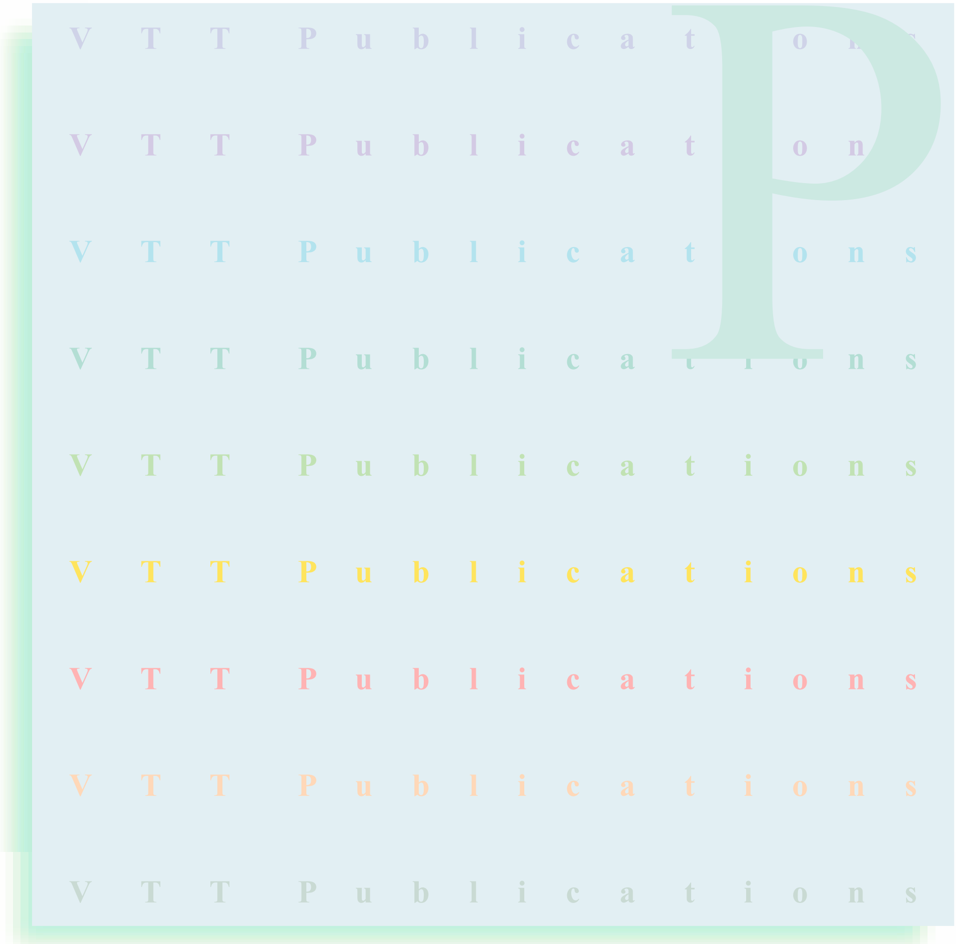
438

VTT PUBLICATIONS

Antti Pasanen

Phenomenon-Driven Process Design methodology

Computer implementation and test usage



TECHNICAL RESEARCH CENTRE OF FINLAND ESPOO 2001

VTT PUBLICATIONS 438

Phenomenon-Driven Process Design methodology

Computer implementation and test usage

Antti Pasanen

VTT Chemical Technology

Dissertation for the degree of Doctor of Technology to be presented, with the permission of the Department of Process and Environmental Engineering of the University of Oulu, for public discussion in Auditorium L10, on September 8th, at 12 noon.

TECHNICAL RESEARCH CENTRE OF FINLAND
ESPOO 2001

ISBN 951–38–5854–5 (soft back ed.)
ISSN 1235–0621 (soft back ed.)

ISBN 951–38–5855–3 (URL:http://www.inf.vtt.fi/pdf/)
ISSN 1455–0857 (URL:http://www.inf.vtt.fi/pdf/)

Copyright © Valtion teknillinen tutkimuskeskus (VTT) 2001

JULKAISIJA – UTGIVARE – PUBLISHER

Valtion teknillinen tutkimuskeskus (VTT), Vuorimiehentie 5, PL 2000, 02044 VTT
puh. vaihde (09) 4561, faksi (09) 456 4374

Statens tekniska forskningscentral (VTT), Bergsmansvägen 5, PB 2000, 02044 VTT
tel. växel (09) 4561, fax (09) 456 4374

Technical Research Centre of Finland (VTT),
Vuorimiehentie 5, P.O.Box 2000, FIN–02044 VTT, Finland
phone internat. + 358 9 4561, fax + 358 9 456 4374

VTT Kemiantekniikka, Prosessit ja Ympäristö, Biologinkuja 7, PL 1401, 02044 VTT
puh. vaihde (09) 4561, faksi (09) 456 7026, (09) 456 7021

VTT Kemiteknik, Processer och miljö, Biologgränden 7, PB 1401, 02044 VTT
tel. växel (09) 4561, fax (09) 456 7026, (09) 456 7021

VTT Chemical Technology, Processes and Environment
Biologinkuja 7, P.O.Box 1401, FIN–02044 VTT, Finland
phone internat. + 358 9 4561, fax + 358 9 456 7026, + 358 9 456 7021

Technical editing Leena Ukskoski

Otamedia Oy, Espoo 2001

Pasanen, Antti. Phenomenon-Driven Process Design methodology. Computer implementation and test usage. Espoo 2001. Technical Research Centre of Finland, VTT Publications 438. 140 p. + app. 26 p.

Keywords research and development, process design, systematization, conseptual design, models, modelling, simulation, computer tools, PSSP, chemical engineering

Abstract

This work focuses on how to develop a generic computer tool support for a design methodology proposed for chemical engineering purposes. The applied methodology is called Phenomenon Driven Process Design methodology (PDPD), and the formal language of the methodology is called PSSP language (an acronym for Purpose, Structure, State and Performance attributes). The focus of this work is divided into two theses

1. "Multi-characteristic and creative chemical process research and development work can be supported in a systematic manner – based on PDPD – and implemented as a computer system."
2. "The use of this computer system is expected to result in comprehensive process models and modelling work with efficient documentation and data management. These are prospective advantages for any chemical engineering R&D organisation."

The developed tool is tested and evaluated by employing it in a few academic and industrial pilot projects, but the methodology has not been taken into practice by industrial companies. The introduced test cases illustrate the formalisation of research data and knowledge of both the project and process models. The test cases also illustrate a phenomenon-driven way to argument process models and project management. The analysis and argumentation of both the developed tool and the test cases are written transparently for discussion. In addition, the usage and characteristics of the meta-level tool for capturing the methodology is analysed in an open-formatted way. The use of the developed prototype application is expected:

a) to save project meeting time and to improve meeting performance,
b) to enhance data and knowledge exchange among project staff,
c) to improve project and process model data retrieval,
d) to ease the utilisation of various process modelling software and
e) to improve the comprehension of the linkages between numerical results, conceptual process models and the set project goals.

The research work behind this thesis has been carried out during 1994 to 2000. PDPD and PSSP–based research work for building computer tool support are still ongoing.

Acknowledgements

The author acknowledges Professor Veikko Pohjola for giving the opportunity to undertake research on systematising chemical engineering. During the years of research, the author has started to perceive the physico-chemical systems in a new way.

The post-graduate studies ware mainly carried out under the Finnish Graduate School of Chemical Engineering (GSCE), which provided a very enjoyable group of people.

During the final work the author was employed by VTT Chemical Technology, where the colleagues gave true encouragement. Special thanks are given to docent Markus Olin, who has advised the author with his valuable comments during the writing of this dissertation. He also has helped to utilise the results of this research and to take on new challenges beyond the academic work.

If not for other researchers willing to provide test data and comments, there would not have been any testing procedures, and therefore no dissertation. In this respect, the author acknowledges Juha Ahola (University of Oulu), Juha Anttila (University of Oulu) and Kemira Chemicals for giving the opportunity to utilise their projects as case studies for this thesis. Also MetaCase Consulting is acknowledged for guiding the use of special computer aids during the test procedures.

My sincerest thanks and acknowledgement, however, belong to my family, Sanna and Iida. Without my wife's full support I would probably have never finished this work.

Espoo, 13.8.2001

Antti Pasanen

Contents

Glossary

A class: Defines a set of general concepts. Individual objects can be named for a class. *E.g.*, car, house, chemical plant, tube reactor, batch reactor or input flow.

Attribute: Property of on object. In the case of PSSP language, an object always has properties: Purpose, Structure, State and Performance.

An object: Named and focused (set of) target, or property of a system under observation. There are various sets of object-oriented theories, which cover objects and attributes, or properties, differently.

Aggregation and disaggregation: Terms for building a more detailed (disaggregation) or more abstracted (aggregation) description for an object.

CAME: Computer-Aided Methodology Engineering. New computer engineering area, where most applications are made for software development, business modelling and electronics. Through a CAME tool a user can follow and apply a methodology (or a method) more efficiently.

Conceptual model: An abstract model, which describes an object with certain symbols, and with a certain notation. *E.g.* selected chemical reaction pathway and means to control the reactions, block diagram and a flowsheet. Conceptual models are the result of conceptual design. Conceptual design does not usually cover detailed design, dimensioning and layout design.

Data model: A data model describes static abstractions of the system based on predefined modelling concepts. *E.g.* in this thesis PSSP is proposed and used as a modelling language for the data model of physico-chemical processes.

GOPRR: Graph, Object, Property, Relationship, Role of a relationship. GOPRR is a metamodelling language proposed for the modelling of any method or methodology. MetaEdit+ is based on this meta-modelling language and provides tools for building the graphical notation of the method or methodology.

GUI and UI: User Interface (UI) and Graphical User Interface (GUI) refer to computer tools, and to the screen appearance. GUI usually means a simple (higher level) notation and a form to perceive and to edit output and input data, *e.g.* for a complex process simulation code.

Mathematical model: More formal description of an object, or object behaviour, with operands and operators, which normally also has a numerical representation. *E.g.* (set of) algebraic equation(s) and (set of) differential equation(s)

Metadata model: Metadata models describe static aspects of methodologies, such as their conceptual structure and notation. In this thesis, GOPRR is used as a meta-data model for PSSP.

Metamodel: Metamodelling can be defined as a modelling process which takes place one level of abstraction and logic higher than the standard modelling process (van Gigch 1991). The result of metamodelling is a metamodel. *E.g.* PDPD and PSSP language are proposed and used in this thesis as a metamodel for physico-chemical systems, describing both the structure and behaviour of a chemical process.

MetaEdit+ (ME+): is a commercial computer program proposed for method(ology) engineering and method usage. ME+ is a graphical user interface, which applies GOPRR language.

Method: Set of procedures for achieving an objective. Analogous to methodology, but normally associated with a more reproducible (or simpler) set of actions. *E.g.* an enzymatic method to hydrolyse starch, and an acid-based method to hydrolyse starch.

Methodology: Set of procedures, or methods, for achieving defined objectives. A methodology usually has various methods. *E.g.* Process designs can be built by following Phenomenon-Driven Process Design methodology.

Model: Description, or prescription, of an object (real or imaginary) with various types of symbols, or physical parts. *E.g.* miniature model of an aeroplane and mathematical model of a mass balance for a process (the model is built with symbols and with a certain mathematical notation)

Notation: Notation means how some symbol system is used. Communication with some symbols is difficult without a clear notation for the symbols. *E.g.* notation for process flowsheeting with a flowsheet editor, notation for batch reactor design with a batch reactor design tool.

Object-oriented: Refers to a certain way to perceive and model the world. There are various object-oriented modelling techniques (see *e.g.* OSA), each having a certain notation for classes, properties and relationships. It is claimed (*e.g.* Embley *et al.* 1992) that man naturally reasons according to some object-oriented way.

OSA: Object-Oriented System Analysis (Embley *et al.* 1992). A method to model systems (imaginary or real) with a certain notation.

PDPD: Phenomenon-Driven Process Design. PDPD refers to a certain way of perceiving and building models for real and imaginary objects. PDPD is originally proposed for conceptual chemical process design.

PSSP: An acronym for attributes; Purpose, Structure, State and Performance. PSSP language uses a certain set of modelling elements, each having the PSSP attributes. PSSP language has a notation for how to use the modelling elements, and how to build process models.

Process model: Structural and behavioural description (usually as mass and energy balances) of a physico-chemical process.

Process metamodel: A metamodel which has building blocks and principles of how to describe physico-chemical processes. *E.g.* a process model always has some predefined specifications of its structure.

Relational Data Base (DB) and Object repository: Relational Data Base is a database where data is stored in tables. Through tables the data can be queried in various ways, and provide information of how individual data elements are related in various data sets. Object Repository means here object data-base. Object repository has some predefined ways of how the data is stored, manipulated and queried. In the case PDPD and PSSP language, the data structure can be built both with various object repositories or with relational databases.

1. Introduction

This thesis focuses on the systematisation of the conceptual design of chemical processes by utilising a process design methodology. The applied methodology is called Phenomenon-Driven Process Design methodology (PDPD), which is developed for helping to manage and document conceptual chemical process models, as well as design and research activities. The development work has been performed during the past ten years at the University of Oulu by Professor Pohjola with his group[1] (Pohjola *et al.* 1993a, 1993b, Tanskanen *et al.* 1995, Pohjola and Tanskanen 1998).

Thus, this thesis is part of a larger undertaking for developing, analysing, computer implementing and testing the methodology. Figure 1 lists the studies on development and testing of PDPD. All the studies in the figure are projects in them-selves with academic and industrial partners, and so far they have resulted in a few licentiate and doctoral theses with a few more theses due to be finished in the near future. PDPD methodology has been developed within two Tekes (the National Technology Agency of Finland) projects, CARD (Computer-Aided Reactor Design, 1987–1991) and CARD-2 (1991–1994), and within two projects called MetCa (Methodologically Systematised, 1996–1998), within the MATRA programme (Academy of Finland), and MeCaNOx (1998–2000), which was financed by Tekes (under the Nano-Technology programme). Currently there is an ongoing EU project called POEM[2] (Process Object Engineering Methodology, 1999–2000) and a project on the drawing board called POEM DESC[3] for international Intelligent Manufacturing Systems (IMS). These two projects are aimed at applying the methodology and developing tool support for the use of the methodology. At the University of Oulu there are two ongoing projects: "Theory of waste management" and "Development of non-wood pulping process", which both utilise and develop PDPD further. There is also a smaller ongoing project for modelling geo-chemical systems, in which the comprehension of the models and the modelling work are enhanced by applying formal PSSP language behind PDPD. In this project, PSSP is to be utilised for the purposes of POSIVA, but the work is beyond the scope of this thesis. So far,

---

[1] http://cc.oulu.fi/~pokemwww/

[2] http://www.vtt.fi/aut/rm/projects/poem/

[3] http://www.ims.org/project/projinfo/poemdesc.htm

PDPD has not been taken into practice by industrial companies, but it is still under refinement and test usage.

The developer group of PDPD has experienced that it is truly challenging to make an engineer, or any person, understand the methodology and the ideas behind it. This can be reasoned by the fact that PDPD invites users to take an objective and holistic view of their actions and of things that they work with, such as process models, data and numerical methods.

The use of PDPD is expected to result in a conceptual model of processing systems, linked with mathematical models, and modelling activities, which together simultaneously are not usually taken under formalised and systematised documentation and evaluation. At first glance, PDPD might make the user feel uncomfortable by making him analyse his actions and his way of doing things. But it has been noticed that, if a person puts some effort into understanding PDPD, then he will probably adopt the methodology also beyond chemical engineering. The use of PDPD does not necessarily result in new knowledge about the subject – process models, numerical techniques or project management – but it is expected to enhance a systematic way to document decisions and data, assisting team work and project management. The effect of PDPD usage depends on how an engineering community originally works. The way in which PDPD invites the user to view the real world – especially chemical processes and conceptual design – is unique, since it requires making a fundamental and holistic analysis of conceptual design work, which is also integrated with process models, data and knowledge.

For this work a mission has been posed "to give a comprehensive report and study about the status of the methodology development work and the applicability of PDPD as a computer tool". This mission has been formalised further as two theses:

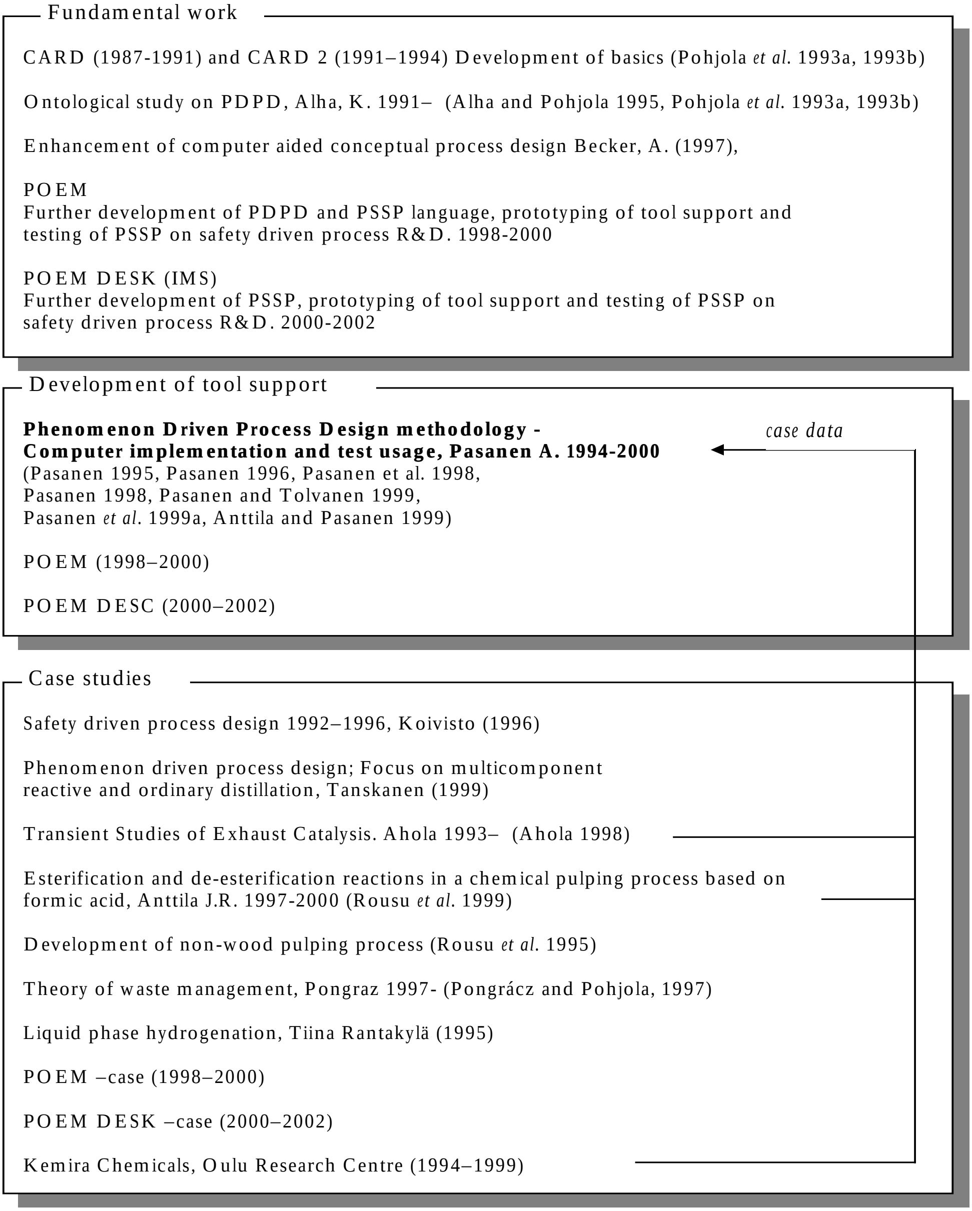
1. "Multi-characteristic and creative chemical process research and development work can be supported in a systematic manner – based on PDPD – and implemented as a computer system."

2. "The use of this computer system is expected to result in comprehensive process models and modelling work with efficient documentation and data management. These are prospective advantages for any chemical engineering R&D organisation."

The author defends these theses by analysing and characterising the development of a computer tool support and by reporting the pilot projects in which PDPD and the tool are applied. Demonstrating the methodology with pilot projects probably eases the application and understanding of PDPD, as well as the philosophical thinking behind it. The demonstration is given as a thorough analysis of the methodology for computer implementation purposes, and the demonstration is also given as a test use of the implemented computer tool. The formal language behind the methodology, called PSSP language, and the modelling principles – performance-driven strategy and phenomenon-driven ordering – have been combined within a prototype tool which has been used for the pilot projects.

The author has tried to formulate the thesis so that it provides an objective demonstration and evaluation of the implementation of the prototype. The working procedure and the evaluation criteria of the prototype tool that have been used, as well as the case projects are made as open and transparent as possible. In this way it is expected that the conclusions of the performed research work are not so much dependent on the author's opinion, but are objective observations.

Chapters 2 to 4 introduce and pinpoint this thesis to research work done in the area of systematisation of chemical process R&D. This is done by mapping the other studies and by trying to indicate how similar or different they are. In this way the reader is introduced to the viewpoint behind this thesis. Chapter 5 introduces the methodology and Chapter 6 introduces the computer implementation. Chapter 7 briefly describes three test cases, and the overall evaluation is given in Chapter 8.



Fundamental work

CARD (1987-1991) and CARD 2 (1991–1994) Development of basics (Pohjola *et al.* 1993a, 1993b)

Ontological study on PDPD, Alha, K. 1991– (Alha and Pohjola 1995, Pohjola *et al.* 1993a, 1993b)

Enhancement of computer aided conceptual process design Becker, A. (1997),

POEM
Further development of PDPD and PSSP language, prototyping of tool support and testing of PSSP on safety driven process R&D. 1998-2000

POEM DESK (IMS)
Further development of PSSP, prototyping of tool support and testing of PSSP on safety driven process R&D. 2000-2002

Development of tool support

Phenomenon Driven Process Design methodology - Computer implementation and test usage, Pasanen A. 1994-2000
(Pasanen 1995, Pasanen 1996, Pasanen et al. 1998,
Pasanen 1998, Pasanen and Tolvanen 1999,
Pasanen *et al.* 1999a, Anttila and Pasanen 1999)

POEM (1998–2000)

POEM DESC (2000–2002)

Case studies

Safety driven process design 1992–1996, Koivisto (1996)

Phenomenon driven process design; Focus on multicomponent reactive and ordinary distillation, Tanskanen (1999)

Transient Studies of Exhaust Catalysis. Ahola 1993– (Ahola 1998)

Esterification and de-esterification reactions in a chemical pulping process based on formic acid, Anttila J.R. 1997-2000 (Rousu *et al.* 1999)

Development of non-wood pulping process (Rousu *et al.* 1995)

Theory of waste management, Pongraz 1997- (Pongrácz and Pohjola, 1997)

Liquid phase hydrogenation, Tiina Rantakylä (1995)

POEM –case (1998–2000)

POEM DESK –case (2000–2002)

Kemira Chemicals, Oulu Research Centre (1994–1999)

Figure 1. Development of PDPD methodology and PSSP language.

2. Status of systematising chemical process R&D

The systematisation of conceptual chemical process design is now briefly reviewed. Systematisation of the conceptual design phase of chemical processes can be approached in various ways: development of better numerical solvers, development of better systems of tool kits for various design problems, and development of information systems for enhancing the integration of design data and models. On the other hand, industrial chemical engineering companies are also utilising the development of new methods and tools for project management, as any other project-oriented industrial organisation. This brief review is not intended to give a thorough analysis of systematising chemical engineering work, but to introduce the commercial status and a few academic undertakings that have not yet been commercialised.

2.1 Commercial status

Established computer tool kits for specific modelling and numerical techniques for single, and often well-defined chemical engineering problems exist. There are tools for separate design problems, such as numerical simulation, optimisation, flowsheeting and drawing. The most applied commercial tools are often based on some predefined building blocks of process models (or system models), providing enhancement for modelling work and helping to gain results faster. However, often the modelling and calculation of physico-chemical phenomena, such as some specific mass transfer mechanism, are performed with a proper tool and the result is either manually or by a dynamic link returned to a separate flowsheeting and simulation tool. There are also various commercially available project management applications that are applied extensively by industrial companies[4,5]. Project management tools help to schedule, to document the resource allocation and to update the activity network. Project management applications are proposed mainly for use as documentation tools, such as documenting R&D projects of production processes. Project management and documentation tools, say MS Project® and Lotus Notes®, are generic, and usually as such they do not necessarily link created engineering knowledge and data with project definitions, such as goal specifications.

---

[4] http://www.pmi.org/

[5] http://www.apmgroup.co.uk/pprof.htm

Next, characteristics of commercial process flowsheeting and simulation tools are briefly introduced, and a brief review is given of the project management tools used. This introduction is not intended to review all the simulation programs and project management tools, but to cover some general aspects relevant for the academic research work as well as for this thesis.

ASPEN PLUS®

ASPEN PLUS[6] (1995) is a widely used chemical engineering tool proposed for steady-state and dynamic simulations, process synthesis and design with predefined models for process units, thermodynamic data library and physical property models. An item of the Aspen Tech's tool kit, Zyqad, is a commercial system for integrating and managing the data, knowledge, and activities in the engineering work process. The open standard Data Model Aspen Zyqad also provides a commercial application of the emerging open international process design data model standard pdXi/STEP Application Protocol AP231 (see STEP ISO 10303 1994 in Chapter 2.2). The use of a standard data model facilitates integration between applications and between organisations, such as operating companies and process licensors. ASPEN can 'file-in' the complete problem definitions as input files with ASPEN input language, and user-specified process models can be loaded as well.

HYSYS®

HYSYS[7] provides a set of tools proposed for steady-state and dynamic simulation, flowsheeting and flowsheet analysis, analysis of operative troubleshooting, and real-time multivariate optimisation. HYSYS enables the integration of unit operations, reactions and property packages, and interaction with other applications to create hybrid programs. HYSYS is designed for integrating other applications and it allows the user's own models to be filed in as C++, Visual basic or FORTRAN files. HYSYS can be applied for ill-defined problems, which means that the whole model does not have to be ready before the user can see results. This also means that there is no input or output file, but all data and model topology are stored in one and the same case file.

---

[6] http://www.aspentec.com

[7] http://www.hyprotech.com/products/default.htm

SUPERPRO DESIGNER®

SUEPERPRO DESIGNER[8] is an example of a more domain-focused modelling tool kit compared to the above-mentioned. It is proposed for bioprocess design and the design of speciality, pharmaceutical and food chemicals, and in designing end-of-pipe treatment processes. This designer has many predefined specific unit operation models and procedures and a specific chemical component database. However, the capabilities to modify the predefined process models or to file-in some in-house models is very limited in the current version.

MATLAB® and MS EXCEL®

MATLAB[9] and EXCEL are probably the most applied calculation tools. In chemical engineering, these tools are utilised for data analysis, mass and energy balance calculations and for numerically solving, e.g. a detailed sub-model of a process system. There are many additional tool boxes (m-files for MATLAB, VisualBasic and .xls-files for EXCEL) available for helping to enhance, to compile and to solve scientific models, such as partial and ordinary differential equations (PDE and ODE). An introduction to scientific and engineering calculation with Excel is given by Orvis (1996). Both these applications provide good data-sharing properties. These tools are generic in nature – as they are supposed to be – and there are not much higher-level elements for systematising modelling work.

MS PROJECT®

MS PROJECT[10] is a widely used application and it is purely a project management tool proposed for task and resource scheduling, resource allocation and project monitoring. Project state can be monitored by using, for example, PERT (Program Evaluation and Review Technique) and GANTT charts[11]. This tool also provides data sharing with other applications with Dynamic Data Exchange (DDE) and Object Linking and Embedding (OLE) capabilities. However, linking tasks with engineering data or textual project definitions is a little complicated and it is not systematically supported.

---

[8] http://www.intelligen.com/in/prod/superpro/

[9] http://www.mathworks.com/products/matlab/

[10] http://www.microsoft.com/office/project/default.htm

[11] http://www.microsoft.com/office/project/default.htm

OTHERS
There are no practical commonly available methods or computer tools for integrating project management definitions and the data created and possessed by a project. Hence, companies have started to compile their own in-house tools for combined project and data management. Such tools can be structured, for example, as hypertext documents, www-databases or macro-supported MS Excel® and Word® documents. Companies have also applied commercial tools, which provide some framework for how to locate and manage the project data, such as with Lotus Notes®[12]. In any case, compiling a tool necessitates analysing the working methods and data structures, and after that it is possible to develop fully functioning tool support. However, there are organisations that aim at developing "standardised" systematisation of project management, such as "a guide to the Project Management BOdy of Knowledge – PMBOK® Guide[4] – in the USA, and PRojects IN Controlled Environments – PRINCE[5] – in the UK. Solid tools have been reported for pure project management purposes (PMBOK reports 200 applications), but no methods or tools have been reported for solid integration of project definitions and activity target data.

An established way to achieve systematisation and more standardised working practices is to apply and follow qualified standards. Current standards, such as ISO 9001 (1994), do not necessitate compiling any predefined systematic procedures for design work, but require drawing up a proper documentation procedure for the design practice 'as is' in an organisation. These standards also support backtracking the applied working procedure after the design work is finished. However, there is no commercial computer tool support for ISO 9001 known to the author.

2.2 Academic undertakings

Winograd (1986) gave a good analysis of how to utilise computers in complex engineering work. This analysis is partly still valid. A computer system called "co-ordinator" was reported as attractive to be applied and developed further for generic engineering purposes. Co-ordinator is supposed to share design data and to support team working. However, this description is at a very general level,

---

[12] http://www.lotus.com/home.nsf/welcome/lotusnotes

and in chemical engineering practice the objectives of co-ordinator should be specified further. These specifications should obviously include some kind of formalisation of the target system, generic model for chemical process and generic model for project activities. Compiling a formal conceptual language, or concept system, eases and enables building a systematisation that supports some specified working practices and data sharing within working teams. These kinds of conceptual models and modelling languages for chemical process models and modelling activities have been under development. Some of them are briefly introduced below, but this introduction is not intended to review all the projects on systematising chemical engineering but cover a few aspects relevant to this thesis.

