX(1) = u0_Fhac;
DYDX(2) = - u0_rhac * u0_V - u0_Fh2o;
DYDX(3) = u0_rhac * u0_V - u0_Fhac;
DYDX(4) = - u0_rhac * u0_V - u0_Fmeac;
DYDX(5) = u0_rhac * u0_V + 3.0 - u0_Fmeoh;
DYDX(6) = u0_vo - ( ( ( u0_Fmeac * 74.08 ) + ( u0_Fmeoh * 32.04 ) + ( u0_Fhac * 60.05 ) + (
u0_Fh2o * 18.0 ) ) ) / ( 943.4 );
DYDX(7) = u0_T - ( 325.0 );
DYDX(8) = u0_Ptotal - ( 101.3 );
DYDX(9) = u0_Ftotal - ( 100.0 );
DYDX(10) = u0_Pvmeoh - ( 8092.0 * exp( 14.413 - ( ( 14.8248 ) / ( ( ( u0_T ) / ( 512.6 ) ) ) ) ) - 8.8032
* ( log( ( u0_T ) / ( 512.6 ) ) ) + 0.4118 * ( ( ( u0_T ) / ( 512.6 ) ) ) ^ ( 6.0 ) ) );
DYDX(11) = u0_Pvhac - ( 5786.0 * exp( 12.446 - ( ( 12.8016 ) / ( ( ( u0_T ) / ( 592.7 ) ) ) ) ) - 7.1135 *
( log( ( u0_T ) / ( 592.7 ) ) ) + 0.3556 * ( ( ( u0_T ) / ( 592.7 ) ) ) ^ ( 6.0 ) ) );
DYDX(12) = u0_Pvh2o - ( 22060.0 * exp( 11.06 - ( ( 11.376 ) / ( ( ( u0_T ) / ( 647.1 ) ) ) ) ) - 5.9233 * (
log( ( u0_T ) / ( 647.1 ) ) ) + 0.316 * ( ( ( u0_T ) / ( 647.1 ) ) ) ^ ( 6.0 ) ) );
DYDX(13) = u0_Pvmeac - ( 4750.0 * exp( 10.703 - ( ( 11.0088 ) / ( ( ( u0_T ) / ( 506.5 ) ) ) ) ) - 5.4361
* ( log( ( u0_T ) / ( 506.5 ) ) ) + 0.3058 * ( ( ( u0_T ) / ( 506.5 ) ) ) ^ ( 6.0 ) ) );
DYDX(14) = u0_vo - ( ( 3.0 ) / ( 5.0 ) );
DYDX(15) = u0_X - ( ( ( 300.0 - u0_Nhac - u0_I ) ) / ( 300.0 ) );
DYDX(16) = u0_Cmeoh - ( ( u0_Nmeoh ) / ( u0_V ) );
DYDX(17) = u0_Chac - ( ( u0_Nhac ) / ( u0_V ) );
DYDX(18) = u0_Cmeac - ( ( u0_Nmeac ) / ( u0_V ) );
DYDX(19) = u0_Ch2o - ( ( u0_Nh2o ) / ( u0_V ) );
DYDX(20) = u0_Nt - ( u0_Nmeoh + u0_Nhac + u0_Nmeac + u0_Nh2o );
DYDX(21) = u0_Xmeac - ( ( u0_Nmeac ) / ( u0_Nt ) );
DYDX(22) = u0_k - ( ( 8.88E8 ) * exp( - ( 7032.1 ) / ( u0_T ) ) );
DYDX(23) = u0_K - ( ( 5.2 * exp( ( - ( 8000.0 ) / ( 1.978 ) ) ) * ( ( ( 1.0 ) / ( 298.0 ) ) ) - ( ( 1.0 ) / ( u0_T ) )
) ) );
DYDX(24) = u0_rhac - ( - u0_k * ( ( u0_Cmeoh * u0_Chac ) - ( ( ( u0_Cmeac * u0_Ch2o ) ) / (
u0_K ) ) ) );
DYDX(25) = u0_Xmeoh - ( ( u0_Nmeoh ) / ( ( u0_Nt ) ) );
DYDX(26) = u0_Xhac - ( ( u0_Nhac ) / ( u0_Nt ) );
DYDX(27) = u0_Xh2o - ( ( u0_Nh2o ) / ( u0_Nt ) );
DYDX(28) = u0_Fmeoh - ( u0_Xmeoh * ( ( u0_Pvmeoh ) / ( u0_Ptotal ) ) * u0_Ftotal );
DYDX(29) = u0_Fmeac - ( u0_Xmeac * ( ( u0_Pvmeac ) / ( u0_Ptotal ) ) * u0_Ftotal );
DYDX(30) = u0_Fhac - ( u0_Xhac * ( ( u0_Pvhac ) / ( u0_Ptotal ) ) * u0_Ftotal );

```


APPENDIX

DYDX(31) = u0_Fh2o - (u0_Xh2o * ((u0_Pvh2o) / (u0_Ptotal)) * u0_Ftotal);

DYDX=DYDX';

end

7.3 Flowsheet template

7.3.1 Flowsheet model transferred from ModDev