MODEL.LA

MODELing LAnguage – MODEL.LA[13] – is proposed by Stephanopoulos (1990a and 1990b) and Christopher *et al.* (1995) for the conceptualisation of processing systems. MODEL.LA is an extension from an earlier work on DESIGN-KIT (Stephanopoulos *et al.* 1987). The expected advantages of using MODEL.LA are a) enhanced procedure for defining process models, b) enhanced documentation of contextual data and knowledge, such as assumptions and simplifications, and c) a procedure to build process models without dictating the modelling work too early with some algorithm solving the problem. These three enhancements are analogous with PDPD. MODEL.LA has been compiled as software systems. The language is developed as a system of objects and it is expected to capture and describe all the available technical data and knowledge of process systems. MODEL.LA provides a framework for declarative knowledge ('what is') but it does not model design process ('how to'). However, MODEL.LA has been taken beside the procedural model of Douglas (see below) and beside Hierarchical Decision Language – HDL (Han, 1994). MODEL.LA, as a declarative language, captures the data involved in processing systems and it is semantically very similar to the congruent part of PDPD methodology – generic process model. For the purpose of modelling processing systems, a set of predefined modelling elements has been compiled. The basis of the MODEL.LA language for describing process systems consists of six modelling elements. In addition, there are eight semantic relationships for capturing the links between the modelling elements. As an example, one detailed capability in MODEL.LA

---

[13] http://web.mit.edu/afs/athena.mit.edu/org/c/cheme/www/Research/Process_Systems.html

is that the generic modelling elements can semi-automatically update compatibility among abstractions of a process.

Six modelling elements:
Generic unit; such as plant, plant section, reactor vessel, tube
Port; material-, energy- and information flows between units
Stream; connecting ports
Modelling scope; contextual assumptions, hypothesis
Variable; such as mass-flow, enthalpy, Reynolds-number
Constraint; equations, order of magnitude

Eight relationships:
is-a; between subset and superset
is-member-of; between instance and class
is-composed-of; between modelling object and other modelling objects
is-attached-to; symmetric to *is-composed-of*
is-connected-by; is symmetric to *is-attached-to*
is-described-by; variable is described by, for example, as set of equations
is-disaggregated-in; to build for an object type a separate context
is-characterized-by; a class characterised by a property

KBDS

Knowledge-Based Design System (Bañares-Alcántara and Labadidi 1995) and GUI-KBDS (Graphical User Interface) are experimental prototype information systems for enhancing the design process by providing generic modelling elements and tools for representing process systems. The ideas of improving the comprehension of process models and modelling work are analogous to the ideas behind PDPD. KBDS consists of a system of object classes to be used by a KBDS user group. KBDS is not intended to be a methodology, but to serve as a platform where design objectives, alternatives and design history can be brought together and stored to support decision making. The predefined modelling elements in KBDS are:

Scheme; a design alternative
Unit; one chemical engineering unit operation

Item of Equipment; e.g. boiler
Variable; e.g. T, p, pH, …
Metascheme; e.g. the best group of schemes
Project; keeps track of evolution of schemes and metaschemes
Objective; a set of design objectives
Equivalence; equivalence (or compatibility) of units in different alternatives

ÉGUIDE

Ecosse Graphic Integrated Design Environment (Bañares-Alcántara *et al.* 1995) is a prototype design support system for conceptual design of chemical processes. It is the result of extending the KBDS to enable it to record design rationale. Like the developer group of PDPD, Bannares-Alcántara *et al.* have also noticed that very often the most crucial data and knowledge are behind the reported designs. The Équide tool, as well as the argumentation on models and alternatives, is embedded into the GUI-KBDS models.

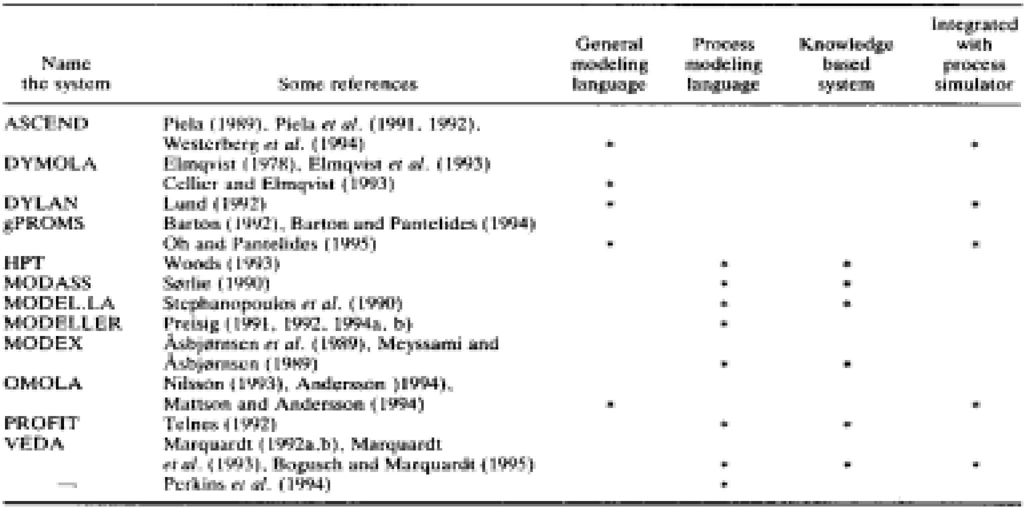
ÉPÉE

Ecosse Process Engineering Environment (Costello *et al.* 1996) is a distributed environment for the integration of software applications. It is based on the modelling elements of KBDS and the integration is based on the client-server approach. The data models in ÉPÉE support ISO PI-STEP and pdXI format (see ISO 10303). These formats are also acknowledged when implementing PDPD but PDPD models (formatted with PSSP language) cannot be automatically transformed into PI-STEP or EXRESS specifications (a general language behind STEP).

MODELLING METHODOLOGY OF MARQUARDT

Marquardt (1992, 1996) has proposed a conceptual model for process systems providing aids for building structural and behavioural models of processes. For example, ASPEN and HYSYS are model-based tools, but modifying the unit-based models is often restricted to a small group of experts. Thus, process models should be made more comprehensive, which enhances and supports engineering activities. Marquardt categorises some modelling tools as shown in Table 1. This table provides links to further studies, beyond the reviewed tools in this thesis. Marqurdt's modelling methodology provides a framework for building and documenting process models but it is not intended to guide how to design.

Table 1. Some generic modelling tools, taken from review of Marquardt (1996).



Name the system	Some references	General modeling language	Process modeling language	Knowledge based system	Integrated with process simulator
ASCEND	Piela (1989), Piela *et al.* (1991, 1992), Westerberg *et al.* (1994)	•			•
DYMOLA	Elmqvist (1978), Elmqvist *et al.* (1993) Cellier and Elmqvist (1993)	•			
DYLAN	Lund (1992)	•			•
gPROMS	Barton (1992), Barton and Pantelides (1994) Oh and Pantelides (1995)	•			•
HPT	Woods (1993)		•	•	
MODASS	Sørlie (1990)		•	•	
MODEL.LA	Stephanopoulos *et al.* (1990)		•	•	
MODELLER	Preisig (1991, 1992, 1994a, b)		•		
MODEX	Åsbjørnsen *et al.* (1989), Meyssami and Åsbjørnsen (1989)		•	•	
OMOLA	Nilsson (1993), Andersson)1994), Mattson and Andersson (1994)	•			•
PROFIT	Telnes (1992)		•	•	
VEDA	Marquardt (1992a,b), Marquardt *et al.* (1993), Bogusch and Marquardt (1995)		•	•	•
—	Perkins *et al.* (1994)		•		

The basic elements in Marqurdt´s modelling language for process structural description are:

Decomposition; decomposition of a plant, a process, a reactor, tube or an equation
Device; reactor or a phase
Connection; realises the decomposition

The behaviour of a processing system is described with the following modelling elements:

Generalised fluxes
Thermodynamical states and state functions
Phenomenological coefficients
Equation; balance equations, constitutive equations and constraints

An example of decomposition of an equation is given in Figure 2, the taxonomy of some elementary phase connections is given in Figure 3, and the taxonomy of equations is given in Figure 4.

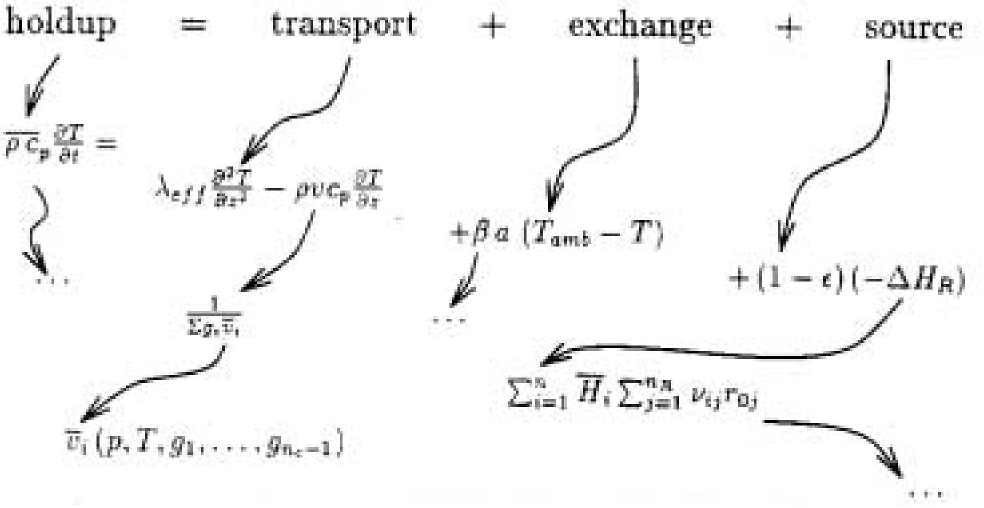


Figure 2. The energy balance of a reactor tube and an example of decomposition (Marquardt 1996).

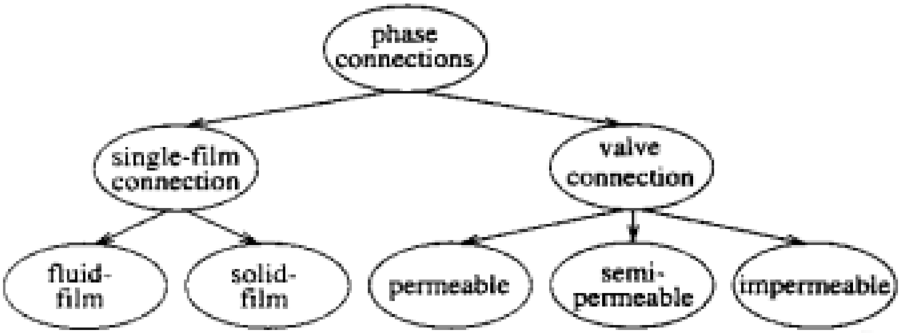


Figure 3. Taxonomy of some elementary phase connections (Marquardt 1996).

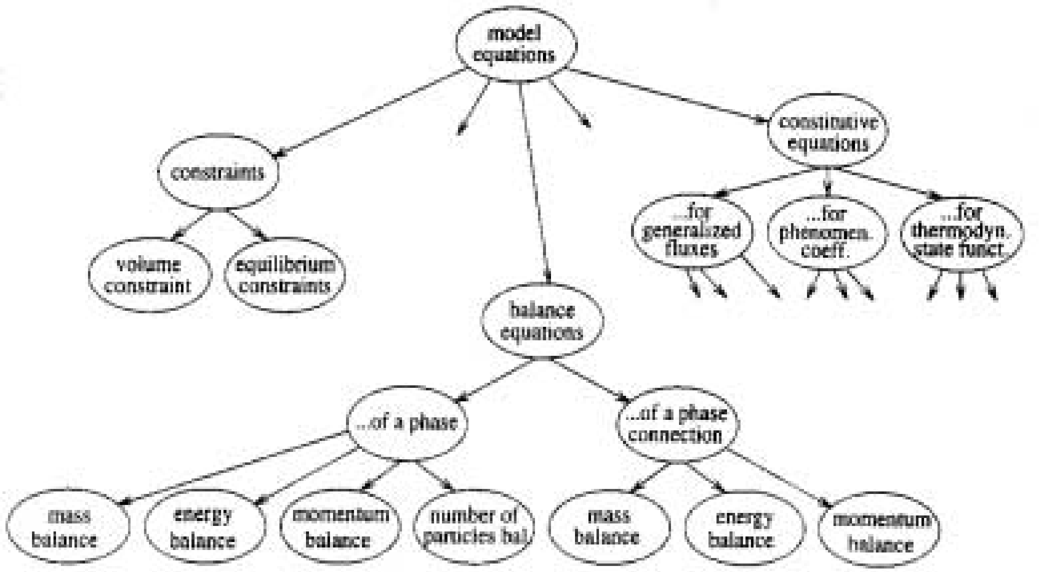


Figure 4. Taxonomy of model equations (Marquardt 1996).

PROTOTYPE SYSTEM FOR AUTOMATIC MATHEMATICAL MODEL GENERATION

Perkins *et al*. (1996) introduced a prototype tool for automatically generating mathematical models for process models. The tool is expected to facilitate the formulation of appropriate and consistent models. The tool keeps numerical solving separated from the system resulting from flowsheeting. In this way, some solver system does not overly dictate the designs. The prototyped tool has analogous features to MODEL.LA. The tool is built on the following building blocks of models:

Transfer-law; for linking variables, such as “phase equilibrium” or “bubble rise”
Vessel; such as “Flash vessel”
Phase; “liquid – gas”
Port; in | out | in and out
Connection; material and energy flows e.g. “Feed vessel –> Flush drum; pressure-driven flow”
Geometry; describing shape
Variable; T, p, etc.
Aggregation; uniform mixture of material

Some proposed solver systems for the prototype tool are SPEEDUP (Perkins and Sargent 1982) and gPROMS (Pantalides *et al.* 1992). The prototyped system has been programmed with Smaltalk (see Smaltalk from Goldberg and Robson, 1983), using WINDOWS facilities. During the prototyping the importance of clear documentation of the concepts used in modelling has been emphasised, such as the physical concepts and underlying assumptions, as well as the capability of various modelling situations. These notations have been considered relevant also for PDPD.

HIERARCHICAL DECISION PROCEDURE FOR PROCESS SYNTHESIS

Douglas (1985) created a hierarchical process design procedure in which the evaluation of a process model and the design steps are carried out in a certain sequence. The economic potential is calculated at each level. The procedure is as follows:

Continuous vs. batch type of process	(Level I)
Input-Output structure of the flowsheet	(Level II)
Recycle structure	(Level III)
General structure of the separation system	(Level IV)

In this methodology, the route of setting up processing systems is bound to the four levels. This procedure has been criticised as restricting designers' creativity, since combining reaction and separation, for example, is not supported by the procedure. Recently, some modifications have been reported (Douglas 1995), which also enable integrated solutions, such as integration of reaction and separation. MODEL.LA has also been used to formalise the process model data ('what is' knowledge) of this hierarchical process design procedure ('how to' knowledge).

GENERAL PURPOSE FLUID DYNAMICS SOFTWARE

This is an example of a more focused tool (Peskin and Hardin 1996), but it still has generic features. The tool is developed with an object-oriented approach and its programming is proposed for scientific and engineering analysis. The justification for this approach originates from an observation that a set of generic higher level modelling elements can be drawn up to enhance solving complex moving boundary problems in chemical engineering, such as in flow dynamics. The use of these elements is applied in the finite element method (FEM) to solve transport phenomena problems, continuum equations of fluid dynamics together with heat transfer, mass transfer and chemical reactions. The modelling elements are:

General_element; *e.g.* species in the system and set of equations describing the phenomena in the system
FEM_Mesh; finite element mesh
Moving_boundary_entity; any moving boundary in a vessel, *e.g.* one particle of a fluid mesh in a packed column
General_solver; generic equation solver
Time _integrator; integrates fluid movement in time (dynamics)

An example function:

Moving_the _entity_forward; an example of the type of call of an entity

The approach is expected to provide a flexible problem formulation. However, the tool may gain benefit from the provision of some procedural guidance on method usage. This approach can be seen as a solver for values for attributes of process objects in PDPD; a) process interior material (∇c_i and $\nabla \text{velocity}_{xyz}$) and b) separate items of process interior material, being disaggregated according to some imaginary boundaries (FEM_Mesh). Even if this approach with FEM is claimed to enhance obtaining numerical results, the problem formulation still remains complex. This is an example in which PDPD is expected to provide extra systematisation and comprehension by expanding the modelling views thoroughly from mathematical models to aspects in the real world. PDPD also provides a platform for documenting model and tool argumentation used by method developers and by users during method usage.

PROCESS INDUSTRIES MANUFACTURING ADVANTAGE

PRocess Industries Manufacturing Advantage – PRIMA[14] – provides enhancement to engineering work through information technology, and it’s development has been started in the ESPRIT 3 program. PRIMA (or Prima process) facilitates co-operation between industrial parties, such as between process equipment vendors and customers. The objectives of PRIMA, being summarised as data and knowledge transfer inside an organisation and among industrial partners, necessitates the formalisation of the subject as well as the

---

[14] http://albion.ncl.ac.uk/esp-syn/text/8234.html#top and PRIMA project report CD-ROM, April 1999 (via Paul Winstone EUTECH, 2001)

activities involved. For this purpose a concept system, called Prima Map, has been compiled, and it consists of following upper level elements, wchich are working together:

Business driver model; detaild features for analysing *e.g.* potential business advantage
Business process model; detailed features for analysing *e.g.* key information flows within business process
Application reference model; compendium of available resources , *e.g.* software tools
IT reference model; detailed software architecture which combines *e.g.* production process and vendor data as a computing framework
Information management strategy; detailed features for analysing strategic response of the used information flow used between *e.g.* production process and marketing
Enterprise model; upper level description of the enterprise having links to detailed production process and business process models.

In order to use Prima Map, STEP ISO 10303 (see below) and also EPISTELE (a language that combines ISO STEP and PRIMA for process industries) has been applied.

SOCIETY FOR PROMOTION OF LIFE-CYCLE DEVELOPMENT

The Society for Promotion of Life-Cycle Development (SPOLD) program[15,16] aims to produce an unified input-data format for use in Life-Cycle Assessment (LCA) calculations. In this way, the data and knowledge transfer at an international level would be easier. SPLOD contains a list of predefined modelling elements used in LCA calculations (*e.g.* material balances and effect of transportation) and it describes the procedures involved in LCA (*e.g.* flowsheeting and uncertainty evaluation). SPOLD is expected to follow the coming ISO 14048 environmental standard and has been planned to be transformed as compatible with ISO STEP 10303. SPOLD is not an LCA methodology but an ASCII format with predefined building blocks of LCA, both for data and activities.

---

[15] http://www.spold.org/publ/FILEFORM.zip
[16] http://deville.tep.chalmers.se/public/SPINE_TECH/SPINEandSPOLD.htm

STANDARD FOR THE EXCHANGE OF PRODUCT MODEL DATA

STandard for the Exchange of Product model data, STEP ISO 10303[17] (1994), has been set up by the British Department of Trade and Industry. STEP is proposed to capture the plant information from design data to operating data.

There is a formal language, EXPRESS (language reference ISO 10303 Part 11), for data and model formalisation, regardless of what programming language is used for computer implementation. STEP applies application protocols: functional data and their semantic representation ISO 10303 Part 221 and Application protocol Process engineering data ISO 10303 Part 231. The process design data and process specification of major equipment, which are built on established unit operations, are as follows:

Some general EXPRESS concepts:

Schema; a wrapper for collections of related information
Entity; a definition of an object – a real-world concept
Attribute; the property for an object
Type; a representation of value domains
Rule; for handling of full and partial constraints
Data types; represent domains of values

These general concepts are further detailed. Some chemical engineering application concepts are as follows[18]:

Chemical_reaction_data
Material_data
Stream_data
Process_simulation_data
Process_port
…

There are about 40 categories of model items, each with a few sub-items. To use EXPRESS and Part 231 obviously necessitates computer aids. This standard has partly the same model items as PDPD's declarative process model (see Section

---

[17] http://www.ukcic.org/step/step.htm
[18] http://www.nist.gov/sc4/step/parts/part231/current/part231.pdf

5.1) and MODEL.LA. However, STEP does not aim to model design activities, but to "specify conceptual process design information".

The Process Data Exchange Institute (pdXi)[19], initiated in 1989 by AIChE's Computing and Systems Technology Division, establishes written protocols (AP 231) and specifications for the exchange or process engineering data among computer programs, databases, and organisations within process engineering. pdXI is an equivalent initiative for STEP ISO 10303 in the USA, proposed for data sharing regardless of applications and organisations. There are some commercial tools based on pdXI[6].

An example of an application using STEP format in the CAD environment is given by Han et al. (1999). In this work, also an activity model has been included besides STEP. The following generic building blocks are used for the activity modelling:

Process design activity; e.g. requirement setting on production rate and quality
Engineering design activity; e.g. requirement setting on safety and maintenance
Required function; activities for satisfying the requirements
Request; a more specified requirement on a facility (a process to be designed)
A facility (a process to be designed) has the following generic modelling blocks in addition to the ISO 10303 format:

Process design features
  Operation variables; *e.g.*, p, T, flow rate
  Preliminary equipment specifications; e.g. capacity, number of inputs and outputs
  Logical connections; e.g. connections, inclusions

Engineering Design features
  Unit of equipment; *e.g.* shell, head, nozzle, agitator
  Detail equipment specification; *e.g.* diameter, length, material
  Spatial representation; *e.g.* orientation, assembly, position

---

[19] http://www.aiche.org/industry/pdxi/index.htm

This combined model of activities and processes can be imagined to be capable of documenting the same kinds of features as KBDS (see above), since it is not supposed to guide the design work explicitly.

Another example of a project where the STEP format is acknowledged is KACTUS[20] (modelling Knowledge About Complex Technical systems for multiple Use). The overall goal of KACTUS is to enhance the re-use of knowledge about technical systems during their life-cycle area by developing standards for the re-use of knowledge and by providing re-usable application-independent domain knowledge bases. One of the knowledge bases is for the production processes of offshore oil platforms. KACTUS has resulted in a computer tool kit that utilises EXPRESS language and CML (Conceptual Modelling Language) for transforming domain expressions into other formats, such as C++. The KACTUS tool kit facilitates the documentation of concepts and data used for describing the target system. This documentation has been extended via CML to the mathematical models of systems, which is analogous to PDPD and to the prototyped PDPD tool. CML has been described in BNF[21] (Baccus Naur format).

US DEFENCE SIMULATION, TEST AND EVALUATION PROCESS
Simulation, Test, and Evaluation Process (STEP) Guidelines (DoD 5000 series)[22,23] is a USA defense office document, proposed to guide and systematise any design and research activity. DoD STEP has been described in a semi-formal way, demonstrated with a few examples of how to use the methodology. It has features analogous to the 'design cycle' of PDPD. The path, or loop, for generating new knowledge is as follows:

Requirements –>
Simulation and Testing –>
Analysis –>
Evaluation –>
Knowledge –> … feedback *e.g.* edit *Requirements*

---

[20] http://www.swi.psy.uva.nl/projects/NewKACTUS/home.html.
[21] http:/swi.psy.uva.nl/projects/Kactus/toolkit/cml.html.
[22] http://www.acq.osd.mil/te/programs/tfr/final_step.pdf
[23] www.acq.osd.mil/te/programs/tfr/step.htm

DoD STEP is more generic than PDPD and it provides general guidance on numerical and computer-based simulation approaches for designing. However, representing the models produced via modelling work according to DoD STEP has not been supported with any higher level modelling elements.

PROCESS INFLUENCE DIAGRAM

Process Influence Diagram (PID) is proposed as a graphical tool for analysis and performance assessment of a hypothetical spent nuclear fuel repository. The PID method has been developed and applied in the SITE-94 project within the Swedish Nuclear Power Inspectorate, SKI, STATENS KÄRNKRAFTINSPEKTION (SKI, 1996). PIDs are composed of the following Performance Assessment (PA) elements:

General performance criteria for the repository; e.g. radiation doses per individual < 0.1 mSv/year
Technical criteria; e.g. requirements on the repository components in the licensing process
Scenario definitions; reference cases, central scenario
Expected output from a PA; knowledge required by a decision maker
Expert group identity; group identification
Model selection; proper structure models for site evaluation
Consequence Analysis; selection of input numbers of the models
Uncertainty analysis; conceptual uncertainty and parameter uncertainty

For systematic model building, the following modelling elements are used (indentation denotes hierarchy of the elements):

- *Boundary identification*
- *FEPs (Features, Events and Processes)*
  - *Process systems*;
    - *Engineered barriers*; fuel & canister, bentonite buffer, tunnel backfill, near-field rock, far-field rock
    - *Biosphere*
    - *Geosphere*
  - *Event*; e.g. near-field mass transport, far-field mass transport
  - *Features;* water chemistry of the buffer

Link (between FEPs); link descriptions and protocols, coupling to Assessment Model Flowcharts (AMF) of reference cases, coupling to AMF central scenario

This methodology has analogue features to PDPD by aiming to systematise modelling and analysis activities and by making process models more comprehensive. In respect of the ambiguous goals of the SITE-94 project, the modelling elements of PID may gain benefit, however, from further formalising the concepts of the methodology, as done in PDPD and MODEL.LA. However, FEPs are applied as a solid computer database with a master list of 150 generalised descriptions. The database administrator is the Nuclear Energy Agency (NEA) of the OECD, and the database has also been evaluated for the purposes of POSIVA in Finland (Vieno and Nordman 1997).

GRAPH, OBJECT, PROPERTY, RELATIONSHIP and ROLE MODEL

The Graph, Object, Property, Relationship and Role (GOPRR) model has been developed and extended from the OPRR data model (e.g. Smolander, 1992 and Tolvanen et al. 1993). GOPRR is proposed specifically for metamodelling or method engineering; and in the case of this thesis, it is used for building a prototype platform for how to model chemical process models and how to model R&D activities according to PDPD methodology and PSSP language. The GOPRR model is implemented in a MetaEdit+® metaCASE (Computer-Aided Software Engineering) tool (Kelly *et al.* 1996, MetaCASE 1996, 1999). MetaEdit+ is described in more detail in Section 6.2.2. Definitions on how MetaEdit+ is applied to capture PDPD are described in Sections 6.3.1 and 6.3.2.

There are few ongoing programs and projects systematising chemical engineering. To name few:

- ❑ EU projects on Information technology in the ESPRIT programme[24]
- ❑ Intercontinental IMS projects (Intellectual Manufacturing Systems)[25]
- ❑ Process Integration programme for applied and industrial R&D organised by the National Technology agency of Finland (Tekes)[26]

---

[24] http://www.cordis.lu/esprit/home.html

[25] http://www.ims.org/index2.htm

[26] http://www.tekes.fi/teknologia/tekno_tiedot.asp?id=56

3. Problems addressed in this thesis

In the tool development and testing of phenomenon-driven process design methodology it is very important to define the problem space carefully in order to make a successful study on such a large and holistic problem: building a tool support for systematising chemical process R&D work. "Systematising chemical R&D work" can be seen to involve almost everything that engineering is made of: model building, generation of design data and management of activities. In this thesis, the problem space has been divided as follows:

- ☐ improving the management of modelling concepts,
- ☐ improving the modelling work practices and management,
- ☐ improving the building of process models,
- ☐ improving the sharing and utilisation of design data and
- ☐ improving computer tools for implementing the systematisation.

Suggesting and demonstrating the solutions for these problem classes are approached by showing how characteristics of PDDP have been implemented as a computer tool (in Chapter 6) and how the use of this tool facilitates R&D work in a few pilot projects (in Chapter 7).

3.1 Need for metamodels

In order to enhance the integration of modelling work, to enhance the reuse of modelling data, and to set-up some generic modelling or documentation framework higher level modelling elements – called here metamodelling elements are necessary. This means that all the concepts that modelling or a project management tool are intended to deal with should be explicitly taken into the data structure of the tool.

The observation in respect of commercial tools is that if a tool is expected to enhance, say, data and knowledge sharing (‘what is’ knowledge), still argumentation behind the modelling decisions (‘how to’ knowledge) often remains hidden. These kinds of argumentation may answer questions like: “For what purpose the process is to be designed?”, and “For what purpose is the model to be created?”. The answers should be properly documented and some metamodelling elements are required to ensure the proper documentation.

Another observation is that the commercial tools, and often the prevailing practice, involve very complex mathematical models of processes. Still, the model structure remains hidden; in other words, metamodelling elements are not shown when building and documenting the models. However, every modelling tool has a purpose and a set of predefined building blocks that represent reality, and which are used for model building. Often, only half of the metalevel commitment behind a tool can be seen, which may in the worst scenario result in false usage of the tool. Why not make these metalevel aspects in process modelling tools visible?

3.2 Need for activity modelling

The knowledge and task complexity as well as the ever-increasing requirements on effectiveness presume systematisation in chemical engineering – as in other engineering domains. The systematisation of any engineering domain presumes an Information System (IS) providing principles and procedures for an efficient way to find design solutions. The IS should also provide unified frames for knowledge representation, processing, documentation and sharing. The approaches to improve effectiveness in chemical engineering can be categorised, for example, as follows:

- utilising progress in deep knowledge, such as application of new catalysts, application of new control systems and application of new analysis methods
- utilising progress in computer technology, such as new hardware and software reducing CPU time
- development of advanced knowledge representations, such as symbolisation and modelling languages for helping to build models and to gain results faster
- development of engineering methodologies to enhance and maximise the utilisation of domain and deep knowledge and the resources available.

Many of the prevailing studies to enhance chemical engineering are focused on utilising progress in specific disciplines or computer technology. Also a few alternative ways of building tools for knowledge representations have been introduced, see *e.g.* Bañares-Alcántara and Labadidi (1995). The methodological approach is less popular, even if Douglas (1985) presents an example procedure. One reason for avoiding methodological approaches may be the need to tackle a large variety of engineers' requirements for a multipurpose information system. The multipurpose characteristics may result in too many difficulties in the development and testing of the methodologies. Until now, also the development and prototyping of methodology-based computer tools have been very time-consuming (Tolvanen 1999).

Today, industries have started to seek ways of 1) how to document and utilise the data and knowledge of their processes, 2) how to manage creative R&D work more efficiently, and 3) how to provide methods for personnel about how to carry out R&D projects and process modelling. The implementation of these requirements necessitates some holistic system – now computer-based information system – supporting and integrating the engineering work and project management. Still, there are no commercial computer tools, nor established practices, for holistic activity and engineering data modelling, even if there are a few ongoing projects (see Section 2.2). Fundamental introduction on Information Systems (IS) development is given by Hirschheim *et al.* (1996) and Tolvanen (1999). IS development has established systematic practices and acknowledges holistic analysis and modelling of activity and activity target (software). Chemical engineering may benefit from transforming experiences of IS development to chemical engineering practices – this thesis is a modest step towards this end.

Usually, design activity is not properly modelled and documented, and for this reason R&D companies are still seeking methods and platforms for proper documentation and guidance of R&D work. However, during this research work both academic and industrial partners expressed their concern that this should not result in over formalisation, which may restrict individual creativity. That is why the IS supporting some specified activity and model documentation should have sufficient and still generic modelling elements for proper formalisation. Often, the systematisation of R&D work has been considered too complicated, and even unnecessary. However, documenting and linking activities with the specific process models properly is expected to result in better project management. Frezza et al. (1996) have noticed this potential improvement on prevailing practices, and have set up a decision support system for linking project goals with the activities involved and properties of the activity target, micro-electronic systems. This system produces suggestions on critical data to be created. Also Herder and Weijnen (1998) have set up a support system through panel discussions to link quantitatively the project evaluation criteria with process models. These studies would probably gain benefit from applying a formal language of activity and activity target.

"How to monitor what is the status of the project?" is the question that has been addressed by Frezza *et al.* (1996). This question is also relevant for PDPD methodology. In chemical engineering, a part of the answer is to define what the status and performance are of the chemical process model under development. In other words, this necessitates building a system for helping to define and evaluate the whole data and knowledge created during the project.

3.3 Need to enhance holistic process modelling

Engineering problems are often addressed with groups of people who have different backgrounds. In order to enhance and systematise modelling and development work, say the development of new processes or improvement of existing processes, unified process modelling elements are required. These elements should be applicable for various modelling views and modelling techniques. Figure 5 shows how to build models and modelling views, by naming the features and iteratively ending up with numerical values. Figure 5 is adopted from the work of Olin (1994), which is also valid for chemical process design purposes. The modelling levels in Figure 5 depict how reality is viewed and simplified into numeric data. Very often capturing all the simplifications, and assumptions made, during the modelling work would be beneficial. Such knowledge could be utilised when backtracking or evaluating the model, or when presenting the results to non-experts. These advantages are in prospect for any kind of modelling work. Careful documentation may improve the performance of large modelling systems and also more focused physico-chemical systems or equipment, even with environmental modelling aspects.

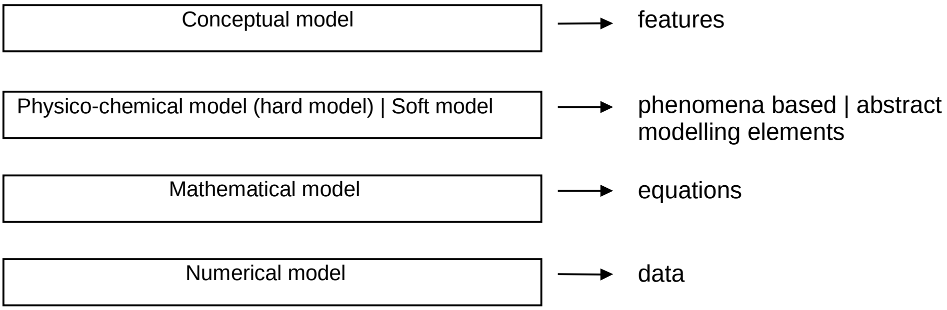


Figure 5. A few example modelling views.

Phenomenon-based, or an other type of conceptual chemical process modelling can be divided into routine modelling and calculation tasks, and tasks that necessitate building new model frames and some new calculation procedures. The routine design work normally utilises commonly used and available commercial tools. However, when solving a non-routine problem, some in-house methods and software might be required. Constructing new model frames, software or modelling methods may result in difficulties to communicate about the modelling elements within an organisation. Also sharing modelling and the result data of some new in-house method could be difficult and error prone. The framework behind PDPD is expected to (or is designed to) provide a proper environment with thorough elements for model building and model documentation. In Figure 2 it was shown how material and energy balances might expand into complex model hierarchies, which may result in some non-routine modelling. In Figure 2 the example models presented by Marquardt are analogous to what can be done with PDPD. Building up these equations and solving their numerical values can also be seen to fit into Figure 5. However, Figure 2 deals only with the data and models, but Figure 5 invites us to extend the models also with assumptions, and with predefined structural building blocks through "features". Still, Figures 2 and 5 do not capture the activities involved, in order to manage the performance of models, data, activities and the whole project. Furthermore, Figures 2 and 5 do not provide explicit frames for conceptual models, or guidance for naming the relevant features.

Many kinds of process models and modelling techniques can be used for a design problem, *e.g.* for optimisation or simulation. Phenomenon-driven approaches may be efficient in one case but a soft model, say a chemometric model[27,28] may solve the problem in another case. The modelling techniques and the tools available depend on the project resources. Perkins *et al.* (1996) give an example tool on separated physico-chemical process model generation and process flowsheeting. In the prototyped system of Perkins it is expected that the process model computation does not dictate process flowsheeting. In any case, conceptual frames facilitating holistic and generic views should be able to support various modelling practices. PDPD, and this thesis, do not give methods or guidelines of what technique to apply to solve the numerical values of process modelling objects, say compositions and fluxes. However, design and modelling work will probably benefit if the representation of the real world is done together with mathematical model, in other words, how the model structure is linked to flows, material, energy, phenomena and so on. This procedure is expected to be beneficial, no matter what type of modelling or numerical technique is applied.

For industrial organisations the alternative for commercial tools is to compile modelling platforms and numerical solvers by them selves, which is usually very time and resource consuming. To improve the data sharing and model comprehension, a higher level representation for mathematical models is often in order. In an ongoing project, POSIVA PSSP language is applied for this purpose on adsorption modelling in geo-chemical systems. The use of PSSP is expected to result in improved comprehension of how real phenomena and material are captured into a conceptual model and into a mathematical model, and what modelling activities are involved.

Research work on holistic modelling is not very popular and publications on the subject are very few. During this research work, the author has experienced that despite the unpopularity of holistic models, they are requested among researchers and project managers. This means that holistic views are expected to be taken into consideration more seriously both in academic and industrial organisations.

3.4 Need for data and activity transparency

Industry is still seeking tools and methods to integrate the design work and data (see e.g. PMBOK® in the USA and PRINCE in the UK). There are commercial

---

[27] http://www.wiley.co.uk/wileychi/chemometrics/

[28] see more Wold *et al.* (1987)

tools for integrating engineering data but no tools for integrating the documentation of design work with modelling data. Furthermore, commercial tools (known to the author) do not provide a conceptual framework of how to build process models and how to link the modelling elements to proper solver tools. This kind of upper level tool would be very beneficial for R&D projects, resulting in time savings, improved project management and probably a more efficient way to find solutions to problems. Just to bring process modelling data, such as process simulation data and economic data, together with some database application is not enough. These aspects were already noticed ten years ago by Wozny (1990). It is obviously a question about metamodelling elements – "What should be taken explicitly into models?". If proper metamodelling elements (Section 3.1) were available, then a solid data integration and enhancement for working procedures would be possible to compile by using the metamodel. PDPD provides metamodels for chemical process models and for R&D projects.

Creativity is understood to have an important role in process development. Creativity refers to an ability to find novel ways to utilise the degrees of freedom by applying various domain technologies. The resulting designs and the simultaneous consumption of resources are dependent both on the skills of the designers and on the management of the design work. Usually, project staff represent expertise in various areas and they may have large heterogeneity in their way of thinking. Hence, there is a common fear that any efforts to systematise process development tend to stifle creativity (Roussel *et al.*, 1991). That is why "How to systematise?" arises as the crucial question (Pohjola *et al.* 1997). The solution that PDPD provides is based on 'phenomenon-driven ordering', on data sharing and on support for expressing argumentation. Another feature in PDPD is that process synthesis does not dictate mathematical modelling. All this is based on sufficient elementary modelling elements, which is to be applied by all the staff in a project.

Project management seeks ways to backtrack goals, activities, design and modelling rationale and the argumentation behind them. On the other hand, the modellers may find it beneficial if they could follow how "separate" models are connected to each other, how the separate work sets, say experimental work, parameter fitting and equipment design, are connected to each other with specific data flows. Providing enhancement to these aspects is expected to result in better information exchange, improved documentation, reduced meeting time and in the best scenario, improved models and model performance. Stephanopoulos (1990a and b) and Christopher *et al.* (1995) have also noticed that knowledge transparency is important. In addition, the industrial and academic partners in this work have both stated the importance of the knowledge and data transparency. It is a question of how to support an "open formatted information system" and how data integration is harnessed to give extra enhancement to working practices, such as some extra motivation. Benson (1995) and Roussel *et al.* (1991) also notice this.

The SELME, TIESU, CICERO and LEPO projects were carried out at VTT Electronics and at the European Center of Nuclear Research (C.E.R.N) (1990–1996). These projects have addressed the question of data transparency between plant design and plant operation; design data and knowledge transfer to the operator is expected to improve the plant control and operation (Huuskonen 1997). This transparency is expected to result in fast and efficient trouble-shooting and correction actions of operation faults whilst simultaneously minimising production losses. The starting point of the projects was that the plant design was considered to be composed of various 'interconnected design areas', without any systematic ordering. The test projects followed some management practices that were lacking in their conceptual framework for integrating design activities and design data (not reported). Thus, the 'means end model' with tool support was developed and introduced. 'Means end models' contain elements to capture the design knowledge, such as 'purpose' and 'rationale' attributes for explaining design data. These projects have analogous objectives with PDPD by making the argumentation behind the designs and constructions visible, and in these projects especially for the plant operator. However, PDPD and PSSP language may provide the missing framework for making the integration of the design data easier. Perhaps then more efficient model-based explanation tools of plant knowledge could be developed.

3.5 Need for computer tool support for design methodologies

There are very few commercially available computer environments that are intended for methodology engineering and methodology tool development. Still, computer tool support is usually necessary for the efficient use and management of complex design methodologies (Hirschheim *et al.* 1996 and Tolvanen 1999). However, there are metalevel tools for software development facilitating the development work – Rational Rose®[29], Graphical designer®[30], and Visual studio®/Visual modeller®[31] to name a few. It is challenging to provide computer support to aid following an expert's methodology with a specified documentation format. Building tool support for methodologies has been reported (see Tolvanen 1999) as very time-consuming, which has been observed also during this work. It is also technically challenging to support multi-user requirements, to link data and solvers with conceptual models and to link models and modelling activities with the objectives involved. Furthermore, tool support development often necessitates simultaneous system development and the use of the system. Tolvanen (1999) has also noticed this.

Software engineering provides a well-established reference point on systematising working practices, which can also be utilised for chemical engineering. Tolvanen (1999) and Kelly *et al.* (1996) give a good introduction to tool support development for software engineering, which can be utilised for systematising chemical process R&D, and has been now utilised for building a computer tool support for PDPD.

---

[29] http://www.rational.com/products/rose/index.jtmpl

[30] http://www.advancedsw.com/

[31] http://www.microsoft.com/WorldWide/NZ/current/pressreleases/1998/Sept07-09-vstudio.stm

4. Purpose of the work

The purpose of this work is to explore how to make a useful computer tool support for Phenomenon-Driven Process Design methodology. The author seeks ways to defend the theses:

1. "Multi-characteristic and creative chemical process research and development work can be supported in a systematic manner by applying Phenomenon-driven process design methodology, implemented as a computer system."
2. "The use of this system is expected to result in comprehensive process models and modelling work with efficient documentation and data management. These advantages are relevant for any chemical engineering R&D organisation."

These theses have resulted from the observations given in Chapters 2 and 3. During the work, an important observation has been made; R&D organisations are still lacking holistic modelling views, and activities in R&D projects are not linked properly with the data they produce. The theses and the observations are partly the same as described in Section 2.2 in the works of Stephanopoulos *et al.* (1990a and 1990b), Christopher *et al.* (1995), Bañares-Alcántara and Labadini (1995) and Marquardt (1996). However, all these works have their own approach and different computer tool requirements. In this thesis, the chosen unique metamodel on systematising R&D – PDPD methodology and PSSP language – is expected (*a priori*) to be sufficient for the most observed problems listed in Chapter 3.

4.1 Working procedure

The questions to be answered in this thesis, which is the new knowledge, are indicated in Chapter 3. This knowledge is produced in the following way, as described in Figure 6.

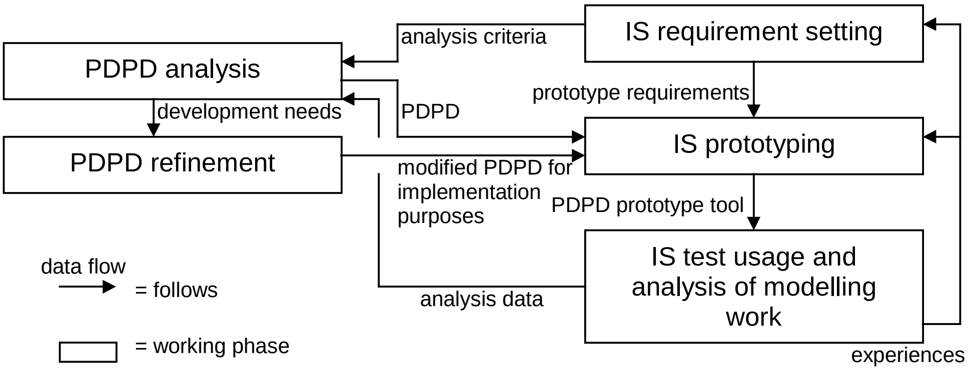


Figure 6. Working procedure.

Now, after finishing the procedure described above, the author demonstrates the development of a methodology-based process modelling and project management computer tool for systematising R&D work in chemical engineering. More detailed descriptions of PDPD analysis and Information System (IS) prototyping are given in Chapters 5 and 6. Some computer environments for the tool development have been tested during the work. In addition, a few test cases and pilot projects have been carried out during the work, while a few are still ongoing.

4.2 Limitations of the study

This study has several limitations. First of all, it is cumbersome to "measure" the performance of the PDPD tool in respect of the set tool requirements. Even though the prototyped tool has been tested in a few pilot projects, still the preliminary evaluation reported in this thesis is mainly based on the tool developer's (the author's) observations. More user tests with larger user groups and with various types of R&D projects should be arranged in the future. However, basing the demonstrations purely on user tests is truly a long process. Thus, the comparison of the expected (*a priori*) advantages from the use of PDPD with the prevailing practices has been emphasised in this thesis.

Another limitation is lack of categorisation of modelling problems in chemical engineering. This categorisation may result in a more careful analysis of how to use and benefit from the PDPD tool in R&D projects. Utilising the categorisation, on the other hand, possibly requires evidence about the categories through test cases, which would be in that sense truly a large project. In this study, the evaluation of the usefulness of the PDPD tool is not based on numerous case studies, but on the argumentation of expectations, and on emphasising the generic applicability originating from the concept system of PDPD.

4.3 Expected value

The generic value of this research work is to provide a procedural model of how to build tool-supported PDPD methodology for organisations working with the R&D of chemical processes. In addition, the prototype computer tool development itself possesses a newness value – the application requirements are specified later in Section 6.1. The pilot projects and test cases are expected to identify the advantages of the methodology usage, as well as the industrial relevance. The pilot projects are also used as feedback for developing the methodology further.

As a side-product, this work provides a reference procedure for how to build and evaluate a computer tool for any design methodology. In other words, this thesis provides reference material about the issues of how to systematise, how to build computer tool support for a methodology, and even how to organise test usage. In this sense, the work can be seen as one kind of procedural model for methodology development work.

5. Phenomenon-Driven Process Design methodology

This thesis is based on Phenomenon-Driven Process Design (PDPD) methodology. PDPD is still under development and test usage. It has been under development for over 10 years at the University of Oulu by Professor Pohjola and his group (Pohjola *et al.* 1993a, 1993b, Tanskanen *et al.* 1995, Pohjola and Tanskanen 1998). Various projects have been carried out during this period with academic and industrial partners. Before a brief introduction of the methodology, a list of expected advantages of the use of the methodology is given as an orientation for the reader. The list can be seen as set of methodology requirements and as an initial answer (basis for the thesis) for the addressed problems (see Chapter 3) in process R&D systematisation.

- ☐ The PDPD-formatted documentation records properties and characteristics of a project, which may easily be omitted and lost without proper formalisation.
- ☐ The use of PDPD provides a platform for sharing information on engineering data and project definitions among researchers, engineers and management.
- ☐ The formalism facilitates integrating engineering and natural sciences under the same information system, supporting concurrent engineering if required.
- ☐ The use of PDPD methodology ensures consideration of phenomenon-driven ordering and argumentation in any process development project.
- ☐ PDPD provides a platform that specifies and accelerates the goal setting, which is expected to result in increased efficiency of a project.
- ☐ The PDPD platform is expected to enhance the comprehension of design or modelling problems; content and linkages among goals, working procedure and the process models involved
- ☐ PDPD is expected to help following and evaluating the knowledge created in a project.

Making a serious attempt to realise these kinds of methodological advantages as applicable as possible requires the development and use of a formal concept system, which has also been noticed by Marquardt (1996) and Stephanopoulos *et* *al.* (1990a and b), as well as by Tolvanen (1999). The concept system behind PDPD is built and described object-orientedly (Pohjola and Tanskanen 1998). The formalisation has resulted in a language called PSSP, which is to be introduced. Sections 5.1 and 5.2 describe the methodology: the concept system for describing 'what is' knowledge and the methodological guidance for describing 'how to' knowledge. From the beginning of the methodology development it was assumed that computer tool support is necessary. Chapter 6 describes the computer tool development.

5.1 The concept system; metamodel and PSSP language

The development of a practical methodology necessitates a solid information system. The development of an information system, on the other hand, requires conceptualisation and analysis of real world. The capability of PDPD to make a systematic representation of any process development project stems from the expressive power of the concept system behind the methodology. In this concept system – called the PSSP language – any real thing is represented as an object with four attributes: Purpose, Structure, State, and Performance (PSSP). These attributes and the elementary modelling elements selected are expected to be sufficient for the modelling of any chemical R&D project. This representation produces a transparent object hierarchy with a highly unified format. In this chapter the PSSP language is briefly described and some argumentation behind the concept selection is given.

In order to ease the understanding of PDPD development work a concept metamodel is introduced. A metamodel is one higher level of abstraction than an ordinary model (Tolvanen 1999 and Kelly *et al.* 1996). In this work it is meta-level commitment, for example, to use an object-oriented paradigm (structure of PSSP language) and how the concept selection is argued (purpose of the PSSP language and its elements). The end user does not necessarily have to know anything about meta-level decisions of the methodology. However, knowing the meta-level commitments may facilitate understanding the methodology. For a methodology user, PSSP provides a metamodel of a project and for a chemical process. This means that, according to PSSP, both an R&D project and a chemical process are viewed (or modelled) in a specified way, with pre-specified modelling elements. Even though the word 'metamodel' is avoided, still 'metamodel of a process model' is taken explicitly into the PSSP language because of its special importance. 'Metamodel of a process model' captures, e.g. the purpose of the process model, which is not usually properly documented. Figure 7 clarifies the concept of metamodel and 'metamodel of process model'. Note that now PSSP attributes are applied also for capturing the relevant features (meta-level commitments) of the methodology development phase.

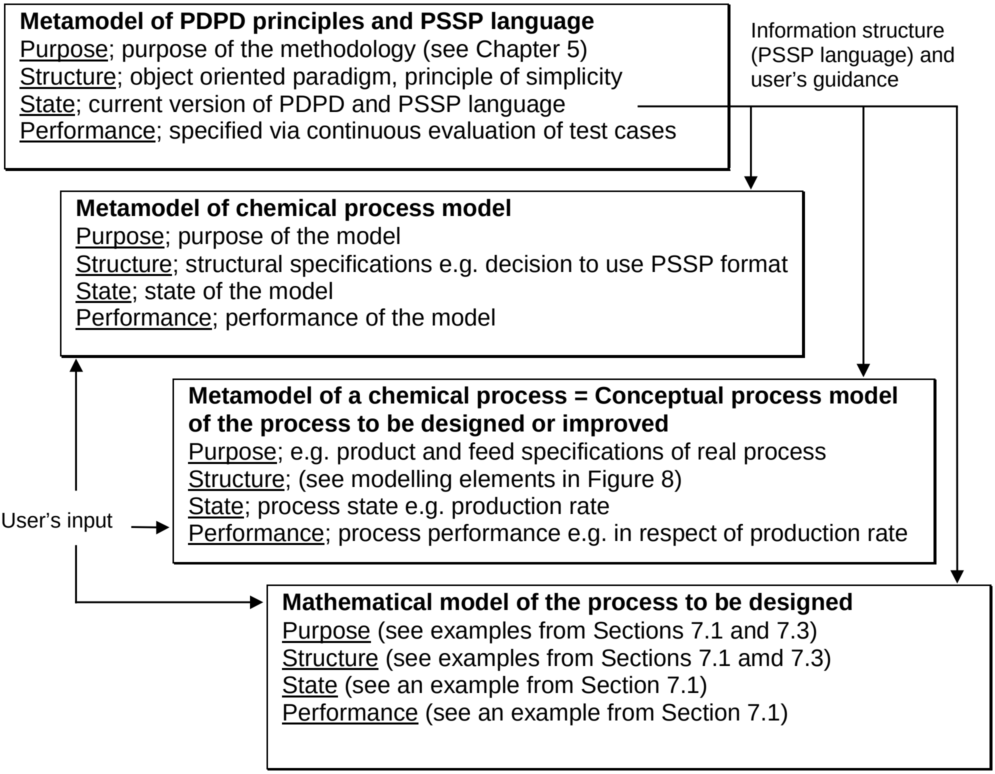


Figure 7. Metamodels involved in PDPD and PSSP language development and usage.

5.1.1 Generic Real Thing

Generic Real Thing (GRT) is used as storage for the generic properties identified from the objects in PSSP (see Figure 8). In other words, the role of GRT is to simplify the information structure, but the term GRT is invisible for methodology users. Next, the structural elements of GRT are described in short.

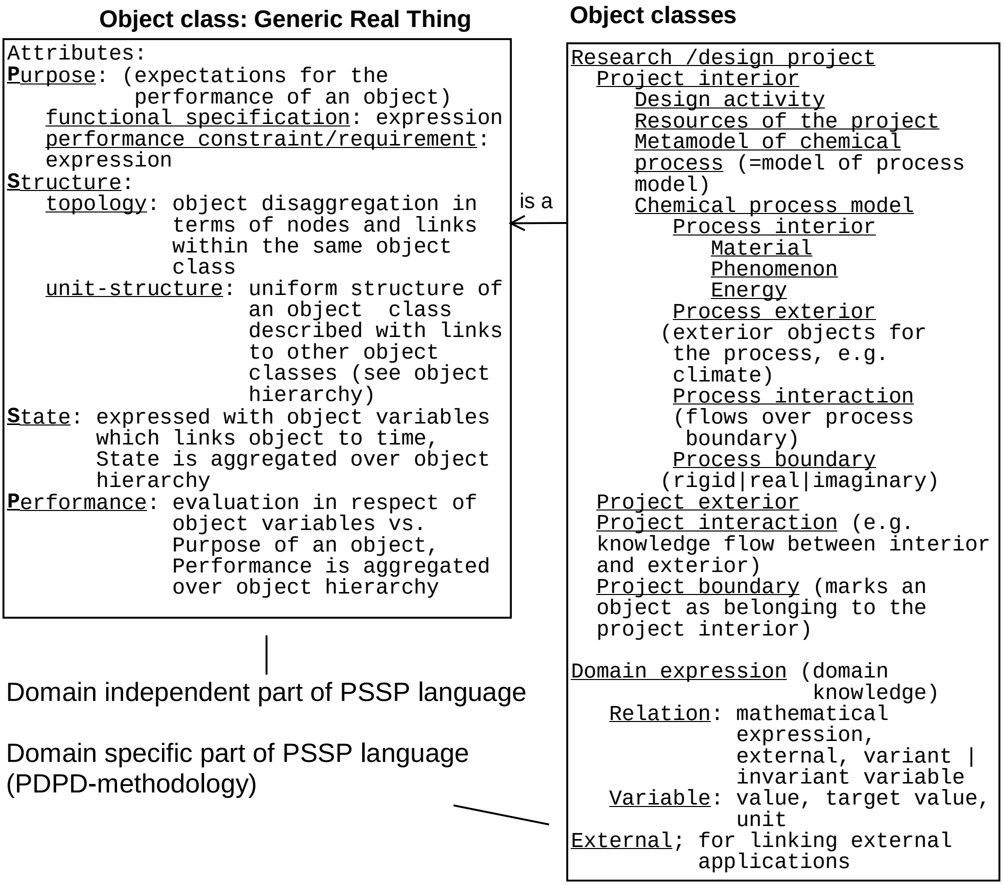
'Purpose'

The first element of GRT is 'purpose'. Through the 'purpose' attribute the methodology user is supposed to give a formalised answer to the questions of why an object is taken into the context and what constraints are assigned for it. The purpose for an object is given as a set of purpose expressions, being verbal statements and mathematical assignments for target values for object variables. For example, the 'purpose' of a 'project' (any engineering project) is to produce knowledge about the project target, say, how to produce product material and how to control the process behaviour. In turn, the 'purpose' of the target – being a 'chemical process' – is to, say, produce product material. The 'purpose' of the process interior 'phenomenon' is, for example, to convert raw material into the product. The content of the 'purpose' attribute is nothing new for engineering projects, but very often it has not been properly documented and comprehensively formalised, thus 'purpose' has been selected in PSSP (see Object classes in Figure 8).

In PSSP language, 'purpose' has relevance in the context of a chemical process R&D project. In other contexts, for example, a research work of some natural system, the modelling targets, such as species and mass transfer phenomena in the bedrock, do not necessarily have a 'purpose' in the same sense as in industrial processing systems. On the other hand, in biology the knowledge to be created may answer the question of what the function and purpose are of material, species and phenomena in living (biological) systems. A biochemist might try to find ways of controlling biological material and phenomena as a bioprocess. Thus 'purpose' might be applied also for studying natural physico-chemical systems “as is”. The applicability of 'purpose' is obviously very generic, but in this work the focus is on chemical process R&D projects.

'Structure'

The structure of any object (see Figure 8) has two dimensions, one for disaggregating an object topologically relative to similar objects, such as disaggregating a project into sub-projects or a chemical process into sub-processes. The other dimension is the unit-structure for linking an object with objects from other classes; see indentation in the box at right side in Figure 8.



```
Attributes:
Purpose: (expectations for the
          performance of an object)
   functional specification: expression
   performance constraint/requirement:
   expression
Structure:
   topology: object disaggregation in
             terms of nodes and links
             within the same object
             class
   unit-structure: uniform structure of
                   an object  class
                   described with links
                   to other object
                   classes (see object
                   hierarchy)
State: expressed with object variables
       which links object to time,
       State is aggregated over object
       hierarchy
Performance: evaluation in respect of
             object variables vs.
             Purpose of an object,
             Performance is aggregated
             over object hierarchy
```

```
Research /design project
  Project interior
     Design activity
     Resources of the project
     Metamodel of chemical
     process (=model of process
     model)
     Chemical process model
        Process interior
           Material
           Phenomenon
           Energy
        Process exterior
       (exterior objects for
        the process, e.g.
        climate)
        Process interaction
        (flows over process
         boundary)
        Process boundary
       (rigid|real|imaginary)
  Project exterior
  Project interaction (e.g.
  knowledge flow between interior
  and exterior)
  Project boundary (marks an
  object as belonging to the
  project interior)

Domain expression (domain
                   knowledge)
   Relation: mathematical
             expression,
             external, variant |
             invariant variable
   Variable: value, target value,
             unit
External; for linking external
          applications
```

Figure 8. Concepts of Phenomenon-Driven Process Design methodology (indentation denotes the ‘consists of’ hierarchy).

In the same way as 'purpose', also the 'structure' attribute is very generic, being easy to apply for various types of scientific research work. Pilot projects in Sections 7.1 and 7.2 provide examples of how to use the structure attribute for modelling purposes and for project activity documentation.

'State'

The third attribute, 'state', captures the current values of the object properties. State properties come automatically initiated and named when setting the object purpose attribute, but additional state properties might be required later when solving a value for some state variable or performance variable. In PSSP format, solving the value of a 'state' variable often necessitates solving a set of 'states' of other objects. The state variables of material are defined in an established way in thermodynamics and GRT provides a location, for example, for pure state properties p, T, V and $c(i)$, but also for other object properties, such as some derived state properties, enthalpy, viscosity and so on. In addition, for a project there are common and in-house state variables (or parameters) applied by organisations and GRT provides a location for all of them. Numerical or Boolean values of 'state' variables (of any object) cannot be given without first specifying fully, or roughly, the 'structure' attribute. The numerical value for a 'state' variable in turn usually originates from a 'mathematical relation'.

'Performance'

The performance properties for an object become automatically specified when giving the 'purpose' for the object. The 'performance' attribute can be expressed as 'in respect of' and 'overall'. This means that the 'performance' of an object can be specified separately in respect of an individual object property, or as an aggregated overall performance. The value of a 'performance' variable is aggregated from the 'state' and 'performance' attributes of lower level objects. 'Relation' is used for capturing the relationship behind the aggregation. Often, setting a value for a 'performance' variable necessitates solving the numerical or Boolean value of some 'state' variables. If a 'purpose' can be given for an object then, and only then, 'performance' can be specified. It should be noted that 'performance' is not the same as the value of a 'state' variable; for example, the performance of a process model is not the same as the rate of product material. If "rate of production" is one of the 'performance' variables then the value 'in respect of' "production rate" may depend – through aggregation relation – for example on the achieved rate of production value, credibility of the value, distance from target value, or even the resources used of the R&D project. The aggregation relation can be just an assignment with verbal argumentation or a more specified mathematical equation.

The applicability of the 'performance' attribute is analogous to the 'purpose' attribute. The usage of all four attributes can be best understood via case projects (Sections 7.1, 7.2 and 7.3).

5.1.2 Project model

A process development project can be viewed as an analogue to a batch process. The product of a project, being knowledge formalised as a conceptual model of the process under development, evolves as the project advances. The development activity refines the knowledge and is driven by the gap between the present state of the knowledge and the present sub-goal, and by the motivation of the project team to reduce the gap. The goals are set dynamically by the management activity. In PDPD, the triplet of development activity, management activity and chemical process is represented object-orientedly as a set of mutually communicating objects. This is called the design cycle (Pohjola and Tanskanen 1998) and it is shown in Figure 9.

PDPD methodology provides procedural guidance on how to start and proceed with design projects, and on how to develop and model the chemical processes in a systematised manner. The usage of this methodology for supporting process development is expected to improve the quality of design projects by offering a framework for enhanced communication, for creative solutions, for enhanced comprehension, and for the documentation of all the relevant knowledge.