```

#Imported from XML format
#=====
#
ft_1=sum_i(f_1[i])
#
ft_2=sum_i(f_2[i])
#
ft_3=sum_i(f_3[i])
#
ft_8=sum_i(f_8[i])
#
0=f_1[i]+f_2[i]-f_3[i]+f_8[i]
#
0=ft_1*H_1+ft_2*H_2-ft_3*H_3+ft_8*H_8
#
ft_4=sum_i(f_4[i])
#
RKforward_Reactor[m]=A1[m]*exp(-Ea1[m]/(Rg*T_Reactor))
#
RRForward_Reactor[m]=Vt_Reactor*RKforward_Reactor[m]*XR_Forward_Reactor[m]
#
RKbackward_Reactor[m]=A2[m]*exp(-Ea2[m]/(Rg*T_Reactor))
#
RRBackward_Reactor[m]=Vt_Reactor*RKbackward_Reactor[m]*XR_Backward_Reactor[m]
#
RR_Reactor[m]=RRForward_Reactor[m]-RRBackward_Reactor[m]
#
RN_Reactor[i]=sum_m(nu[i][m]*RR_Reactor[m])
#
x_Reactor[i]=f_Reactor[i]/ft_Reactor
#
XR_Reactor[m]=x_Reactor[1:Nc]*x_Reactor[1:Nc]
#
0=f_3[i]-f_4[i]+RN_Reactor[i]
#
0=ft_3*H_3-ft_4*H_4
#
f_6[i]=SplFac_Splitter[i]*f_Splitter[i]
#
ft_6=sum_i(f_6[i])
#
f_5[i]=(1-SplFac_Splitter[i])*f_Splitter[i]
#
ft_5=sum_i(f_5[i])
#
0=f_4[i]-f_5[i]-f_6[i]
#
0=ft_4*H_4-ft_5*H_5-ft_6*H_6
#
ft_7=sum_i(f_7[i])
#
0=ft_Purge-DivFac_Purge*ft_7
#

```

```

0=ft_Purge-DivFac_Purge*ft_8
#
0=f_6[i]-f_7[i]-f_8[i]
#
0=ft_6*H_6-ft_7*H_7-ft_8*H_8

```

7.3.2 Flowsheet model implemented in MoT

```

#Imported from XML format
#=====

0=f_in1[i]+f_in2[i]-f_mixer[i]+f_recycle[i]
#
ft_reactor=sum_i(f_reactor[i])
#
RKforward_Reactor[m]=A1[m]*exp(-Ea1[m]/(Rg*T_Reactor))
#
RRForward_Reactor[m]=Vt_Reactor*RKforward_Reactor[m]*XR_Forward_Reactor[m]
#
RKbackward_Reactor[m]=A2[m]*exp(-Ea2[m]/(Rg*T_Reactor))
#
RRBackward_Reactor[m]=Vt_Reactor*RKbackward_Reactor[m]*XR_Backward_Reactor[m]
#
RR_Reactor[m]=RRForward_Reactor[m]-RRBackward_Reactor[m]
#
RN_Reactor[i]=sum2_<m>(nu[i][m]*RR_Reactor[m])
#
x_Reactor[i]=f_reactor[i]/ft_reactor
#
XR_Forward_Reactor[m]=x_Reactor_0*x_Reactor_1
XR_Backward_Reactor[m]=x_Reactor_2*x_Reactor_3

0=f_3[i]-f_4[i]+RN_Reactor[i]
#
f_splitter[i]=SplFac_Splitter[i]*f_reactor[i]
#
f_product[i]=(1-SplFac_Splitter[i])*f_reactor[i]
#
0=f_purge[i]-DivFac_Purge*f_splitter[i]
#
0=f_recycle[i]-(1-DivFac_Purge)*f_splitter[i]

```

7.3.3 Williams-Otto plant model based on the template

```

#Imported from XML format
#=====
#Mixer

0=f_in1[i]+f_in2[i]-f_mixer[i]+f_recycle[i]

#Reactor

0=f_mixer[i]-f_reactor[i]+RN_Reactor[i]

RKforward_Reactor[m]=A1[m]*exp(-Ea1[m]/(T_Reactor))

RRForward_Reactor[m]=Vt_Reactor*rho*RKforward_Reactor[m]*XR_Forward_Reactor[m]

```

APPENDIX

```
RR_Reactor[m]=RRForward_Reactor[m]

RN_Reactor[i]=sum2_<m>(nu[i][m]*RR_Reactor[m])

x_Reactor1_0=x_Reactor_0
x_Reactor2_0=x_Reactor_1
x_Reactor1_1=x_Reactor_2
x_Reactor2_1=x_Reactor_1
x_Reactor1_2=x_Reactor_4
x_Reactor2_2=x_Reactor_2

XR_Forward_Reactor[m]=x_Reactor1[m]*x_Reactor2[m]

ft_reactor=sum_i(f_reactor[i])

x_Reactor[i]=f_reactor[i]/ft_reactor

#Splitter

f_decanter[i]=SplFac_decanter[i]*f_reactor[i]
f_waste[i]=(1-SplFac_decanter[i])*f_reactor[i]
f_column[i]=SplFac_column[i]*f_decanter[i]
f_product[i]=(1-SplFac_column[i])*f_decanter[i]

f_column_4=SplFac_P*f_decanter_3
f_product_4=f_decanter_4-f_column_4

#Purge

0=f_purge[i]-DivFac_Purge*f_column[i]