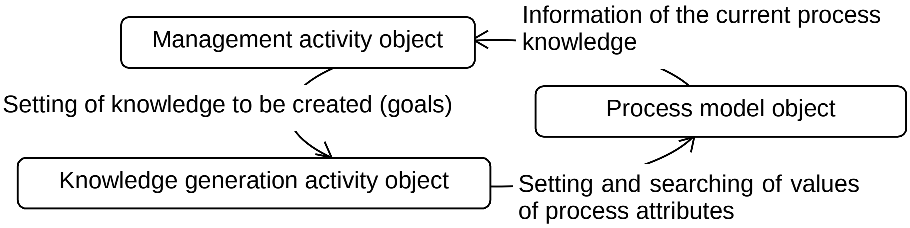


Figure 9. Design cycle.

Thus, 'project' describes a unique set of design activities in conceptual design. 'Project' in PDPD is meant to be used for project management decisions, i.e. setting the purpose (goal) and the initial knowledge, as well as making decisions on project resources, design actions, project disaggregation and project termination. The methodological definition of a design project is:

Methodological definition 1 *A design project is the control of resources and design activities for a purpose,* where the purpose falls into goal and functional specifications and performance requirements, expressing the knowledge to be created in the project. An example of the design cycle and usage of the project model is given in Section 7.2.

5.1.3 Process model

One structural element of a 'project' is a 'chemical process', where engineering knowledge, data and results of design decisions on the chemical process to be designed are located. As described in Section 3.1, the project also has to specify the metamodel of the chemical process (=prescriptive model of a process model), which initiates the model to be created as predefined purpose, structure and state attributes. Then, and only then, the performance of the process model can be evaluated in a systematic manner during the modelling work. A 'chemical process', in turn, consists of the following structural parts: 'process interior', 'process exterior', 'process interaction' and 'process boundary'. 'Process interior' is composed of 'material', physico-chemical 'phenomenon' and 'energy'. 'Process exterior' captures instances outside the process. 'Process interaction' is composed of material, energy and information flows. 'Process boundary' can be specified according to 'real boundary', *e.g.* phase interface, or according to 'rigid boundary', *e.g.* the walls of a chemical reactor. Process boundary can also be specified according to some 'imaginary boundaries', *e.g.* disaggregating homogenous material inside a process interior into discrete material elements to ease numerical calculation. All these structural elements of the 'chemical process' are sub-classes (objects) of GRT. The methodological definition of a chemical process is as follows:

Methodological definition 2 *A chemical process is the control of physico-chemical phenomena for a purpose,*

where purpose falls into the functional specifications and performance requirements of the process to be designed. As mentioned earlier (Sections 2.1 and 2.2), chemical process designs are often generated from existing designs and with ‘unit operation’-based building blocks, which may enhance gaining numerical results, but possibly stifle finding new designs. 'Process boundary', 'process interaction', process interior 'material' and physico-chemical 'phenomenon' are selected as primitive building blocks of a chemical process. With these concepts and with the methodological definition 2, the creativity of engineers is stimulated. This is called phenomenon-driven modelling and design.

Gavrila and Iedema (1996) give a brief evaluation and review of a few other phenomena-driven approaches.

As mentioned in Section 5.1, there can be many levels of models and many levels of abstractions. Disaggregation procedures (see Section 5.2) offer a rich variety of ways to build and to view structural descriptions of processes. PSSP language has been applied even to capture molecular level features of material, and PSSP language has been used as an integrator for calculating the properties of continuous material from molecular and atomic level data (Pohjola *et al.* 1999). In order to avoid confusion in using the disaggregation procedure, a more detailed description is given. Figure 10 demonstrates how a chemical process may be disaggregated by using real, rigid and imaginary boundaries. Disaggregation could be used for developing a new process structure by creating rigid boundaries in order to gain better performance of the process to be designed (see design version 2 in Figure 10). Disaggregation may also be performed in order to gain modelling advantages, without changing the structure of the artefact if constructing the design. Modelling advantages may be achieved by disaggregating a process through real boundaries, i.e. through a phase interface, without separating them from rigid boundaries (see modelling view 1 in Figure 10). Modelling advantages may also be achieved by using imaginary boundaries. Using imaginary boundaries, where the process interior material is disaggregated in order to gain modelling advantages, creates the modelling view 2 in Figure 10. There can be many levels of abstractions for various design and modelling purposes. Some hypothetical situations and modelling views are sketched in Figure 10. Figure 10 shows how the 'disaggregation' operator can be utilised for various purposes resulting in various structural descriptions of a process.

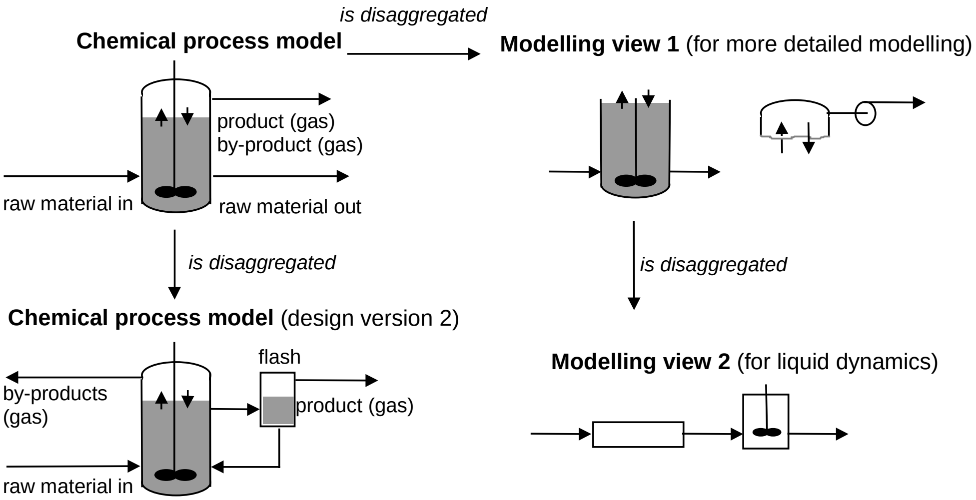


Figure 10. Hypothetical disaggregations of a chemical process.

The behaviour of a process, or processing systems, means how an object property (under 'state' and 'performance' attributes in Figure 8) is dependent on other properties of process objects and on time and space. The behaviour of chemical processes is usually modelled mathematically. The models describing the behaviour of a process are located under the state attribute of 'chemical process'. The state models relate the values of state variables to time and to other variables; StateVar = f(t,x). The concept of 'mathematical expression' is meant to capture the purpose, structure, state and performance of mathematical models, such as mass and energy balances. The structure of a 'mathematical expression' consists of 'topology' and 'unit-structure' – as in the case of any other object. The unit-structural elements of a 'mathematical expression' are 'variable' (acting as an operand in a mathematical relation), 'variable-type in an external: variant | invariant', and 'externals' (external computer environments). The PDPD knowledge base is not supposed to do the mathematical solving by itself, but it enables linking PDPD-formatted knowledge to 'external' applications, where the actual numerical manipulations are performed. In this way, conceptual process design is not dictated by mathematical modelling. A few examples are shown in Sections 7.1, 7.2 and 7.3.

5.2 Procedural model – principles and procedures

The following describes how the concepts of PDPD should be used. This means how 'purpose' is set, how objects are instantiated, how disaggregation and aggregation are performed, and how 'performance' is evaluated resulting in design cycle. The description of the procedural model is divided into performance-driven strategy to manage any object (generic), and phenomenon-driven ordering to model chemical processes (domain-specific).

5.2.1 Performance-driven strategy

PDPD methodology guides project management to monitor particularly the value of process performance in order to update the prospects of the commercial viability of the target process. This continuous information retrieval necessitates the documentation of all the modelling and design decision rationale. PDPD provides a platform for this purpose.

The process performance and knowledge credibility are used as input for management, as described in Figure 9. The resulting further goals and the resource allocation are driven by the impression of the performance of a process. According to this strategy, setting sub-goals should aim to reveal the features with a dominant effect on the current process performance, resulting in proper knowledge credibility at the same time. The goals are set dynamically and documented into metamodels of a process, accessible to the entire project staff.

The guidance for carrying out engineering tasks according to the performance-driven strategy is depicted in Figure 11. The description in Figure 11 applies the Object-Oriented Systems Analysis (OSA) notation (Embley 1992). A general principle for object usage in PDPD is given as Principle 1:

Principle 1 *Use performance-driven strategy in project, design and modelling decisions (see Figure 9 and 11)*

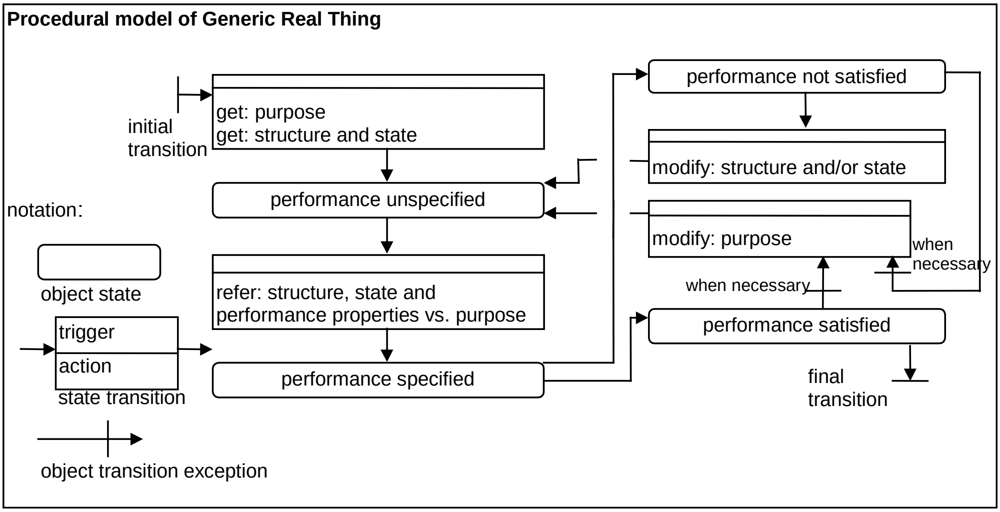


Figure 11. Purpose- and performance-driven strategy of editing objects (in OSA format).

The idea of Principle 1 is to express the 'purpose' for every object, and after that specify the 'structure' and 'state' of the object. It is then possible to evaluate the 'performance' (or goodness) of the object and to make, say, design decisions on the 'structure', 'state' or 'purpose' attribute of a process object. The 'purpose', such as functional specifications and performance requirements, is given as a set of 'purpose expressions, according to which the 'performance' is evaluated continuously throughout the lifetime of the object. This is analogous to what has been described as the design cycle in Figure 9.

The performance-driven strategy to edit objects, for example, to design and model chemical processes, is compatible with phenomenon-driven ordering, which is described below. Pohjola (1998) gives a fundamental introduction of the principles. The usage of these principles should result in systematic, controlled and formalised project actions on knowledge of how to control the phenomena and material in such a way that the process requirements are fulfilled.

5.2.2 Phenomenon-driven ordering

Project goals are set to make the process developers reveal the structural and behavioural features of the process in order of relevance with respect to process performance. The general principle for finding out the order of relevance emerges from the definition: "Chemical process is Control of Phenomenon for a Purpose". It formulates the challenge of any development project as: "How to make the named phenomena advance as desired?" This necessitates characterising:

(2.1) necessary material, phenomena and energy in the process

(2.2) the desired and proper thermodynamic state for a process, approached by studying of the inherent attainability and maintainability of the state distributions

(2.3) structural disaggregation | aggregation of a process in such a way that it possibly results in the required process state with the specified process performance criteria

(2.3.1) structural disaggregation | aggregation are performed in respect of process boundary and interaction specifications and further material and phenomena characterisation (e.g. by initiating some auxiliary material and phenomena)

Any structural detailing should aim at higher process performance via more specific characterisation of process behaviour. These sub-principles are aggregated as Principle 2:

Principle 2 *Use phenomenon-driven ordering on structural detailing and on process disaggregation or aggregation*

Structural detailing should also aim at higher credibility of the process performance assessment. Furthermore, as already mentioned (Section 3.3), the structural detailing of process models often makes the models more complex; thus, detailing procedure should simultaneously facilitate comprehension of the linkages between real-world, conceptual process models and mathematical models.

Principle 2 is expected to promote the discovery of new designs and lead to a more fundamental modelling and evaluation of existing designs. Furthermore, following Principle 2 does not dictate the solver tools too early in the design phase. These advantages are based on the idea of grouping phenomena and material as a controllable unit. For example, if the process material and the set of desired phenomena within one process boundary do not result in the required process performance, then the process may be disaggregated. The resulting sub-processes can be designed to have different thermodynamical states and differently grouped sets of phenomena and material, which may improve the process performance. These kinds of ideas for promoting process intensification are supported, e.g. under Process Intensification conferences[32], where the intensification often applies a phenomenon-driven way of thinking. However, usually a phenomenon-driven way of thinking is not systematically supported by any method, or by any steps like 2.1 – 2.3.1. Enhancement of model comprehension is based on the idea that the argumentation behind complex models and how the models represent reality should be explicitly documented. The documentation is supported with sufficient and generic modelling elements (Figure 8), with an attribute list: 'purpose', 'structure', 'state' and 'performance', categorising variables in a systematic manner.

Let us take a look at a simple design example. If a designer's task is to design proper equipment to produce bioethanol via fermentation, then a usual way is to construct a batch fermentation plant, and a separate distillation plant[33]. However, when applying phenomenon-driven ordering, firstly the designer names the required phenomena: ethanol formation (*e.g.* via *saccharomyces cerevisiae* microbe) and ethanol recovery. After that, the designer should decide whether to "buy" an existing design solution and equipment, or try to find some other solutions. According to phenomenon-driven ordering, a designer may keep ethanol formation and ethanol recovery phenomena occurring in the same vessel (e.g. via simultaneous fermentation and evaporation[34]), in the same interior material, *i.e.* not directly disaggregating phenomena that occur in separate vessels. In this way, the designer may end up with a solution, whereby fermentation and ethanol recovery occur in the same vessel, as shown in Figure

---

[32] A Process Intensification conference; see an example given by Turunen (1997).

[33] 75 % of the world's bioethanol processes are batch processes (http://www.nal.usda.gov/ttic/biofuels/cont.htm).

[34] A few other alternative techniques for bioethanol production are given, *e.g.* by Maiorella (1984). One of them, integrated fermentation and evaporation, is argued and derived by applying phenomenon-driven ordering. In the report of Pohjola (1997) there are also a few other examples in which phenomenon-driven ordering has been applied for various process design exercises.

12 (bottom). In this way, ethanol inhibition is diminished[35]. In some cases, this type of solution may result in lower investment and operation costs than the traditional solution. Even if a designer decides to apply the traditional solution, the phenomenon-driven ordering can be applied for documentation of the designed process, which may enhance the understanding of the design solution, as well as the research activities involved[36].

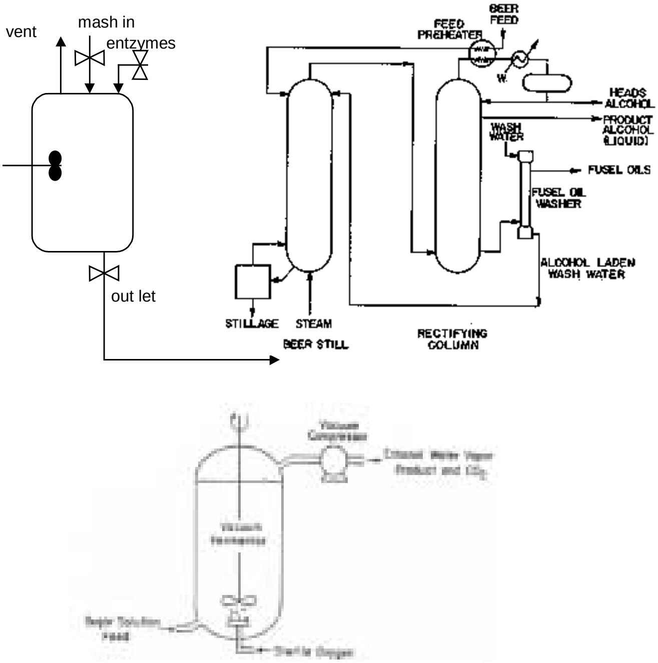


Figure 12. Traditional batch fermentation process (simplified from Kelsall 1995) and a distillation unit (Maiorella et al. 1984), and (bottom) an alternative integrated continuous fermentation and ethanol recovery process via evaporation (Maiorella et al. 1984).

---

[35] Ethanol in the fermenter beer (interior liquid material) inhibits ethanol formation in the microbes, and hence decreases the final ethanol production rate. This undesired phenomenon is called ethanol inhibition (see *e.g.* Maiorella *et al.* 1984).

[36] See an example given in Chapter 7.1

6. Computer tool development

This thesis not only aims to show that PDPD can be implemented, but through computer tool development also aims to help the usage (*e.g.* data management of process design projects) of PDPD methodology – and through test usage the tool can be evaluated and illustrated. Building a computer tool support for methods and methodologies, as in this case for PDPD, is normally initiated by setting the purpose of the tool. This means compiling a list of computer tool requirements. After that, it is possible to select a proper computer environment for tool development. Once thorough requirements are set, it is possible to build a proper information structure, the methods and user interface, as well as make a continuous evaluation of the performance of the tool. The working procedure behind this thesis, which is shown in Figure 6, is not in conflict with some other general methods of software engineering for knowledge-based systems (*e.g.* Cuena 1993). Computer tool development can be formalised also with 'design cycle', analogously to process design as described in Section 5.1.2.

6.1 Purpose and requirements of the tool

This section can be considered as 'purpose' setting for the target software system. It is difficult, and even unnecessary, to make a distinction between a user's generic platform requirements and a user's requirements in respect of PDPD. This distinction has not been made in detailing the tool requirements. The list of requirements has been built as further specifications for the problems observed from the current knowledge of systematisation of R&D, as described in Chapter 3. The list has been initiated right at the beginning of the thesis but the list has also been edited during the implementation work.

Support for documentation

- From the user's perspective, the usage of the tool should "automatically" result in PSSP-formatted process models and project activity descriptions.
- The tool should "automatically" convert PSSP-formatted target (chemical process) and activity representations into user-specified format, if necessary, for example, into a textual report with a specified format of the organisation.
- The tool should support visualisation, like having editors for graphical representation of the target and activity.

- Ability to capture (or to link) the rationale behind the modelling and management decisions while using the tool.
- Ability to link various attribute data-types of objects:
    - Text and string
    - Fixed and editable lists and tables
    - Word.doc
    - Excel.xls
    - Input and output files of external solvers and applications
    - Pictures and figures
- The tool should enable a user to design his own data-query algorithms, such as checks and reports.
- The tool should enable the documentation of unexpected and creative solutions

Communication

- Support for communication between management, researchers and designers by enabling the storage and visualisation of information flows between activities.
- To provide a platform and guidance for making the communication follow the ‘design cycle’.
- Support for direct communication of project staff, such as providing a platform for making comments, notes, requests and replies.

Interactivity

- Dialog boxes for user’s input.
- Fields for program’s output.
- Graphical editors for editing process, project and activity flowsheets and networks.

Guidance to act as a checklist for what to do

- Guidance on knowledge retrieval according to the fixed modelling elements, structure and hierarchy of PSSP language to view project data and knowledge.

- Ability to generate suggestions on objects to be primarily edited according to performance-driven strategy.
- The tool should ensure that users are aware of the PDPD's guidelines and suggestions: performance-driven strategy and phenomenon-driven ordering.
- As PDPD is intended, the tool should support finding creative solutions.

Linking ability

- Ability to associate external applications with the object repository; data and executable files.
- Ability to transfer data between the tool and external applications. It is expected that 'one-way' data flow – from the PDPD platform to external tools – is sufficient.
- The tool should provide a database (or object repository) and have the ability to link external representations without retyping data-files.

Functionality of object repository

- Ability for automated report generation from the database (or object repository) with user's specified format.
- Ability to generate input-files for external applications. This means, for example, transforming PSSP-formatted models into ASPEN PLUS models.
- Tool-development environment and usage platform should enable simultaneous method modification and test usage, without losing or retyping the test data. This requirement was added during the research work resulting from bad experiences with the first development environments (see Section 6.2.1).
- The tool should be capable of multi-user sessions.

There is also a set of more detailed user's requirements listed during pilot projects. Now, after systematic specification of IS requirements, it is possible to realise the implementation procedure. The evaluation of IS versions is impossible without a properly setting the requirements. Next, how these requirements are realised in the tools prototyped during this study is presented.

6.2 Implementation

PDPD can be considered as a complex and abstract methodology, and one way to make PDPD more concrete is to develop a computer tool support. The more inexperienced the methodology user, the more helpful the computer tool is. For any user group, a computer tool is expected to provide aids for true data sharing, as well as aids for systematic view and procedure on research and modelling. In the following, the implementation work and platforms applied are briefly reviewed.

6.2.1 Brief history of tool development platforms used

After the preliminary IS requirements have been stated, it is then possible to argue for the prototyped IS versions. Next, a brief history is given of some early versions. Section 6.2.2 describes more carefully the latest prototyped version of the PDPD tool. The author has programmed all the PDPD tools presented here.

First version

The implementation work was started with a hypertext tool (Pasanen 1995) – ToolBook® 3.0 (Asymmetrix, 1994). The advantage in ToolBook was the “unlimited“ capability to develop the user interface. It was also easy to link external applications as attribute values with dynamic data exchange (DDE) and object embedding and linking (OLE) protocols, and with ToolBook Open Script language. However, linking the user interface and database was time-consuming when making methodological modifications. The database was maintained via ToolBook OpenScript programming language; the user data was read via input fields to ASCII files, which was a clumsy method. The first PDPD version developed with ToolBook was briefly tested in a finished project at Kemira Chemicals in Oulu (Pasanen 1995).

Second version

The next version was realised with a relational database tool (Pasanen 1996) – MS Access (Microsoft 1994). This environment provided better support for maintaining the database, but the modifications in the database structure resulted in time-consuming modifications in the user interface. The user interface was constructed as hypertext – AccessBasic and SQL were used for querying the data into text fields. AccessBasic and SQL were sufficient for the database queries, for input-file generation to external applications, as well as for reporting and checking the design specifications and data. MS Access also provides DDE and OLE capabilities. There was not any particular project-based test usage with MS Access. The final evaluation was that the relational database, such as MS Access, might be applicable in supporting an object-oriented methodology. However, building the PDPD tool with MS Access necessitates “the final” version of PDPD and PSSP language in order to save time when constructing the database and user interface.

Third version

The third and later versions of PDPD tool support are realised with MetaEdit+® (ME+) version 2.5 (Pasanen *et al.* 1998) and 3.0 (Pasanen 1998, Pasanen and Tolvanen 1999, Pasanen *et al.* 1999a and Anttila and Pasanen 1999). ME+ is proposed for methodology engineering, supporting the object-oriented features of the methodology, and enabling fast prototyping and testing. ME+ has an object repository and a meta-language (GOPRR) for describing the methodology. The data in the object repository (=database) can be queried and formatted into the user interface via MetaEdit+ query language (C++-like). The PSSP-formatted knowledge in ME+ can be linked to various external representations. One disadvantage of ME+ is in the limitations on designing the user interface. ME+ is a static environment, which means that the methodological guidance has to be transformed into GOPRR definitions and into database queries, such as automated reports on the missing parts of a process design, and suggestions on the objects to be primarily edited. However, for the purpose of this work (Chapter 4) ME+ proved sufficient for prototyping and for arranging the user tests, which are described in more detail in Section 6.2.2 and in Chapter 7.

Other environments

Visual Prolog®[37] was considered but then rejected since it could offer no additional value to the current situation. VisualBasic was considered and tested very briefly in MS Excel. The idea was to compile a 'self expanding folder and file system'. When initiating a project, then the folder structure, with the object hierarchy shown in Figure 8, is built automatically according to the VisualBasic macro in Excel. The folders are supplied with proper files, which have a predefined structure according to PSSP for each class. VisualBasic might be

---

[37] http://www.visual-prolog.com/

used in the same way as MS Access for the “final“ PDPD version. Visual C® or Visual C++® would also provide platforms for the final version, but for the purpose of this work – fast prototyping and test usage – it has been evaluated as a time-consuming environment. As the latest prototype, the author has compiled a documentation platform for PDPD with a www-browser. It is used as a documentation and report formaliser of research work, conceptual models (physico-chemical), mathematical models and numerical models, and data as well. The users are supposed to edit the intranet database with a commercial www document editor (Netscape® 5 and Netscape 5 Composer®). This approach is adopted in a project of POSIVA for documenting the modelling work and adsorption models in geo-chemical systems. The use of PDPD and the www-platform in this project is expected to make the working procedure and the models more comprehensive.

6.2.2 Platform for implementation – MetaEdit+

For the computer implementation of PDPD, a program called MetaEdit+® (MetaCase Consulting 1999 and 1996) has been applied, which was experienced as an efficient tool for the purpose of this thesis. MetaEdit+ is a multi-user metaCASE (Computer-Aided Software Engineering) environment, in which designers and researchers can use methodologies and methodology engineers can develop methodologies (Kelly *et al.* 1996). Thus, ME+ operates with both models and metamodels, *i.e.* a model of a method or methodology. In this case, it operates with chemical process models based on PDPD, and metamodels describing PDPD methodology. The (meta)methodology behind ME+ is GOPRR (Graph, Object, Relationship and Role) language, which is expected as sufficient for modelling design methodologies. Although MetaEdit+ has been designed to support system engineering methods, it has also been applied in other domains, such as business modelling and analysis of logistic processes. Now, during this thesis ME+ and GOPRR have been applied for modelling a design methodology for chemical engineering purposes.

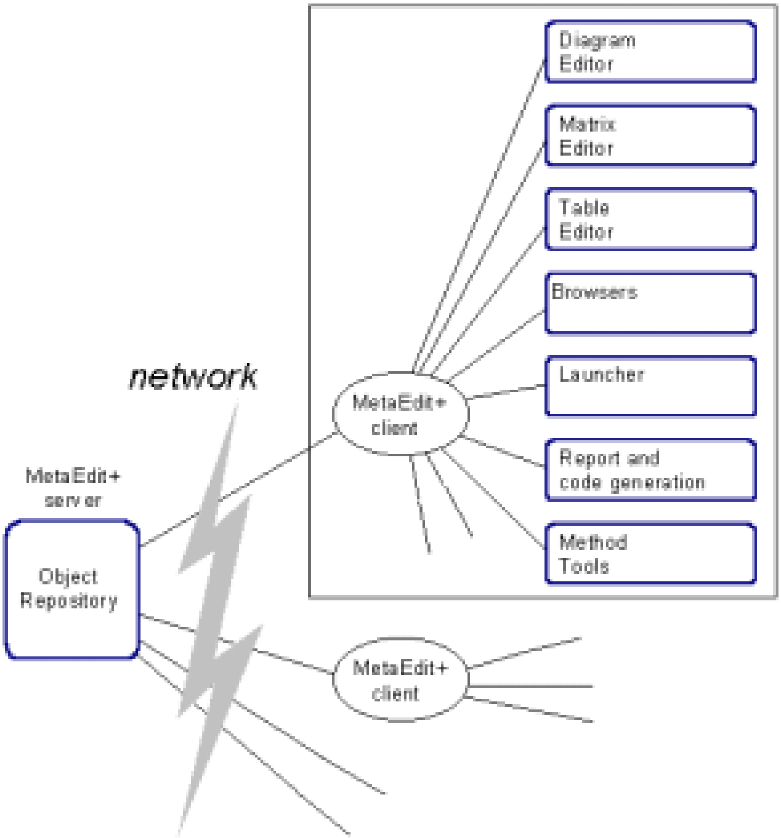


Figure 13. Architecture of MetaEdit+ (Rossi and Tolvanen 1995, and MetaCase Consulting 1999).

ME+ results from an academic research project; MetaPHOR (Lyytinen et al. 1997). ME+ provides a fully functioning object repository and the implementation code behind ME+ is Smalltalk (see e.g. Goldberg and Robson, 1983). ME+ evaluation version can be downloaded from the web[38]. ME+ provides tools and editors for facilitating the use of GOPRR for a methodology engineer during the implementation work. Those tools are introduced in Applendix 6.1. The current version of the PDPD tool implemented with ME+ (3.0) applies all the methodology engineering tools and editors described in Appendix 1, and users can utilise all the described editors and procedures except Table- and Matrix editors,

---

[38] http://www.metacase.com/

and explosion and decomposition procedures. The high-level architecture of ME+ is depicted in Figure 13.

6.3 Structure of the PDPD tool

The efficiency of the prototyped tool is obviously dependent on the effort put into the computer implementation work. It can be claimed that no matter what the computer implementation environment is, the resulting PDPD tool of two programmers will always have differences. The main concern when analysing the PDPD tool reported in this thesis is on how the relevant features of PDPD are supported by the features of ME+. In this thesis, the features that are relevant for PDPD were listed in Section 6.1. is the subject of another research project, which is beyond the scope of this thesis, is how the prototyped computer implementation reflects the user requirements according to massive user tests. However, building up a user interface of the final and analysed method for an organisation is "routine" work in software development and, according to the author, the final computer implementation of PDPD is not an exception. Why is it necessary to describe the structure of the tool? The effectiveness of the prototyped tool is dependent on the structure, which implies, *e.g.* the required functionality of the database, and the required functionality of the user interface, as given for the PDPD tool in Section 6.1. On the other hand, chemical engineers are usually not familiar with this kind of computer tool development work. However, designing computer tool support for other methods or methodologies might be relevant for an engineer. Still, this kind of "method guide book" for building tool support for an engineering method, or methodology, is unique, and could be very useful.

6.3.1 Realisation of the requirements

The implementation of a modelling tool for any design methodology necessitates normalisation of the methodological knowledge. The implementation of the PDPD methodology is not an exception, even though PDPD has been supported with a formal PSSP language. Modelling the methodology into metamodels – GOPRR models of ME+, carries out the normalisation of the conceptual structure of PDPD. A metamodel in computer implementation means a descriptive model of a methodology to be implemented into computer, which can be seen as one higher level of abstraction than ordinary models (see Section 5.1 for metamodel of process models). The meta-models of ME+ describe the concepts, rules and procedural guidelines of the methodology producing a model of PDPD. The process of developing metamodels is called metamodelling. Metamodels can be further divided into meta-data models and process models (in this section ‘process model’ should not be confused with ‘chemical process models’). Meta-data models describe static aspects of methodologies, such as their conceptual structure and notation. Process models specify the relationships between the design tasks targeted to achieve certain objectives or to emphasise the specification of modelling products and the activities needed to make it evolve. Figure 11 shows an example of the process model describing the tasks and their precedence when using 'Generic Real Things'.