0=f_recycle[i]-(1-DivFac_Purge)*f_column[i]
```

NOMENCLATURE

A_{cs}, A'_{cs}	cross-sectional area of shell and tube sides [m^2]
a_i	constant, 1 for hydrogen, 0 otherwise [-]
a_{dr}	specific surface of the droplet [cm^2]
Cp_i, Cp'_i	heat capacity of component i on shell and tube sides [J/mole/K]
c_i^k	concentration of compound i in phase k [mole/ m^3]
D_{ij}, D'_{ij}	binary diffusivity of i into j on shell and tube sides [m^2/s]
D_{ik}, D'_{ik}	Knudsen diffusivity of i on shell and tube sides [m^2/s]
D_{ij}^e, D'_{ij}^e	effective binary diffusivity of i into j on shell and tube sides [m^2/s]
D_{ik}^e, D'_{ik}^e	effective Knudsen diffusivity of i on shell and tube sides [m^2/s]
D_p, D'_p	diameter of catalyst particles on shell and tube sides [m]
D_t	diameter of tube [m]
$DivFac_Purge$	separation factor of component i in the divider
E_i, E'	activation energy of reaction i on shell side and hydrogenation reaction on tube side [J/mole/K]
Ea_i	activation energy of reaction i [J/mole]
F_0	hydrogen peroxide solution flow rate [m^3/s]
$f_in1[i]$	component flow rate of compound i in the first inlet stream
$f_in2[i]$	component flow rate of compound i in the second inlet stream
$f_mixer[i]$	component flow rate of compound i in the mixer outlet stream
$f_product[i]$	component flow rate of compound i in the product stream
$f_purge[i]$	component flow rate of compound i in the purge outlet stream
$f_recycle[i]$	component flow rate of compound i in the recycle stream

NOMENCLATURE

$f_{reactor}[i]$	component flow rate of compound i in the reactor outlet stream
$f_{splitter}[i]$	component flow rate of compound i in the splitter outlet stream
$H_{H_2O_2}$	partition coefficient
H_i, H'_i	enthalpy of component i on shell and tube sides [J/mole/K]
J_i	molar flux of component i [mole/m ² /s]
k_i	reaction i rate constant
k_{i0}	reaction i pre-exponential factor [mole/K/kg cat/s/bar]
k_L	mass transfer coefficient for hydrogen peroxide [m/s]
k_i^{ref}	kinetic constant [-]
k_{ti}	thermal conductivity, [Jm/s]
N	number of tubes in hybrid reactor [-]
N_i, N'_i	molar flux of component I inside catalyst particles on shell and tube sides [mole/m ² /s]
NR	
n_i, n'_i	molar flow rate of component i on shell and tube sides [mole/s]
P, P'	total pressure on shell side and tub side [bar]
p_i, p'_i	partial pressure of component i on shell and tube sides [bar]
R	universal gas constant [J/mole/K]
Q	heat transferred from tube side to shell side [W/m]
Q_0	pre-exponential constant of hydrogen membrane [mole/m/s/bar ^{0.5}]
r_j	rate o reaction j on shell side [mole/kg cat/s]
r'	rate of hydrogenation reaction on tube side [mole/kg cat/s]
r_{pore}	catalyst pore radius [m]
r_1, r_2	inner and outer radius of hydrogenation tube [m]
$r_3 - r_2$	thickness of palladium membrane [m]
$SplFac_Splitter[i]$	separation factor of component i in the splitter
T, T'	temperature on shell and tube sides [K]

NOMENCLATURE

T_k	reaction temperature [K]
t	time [s]
V^k, V_k	volume of the phase k (aqueous, organic) [m ³]
x_i, x'_i	mole fraction inside catalyst pellet on shell and tube sides [-]
y, y'	radial coordinate inside catalyst pellet on shell and tube sides [-]
z	axial coordinate along reactor [m]

Greek letters

$\varepsilon, \varepsilon'$	catalyst porosity [-]
ν_i	stoichiometric coefficient of the reaction j
η	effectiveness factor [-]
ρ	catalyst density [kg/m ³]
ρ_{H_2O}	water density [kg/m ³]
τ	catalyst tortuosity
ω, ω'	dimensionless radial distance inside catalyst pellets on shell and tube sides [-]
δ_{H_2}	thickness of hydrogen permeation membrane [m]
σ_{ij}	stoichiometric coefficient of reactant i in reaction j [-]

Acronyms

AC	Acid
AcAcid	Acetic Acid
AD	Aldehyde
AE	Algebraic Equation
AL	Alcyloln
D	Diol
DAE	Differential Algebraic Equation
EP	Epoxide

NOMENCLATURE

gPROMS	General Process Modelling System
HPAE	Hydroperoxystearic acid methyl ester
HWM	High Weigh Molecule
ICAS	Integrated Computer-Aided System
KE	Ketone
MeAC	Methylacetate
MeOH	Methanol
ModDev	Model Development
MoT	Modelling Testbed
ModTem	Modelling Templates
ODE	Ordinary Differential Equation
PDAE	Partial Differential-Algebraic Equation
PDE	Partial Differential Equation
UFA	Unsaturated Fatty Acid

REFERENCES

Abo-Ghander N.S., Grace J.R., Elnashaie S.S.E.H., Lim C.J., 2008, Modeling of a novel membrane reactor to integrate dehydrogenation of ethylbenzene to styrene with hydrogenation of nitrobenzene to aniline, *Chemical Engineering Science*, 63(7), 1817-1826.

Abo-Ghander N.S., Logist F., Grace J.R., Van Impe J.F.M., Elnashaie S.S.E.H., Lim C.J., 2012, Comparison of diffusion models in the modeling of a catalytic membrane fixed bed reactor coupling dehydrogenation of ethylbenzene with hydrogenation of nitrobenzene, *Computers and Chemical Engineering*, 38, 11-23.

Andersson M., 1994, Object-oriented modelling and simulation of hybrid systems. PhD thesis, Department of automatic control, Lund Institute of Technology, Sweden.

Amon, B., Redlingshöfer, H., Klemm, E., Dieterich, E., & Emig, G. (1999). Kinetic investigations of the deactivation by coking of a noble metal catalyst in the catalytic hydrogenation of nitrobenzene using a catalytic wall reactor. *Chemical Engineering and Processing*, 38(4–6), 395–404.

Ausbrooks R., Buswell S., Carlisle D., Chavchanidze G., Dalmas S., Devitt S., Diaz A, Dooley S., Hunter R., Ion P., Kohlhase M., Lazrek A., Libbrecht P, Miller B., Miner R., Rowley C., Sargent M., Smith B., Soiffer N., Sutor R., Watt S., 2014, Mathematical markup language (MathML) – Version 3.0 2nd edition, available online: <http://www.w3.org/TR/MathML/> (accessed 28.03.2015)

AspenTech, 2015a, Aspen Plus, Aspen Technology, Inc. Burlington MA, USA, <http://www.aspentech.com/products/engineering/aspen-plus/> (accessed 07.07.2015)

AspenTech, 2015b, Aspen Custom Modeller, Aspen Technology, Inc. Burlington MA, USA, <http://www.aspentech.com/products/aspen-custom-modeler.aspx> (accessed 07.07.2015)

Barton P., Pantelides C., 1994, Multipurpose process modeling environments, *Computer-Aided Process Design*, FOCAPD'94, Snowmass CO.

REFERENCES

- Behr A., Tenhumberg N., Wintzer A., 2012, An efficient reaction protocol for the ruthenium-catalyzed epoxidation of methyl oleate, *European Journal Lipid Science Technology*, 114, 905-910
- Behr A., Tenhumberg N., Wintzer A., 2013, Efficient ruthenium-catalysed oxidative cleavage of methyl oleate with hydrogen peroxide as oxidant, *RCS Advances*, 3, 172-180.
- Biegler L.T., Grossmann E., Westerberg A.W., 1997, *Systematic methods of chemical process design*,
- Bieszczad, J.; Koulouris, A.; Stephanopoulos, G., 2000, MODEL.LA: A phenomena-based modeling environment for computer-aided process design, *FIFTH INTERNATIONAL CONFERENCE ON FOUNDATIONS OF COMPUTER-AIDED PROCESS DESIGN*, 96(323), 438-441.
- Bogusch, R., Lohmann, B., Marquardt, W., 2001, Computer-aided process modeling with MODKIT, *Computers and Chemical Engineering* 25, 963-995
- Cameron I., Gani R., 2011, *Product and process modelling. A Case Study Approach*. 1st ed., Elsevier, Amsterdam.
- Cameron I.T., Ingram G.D., 2008, A survey of industrial process modelling across the product and process lifecycle, *Comp. Chem. Eng.*, 32, 420-438.
- Codd E.F., 1970 A Relational Model of Data for Large Shared Data Banks, *Communications of the ACM* 13, 6, 377-387. doi:10.1145/362384.362685.
- Elnashaie, S. S. E. H., Abdalla, B. K., & Hughes, R. (1993). Simulation of the industrial fixed bed catalytic reactor for the dehydrogenation of ethylbenzene to styrene: Heterogeneous dusty gas model. *Industrial & Engineering Chemistry Research*, 32(11), 2537-2541.
- Fedorova M., Sin G., Gani R., 2013, Computer-aided modeling framework – a generic modeling template for catalytic membrane fixed bed reactors, *Computer Aided Chemical Engineering*, 32, 775-780.