The usefulness of metamodelling not only lies in systematising methodologies, but especially in the possibility of applying meta-models as specifications for computer implementation. This means that the instances of the metamodel act as class-level definitions for methodologies. In an optimal scenario, metamodels are readily applicable specifications for the computer-aided environment. This requires that the methodology is specified formally enough and the computer program is able to interpret, compile and execute the metamodels.

A procedure applied in the implementation of PDPD, can be described as shown in Figure 14. During the PDPD tool development, a number of projects have been undertaken in order to evaluate the tool for project management and for design purposes.

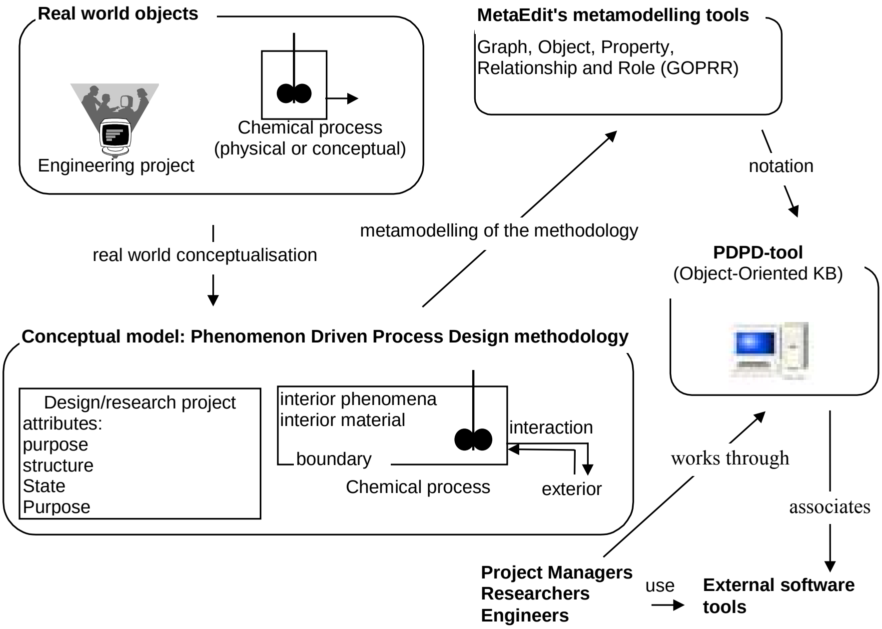


Figure 14. System principle and procedure for building a tool-supported methodology.

6.3.2. Structure of the information system

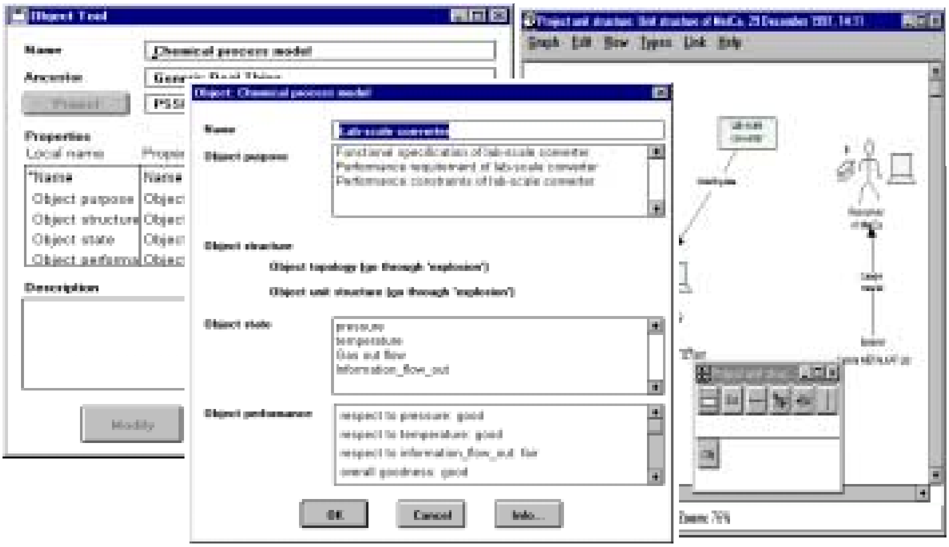
For the purpose of implementation and systems analysis, the PDPD methodology is specified with OSA (Embley et al. 1992). OSA is an Object-oriented Systems Analysis method, which supports data modelling for the purpose of meta-data models and behaviour modelling for the purpose of describing the PDPD process. Some examples of these metamodels are shown in Figures 8 and 11. For the purpose of implementation, we have transformed the metamodels into a GOPRR metamodel applied in MetaEdit+. In other words, all the methodologies to be implemented with MetaEdit+ must be specified with GOPRR. Since GOPRR has been specifically developed (Kelly *et al.* 1996 and Tolvanen 1999) to support metamodelling, the use of GOPRR is expected to specify all the relevant aspects of the methodology with sufficient formality. The word “sufficient” means enough formalisation to carry out the implementation of tool requirements (see Section 6.1). The transformation from OSA was carried out manually, but most of the transformations were relatively straightforward. For example, an object in OSA is normally also an object in GOPRR. It should be noted that the usage of OSA is not necessary, but is expected to ease the methodology transformation into GOPRR. Pohjola and

Tanskanen (1998) describe the formal language behind PDPD as PSSP language, in Baccus Naur Format (BNF). BNF formalisation might provide another type of support for computer implementation. However, BNF formalisation is not necessary when implementing the methodology with ME+.

The implementation procedure of PDPD with MetaEdit+ is fairly straightforward and provides fast prototyping. With MetaEdit+'s Object-, Property-, Relationship- and Role-tool – the classes, class properties and property inheritance (see Figures 8, 13 and 14) are set up. This stage also includes symbolisation of the modelling elements and setting up the dialogues used for capturing chemical process design data and knowledge. The object hierarchy is set up with Graph-tool and it guides PDPD users to trigger and view objects in a particular order with graphical editors.

These GOPRR definitions ensure that engineering data and knowledge are represented in PDPD format with PSSP language. However, these definitions ensure only partly that users follow the methodology. The object database has to be queried in order to check users' actions and the current attribute values of objects. Then, the PDPD tool is capable of suggesting what to do next, requesting the missing data and evaluating the current states of design tasks and chemical process models. The data queries are described more precisely in Section 6.3.3.

The metamodelling tools of MetaEdit+ are used to capture the concepts, the rules between the concepts, and the notations of PDPD. Hence, the metamodels form the unified concept structure according to PDPD. For each concept, a representation is added to support communication and documentation. This enables project managers, process designers and researchers to trigger objects and set values for 'purpose', 'structure', 'state' and 'performance' attributes for every individual instance of a class. Chemical process models and project models are represented by graphical tools, which formalise the editing and viewing of knowledge. Sufficient formality is expected to enhance the model comprehension. An example of the GOPRR metamodel, and its instantiation in chemical process modelling, is illustrated in Figure 15 (Pasanen *et al.* 1998). The window on the left defines the chemical process model as a GOPRR object consisting of a set of properties, such as "name" and "purpose". Since chemical process models are subclasses of GRT in the GOPRR metamodel, these properties are inherited. The window on the right describes the unit structure of a project in a diagram editor of MetaEdit+. Finally, the dialog in the middle describes a chemical process according to PDPD. Here, an instance of a 'chemical process' named "lab-scale converter" is viewed through its properties, such as 'name' and 'purpose'.


Object Tool

Name: Chemical process model

Properties

Local name

Name

Object purpose

Object structure

Object state

Object: Chemical process model

Name: Lab-scale converter

Object purpose:
Functional specification of lab-scale converter
Performance requirement of lab-scale converter
Performance constraints of lab-scale converter

Object structure

Object topology (go through 'explosion')

Object unit structure (go through 'explosion')

Object state:
pressure
temperature
Gas out flow
Information_flow_out

Object performance:
respect to pressure: good
respect to temperature: good
respect to information_flow_out: fair
overall goodness: good

Figure 15. A metamodel of a chemical process (left), a model of a chemical process (middle) and a project (right) according to Phenomenon-Driven Process Design methodology.

Figure 15 also illustrates the possibility to use modelling tools and metamodelling tools simultaneously; consequently, chemical process engineers can even apply the methodology while methodology engineers are re-developing it. This feature allows a fast prototyping approach in which metamodels are tested and modified based on the experiences of their use in chemical process modelling.

Figure 16 illustrates the architecture of the PDPD tool, prototyped with ME+.

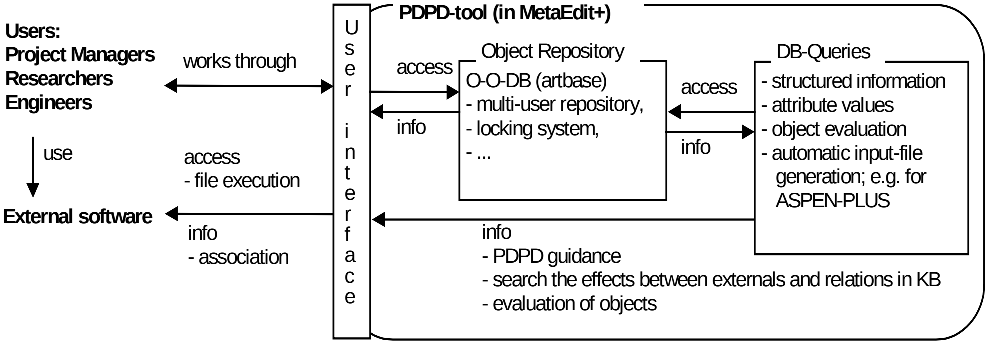


Figure 16. The implemented system description from the user's perspective.

6.3.3 Data processes

Most of the static aspects of PDPD can be supported with MetaEdit+, tools for formalising design targets and design activities into PSSP specifications. It is a more challenging task to capture the behaviour of PDPD for guiding project managers, researchers and designers. The behaviour, or dynamics, of PDPD refers to the questions: "In what order?" and "In what way?" It should be noticed that most of the dynamic part of PDPD could be supported with static features of the tool. Static features means how the design data is structured in the IS (part of the "In what order" –control), and how information on object usage is provided while using the IS (part of the "in what way" –control). MetaEdit+ is currently a static environment, which does not perform object behaviour descriptions dynamically, as automated suggestions during design. One example of such a dynamic feature is PDPD guidance in accordance with the sub-principle (2.3): "Characterise structural disaggregation | aggregation of a process in such a way that it possibly results in the required process state with the specified process performance criteria". Principle (2.3) is part of the procedure of how to create alternative chemical process models. In order to capture Principles 1 and 2, MetaEdit's graph editor has been applied to ensure PDPD-formatted models. This is done by providing the sequence and hierarchy for the object types according to which the instantiation and navigation in the object hierarchy are performed.

Moreover, to ensure that user is aware of Principles 1 and 2, and other dynamic features involved in PDPD usage[39], a number of reports have been compiled

---

[39] Appendix 2: A scheme of controls and data processes

with the MetaEdit query language. These queries are based on PDPD metamodels and allow users to view and report the object repository and to make comprehensive data retrievals. So far the compiled queries are:

1. 'Report an object | set of object system purpose +| structure +| state +| performance attribute values'.
2. 'Search for the terminal topology' to find the most detailed process model (applied in external input-file generation to ASPEN PLUS®).
3. 'Differences between the target (under purpose) and the current (under state) values of object variables'.
4. 'A suggestion on the objects to be primarily edited'.
5. 'Affected set of externals' for a selected part of the mathematical relations to help the users to maintain the knowledge and data consistency among the PDPD environment and external applications.
6. A few examples of automated or semi-automated external input-file generations to ASPEN PLUS®, Mathematica® (Wolfram 1988) and MODEST® (1994).

The first query[40] facilitates reporting the object repository in text-format. The usage of the PDPD tool forms a hierarchical and multidimensional system, and it is time-consuming to extract the data in the object repository into lists or to some textual format. Thus, the author has compiled an automated report generation for the repository according to user's selection (*e.g.* a part of activity network or part of the process system). The second query, 'search for the terminal topology' requires some further explanation. This piece of code searches the most detailed topological representation of a process. An example of a terminal topology is given in Figure 17. The result of the code may enhance the comprehension of a process system. The second code may also be required when compiling an input-file to external applications.

---

[40] Appendix 3: Report an object purpose

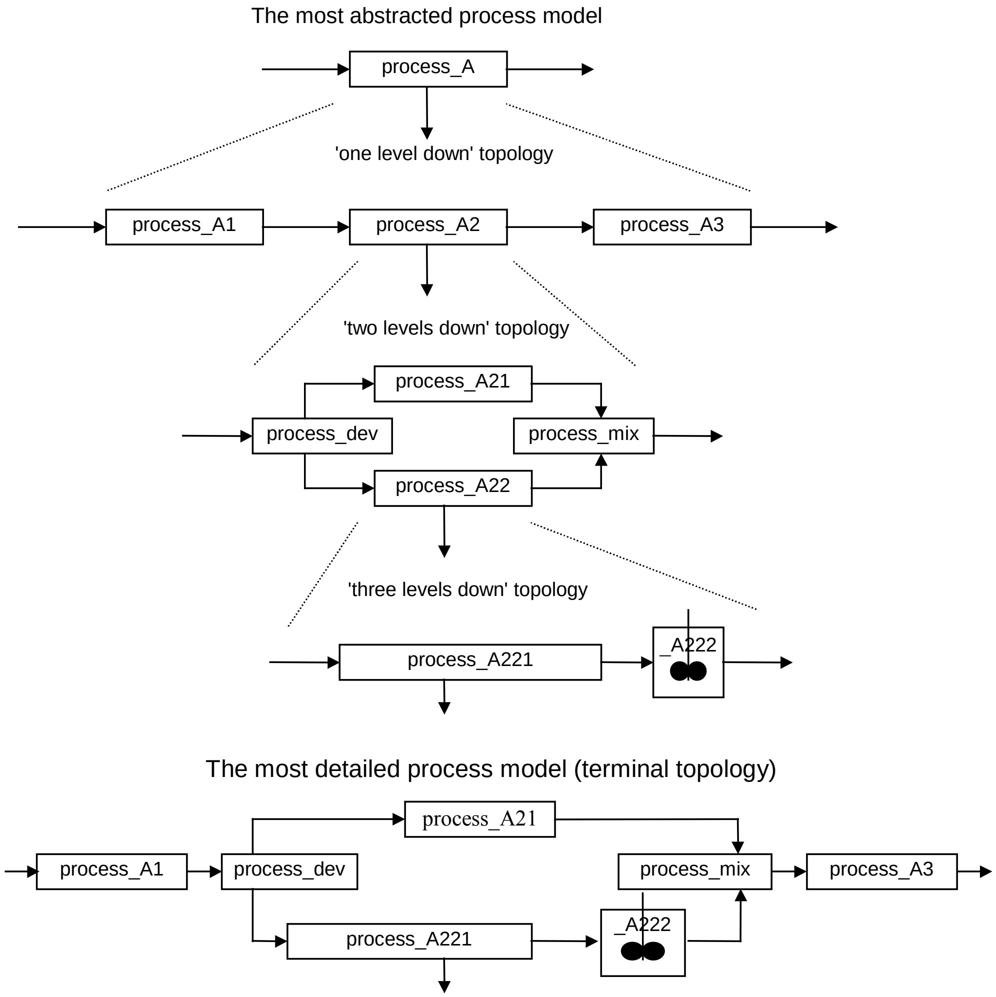


Figure 17. An example of abstraction levels for a process.

The third query[41] extracts the current values of the object properties of a project. It is intended to help to follow Principle 1 (see Figure 11): Use performance driven strategy to edit objects. Extracting all the values of object purpose and state attributes is time-consuming; therefore, it has been supported by the automated report. The fourth query analyses, when necessary, the result of the third query and provides a suggestion of what order to edit objects. The suggestion is based on the object hierarchy (see Figure 8) by a) searching the objects not meeting the target variable values expressed under object purpose

---

[41] Appendix 4: Project property report

(the third query), and b) by ordering these objects in accordance with the hierarchy (application of the second query)[42].

The fifth query[43] is intended to help follow the consistency of the mathematical models. It reports that, if a designer modifies a mathematical relation or external solver, then this code searches all the mathematical models, and external files or applications that are possibly affected. The algorithm is based on using the variable type definition in a mathematical relation: variant | invariant, and on using a tree search algorithm[44]. This fifth search was also programmed into the second version of the PDPD tool (MS Access®) with SQL, VisualBasic and AccessBasic[45].

The sixth query integrates the object repository (attribute values of PDPD objects) with external applications; the query language can be used to generate some specific input files automatically. For example, an ASPEN PLUS (Aspen plus 1995) input file – expressed with the ASPEN Input Language – can be automatically generated from PDPD-formatted engineering data in MetaEdit+[46]. 'Search for the terminal topology' is applied in this input file generation for skipping, if necessary, the abstracted design models in the PDPD tool. However, the linking of externals in some cases cannot be fully automated; manual steps may be required for compiling and using external input and result files, as in the case presented in Section 7.1.

Instead of applying procedural models to guide PDPD usage, MetaEdit's hyper-document tool for linking information about the right procedure of using PDPD has been applied (Pasanen *et al.* 1998). This is done by simple hyperlinks and 'Question-Answer-aRgument' descriptions (QAR) (Oinas-Kukkonen 1996) with a beta version of the Rationale tool in ME+ 2.5. The linking of hyper-documents has also been used to capture the design rationale while using the methodology, including the argumentation of the process design versions (see Figure 21) and argumentation on PDPD concepts. In this way, it is possible to capture the information of how the PDPD methodology should be used and how it is used.

---

[42] Appendix 4: Suggested sub-processes to be checked
[43] Appendix 5: Block diagram of the search program
[44] Appendix 6: Effected set of relations with ME+ query language
[45] Appendix 7: Effected set of relations with MS Access (AccessBasic and SQL)
[46] Appendix 8: Aspen Plus code generation and search for the terminal topology

6.4 Evaluation

The evaluation of the prototyped PDPD tool is given in respect of the set evaluation criteria, as given in Section 6.1. Also the final overall evaluation is given. The value of the evaluations is not formalised, say, as good or poor, but as descriptions of how the requirements have been realised and what the limitations of the solutions are. Objectively setting an exact value for each criterion would be difficult, and it would require some larger test and evaluation arrangements (possibly another project in the future) than those conducted by the author in this research. Now, by briefly reporting the solutions and limitations for each criterion, a transparent and open-formatted evaluation can be obtained.

Support for documentation

- From the user's perspective, the usage of the tool should "automatically" result in PSSP-formatted process models and project activity descriptions.

  PDPD prototype tool

  With ME+'s GOPRR metamodelling language, the PDPD methodology is transformed into a visual object system. The usage of the developed system always results in PSSP-formatted models. In other words, the data and models in the object repository can be viewed and modified via a Graphical User Interface (GUI). The graphical user interface is specified according to PSSP language and PDPD methodology.

- The tool should "automatically" convert PSSP-formatted target (chemical process) and activity representations into user-specified format if necessary, for example, into a textual report with the specified format of the organisation.

  PDPD prototype tool

  With ME+'s query language (C++-like) it is possible to program a query, which converts the data and models in the object repository into a user-specified format. This enables extracting the PDPD object topologies, single objects and their attribute values into text files or graphical files (*e.g.* gif) to be used in a text or document editor. Objects in ME+ can be automatically extracted into HTML specifications. The query language has some limitations, such as lacking variable definition (ME+ 3.0), which makes it impossible to code some nested loops (see an example case in Section 7.3: the Mathematica input-file cannot be created via abstracted representations of a process). ME+ query language is under continuous refinement.

- The tool should support visualisation, such as having editors for graphical representation of the target and activity.

PDPD prototype tool
Symbol-editor allows a methodology engineer to build symbols for the classes, such as chemical process or activity. Dialog-editor is used for building dialog-boxes for editing the content of objects and attributes. With ME+'s graph-editor, a methodology engineer is allowed to build platforms for graphical representations. Via diagram-editor, end-users may edit and view objects and models, such as the topology of a chemical process or network of activities. However, designing a graphical user interface is limited by the ME+ method engineering tools. On the other hand, no further requirements have been set for the graphical user interface. See examples in Chapter 7.

- Ability to capture (or to link) the rationale behind the modelling and management decisions while using the tool.

PDPD prototype tool
ME+ 2.5 with the beta version of the Rationale tool allows storage of the rationale of a methodology (for helping the methodology usage) and the rationale and notations of a user's actions while using the methodology. A user is allowed to link any notations on any object: on management and design activities, on chemical process models, on mathematical relations, etc. (see the examples in Section 7.1.).

- Ability to link various data-types of attribute values of objects:
  - Text/string
  - Fixed/editable lists and tables
  - Word.doc
  - Excel.xls
  - Input and output files of external solvers and applications
  - Pictures and figures

The data-type of a property in ME+ can be a string, text, fixed list, a collection of objects, and external applications. With these data-types, the PDPD tool is expected to be capable of expressing any user-specified attribute value. Attribute value-types in chemical engineering can be distributed, fuzzy, discrete, etc. The numerical values are usually stored in the external data-files of an external solver application. The usage of the prototype allows an attribute value to be associated with an external data-file or application, such as Word, Excel, CFX® (computational fluid dynamics tool) geometry.dat, CFX result.dat, ASPEN+ result.dat, etc. There is no need to retype any existing data.

- The tool should enable a user to design his own data-query algorithms, such as checks and reports.

  PDPD prototype tool
  ME+ query language can be used for this purpose. Compiling the queries, reports or checks requires coding with the ME+ query language (C++ like). After a short period of training, the language is easy to use for any user. See the examples in Appendices 2, 3, 4, 7, 9 and 10)

- The tool should enable the documentation of unexpected and creative solutions.

  PDPD prototype tool
  As the modelling elements in PDPD are intended to support and enable creativity, the tool follows the PDPD, and thus, has all the modelling elements of PSSP language, which are expected to support and enable the documentation of some unexpected and creative models. The argumentation behind the models can also be documented.

Communication

- Support for communication among management, researchers and designers by enabling information flows between activities to be stored and visualised.

  PDPD prototype tool
  Activities, and information flows in between, can be represented graphically. The object repository can be queried for making suggestions about missing

data and activity types. However, ME+ is based on voluntary usage, which means that the user is responsible for expressing and storing all the relevant data.

- To provide a platform and guidance for making the communication follow the ‘design cycle’.

  PDPD prototype tool
  The tool provides a platform, which “enforces” the user to initiate activities and trigger objects according to the first design cycle. However, the tool does not dynamically check whether the user follows giving the values of the objects’ attributes in order of PSSP; nor does the tool dynamically check if project purpose setting, process design activities and process data on process objects are monitored according to ‘design cycle’ later, after the first cycle. See *Guidance to act as a checklist on what to do.*

- Support for direct communication of project staff, such as providing a platform for making comments, notes, requests and replies.

  PDPD prototype tool
  The tool allows documentation of users’ comments: requests, replies, notations and rationale (ME+ 2.5 extended with the beta version of the Rationale tool, e.g. Pasanen et al. 1998). This direct communication – via messages – can be linked to any object, such as activities or process models and data. ME+ automatically stores user-data for any object, such as object creator and date of an instantiation.

Interactivity

- Dialog-boxes for user’s input.

- Fields for program’s output.

  PDPD prototype tool
  With the ME+ dialog-editor, the methodology engineer (the author) has designed dialog-boxes for all PDPD object classes. Users can view and edit the PSSP attribute values of an object via these dialog-boxes. The lists produced by the object repository queries can be designed with ME+ query

language. Designing object repository queries is simple and can be done both by methodology engineers as well as end-users.

- Graphical editors for the editing process, project and activity flowsheets and networks.

  PDPD prototype tool
  The tool provides the graphical tools for initiating and for editing process, project and activity flowsheets and networks. However, designing the GUI (e.g. symbols, dialogs and graphs) is limited by the ME+ method engineering tools (see Section 6.2.2). Modifying the flowsheets and networks is also not as easy as in commercial flowsheeting tools.

Guidance to act as a checklist on what to do

- Guidance on knowledge retrieval according to the fixed modelling elements, structure and hierarchy of PSSP language for viewing project data and knowledge.

  PDPD prototype tool
  Viewing and editing the models and data with the tool can be done in the sequence of the object hierarchy of PSSP. When necessary, it is also possible to skip the hierarchy and to focus directly on a section of, say, process topology, or on the interior of a process.

- Ability to generate suggestions about objects to be primarily edited according to performance-driven strategy.

- The tool should ensure that users are aware of the PDPD's guidelines and suggestions: performance-driven strategy and phenomenon-driven ordering.

  PDPD prototype tool
  The first 'design cycle' forms automatically in the right sequence. After the first 'design cycle' has been made in a design project, and most of the objects are triggered, then the suggestions on continuation are given via ME+ queries[33, 34, 47]. The actions of PDPD user's and data in the object repository

---

[47] Appendix 10: an example query

can be checked via queries based on performance-driven strategy and phenomenon-driven ordering. This means that, object properties – target and current values – can be checked automatically, and object properties deviating from their target and current values can be ordered according to the hierarchy of PDPD. The user may also categorise projects and activities, for example, as ‘creation of background data’, ‘creation of knowledge on existing processes’, ‘creation of knowledge on existing R&D projects’, ‘process development’, ‘process trouble-shooting’, ‘process retrofitting’, ‘process modelling’ and ‘modelling data generation’. The user must select a proper category for a project or activity when instantiating it. In this way, it is possible to develop predefined suggestions on users’ actions based on the users’ selections. A part of the query ‘if … then’ suggestions is given in Appendix 4. The categorisation and the automated suggestions presented here can be extended and specified further.

- As PDPD is intended, the tool should promote finding creative solutions.

  PDPD prototype tool
  The tool does not give direct suggestions of creative solutions according to Principle 2. However, features supporting the initiation and documentation of creative actions are not lost in the prototype tool. According to PDPD, conceptual process models in the tool are not dictated too early with the mathematical models. However, when necessary, it can be directly decided whether to use a special calculation tool. ME queries provide aids for following Principle 2, such as 'a suggestion of the objects to be primarily edited' and 'report an object | set of object system purpose +| structure +| state +| performance attribute values'. The user can also compile a query reporting the design rationale of process models, providing, say, extra information behind the decision on the process structure and state.

Linking ability

- Ability to associate external applications to the object repository; data and executable files.

PDPD prototype tool

ME+ provides the ability to link various external software environments as an attribute value of PSSP objects. This is done by typing the attribute value as:

location of the tool; C:\\ …
executable file name; e.g. "execute C:\\…\model2.xls"
a set of locations of the required set of external files and applications; e.g. C:\\…\model_1.dat, C:\\…\model_2.dat, ….

- Ability to transfer data between the tool and external applications. It is expected that 'one-way' data flow – from the PDPD platform to external tools – is sufficient.

PDPD prototype tool

The linkage of external environments is one-way, which means that the location of a data-file is documented, or an executable file can be opened or executed, but no data is transferred between ME+ and an external. In this way, the values and mathematical equations giving the values for the attributes in the PDPD tool are attached to the source. In this manner, the PDPD tool and external applications are kept separate, without any complicated protocols, such as DDE or DLL. A query has been compiled for checking and reporting the affected set of externals and relations if editing mathematical relation(s) or external application(s). In this way, data consistency can be followed in a semi-automated manner.

- The tool should provide a database (or object repository) and ability to link external representations without retyping data-files.

PDPD prototype tool

Linking external files and applications has been enabled as described above. However, in some cases, avoiding data retyping may require some special programming. For example, the result of an ME query is an input file for an Excel file, which in turn is an input-file for the solver application. This kind of arrangement might be required if a tool, say a simulation tool, requires its own conceptual representations or some extra numerical input-files. If the external tool has some elements not present in PDPD then no automated methods can be compiled merely via ME+.

Functionality of object repository

- Ability for automated report generation from the database (or object repository) with user's specified format.

- Ability to generate input-files for external applications. This means, for example, to transform PSSP-formatted models into ASPEN PLUS models.

  PDPD prototype tool

  Section 6.3.3 describes the compiled queries. Applying and programming other input-file generation is easy for users after a short period of studying the query language. However, in some cases completing the user's input-file specification generated via ME query, requires some manual operations, or e.g. Word macro operations.

- The tool-development environment and usage platform should enable simultaneous method modification and test usage, without losing or retyping the test data.

  PDPD prototype tool

  The tool-development and user tests with ME+ can be made simultaneously, without losing or retyping the test data while modifying the PDPD tool. When modifying the method, ME+ automatically checks the consistency of the user-data in the object repository, and up-dates it.

- The tool should be capable of multi-user sessions.

PDPD prototype tool

ME+ automatically provides locking systems of objects during multi-user sessions. In this study, the PDPD tool was installed into a server system, but it enabled only one-user at a time. Multi-user sessions set some extra requirements, such as keeping track of various model versions. Multi-user requirements were not sufficiently studied during the pilot projects.

The PDPD tool applies a unified structure for the documentation of process models and data, which facilitates documentation and knowledge retrieval. One of the elements requested by the tool is object purpose, which has an unquestioned special importance. This utility has been applied for every object, such as mathematical models. For any user group, the PDPD computer tool is expected to provide aids for true data sharing and a systematic view and procedure for research and modelling. The more inexperienced the methodology user, the more helpful the computer tool is when using PDPD.

Over all evaluation

The prototyped tool has been evaluated against all the set requirements. The ME+ can be evaluated as a good method engineering environment for fast prototyping. The PDPD tool, in turn, can be evaluated as good for demonstrating the required capabilities of a computer tool support for phenomenon-driven process design methodology. However, the “first” round of implementation done during this work resulted in *a postepriori* requirements. Tool development is an iterative process.

The tool development and test use put pressure on carrying out a thorough analysis of the methodology itself; and now, in the case of PDPD tool development, it has revealed features that cause problems in computer implementation: How can the dynamic features of PDPD be supported? What advantages do the concepts and procedural guidelines in the form of computer code provide for users? What kind of user interface would be sufficient? The evaluation of test use is given in Section 7.5.

7. Case studies

The justification of the study (in Chapters 2, 3 and 4), the methodology (in Chapter 5) and the development of the computer tool support for the methodology (in Chapter 6) have been described. This chapter provides demonstrations and user test data for evaluation and argumentation of the practical performance and usefulness of the prototyped tool. However, the case studies are only a modest step towards more extensive user tests; thus, this work cannot be seen as a thorough attempt at making solid inductive deductions. Each case is harnessed to demonstrate certain features of PDPD and PSSP. The cases have been reported earlier in the following papers: Section 7.1 Pasanen *et al.* (1998), Section 7.2 Pasanen *et al.* (1999a) and Section 7.3 Anttila and Pasanen (1999).