- Fedorova M., Sin G., Gani R., 2014a, Computer-Aided Template for Model Reuse, Development and Maintenance, *Computer Aided Chemical Engineering*, 33, 817-822.

REFERENCES

Fedorova M., Papadakis E., Meisler K.T., Sin G., Gani R., 2014b Application of the Generic Modelling Template Approach to Unsaturated Fatty Acid Oxidation and Crystallization Systems, *Computer Aided Chemical Engineering*, 34, 309–314.

Fedorova M., Sin G., Gani R., 2015a, Computer aided modelling template: concept and application, *Computers and Chemical Engineering*,
<http://dx.doi.org/10.1016/j.compchemeng.2015.02.010>

Fedorova M., Tolksdorf G., Fillinger S., Wozny G., Sales-Cruz M., Sin G., Gani R., 2015b, Development of Computer Aided Modelling Templates for Model Re-use in Chemical and Biochemical Process and Product Design: Import and export of models, *Computer Aided Chemical Engineering*, 37, 953–958.

B. Foss, B. Lohmann, W. Marquardt, 1998, A field study of the industrial modeling process, *Journal of Process Control*, 8, 5-6, 325-338.

Fraga E.S., Gani R., Ponton J.W., Andrews R., 2002, Tools integration for computer aided process engineering applications, *Comp. Aid. Chem. Eng.*, 11, 485-513.

Gallot J., Kapoor M., Kaliaguine S., 1998, Kinetics of 2-hexanol and 3-hexanol oxidation reaction over TS-1 catalysts, *AIChE Journal*, 44(6), 1438-1454.

Gani R., Hytoft G., Jaksland C., Jensen A.K., 1997, An integrated computer aided system for integrated design of chemical processes, *Comput. Chem. Eng.*, 21(10), 1135-1146.

Hai R., Theißen M., Marquardt W., 2011, An ontology based approach for operational process modeling, *Adv. Eng. Informatics*, 25, 748-759.

Hangos K., Cameron I., 2011, Process modelling and model analysis, *Process Systems Engineering*, 1st edn., Academic press, Harcourt Place, 32 Jamestown Road, London NW1 7BY, UK.

Heitzig M., 2010, Computer-aided framework for regression and multi-scale modelling needs in innovative product-process engineering, *Comp. Aid. Chem. Eng.*, 28, 379-384.

Heitzig M., Sin G., Sales-Cruz M., Glarborg P., Gani R., 2011a, Computer-aided modeling framework for efficient model development, analysis and identification: combustion and reactor modeling, *Ind. Eng. Chem. Res.*, 50(9), 5253-5265.

REFERENCES

- Heitzig M., Gregson C., Sin G., Gani R., 2011b, Application of computer-aided multi-scale modelling framework – Aerosol case study, *Comput. Aid. Chem. Eng.*, XX, 16-20.
- Heitzig M., 2011, Computer-aided modeling for efficient and innovative product-process engineering, PhD Thesis, Technical University of Denmark, Lyngby, Denmark.
- Heitzig M., Linninger, A., Sin G., Gani R., 2014, A computer-aided framework for development, identification and management of physiologically-based pharmacokinetic models, *Computers and Chemical Engineering*, 71, 677-698.
- Hoang D. M., Barz T., Merchan V. A., Biegler L., Arellano-Garcia H., 2013, Simultaneous solution approach to model-based experimental design, *AIChE Journal*, doi: 10.1002/aic.14145
- Jensen A.K., 1998, Generation of problem specific simulation methods within an integrated computer sided system. PhD thesis, Technical University of Denmark, Lyngby, Denmark.
- Jensen A.K., Gani R., 1999, A computer aided modeling system, *Computers and Chemical Engineering*, 23, 673-678.
- Klatt K.-U., Marquardt W., 2009, Perspectives for process systems engineering - Personal views from academia and industry, *Computers and Chemical Engineering*, 33, 536-550.
- Köchritz A., Martin A., 2008, Oxidation of unsaturated fatty acid derivatives and vegetable oils, *European Journal Lipid Science Technology*, 110, 812-824.
- Körtgen A., Nagl M., 2011, Tools for consistency management between design products, *Comp. Chem. Eng.*, 35, 724-735.
- Kraus R., Merchan V.A., Arellano-Garcia H., Wozny G., 2012, Hierarchical simulation of integrated chemical processes with a web based modeling tool, *Computer Aided Chemical Engineering*, 31, 15-19.