The author's part of the actual work was to compile the tool and the required code generators and automatic report generators, with MetaEdit+. The author has also been the main user of the tool in Case 7.1 and 7.2. However, during the test use of these cases, the author has demonstrated the use of the tool for the project groups involved in the cases, and collected the comments. In Case 7.3, the author has compiled the code generator and the proper user interface, and research scientist Juha Anttila has used, and still uses, the tool independently.

7.1 Modelling of automobile exhaust gas catalysis

This case study is based on a research project titled "Modelling of the surface reaction kinetics with transient response method in automobile exhaust gas catalysis". This case project is run at the University of Oulu, and it belongs to larger projects called MetCa (1996–1998), financed by Academy of Finland (under the MATRA programme), and MeCaNOx (1998–2000) financed by TEKES (under the Nano-technology programme). This project has its own purpose and goal definitions (see Section 7.1.2), but it also has another purpose definition in respect of this thesis. The case project has advanced independently from this thesis, but the author has also harnessed it to demonstrate the features of PSSP language.

7.1.1 Purpose of the case study

The case study is designed to test and demonstrate how PSSP formalisation facilitates the comprehension of modelling work and model networks. In respect of this thesis, this case project demonstrates how the project definitions and the knowledge of transient kinetics modelling are translated into PDPD format and represented in MetaEdit+ (version 2.5). The focus is on how to formalise process models into PSSP language. The PDPD principles, 1) performance-driven strategy, and 2) phenomenon-driven ordering are skipped, but the project purpose, research activities and process models are described according to PDPD with PSSP language.

7.1.2 PSSP formalisation

Project purpose

The purpose of this project was to produce a kinetic model for a real exhaust gas converter (goal) with different catalysts and temperatures with high knowledge credibility (performance requirements). A transient technique, transient experiments and fitting of model parameters to data, has been selected as the modelling technique to capture the behaviour, c_i=f(T), of the catalysis with various catalyst material (functional specification). The performance requirements of the project also define the process model expectations, such as "the kinetic model should be valid in dynamic conditions; possibly to be used in the design of a real exhaust gas converter".

Project structure

This is a single-node project in its topological structure. One of the project unit-structural elements is 'activity target', which is composed of the following objects: "real world"[48], "conceptual model of chemical process" capturing the essential phenomena from the part of the real world under investigation, "lab-scale converter" producing the required measured data (concentrations) from "real world", and "mathematical model" producing the numerical values. The 'project unit structure' also includes a few research activities and allocated resources. Figure 18 shows how few project purpose expressions (textual postulates) and 'activity topologies' are given in MetaEdit+ editors. The required

---

[48] There exists a real, commercially available automobile converter, and its catalyst material was available for the project.

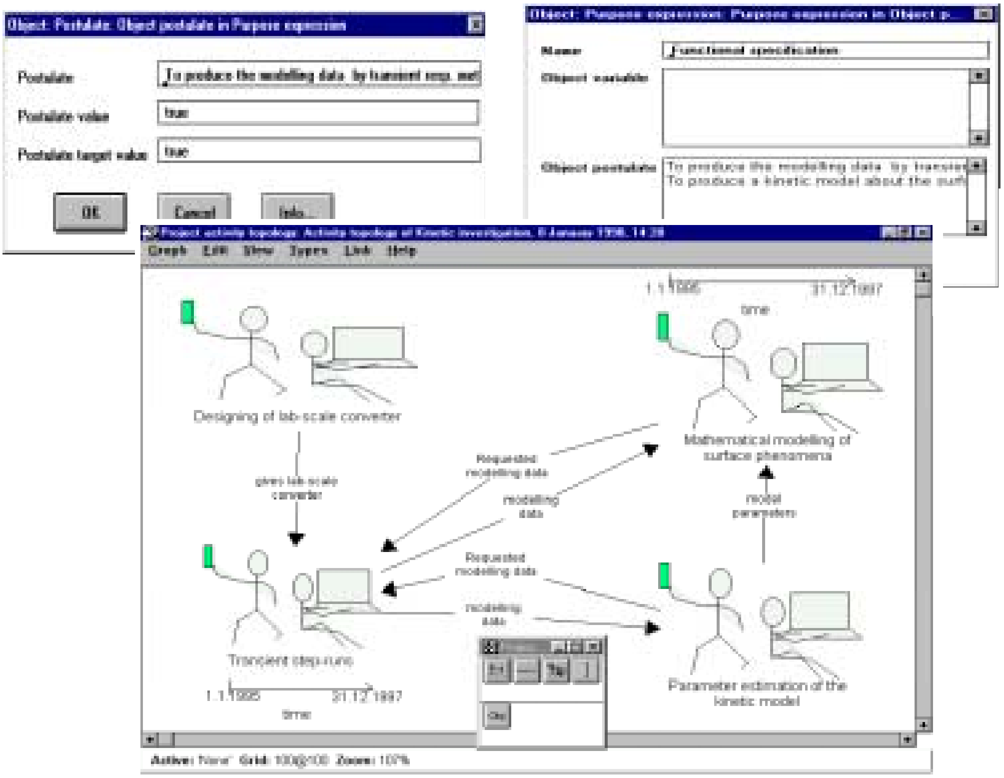
project knowledge (see Project purpose) is produced with the following research actions:

- "test reactor design and construction" with the purpose of producing a tubular lab-scale converter
- "transient step runs in lab-scale converter" with the purpose of producing kinetic data
- "sketching of mechanistic steps of reactions on catalyst surface" with the purpose of finding a mechanistic model that fits the measured data.
- "parameter estimation" with the purpose of producing parameters for kinetic equations

"Transient step runs in lab-scale converter" is disaggregated into two nodes:

  - "Running pure exhaust gas components into inert flow (helium)" with the purpose of "getting the mass transfer and adsorption/desorption kinetics of the components on the catalyst surface"
  - "Running the reacting components with transient steps" with the purpose of "getting data for setting up the reaction mechanism and kinetic parameters".

A step run with a lab-scale converter is a controlled interaction. The result (converter set up and converter data) of the "step runs" (research actions) is represented in MetaEdit+ as a set of material interactions between the exterior and the interior of the "lab-scale converter". These runs are performed with different temperatures, and each step run has the purpose as described above. Furthermore, a few notations are linked to the step runs, such as the reasons for running the transient tests in a particular way. In general, such rationale can be given in MetaEdit+ by linking the text annotations or the methodology rationale (see previous section) for design objects.



Postulate: To produce the modelling data by transient resp. met

Postulate value: true

Postulate target value: true

Name: functional specification

Object variable:

Object postulate: To produce the modelling data by transie
To produce a kinetic model about the surf

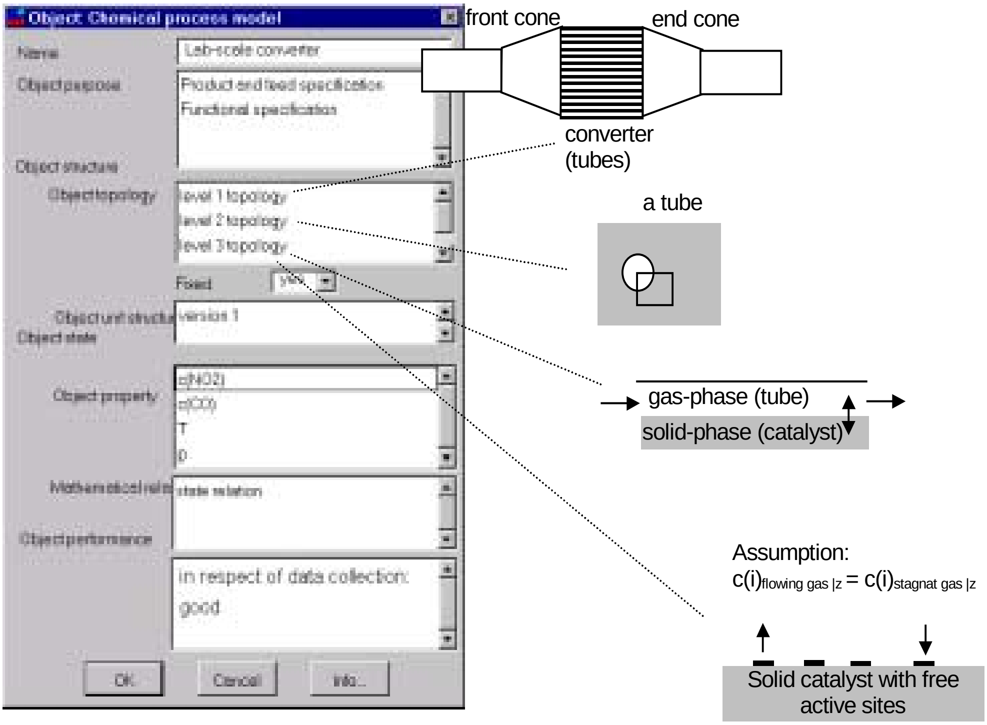
Figure 18. Dialogs for purpose expression (left) and an editor for activity topology (right).

Process-modelling concepts

The introduction of the models involved in this case, starts by naming the features form the real world (automobile exhaust gas converter) to be taken under investigation. These features are formalised into a conceptual model with PSSP language. The goal of the project, expressed above, necessitates mathematically formatted models of the phenomena that occur in the converter. In order to set values for the model parameters, experimental data describing the real converter is required. For this purpose, a lab-scale converter was constructed. The model building is started by first formalising the features of the lab-scale converter with PSSP language. The functional specification of the lab-scale converter is to give the kinetic modelling data (out-flow of information). Structurally, this process is a single-node process and its interior objects are material, phenomenon and energy. The gaseous interior materials of the lab-scale converter have the purposes of "acting as reactant", "being adsorbed" and "being inert". There are also solid materials with the purpose of "supporting the converter" and "serving as catalytically active material". One proposed set of phenomena occurring in the lab-scale converter is NO reduction (Maunula *et al.* 1997):

- NO and H_2 adsorption and dissociation
  (*disaggregated into*: $NO+{*}\leftrightarrow NO{*}$, $H_2+2{*}\leftrightarrow 2H{*}$, $NO{*}+{*}\rightarrow N{*}+O{*}$)
- NO reduction via N_2O route
  (*disaggregated into*: $NO{*}+N{*}\rightarrow N_2O{*}+{*}$, $N_2O{*}+{*}\rightarrow N_2{*}+O{*}$, $N_2{*}\rightarrow N_2+{*}$, $N_2O{*}\leftrightarrow N_2O+{*}$)
- Regeneration of oxygen-covered sites
  (*disaggregated into*: $H{*}+O{*}\rightarrow OH{*}+{*}$, $H{*}+OH{*}\rightarrow H_2O{*}+{*}$, $H_2O{*}\rightarrow H_2O+{*}$)

The structural presentation of the lab-scale converter and the instantiation of a few mechanistic steps are shown in Figure 20. Of course, each instance has also a purpose expression, as described above, but not shown in the figure. A more detailed structural description is given for interior material in the lab-scale converter: "flowing gas", "stagnate gas", "porous catalyst", "active sites on catalyst surface", "noble metal on catalyst", and "covered active sites" on catalyst material. This topological description of material was instantiated into ME+ and linked to the graphical representations shown in Figure 19. The topological disaggregation of the converter was originally drawn-up by Pohjola (1995). This kind of material disaggregation originates from the observed and assumed structural characteristics in the studied catalyst material. The structural description, in turn, results in the model structure as described in Figures 21 and 22.



Name: Lab-scale converter

Object purpose: Product and feed specification
Functional specification

Object structure

Object topology: level 1 topology
level 2 topology
level 3 topology

Fixed: yes

Object unit structu: version 1

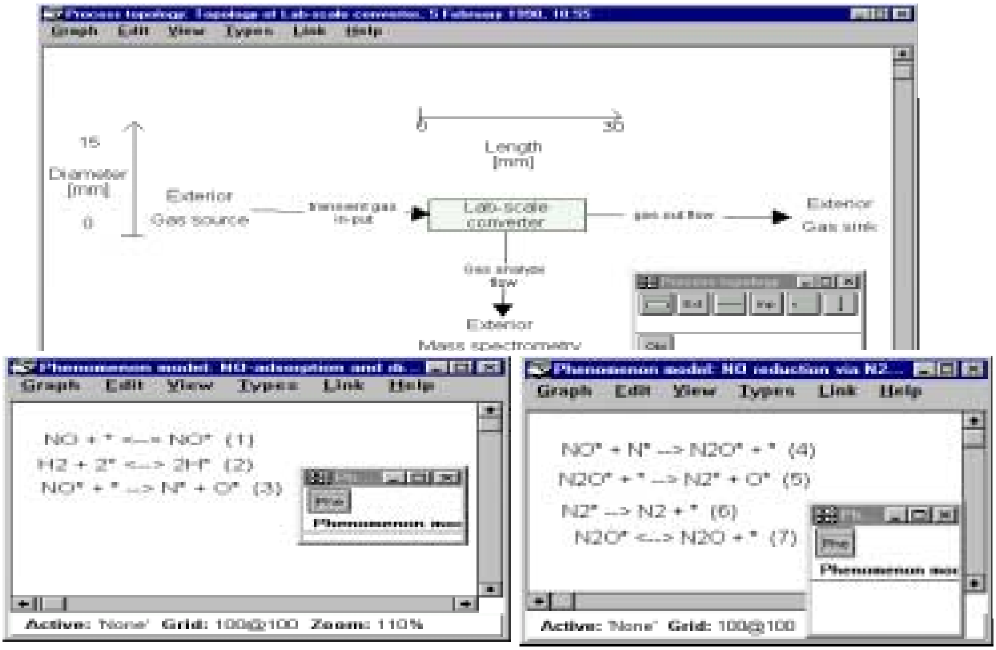
Object state

Object property: c(NO2)
c(CO)
T
p

Mathematical relati: state relation

Object performance: In respect of data collection: good

Figure 19. Topological representation of lab-scale converter viewed through the PDPD tool and through example external presentations.



NO + * <--> NO* (1)
H2 + 2* <--> 2H* (2)
NO* + * --> N* + O* (3)

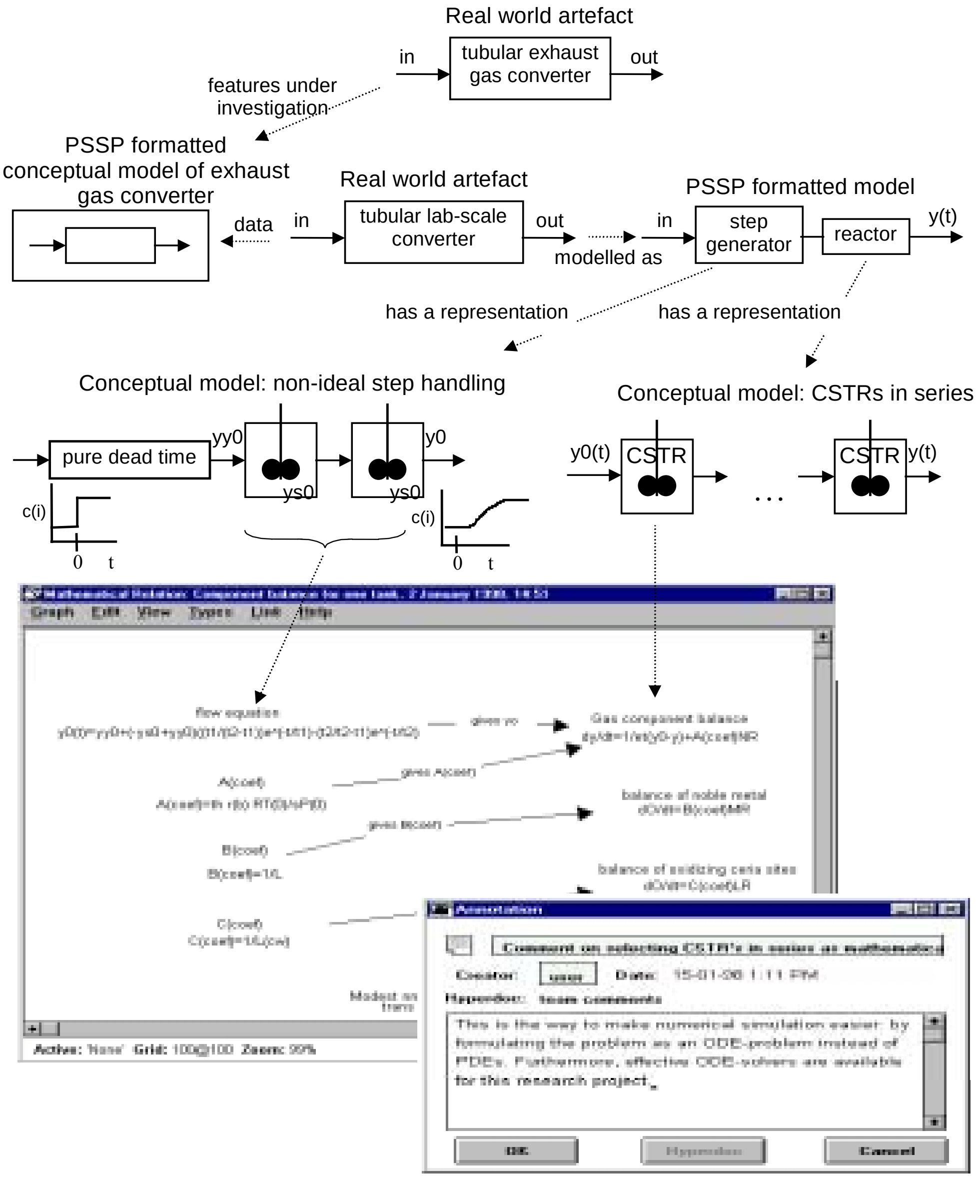
NO* + N* --> N2O* + * (4)
N2O* + * --> N2* + O* (5)
N2* --> N2 + * (6)
N2O* <--> N2O + * (7)

Figure 20. The structure of the lab-scale converter (left) and part of the reaction mechanism for automobile exhaust gas catalysis (right) edited in MetaEdit+.

The model of lab-scale converter, which in turn is a model of reality, is disaggregated into sub-processes: "step generator" (representation for the valve system of lab-scale converter) and "reactor" (representation for the lab-scale tubular converter). The interior material of the "step generator" is disaggregated by using imaginary boundaries in order to create a special modelling view, i.e. modelling the non-ideality of step inputs. Also the interior material of the "reactor" is disaggregated by using imaginary boundaries for creating a special modelling view, i.e. modelling the tubular reactor as CSTRs in series. The purpose of the CSTRs in series model is to make the numerical calculations easier by spatial discretisation of the Partial Differential Equation (PDE) system into Ordinary Differential Equations (ODEs). The purpose of the non-ideal step model is to transfer the ideal step to a real generated input, which is to be combined with tanks in the series model. These special models are shown in Figures 21 and 22.

Mathematical models – representation with PSSP

Mathematical modelling and model validation achieves the goal for this particular project: "kinetic model for a real exhaust gas converter, possibly to be used in the design of a real exhaust gas converter". PDPD, as well as MetaEdit+, does not provide any numerical solvers. However, the expression-editor built in MetaEdit+ was applied for expressing the purposes, symbolical representation, topological structure and links to external software for the mathematical relations (see Figure 21). The actual numerical parameter estimation and the model validation by simulations are realised in an external application, MODEST® (1994). The parameter estimation affects the data consistency of the simulations for the gas component balances. In MetaEdit+, such effects can be reported automatically by the query: "report affected externals". The MetaEdit query language was also applied to generate automatically part of an executable MODEST input-file from the PDPD knowledge-base. However, FORTRAN coding of state equations of the reactor model in a suitable form for numerical calculations was not fully automated. Figure 21 shows a view of the model hierarchy in the project. The kinetic model for the phenomena occurring in the lab-scale converter is simulated with respect to experimental data. After that, the contribution of the kinetic model to the project performance can be evaluated and a decision on how to proceed with the project can be made.



Comment on selecting CSTR's in series as mathematica

Creator: user Date: 15-01-98 1:11 PM

Appendix: team comments

This is the way to make numerical simulation easier: by formulating the problem as an ODE-problem instead of PDEs. Furthermore, effective ODE-solvers are available for this research project.

Figure 21. A model hierarchy and part of the ME+ documentation.

The described modelling technique results in a system of mathematical representation, as depicted in Figures 21 and 22. The hierarchy of the mathematical models can be captured with PSSP language, which is expected to make the models more comprehensive by integrating the higher-level expectations of the model, the constraints assigned and the techniques used.

Figure 22 is a more detailed description of the mathematical representation of the PSSP-formatted conceptual model of the lab-scale converter. The set of equations and all the individual equations can be viewed through the P, S, S and P attributes.

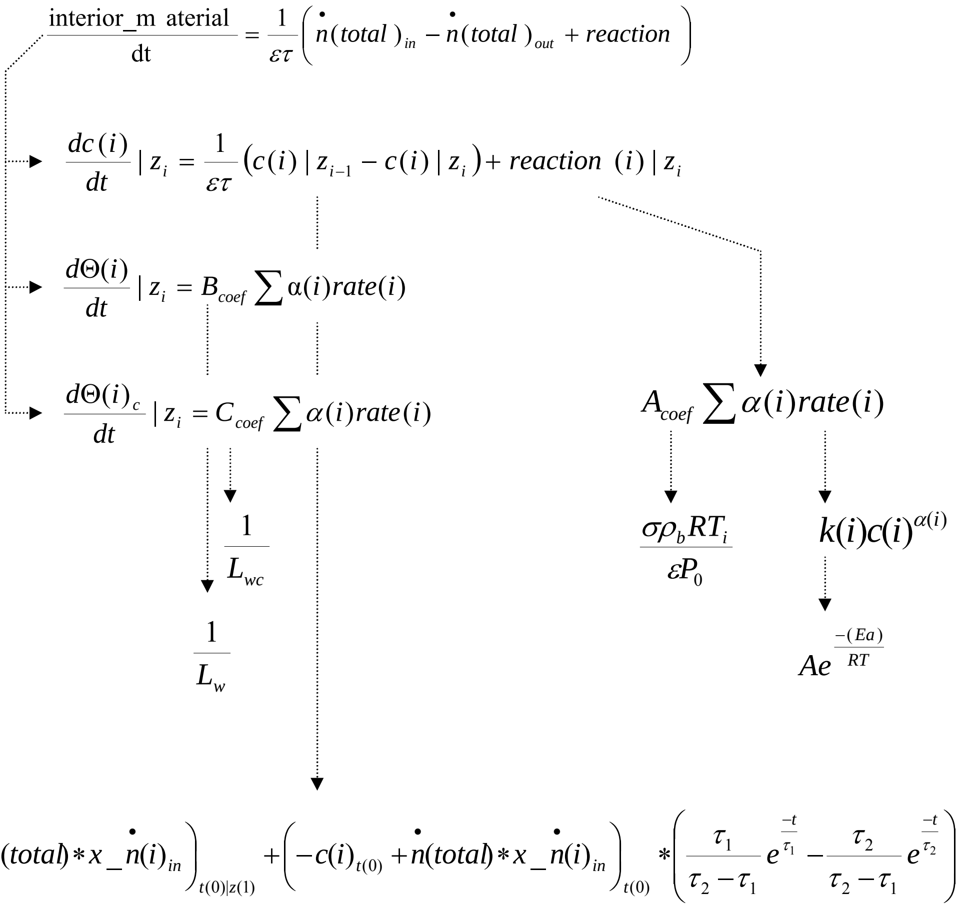


Figure 22. Model hierarchy of the mathematical model describing the target system.

In Figure 22, *t* is continuous time space and $|z_i$ is discrete space co-ordinate (converter is discretised into, *e.g.* four and eight CSTRs in series; 27mm: 4 and 27mm: 8), *c(i)* denotes species and *reaction(i)* denotes the proposed mechanistic steps of overall catalysis. T_0 is 30°C, 90°C, 125°C, 160°C and 240°C. *c(i)|z(out)* is calculated form the model and referred to the measured data from the output of the lab-scale converter. The measurement is performed with an on-line massspectrometer with a time interval of 500ms. The numerical values of the model parameters (k_i and B_{coef} and C_{coef}) are sought by fitting (with MODEST) the model with the measured data.

Project state

Part of the project state – state of the created process knowledge – can be defined by aggregating the performances of the process models. Also the modelling work involved is taken under investigation simultaneously with the process models. The performance of the models results in the following aggregated model performance: "Few feasible models have been achieved for some catalysed reactions (CO oxidation), explaining the kinetic behaviour". "These models can be utilised for process design purposes". The project activities have been finished.

Project performance

Evaluating the project performance does not play any special role for the purposes of this thesis. However, the overall evaluation of the project is "good"; a more detailed and argued performance assessment can be given and formalised in accordance of PSSP and captured with the prototype tool.

7.2 R&D project on chemical water treatment process

The piece of reality to be modelled in the second case is an industrial process development project at Kemira Chemicals Oy. The case project is titled "Use of fluid dynamics in the R&D of water treatment technologies" – simply called "MIXI". This project has its own purpose and goal definitions, but it also has another purpose definition in respect of this thesis. The case project has advanced independently from this thesis, but the author has monitored it in real time for acquiring data on the management decisions, the knowledge generated and the rationale behind them. The monitoring was performed together with the project group and the author (see Pasanen *et al.* 1999a). The data is being used for modelling the project as an instance of PDPD, and for testing the applicability of the methodology.

7.2.1 Purpose of the case study

The project group comprises different backgrounds and its members reside at different locations in Finland and Sweden. The project is being run by adhering to an established good practice rather than to an explicit methodology. Informality is favoured for giving space to creativity and because the prevailing management practices do not support formal knowledge representation. Communication between the project group and the management takes place via e-mail, phone and meetings. Documentation is in the form of textual reports, mainly without any specified format.

In respect of this thesis, the case study is designed to test and demonstrate how to use PDPD principles in the case R&D project. This case also demonstrates how to formalise the project into PSSP language and how the usage of the PDPD and the prototype tool affect the prevailing practice of the case R&D project. Because the project has advanced independently from this thesis, it is possible to compare the realised project management and the proposed PDPD management actions. The prototype tool (developed with ME+ version 3.0) has been used for documenting the management decisions and R&D data. The capabilities of the tool have been evaluated during the project.

7.2.2 PDPD guidance and PSSP formalisation

Starting of the project

When starting a project, PDPD requires the project management to set the purpose of the project, and to document it into the project object. The PDPD tool asks for specification of the purpose in terms of (1) project product specification (ultimate goal), (2) project functional specification, and (3) evaluation criteria of the project performance. The corresponding part of the design cycle is shown in Figure 23. Some specifications of the MIXI-project are shown in Figure 24. An example specification with the PDPD tool is given in Figure 25.

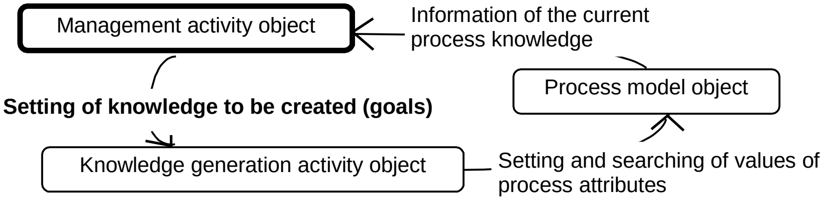
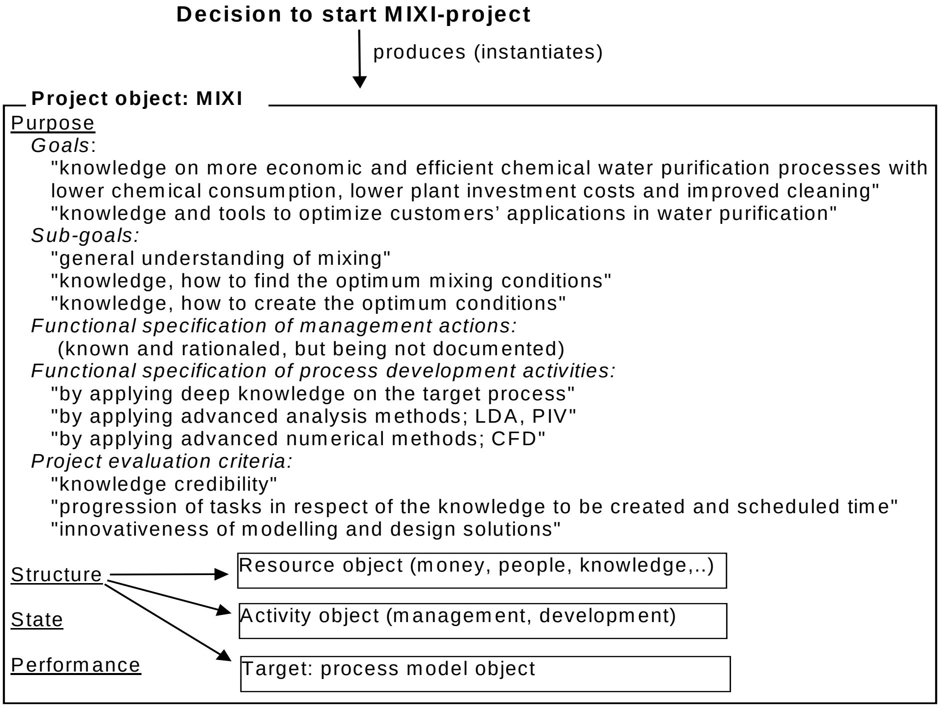


Figure 23. Corresponding part of the design cycle when initiating a project.