- Kraus, R., Fillinger, S., Tolksdorf, G., Minh, Duc H., Merchan-Restrepo, Victor A. and Wozny, G. 2014, Improving Model and Data Integration Using MOSAIC as Central Data Management Platform, *Chemie Ingenieur Technik*, doi: 10.1002/cite.201400007.

REFERENCES

- Kuntsche S., Arellano-Garcia H., Wozny G., 2011, MOSAIC, an environment for web-based modeling in the documentation level, *Computer Aided Chemical Engineering*, 29, 1140-1144.
- Levenspiel O., 1998, *Chemical Reaction Engineering*, Wiley.
- Linninger A., Stephanopoulos G., A natural language approach for the design of batch operating procedures, *Informatica, special issue on natural language processing and multi-agent systems*, 22 (4), ISSN 0250-5596.
- Linninger A., Krendl H., 1999, TechTool – Computer-aided generation of process models (Part 1 – A generic mathematical language), *Comp. Chem. Eng.*, 23, 703-706.
- Linninger A., Chowdhry S., Bahl V., Krendl H., Pinger H., 2000, A systems approach to mathematical modeling of industrial processes, *Comp. Chem. Eng.*, 24, 591-598.
- Linninger A., 2000, Metamodelling – a novel approach for phenomena-oriented model generation, *AIChE Symposium Series*, 96 (323), 462-465.
- Magioglou V., Tsakalova M., Kokossis A., 2014, Natural language modelling in process synthesis and optimization, *Comp. Aid. Chem. Eng.*, 33, 877-882.
- Magioglou V., Garcia E.C., Badr S., Kokossis A., 2015, Model-based analysis of waste management systems through a natural language approach, *Comp. Aid. Chem. Eng.*, 37, 977-982.
- Marquardt W., 1996, Trends in computer-sided process modelling, *Comp. Chem. Eng.*, 20(6/7), 591-609.
- MathWorks, Inc., 2015, Matlab, <http://se.mathworks.com/products/matlab/> (accessed 28.03.2015).
- McGahey S.L., Cameron I.T., 2007, A multi-model repository with manipulation and analysis tools, *Comp. Chem. Eng.*, 31, 919-930.
- Merchan V. A., Tolksdorf G., Kraus R., Wozny G., 2014, Extending Documentation-Based Models towards an Efficient Integration into Commercial Process Simulators, *Chem. Ing. Tech.*, doi: 10.1002/cite.201400014
- Morbach J., Yang A., Marquardt W., 2007, OntoCAPE – A large-scale ontology for chemical process engineering, *Eng. Appl. Of Artificial Intelligence*, 20, 147-161.

REFERENCES

- Morbach J., Wiesner A., Marquardt W., 2009, OntoCAPE – A (re)usable ontology for computer-aided process engineering, *Comp. Chem. Eng.*, 33, 1546-1556.
- Mošat A., Lueshen E., Heitzig M., Hall C., Linninger A., Sin G., Gani R., 2013, First principles pharmacokinetic modeling: A quantitative study on Cyclosporin, *Computers and Chemical Engineering*, 54, 97-110.
- Oh M., Pantelides C., 1995, A modelling and simulation language for combined lumped and distributed parameter systems, *Comp. Chem. Eng.*, 20 (6/7), 611-633.
- Papadakis E., 2013, Modeling the oxidation of Unsaturated Fatty Acids (UFA) by hydrogen peroxide (H₂O₂), MSc Thesis, Technical University of Denmark, Lyngby, Denmark.
- Piela P., 1989, ASCEND: and object-oriented computer environment for modeling and analysis, PhD thesis, Carnegie Mellon University, Pittsburgh, PA.
- Piela P., Epperly T., Westerberg K., Westerberg A., 1991, Ascend – and object-oriented computer aided environment for modeling and analysis: The modeling language, *Comp. Chem. Eng.*, 15, 53-72.
- Piela P, McKelvey R., Westerberg A., 1992, An introduction to ascend: its language and interactive environment, *J. Man. Info Sci.*, 9, 91-121.
- Preisig, H.A., 2010, Constructing and maintaining proper process model, *Comput. Chem. Eng.*, 34, 1543-1555.
- Preisig H., Haug-Warberg T., 2012, Ontology approach to model construction, *Comp. Aid. Chem. Eng.*, 30, 992-996.
- Process Systems Enterprise, 2014, gPROMS, <http://www.psenderprise.com/gproms/> (accessed 28.03.2015).