Decision to start MIXI-project

produces (instantiates)

Project object: MIXI

Purpose

Goals:

"knowledge on more economic and efficient chemical water purification processes with lower chemical consumption, lower plant investment costs and improved cleaning"

"knowledge and tools to optimize customers' applications in water purification"

Sub-goals:

"general understanding of mixing"

"knowledge, how to find the optimum mixing conditions"

"knowledge, how to create the optimum conditions"

Functional specification of management actions:

(known and rationaled, but being not documented)

Functional specification of process development activities:

"by applying deep knowledge on the target process"

"by applying advanced analysis methods; LDA, PIV"

"by applying advanced numerical methods; CFD"

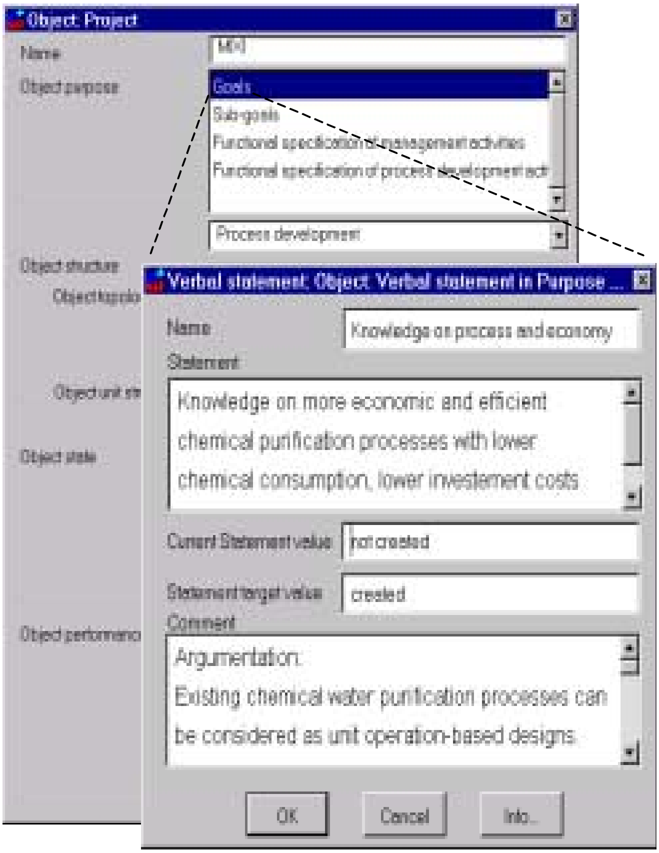
Project evaluation criteria:

"knowledge credibility"

"progression of tasks in respect of the knowledge to be created and scheduled time"

"innovativeness of modelling and design solutions"

Figure 24. Purpose specifications of MIXI-project.



Object: Project

Name: MIXI

Object purpose: Goals

Sub-goals

Functional specification of management activities

Functional specification of process development act

Process development

Verbal statement: Object: Verbal statement in Purpose ...

Name: Knowledge on process and economy

Statement: Knowledge on more economic and efficient chemical purification processes with lower chemical consumption, lower investement costs

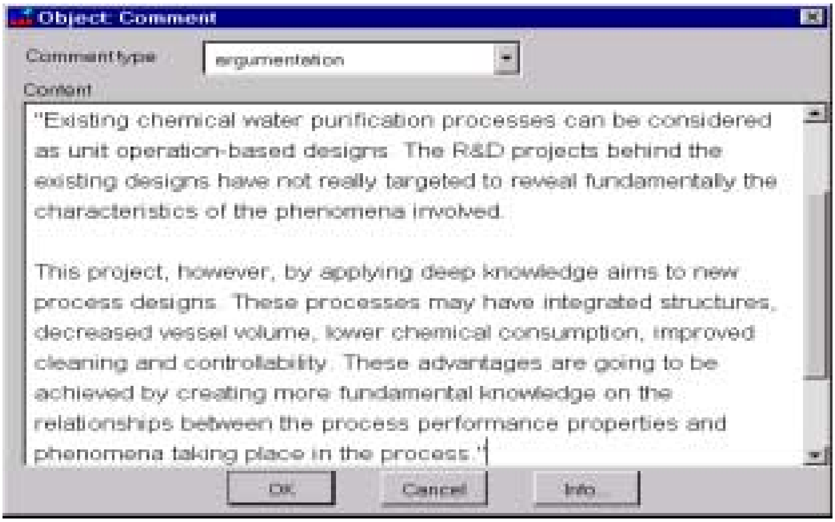
Current Statement value: not created

Statement target value: created

Comment: Argumentation: Existing chemical water purification processes can be considered as unit operation-based designs

Figure 25. A purpose specification viewed through the PDPD tool.

Process developers and project managers are urged to document any rationale behind a specification. The location of a rationale, as a relationship or a statement, is under the structure attribute of the project management or process development activity object in question. The management rationale behind the decision to apply deep knowledge in the MIXI-project is shown in Figure 26.



Commenttype: argumentation

Content

"Existing chemical water purification processes can be considered as unit operation-based designs. The R&D projects behind the existing designs have not really targeted to reveal fundamentally the characteristics of the phenomena involved.

This project, however, by applying deep knowledge aims to new process designs. These processes may have integrated structures, decreased vessel volume, lower chemical consumption, improved cleaning and controllability. These advantages are going to be achieved by creating more fundamental knowledge on the relationships between the process performance properties and phenomena taking place in the process."

Figure 26. An example of management rationale behind a goal specification.

Purpose definitions of the MIXI-project result in the initial settings of the process model to be developed and the model for the process model – called process metamodel. Furthermore, the existing processes and the past development projects serve as a reference for the knowledge to be created. The initial process model contains product and feed specifications, and the selected technology as well. The evaluation criteria of the process to be developed are bound to the performance of the existing processes. In Figure 27 provides a simplified view of an existing chemical water treatment process, and in Figure 28 there is a view of the initial setting of the process model to be developed.

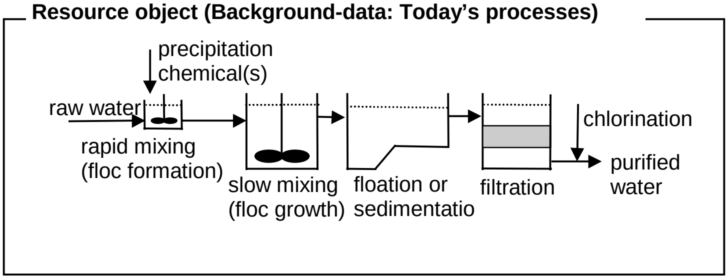
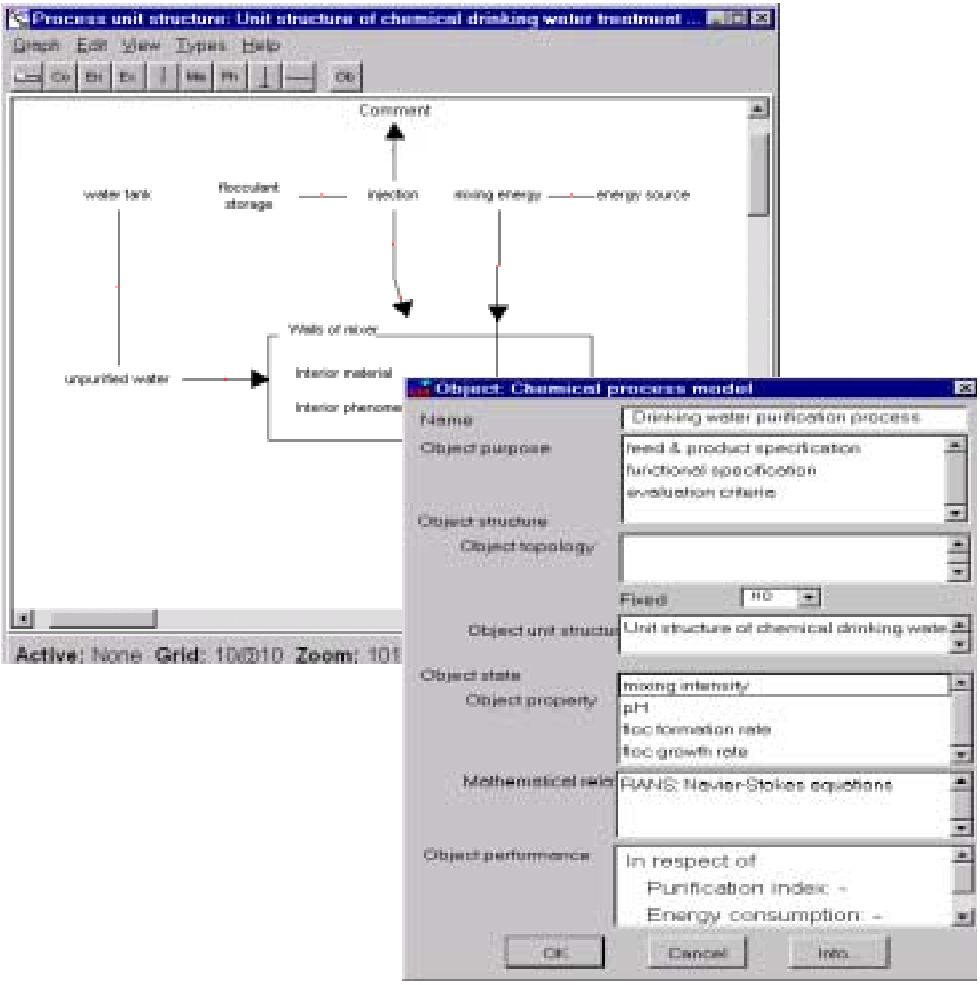


Figure 27. A knowledge resource object of the MIXI-project.



Name	Drinking water purification process
Object purpose	feed & product specification functional specification evaluation criteria
Object structure	
Object topology	
Fixed	no
Object unit structu	Unit structure of chemical drinking wate
Object state	
Object property	mixing intensity pH floc formation rate floc growth rate
Mathematical rela	RANS: Navier-Stokes equations
Object performance	In respect of Purification index: - Energy consumption: -

Figure 28. Initial setting of the process to be developed.

Process metamodel contains the expectations and formalism requirements of the process model to be developed. More accurate and detailed project goals have been specified, more accurate metamodels can be given. Development activities are scheduled, started and they are being continuously evaluated, based on the defined metamodels and goals. An early stage metamodel of the MIXI-project is shown in Figure 29. In the MIXI-project there are data that are not fully documented, for example, knowledge of the model credibility and model evaluation. However, PDPD would have provided a platform for full documentation, as shown in Figure 30.

Process metamodel object

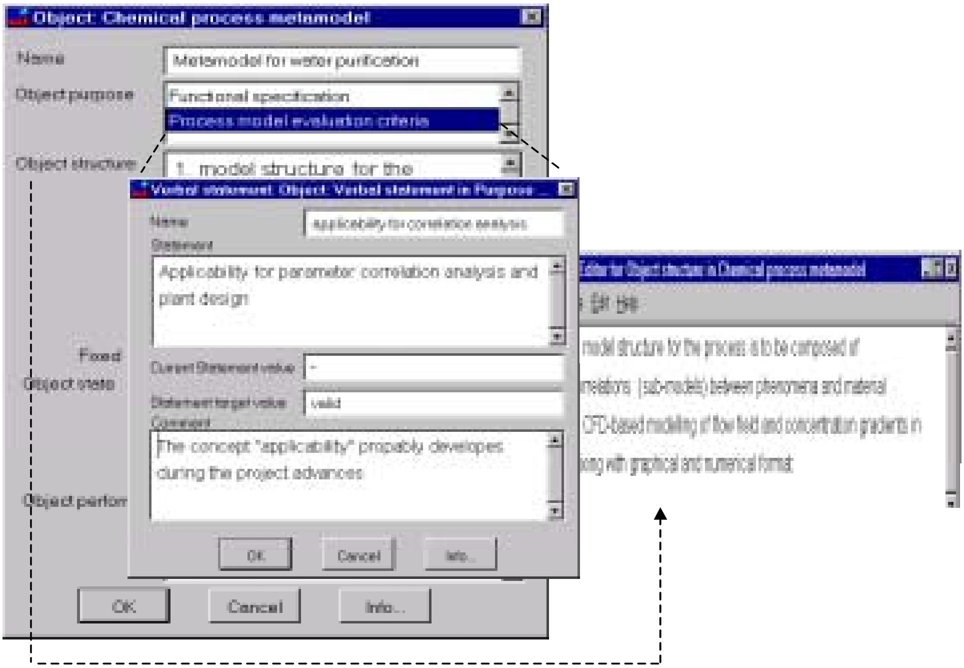
Process model Purpose (process model expectations)
  Functional specification:
    "enough knowledge (relations) to find the optimum state-distribution of the phenomena involved in mixing"
    "knowledge on feasibility assessment of process integration"
    "enough knowledge for plant design"
Process model evaluation criteria:
    "applicability for parameter correlation analysis and plant design"
Structure:
  "model structure for the process is to be composed of correlations (sub-models) between phenomena and material"
  "CFD-based modelling of flow field and concentration gradients in mixing with graphical and numerical format"

Figure 29. An early stage process metamodel of the MIXI-project.

Project Activities

The purpose of the project and the requirements of the process, as well as the process model, have been defined and formatted into PSSP language. Next, PDPD requests project activities (or tasks) to be started for producing the required knowledge. The corresponding part of the design cycle is shown in Figure 31. The way to carry out the tasks is dependent on the priority of the goals and on the available resources.

In the MIXI-project, three concurrent sets of tasks have been started to achieve the goals. The purpose of the first set of tasks is to specify the initial process structure and to create analysis methods needed in further actions. The purpose of the second set of tasks is to produce data and knowledge on the thermodynamic feasibility by creating relationships between the phenomena and material involved. The third set of tasks utilises the produced state-relations and produces knowledge of the technical feasibility. Each of the sets divides into various sub-tasks. The activity sets, and the data-flows between them, are shown in Figure 32 – the structure of MIXI-project.



Name: Metamodel for water purification

Object purpose: Functional specification

Process model evaluation criteria

Object structure: 1. model structure for the

Fixed

Object state

Object perfor

Name: applicability for correlation analysis

Statement: Applicability for parameter correlation analysis and plant design

Current Statement value: -

Statement target value: valid

Comment: The concept "applicability" propably developes during the project advances

model structure for the process is to be composed of

relations (sub-models) between phenomena and material

CFD-based modelling of flow field and concentration gradients in

ong with graphical and numerical format

Figure 30. An example process metamodel specifications with the PDPD tool.

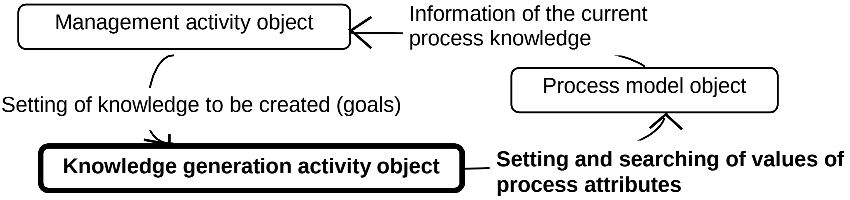


Figure 31. Corresponding part of design cycle for project activities.

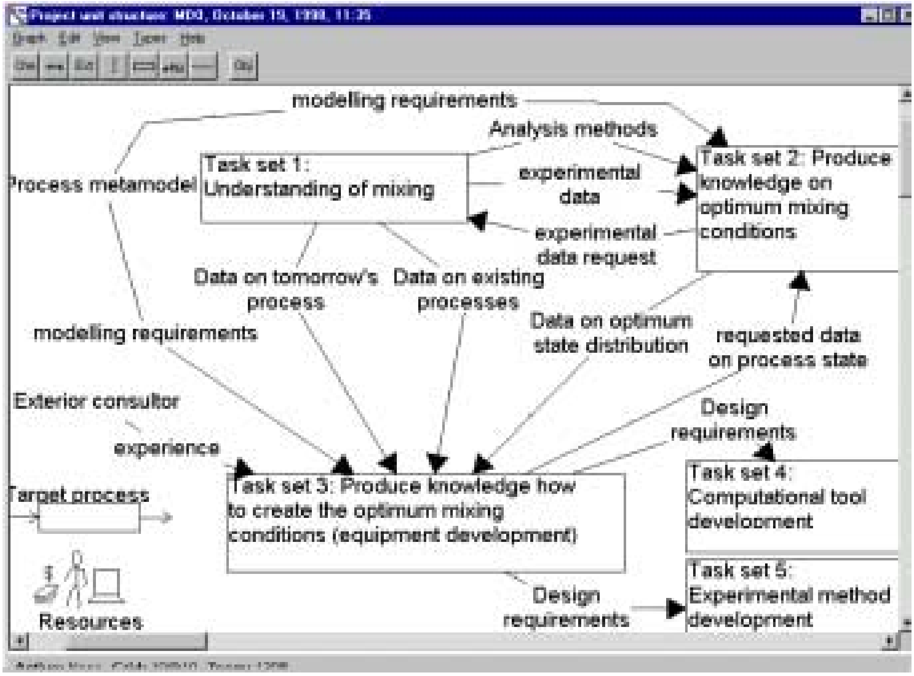


Figure 32. Activity objects and data-flows viewed via the PDPD tool (scheduling and temporal links are filtered).

Process model and knowledge

Process model development and knowledge creation are the result of the project activities and background data. The corresponding part of the design cycle for process knowledge is shown in Figure 33. At this stage, the process model is composed of necessary and auxiliary material and phenomena, and the relationships between thermodynamic variables. Yet, the process model is not associated with any specific process structure or equipment. A view of how part of the data and knowledge is transformed into objects of PDPD is shown in Figure 34.

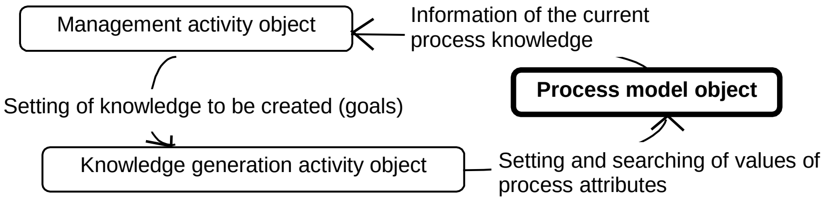
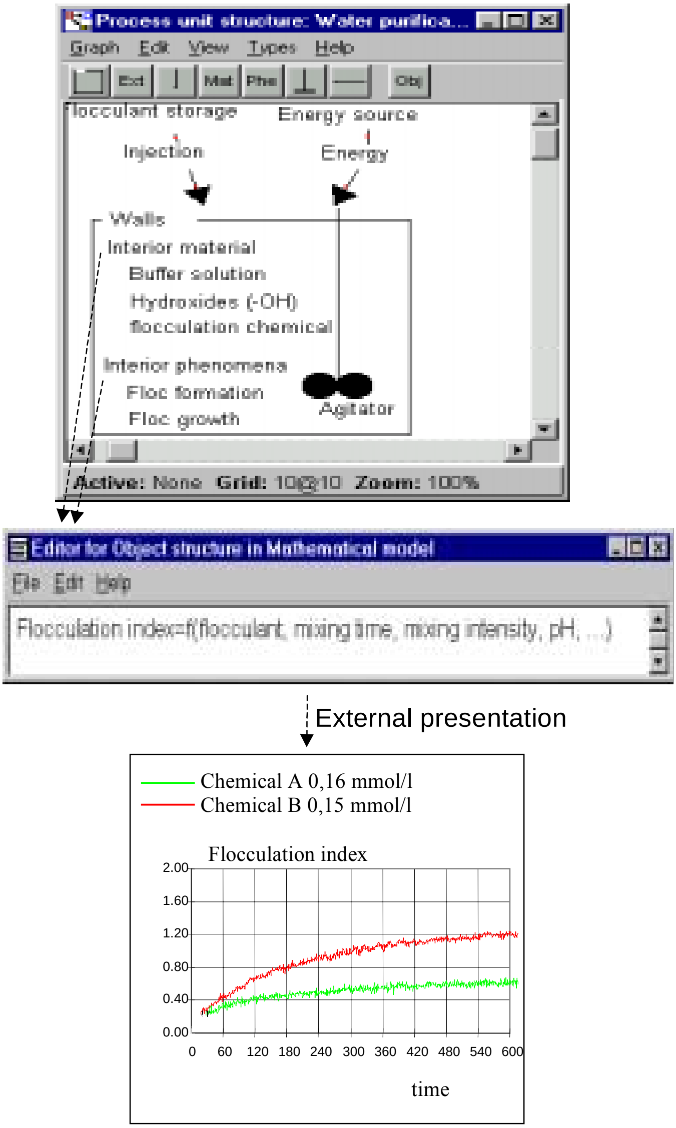


Figure 33. Corresponding part of the design cycle for project activities.

Evaluation of process model and the project

Now, the goals, knowledge created, states and performances of the process models, tasks, and the whole project are evaluated. Figure 35 shows the corresponding part of the design cycle for information exchange when making the project evaluation. The knowledge evaluation is performed by aggregating the states and performances of the structural objects of the process model. This aggregation procedure is started from the low-level objects, ending up with performance of the whole process model. The aggregation of the states and performances is done both quantitatively by using mathematical relations, and qualitatively without making mathematical relationships between the objects. An example aggregation of the process model in the MIXI-project with the PDPD tool is shown in Figure 36.



Flocculation index=f(flocculant, mixing time, mixing intensity, pH, ...)

Figure 34. A process model object.

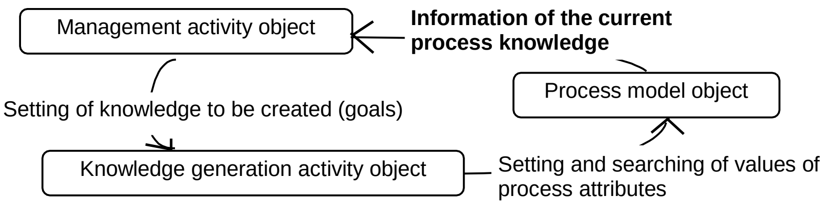
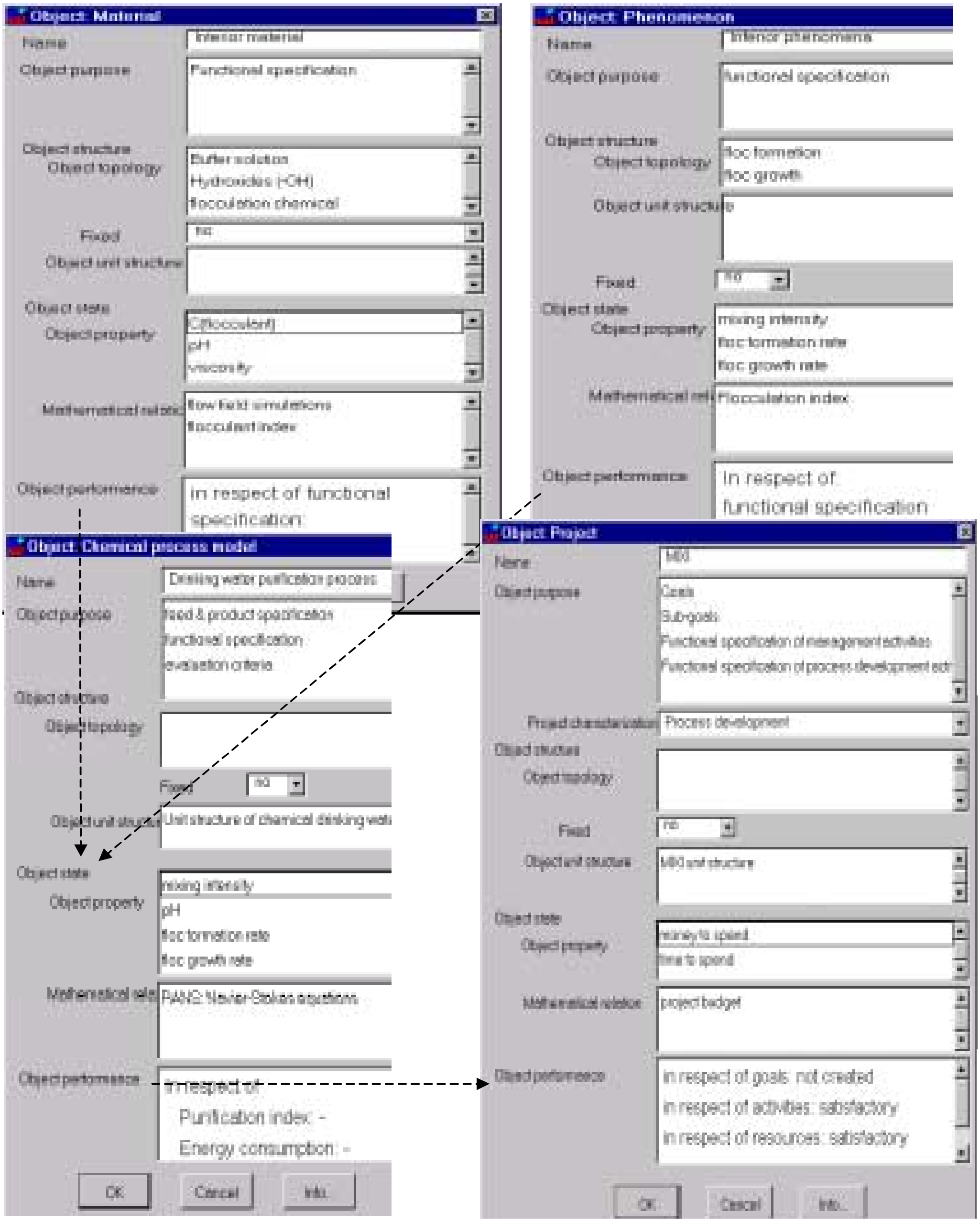


Figure 35. Corresponding part of the design cycle when making evaluation of the project.

Project continuation

After the evaluation, decisions on project continuation can be made: whether to stop the project, add new tasks, or modify the resources or the purpose of the project. In respect of the design cycle, this means that the cycle is closed and the second cycle is started (see Figure 23). In the MIXI-project, the comparison of the process model with the project goals and process metamodel results in new tasks. For example, "knowledge on optimum mixing conditions" is divided into further sub-goals: "effect of mixing intensity, dosing and geometry on flocculation". These sub-goals are produced by a new task: "Flow field simulations". A view of this task and process model is given in Figure 37.



Object: Material

Name: Interior material

Object purpose: Functional specification

Object structure

Object topology: Buffer solution; Hydroxides (-OH); flocculation chemical

Fixed: no

Object unit structure:

Object state

Object property: C(flocculent); pH; viscosity

Mathematical relatio: flow field simulations; flocculent index

Object performance: in respect of functional specification:

Object: Phenomenon

Name: Interior phenomena

Object purpose: functional specification

Object structure

Object topology: floc formation; floc growth

Object unit structure:

Fixed: no

Object state

Object property: mixing intensity; floc formation rate; floc growth rate

Mathematical rel: Flocculation index

Object performance: In respect of functional specification

Object: Chemical process model

Name: Drinking water purification process

Object purpose: feed & product specification; functional specification; evaluation criteria

Object structure

Object topology:

Fixed: no

Object unit structu: Unit structure of chemical drinking wate

Object state

Object property: mixing intensity; pH; floc formation rate; floc growth rate

Mathematical rela: RANS: Navier-Stokes equations

Object performance: In respect of
Purification index: -
Energy consumption: -

Object: Project

Name: MIX

Object purpose: Goals; Sub-goals; Functional specification of management activities; Functional specification of process development act

Project characterisatio: Process development

Object structure

Object topology:

Fixed: no

Object unit structure: MIX unit structure

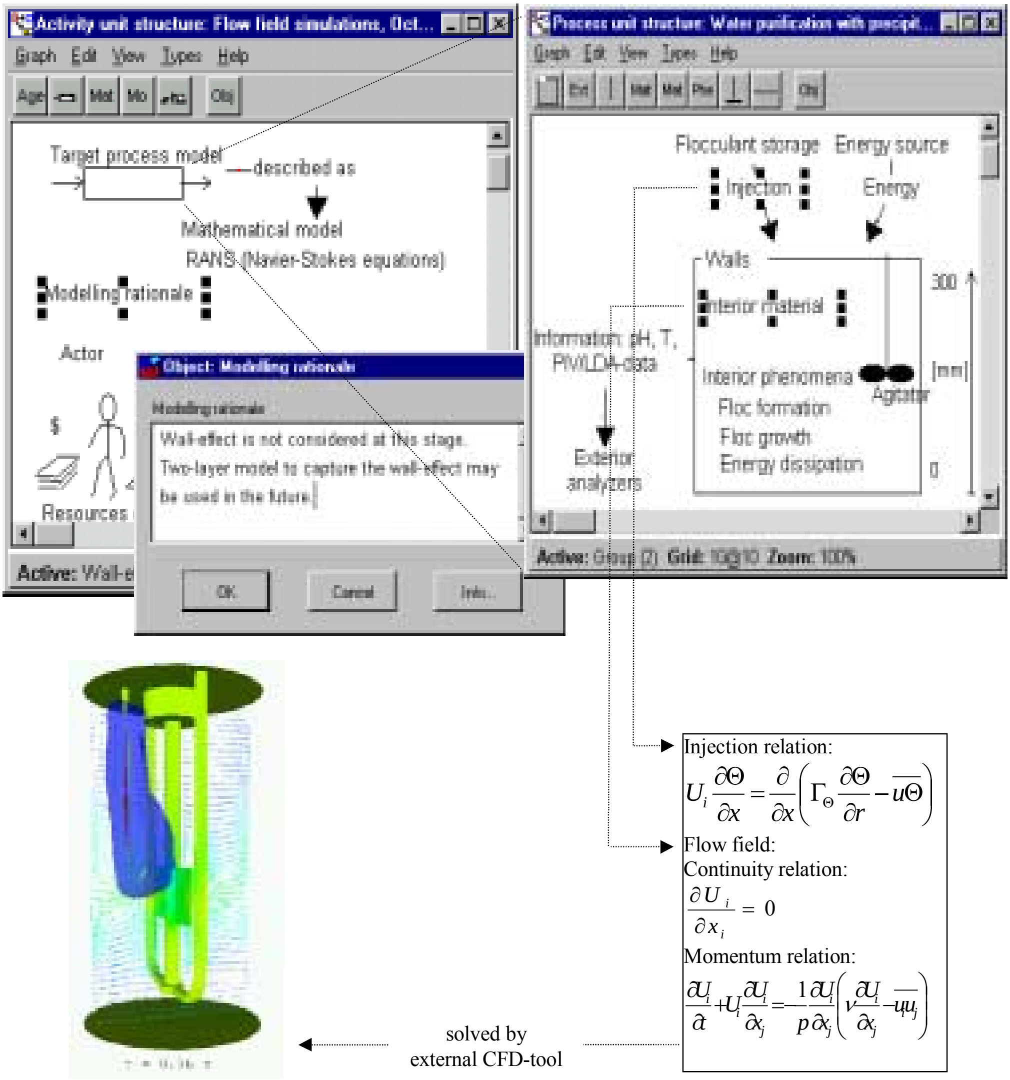
Object state

Object property: money to spend; time to spend

Mathematical relation: project budget

Object performance: In respect of goals: not created; In respect of activities: satisfactory; In respect of resources: satisfactory

Figure 36. Process model aggregation.



Wall-effect is not considered at this stage. Two-layer model to capture the wall-effect may be used in the future.

Injection relation:

$$U_i \frac{\partial \Theta}{\partial x} = \frac{\partial}{\partial x}\left(\Gamma_\Theta \frac{\partial \Theta}{\partial r} - \overline{u\Theta}\right)$$

Flow field:

Continuity relation:

$$\frac{\partial U_i}{\partial x_i} = 0$$

Momentum relation:

$$\frac{\partial U_i}{\partial t} + U_i \frac{\partial U_i}{\partial x_j} = -\frac{1}{p}\frac{\partial U_i}{\partial x_j}\left(\nu \frac{\partial U_i}{\partial x_j} - \overline{u_i u_j}\right)$$

Figure 37. Modelling rationale, process model and a few relations linked with external tools.