- Rico-Ramirez V., 1998, Representation, analysis and solution of conditional models in an equation-based environment, PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania.
- Russel B.M.R., Gani R., 2000, MoT – a modelling test-bed, In: ICAS Manual (CAPEC Report), Technical University of Denmark, Kgs. Lyngby.

REFERENCES

- Sales-Cruz M., 2006. Development of a computer-aided modelling system for bio and chemical process and product design. PhD thesis, Technical University of Denmark, Lyngby, Denmark.
- Sales-Cruz M., Gani R., 2003, A modelling tool for different stages of the process life, *Comp. Aid. Chem. Eng.*, 16, 209-249.
- Schneider Electric, 2015, SimSci PRO/II, Schneider Electric Software, LLC, <http://software.schneider-electric.com/products/simsci/design/pro-ii/> (accessed 07.07.2015).
- Schopfer G., Yang A., von Wedel L., Marquardt W.: 2004, Cheops: A tool-integration platform for chemical process modelling and simulation, *Int J Software Tools Transfer* 6, 186–202.
- Stephanopoulos G., Henning G., Leone H., 1990, Model.II a modeling language for process engineering – i. the formal framework, *Comp Chem. Eng.*, 14(8), 813-846.
- Suwanwarangkul R., Croiset E., Fowler M.W., Douglas P.L., Entchev E., Douglas M.A., 2003, Performance comparison of Fick's, dusty-gas and Stefan-Maxwell models to predict the concentration over potential of a SOFC anode, *Journal of Power Sources*, 122, 9-18.
- Tolksdorf G., Fillinger S., Kraus R., Wozny G., Manenti F., Rossi F., Buzzi-Ferraris G., Pierucci S., Fedorova M., Gani R., 2015, A Posteriori Integration of University CAPE Software Developments, *Chemical Engineering Transactions*, 43.
- Veldsink J.W., van Damme R.M.J., Versteeg G.F., van Swaaij W.P.M., 1995, The use of the dusty-gas model for the description of mass transport with chemical reaction in porous media, *The Chem. Eng. Journal*, 57, 115-125.
- Venturello C., Antonelli E., D'Aloiso R., Gambaro M. and Fiorani T., 1998, Efficient Oxidative Cleavage of Olefins to Carboxylic Acids with Hydrogen Peroxide Catalyzed by MethyltrioctylammoniumTetrakis (oxodiperoxotungsto)phosphate(3-) under Two-Phase Conditions. Synthetic Aspects and Investigation of the Reaction Course, *Journal of Organic Chemistry*, vol. 63, pp. 7190-7206.
- Uschold, M.; Gruninger, M., 1996, Ontologies: Principles, methods and applications, *Knowledge Engineering Review*, 11(2), 93-136.

REFERENCES

Westerweele M.R., 2003, Five steps for building consistent dynamic process models and their implementation in the computer tool modeller. Ph.D. thesis, The Netherlands, Eindhoven University of Science and Technology.

Williams T.J., Otto R.E., 1960, A generalized chemical processing model for the investigation of computer control, AIEE Trans., 79, 458-473.

World Wide Web Consortium, 2008, XML 1.0 specification, available online: <http://www.w3.org/TR/REC-xml/> (accessed 28.03.2015).

Yang A., Marquardt W., 2004, An ontology-based approach to conceptual modelling, Comp. Aid. Chem. Eng., 18, 1159-1164.

Yang A., Marquardt W., 2009, An ontological conceptualization of multiscale models. Comput. Chem Eng., 33, 822-837.

Zerry R., Gauss B., Urbas L., Wozny G., 2004, Web-based object oriented modelling and simulation using MathML, 2004, Comput. Aided Chem. Eng., 18, 1171-1176, DOI: 10.1016/S1570-7946(04)80261-X

Zhan G., Hong Y., Lu F., Ibrahim A., Du M., Sun D., Huang J., Li Q., Li J., 2013, Kinetics of liquid phase oxidation of benzyl alcohol with hydrogen peroxide over bio-reduced Au/TS-1 catalysts, Journal of Molecular Catalysis A:Chemical, 336, 215-221, 2013.

Zhao Y., Jiang C., Yang A., 2011, Towards a Generic Simulation Environment for Multiscale Modelling based on Tool Integration, Computer Aided Chemical Engineering, 29, 76-80.

CAPEC-PROCESS Research Center
Department of Chemical and Biochemical Engineering
Technical University of Denmark (DTU)

Søltofts Plads, Building 229
DK-2800 Kgs. Lyngby

Tlf. +45 4525 2800
Fax +45 4525 4588