7.3 Modelling and simulation of new chemical pulping process

The third case study is based on a research and development project aimed at commercialising a non-wood pulping process. This ambiguous development project is run by a Finnish company, and the project has its own purpose and goal definitions, but it also has another purpose definition in respect of this thesis. The case project has advanced independently from this thesis, but the author has harnessed the project also to demonstrate the features of PDPD and PSSP language. The case study has resulted in a fully functioning integration between a PDPD platform and a mathematical solver – between MetaEdit+ and Mathematica® (Wolfram 1988).

7.3.1 Purpose of the case study

Non-routine modelling work in this case project has placed special requirements on the modelling tools. The case project has not found a proper commercial simulation program that would be easy to apply with the available resources and data. Thus, an in-house software package is applied and a set of special experiments for generating the model parameters has been performed in order to make accurate process simulations.

However, some practical difficulties have occurred during the modelling work. The case project has resulted in numerous sets of simulations, and the way the process is simulated requires compiling numerous matrices and lists manually for the simulator (Mathematica), which is time-consuming and error prone. In addition,keeping track of the sketched process alternatives, various lists and matrices, and results of the simulator is time-consuming.

Now, the PDPD tool is expected to enhance the model building during the sketching of the process alternatives. Moreover, the PDPD tool provides a higher-level platform for the conceptual modelling work. The conceptual modelling elements can be used for expressing the building blocks needed by the external solver. In other words, the purpose of PDPD in this case is "to provide an easy and automatic data integration between the graphical editors of PSSP-formatted process models in ME+ and the external solver (Mathematica)". This purpose has also encouraged setting a more generic purpose: "to test and demonstrate that PDPD and PSSP language can be implemented with ME+ in such a way that it providesa fairly good, high-level integration platform for any external solver".

7.3.2 PSSP formalisation of process models and usage of an external tool

For this case the relevant objects are process models, mathematical relations and external solver, which are documented and operated via ME+. The project object is now intentionally skipped, even though the main features – the purpose and modelling activities – of the project object are shown and documented with ME+.

Background data under 'project resources'
The conventional sulphate pulping process cannot utilise non-wood materials in a way that the process is economically and environmentally feasible. Since 1995, Chempolis Oy has been developing a chemical pulping process, based on formic acid, which can use non-wood plants as raw material. The process can be used to produce fully bleached pulp from non-wood materials, and the pulp can be used, for example, in the manufacturing of fine papers. A simplified flowsheet of the process is shown in Figure 38.

Project purpose
One 'goal' of this case project is the "knowledge of the optimum structure producing the optimum process performance". The goal can be specified further as "knowledge of compositions and fluxes of the process flows for process development purposes".

Modelling activities
The goal of the project is to be realised by the experimental and modelling activities: "batch experiments" and "simulation of process alternatives". These activities can be viewed through the 'design cycle' (see Figure 9 in Section 5.1.2) that has been applied in Figure 39 to describe the role of a simulator tool during "simulation of process alternatives".

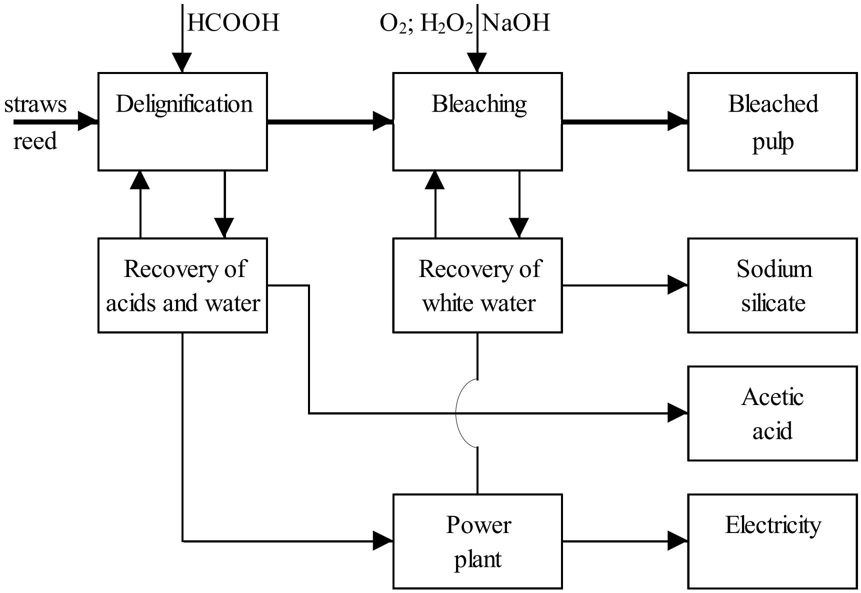


Figure 38. A simplified flowsheet of the process (Rousu 1997).

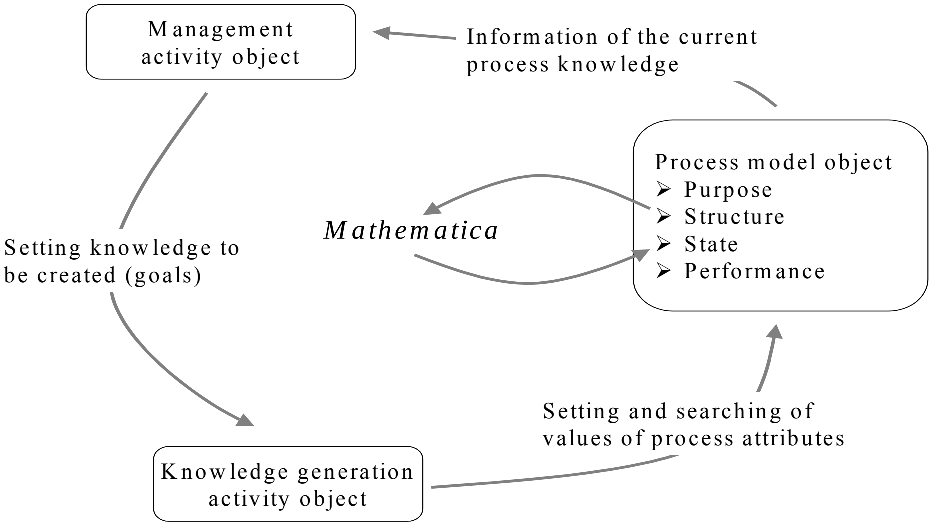


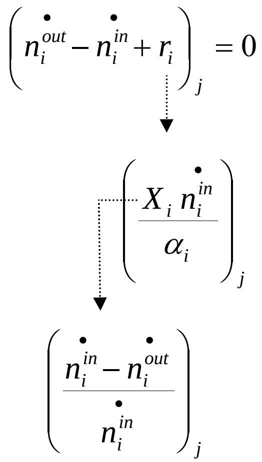
Figure 39. Design cycle according to PDPD and the role of Mathematica®.

Process models

Experiments and modelling produce data and knowledge about a process, which is not yet constructed. In this case, the purpose of the structural process models is to enable mass balance calculations for seeking the optimum structure of the process. Hence, it has been decided to use a steady-state material balance to model the process state of numerous topological alternatives. Calculating the steady-state material balances is based on the general balance for each process unit j:

$$\left(\frac{d\dot{n}_i}{dt}\right)_j = \left(\dot{n}_i^{out} - \dot{n}_i^{in} + r_i\right)_j = 0$$

In this research work, the balance equations are formed by assigning the known flows and by assigning the chemical reactions for the appropriate units. The contribution of the reactions is modelled based on conversion. Defining the conversions in units, in turn, is based on batch experiments in a proper thermodynamic region (p, T, pH, c_i and residence time). The chemical reactions that occur in the units are not fully known, and therefore the reactions are modelled based on conversion. Furthermore, thermodynamic data for the components and species are not available, which is now eliminated by a set of batch experiments producing conversion data. Defining the stoichiometric coefficients of the reactions is performed by fitting the data from the batch experiments into the model. It is most probable that this kind of modelling technique of sketching process alternatives results in ill-defined problems, which are cumbersome to solve with commercial simulation tools, *e.g.* Aspen Plus® or Wingems® (known to project staff). This can be reasoned by the fact that the topological structures, and the way the reactions have been modelled, result in a non-linear set of equations (in this case a few hundred), which does not fully converge numerically. Thus, a symbolic calculation tool – Mathematica – has been applied. The mathematical model is shown in Figure 40.



$$\left(\dot{n}_i^{out} - \dot{n}_i^{in} + r_i \right)_j = 0$$

$$\left(\frac{X_i \, \dot{n}_i^{in}}{\alpha_i} \right)_j$$

$$\left(\frac{\dot{n}_i^{in} - \dot{n}_i^{out}}{\dot{n}_i^{in}} \right)_j$$

Figure 40. Hierarchy of the mathematical model for mass balance calculations used in the project.

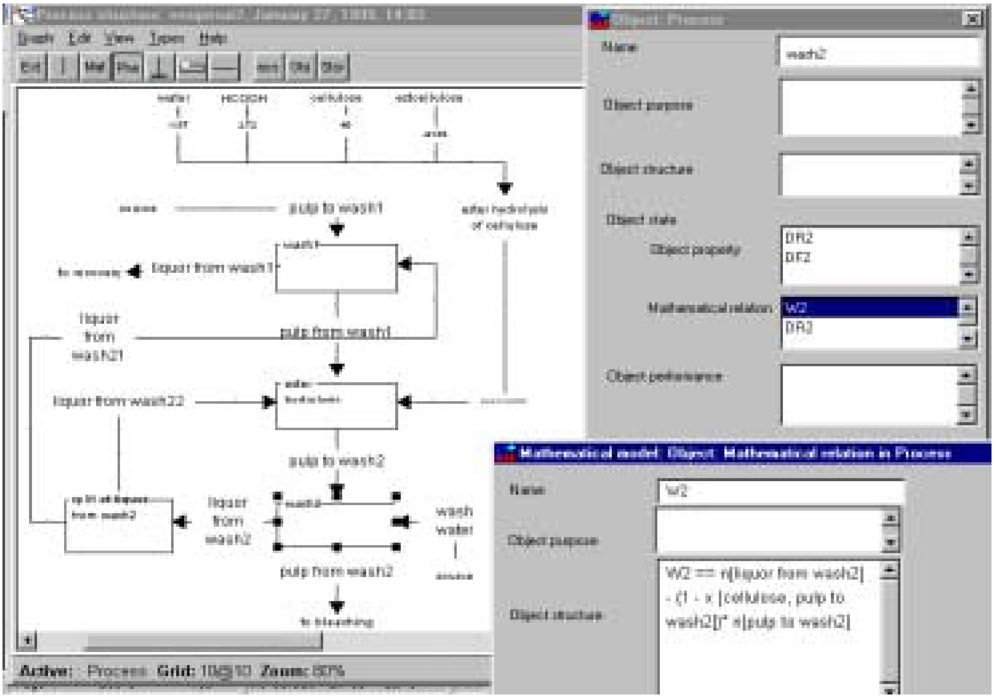
PDPD tool integrated with Mathematica®

Mathematica is a general system for performing mathematical computations and it can be used in many different ways. One way to use it is as a "calculator", but it can be used as a programming environment as well. Mathematica is capable of symbolic manipulation, which is a great advantage when problems are under-defined, to obtain a numerical solution. The process structure was described as sets of lists, matrices, relations and assignments to describe the flows and process units. An in-house program was compiled with Mathematica, which generates mass balance equations – see model equations above – and solves them. Originally, the matrices and lists were set-up manually, which was time-consuming and error prone. In addition, keeping track of the sketched process alternatives, lists, matrices and resulting mass balances, was time-consuming. The prototyped PDPD tool was expected to enhance the Mathematica formalisation and to manage the data of the process model alternatives.

The PDPD tool was applied for sketching the process structural alternatives and setting up the relations between the units and flows. From this graphical model a query[49] of the PDPD tool automatically generates the input for the in-house Mathematica program. PDPD also makes it easier to keep track of the numerous process structural alternatives and numerical simulations involved.

---

[49] Appendix 11



Name: wash2
Object purpose
Object structure
Object state
Object property: DR2, DF2
Mathematical relation: W2, DR2
Object performance

Name: W2
Object purpose
Object structure: W2 == n[liquor from wash2] - (1 - x [cellulose, pulp to wash2])* n[pulp to wash2]

Figure 41. Model of a part of the chemical pulping process in PDPD-IS and the dialog boxes for one process unit.

To demonstrate the use of the tool, Figure 41 shows a part of the actual formic acid pulping process as modelled in the PDPD tool. In this part, formic acid is removed from pulp by washing with water, and the cellulose esters are hydrolysed between the two washing stages. The left side of the picture shows the topological structure of the process. As an example, the dialogue box in the upper right corner of the picture shows the model of the process unit "wash2", and the box in the lower right corner shows the mathematical model object "W2", which is one of the phenomenological relations describing the state of the process unit. This structure is then transcribed into a Mathematica expression by using the database query of ME+. On the basis of this expression, Mathematica generates mass balance equations and solves them. The final result is thus the compositions and fluxes of the process flows.

7.4 Some ongoing projects

The author utilises PDPD under a project called "Adsorption" where PSSP language is applied for documenting a research project. Another type of platform for PDPD has been developed, based on a www-browser. The browser is used as a documentation and report formaliser of research work, conceptual models (physico-chemical), mathematical models and numerical models and data. Users can edit the intranet database with a commercial www document editor (Netscape 5 and Netscape 5 Composer). This approach is utilised in a project of POSIVA for documenting the modelling work and adsorption models in geochemical systems. The use of PDPD and the www-platform in this project is expected to make the working procedure and the models more comprehensive.

The third case in Section 7.3, "Modelling and simulation of new chemical pulping process" is still ongoing. This project utilises the prototyped tool for further simulations of alternative pulping processes.

7.5 Test usage evaluation

The purpose for the evaluation of the case studies is to document the experiences of the few test cases. The evaluation is made against the set purpose and evaluation criteria of each case. The experiences are collected from people to whom the use of the prototype has been demonstrated during the cases, and who have actually used the prototype. All the cases and the test use have taken more than one year, between 1997–2000. The author has collected the users' experiences, and the opinions and comments of the project groups. Implementation of a prototype tool would be meaningless without making a thorough evaluation of the test usage.

7.5.1 General observations

Software development for knowledge-based systems is usually time-consuming (*e.g.* Tolvanen 1999). Hence, the applied CAME (Computer-Aided Methodology Engineering) tool for simultaneous methodology engineering and test use is evaluated as valuable. The three cases provided fairly good and heterogeneous testing environments. These cases are not used as evidence, but as demonstrations on how to use the methodology and the tool. Furthermore, the cases provided a proper demonstration on a few exact enhancements: 1) making the working procedure more comprehensive (case in Sections 7.1); 2) systematising goal setting (case in Section 7.2), and 3) automating code generation into an in-house solver (case in Section 7.3). However, because chemical engineering modelling work has not been categorised in this work, systematic analysis of the representatives of the cases in respect of prevailing practices cannot be done. Based on the experiences from the cases though, PDPD and the tool are expected to be adopted by other types of modelling work.

In particular, the purpose attribute was considered important in all the case projects. PDPD tool support ensures that all the relevant purpose definitions of project management objects and process-modelling objects are to be documented during R&D projects, which was important in all cases. Another useful feature in PDPD, and in the tool, is the ability to capture the topological dimensions of object structures: activity networks, process networks and mathematical model networks. In all cases, the use of the tool provided a more transparent data documentation than without using PDPD. For example, the tool supports transparent documentation of a metamodel of a process model, specifying the purpose of the model to be developed. This feature was considered very useful. Next, a brief analysis is given of how the capabilities of the PDPD tool address the problems highlighted in Sections 3.1 to 3.5. This analysis is based on the experiences gained from test use.

Need for meta-models

The developed tool supports PDPD-formatted documentation of process models and process metamodels with 'purpose', 'structure', 'state' and 'performance' attributes. This utility is unique among commercial process modelling tools and it enables the documentation of features, such as the purpose and building blocks of a model, which are omitted in most the modelling tools.

Need for activity modelling

The developed tool enables the documentation of activities as objects having 'purpose', 'structure', 'state' and 'performance'. The use of the tool eases the systematisation and analysis of activities by accounting for data flows in between, and by helping to aggregate the state and performance of the tasks in a project.

Need to enhance holistic process modelling

The developed tool improves the building of process models by explicitly including all the real-world features, abstraction phases and conceptual models involved in the development of a mathematical process model. The tool supports the documentation of these models, which are often omitted, and especially uses important attributes. purpose and performance. This kind of model and modelling knowledge documentation is expected to improve comprehension of the model. Furthermore, activity has been taken beside process models. This unique ability enhances activity management and the comprehension of activity networks.

Need for data and activity transparency

Through the object repository of the developed tool, project data and process modelling, data comes automatically structured in accordance with PDPD and PSSP language. The tool can be used as a multi-user system in which the objects are operated and external tools can be associated with PSSP objects. However, multi-user sessions were not arranged; during test use only one user at a time was able to operate the repository in the server system. The open-formatted model and activity documentation are expected to enhance the management and efficiency of working teams.

It was successfully demonstrated in Section 7.3 and in Appendix 11 how PSSP-formatted models could be transformed automatically into a Mathematica program. In addition, PSSP-formatted process models can be transformed as input data for more applied applications. Such transformation is presented in Appendix 9, which generates an ASPEN PLUS input-file from PSSP formatted models. ASPEN PLUS is object-orientedly structured, which can be one reason for the successful code generation of PSSP-formatted – object-oriented – process models from ME+. However, during the research case, presented in Section 7.1, an ME+ query code was compiled to produce semi-automatically a MODEST input-file. MODEST is not an object-orientedly structured simulation and parameter estimation program, and this fact supports the argumentation for the expressive power of PSSP language.

Need for computer tool support for design methodologies

The developed tool provides tool support for PDPD. The development environment is MetaEdit+, which enables simultaneous methodology engineering and test use of the tool under development. This environment is unique and saves time during building and evaluating a PDPD tool support.

7.5.2 Detailed evaluation of the case studies

First case

In respect of the set purpose (Section 7.1.1), this case demonstrates the model comprehension. According to the staff of the case project, the tool especially provides a good platform for putting the necessary and specific data under the project purpose attribute and for the project activity topology.

PSSP language and the tool were evaluated by the project group as follows: 1) "PSSP formalisation provides extra facilities on comprehension of modelling work", and 2) "the tool provides some extra enhancement to model comprehension". The most useful feature of the prototype tool was the activity-modelling elements and the capabilities to link activities with the models involved in accordance with PSSP language. Especially, the purpose setting both for project activities and for modelling elements was considered a fruitful way to monitor and document the project and the models involved. The comprehension of the model, enhanced by the modelling elements of PSSP, was considered useful. However, the more complicated the model, and the larger the project group, the more useful the tool. In larger projects the algorithm helping to report the effects when editing the models may provide true enhancement.

The prototype tool provided support for the activity and model documentation. All the relevant features in accordance with PSSP had representation in the prototyped tool. However, the author considers the user interface a bit complex to use. The prototype version for this case was developed with ME+ 2.5 and ME+ version 3.0.

Second case

The case data – management decisions and the created process model data and knowledge – were evaluated as fitting well with the PDPD principles and PSSP language. According to the successful fit of the case project, the main advantages of the methodology were evaluated together with the project representatives and methodology engineers (Pasanen *et al*. 1999a). Most importantly, the goal-setting in the case project is not in conflict with phenomenon-driven ordering. Actually, PDPD provides a platform, which probably would have accelerated the goal-setting. Usage of PDPD ensures consideration of phenomenon-driven ordering in any project. This type of process modelling and project management supports the creation of new and novel designs, with enhanced creativity. Any process model, resulting from any type of designing, however, can be represented with PSSP-language, which is expected to result in better design documentation. Although the linkage between PDPD formalisation and the data produced by external solvers (spreadsheet data-files and a commercial CFD-tool; CFX®) was not fully tested by a larger user group in the MIXI-project, still the linking property was considered useful. According to the project group, the limitation of the prototyped tool was the user interface, programmed with MetaEdit+ Method Workbench 3.0, which was evaluated as too complex for engineers and research scientists. The advantages reported were as follows:

- ☐ The PSSP-formatted documentation records the useful and crucial properties and characteristics of a project, which may easily be omitted and lost in other formats.
- ☐ The use of PDPD and its tool support provide a powerful conceptual framework for information-sharing among researchers, engineers and management.
- ☐ The formalism facilitates the integration of various disciplines under the same information system, which supports concurrent engineering.
- ☐ When necessary, the use of phenomenon-driven ordering may enhance the discovery of new and novel (or creative) designs in any process development project.
- ☐ PDPD provides a platform that accelerates the goal-setting and helps define the problem space, which results in enhanced project efficiency and performance.
- ☐ The unexpected and creative activities and the modelling data can be documented with the tool.
- ☐ The use of the PDPD tool rationalises external data representation and the usage of external solvers. Externals can be used as attribute values of objects.

Third case

This case is more focused than the others. During this case, a researcher took the prototype tool into real use (Anttila and Pasanen 1999). A direct advantage the prototyped tool provided was time-savings when using the in-house solver. This is gained by ME+ code generation features for the automatic transformation of conceptual design models into simulation data programmed with Mathematica. Another direct advantage is the data management of numerous conceptual process versions of the process and the corresponding mathematical simulation files. In other words, the manual data management involved in the conceptual design and in mathematical modelling is time-consuming, but now the conceptual models are stored into the ME+ object repository, which links the mathematical modelling data. This is a demonstration of how to improve the usage of an in-house simulation tool. Commercial process-modelling tools, in turn, have special predefined building blocks to represent the modelling problems, which are transformed into numerical results via predefined mathematical models. The generic features of PSSP language, the experiences of this case and the realised code generations to AspenPlus and Modest support the conclusion that PSSP-formatted process models can be transformed efficiently to many commercial and in-house solvers. Of course, the applicability of the prototyped PDPD tool as a higher-level model data manager is dependent on how the PSSP elements are covered in the external solver.

8. Discussion and conclusions

Phenomenon-driven process design methodology has enough conceptual richness to document the process models, development activities and project management activities of the test cases. According to the feedback of the test cases, PDPD-formatted documentation (with PSSP language) records the properties and characteristics of projects, which may easily be omitted and lost in other formats (see the first case). PDPD provides an alternative platform for information-sharing among researchers, engineers and management (see the second case). The formalism facilitates the integration of various disciplines under the same information system, which supports concurrent engineering (see the second case). These methodological features are captured into the prototyped PDPD tool. Via test cases, the tool has been used for illustrating some advantages of using the methodology. Therefore, the effort put on meta-level analysis and computer implementation of PDPD and PSSP language has paid off in this sense. However, testing and further development of PDPD tools is still under way. The tool development and test cases presented in this thesis are only a modest step towards providing a fully tested and widely used methodology.

8.1 Effect of using the PDPD tool

The effect of using PDPD obviously depends on how the organisation originally adopts phenomenon-driven argumentation in goal-setting. Probably, a "phenomenon-driven" way of thinking is nothing new for any chemical engineer but it is less consciously and less systematically adopted (as noted in the second case), than is suggested according to PDPD. On the other hand, in order to improve information–sharing, an organisation may have its own methods. Often these methods are lacking in consistent data structure, which makes it difficult to develop those methods and may even stifle the information management. Thus, enhancing information-sharing requires both the systematic analysis of the concepts the organisation uses as well as the development of tools and methods to rationalise the project activities. This study is a step towards a more systematised modelling practice utilising phenomenon-driven process design methodology and PSSP language. The resulting tool support has been demonstrated with a few pilot projects. However, the effect of PDPD as a methodology is not totally dependent on computer tool support. Computer tool support eases the usage of PDPD, and it enables proper data management and data sharing.

Process modelling has traditionally focused on developing and applying detailed and specific modelling techniques. Still, at the early phase of process-modelling work, the most crucial decisions are made:

"Why are certain features taken into conceptual model?"
"Why has a certain modelling technique been selected?"
"How should the modelling goals be formulated?"
"How should the modelling activities be initiated?"
"How should a comprehensive model documentation be made?"
"How should the features under investigation from the real world be taken account of?"
"How should representation between mathematical models and tools, and the real world be documented?"

The PDPD tool provides an alternative platform for making sure that these questions are addressed. Thus, PDPD can be seen as an upper-level integrator for Figure 5 presented in Section 3.3.

8.2 Unique features of the tool

Management guidance

The second case in Section 7.2 illustrates how PDPD and the tool would systematise the project management. ME+ queries would have accelerated the project management, for example in goal-setting and by listing the suggested objects to be primarily edited. The acceleration would be achieved by utilising the automated suggestions:

☐ to report the objects being out of target specification,
☐ to organise the "off-spec" objects according to the hierarchy of PDPD and
☐ to draw-up these lists of objects.

Process models and solvers

Perkins et al. (1996) have stated that an optimal modelling method and tool can be used for producing conceptual process models without dictating models too early by some numerical solvers. This feature would support creativity when creating alternative flowsheets for a process. The prototyped PDPD tool does not dictate conceptual models with any kind of solver method or numerical tool, but the objects and their properties in PDPD tool can be associated with proper solver systems. Obtaining numerical values for the objects is dependent on the flowsheeting (i.e. sketching of process topological alternatives). Thus, flowsheeting and simulation are iterative processes, which are usually carried out via commercial modelling tools. Getting the models to converge is dependent on a proper list of input-data. It is another question whether this data is available for the project. Commercial process simulation tools analyse the degrees of freedom, and may help to get the problem to converge. However, the required input data is not always available, and on the other hand, the solver tool might not be able to express and solve some attractive design solution. PDPD and the tool do not restrict any combinatory or topological process alternatives. This feature and ‘phenomenon-driven ordering’ fitted well into the second test case. However, this fit is not a proof that, by following PDPD, a more creative process designs may be found. The second test case only illustrates how the PDPD tool can be used for project documentation, when it is necessary that established (or any) processes be called into question in a systematic manner.

Proper documentation of mathematical models

Mathematical models are structured through some modelling elements (nodes), which are dependent on each other in a certain way (links). Usually, such models have purpose and performance. Performance means more than whether a model converges numerically or not. Often, the purpose and performance specifications of a mathematical model are not properly documented. The PDPD tool provides a platform for model documentation and helps initiate model analysis, *e.g.* as illustrated in the first case. Implementing these features in modelling practice is expected to improve the quality and performance of process models, as was reported by the project groups of all the test cases. To enhance the initiation of process model analysis, a query (part of query in Appendix 7) can be used for listing the terminal elements of a complex model hierarchy. Perceiving the terminal elements of a model is not always easy, as can be seen in Figures 2 and 4. Other queries (Appendices 6, 7 and 8) can be used to help list the dependencies between the relation objects and external solvers. These queries would be valuable if several modellers and designers are involved in developing a process model. However, the use of these queries has not been illustrated in this thesis.

Holistic model evaluation

The evaluation of conceptual and parameter uncertainty appears to be a very important step in almost any research and development work. For example, in Section 2.2, PID methodology and the SPOLD program have a separate 'uncertainty analysis' step for this purpose. PDPD methodology does not include an uncertainty analysis as a predefined activity type, but if a PDPD user considers it important, then he can trigger an uncertainty analysis action. Obviously, the uncertainty of a model is dependent on the simplifications and assumptions made, as well as on the structure of a mathematical model itself. In using PDPD, the concept selection is documented transparently, which may facilitate seeing and evaluating the various model uncertainties more accurately. The implemented queries do not produce a quantitative sensitivity analysis of a model, but a list of relation objects, which might be affected when changing a value of a variable or the structure of some other relation object. However, the usefulness of this feature was not analytically tested in the test cases.

Methodology engineering tool

The tool development environment, MetaEdit+, was experienced as a powerful platform for prototyping methodologies. These experiences have been utilised also in the development and test use of another methodology: "Implementation of IPPC-directive – development of a new generation methodology and a case study in pulp industry". (Olin *et al.* 2000, Pasanen *et al.* 1999b and Pasanen *et al.* 2000).

8.3 Value of the experiences of the computer implementation and test use of PDPD

Would this kind of systematising work pay off if put into practice in an industrial company? Some pilot projects have been presented here, where PDPD has been applied successfully; but what is the return on investment, or payback time of tool development and usage? The figures for competitive advantage have never been calculated. However, any company aims to improve its profitability, which also applies to companies undertaking chemical process R&D work. This thesis aims to promote this endeavour by developing and testing a computer tool, which is expected to systematise and rationalise R&D work. The computer tool development work is based on PDPD methodology, which brings out and documents the relevant concepts in R&D work.

The scientific value of the work is difficult to evaluate, and there are very few publications on methodology development on process R&D. The type of a computer-aided methodology engineering tool (MetaEdit+) presented in this thesis is new for most chemical engineers. The meta-level analysis makes almost any engineer feel uncomfortable by making him analyse his way of conducting engineering and research work. This thesis shows a procedure for demonstrating the use of the methodology, and describes the potential advantages that can be gained by using the methodology as a software system. Perhaps this demonstration can be utilised as a reference procedure for building and evaluating a computer tool for other process design methodologies.

9. Future development work

Further requirements of the prototyped tool

Based on the experiences of the prototype tool development and on the case projects, some *a postepriori* requirements for the PDPD tool can be stated. The use of an open-formatted data structure according to PDPD principles would save more time if the user interface had been designed more carefully. This means that the most relevant concepts to be specified are highly project dependent, which may require, *e.g.* more detailed editors for describing the process equipment. It also means that supporting the dynamic features (see Principles 1 and 2 in Sections 5.2.1 and 5.2.2) in decision-making in R&D projects may necessitate setting up clearer guidance, and perhaps more specified dynamic checks on user's input.

Currently there is an ongoing project where PDPD is applied as an intranet database, which is viewed through a www-browser and edited via a www-page editor. Intranet operability provides a truly shared knowledge base, which would result in saved meeting time and possibly in better model and modelling work comprehension. This kind of www-browser might also be linked dynamically into Internet-based process simulation systems. For example, Zeng, *et al.* (2000) has introduced Internet-based simulation systems.

Further utilisation of the methodology and ongoing projects

If the types of applied modelling practices in chemical engineering were listed and mapped to PDPD and PSSP language, then the usefulness (or non-usefulness) of PDPD and the tool could be evaluated more carefully. PDPD is not a widely used methodology. PDPD methodology and prototyped PDPD tools require further testing (pilot projects and development of better user interfaces). There are a few ongoing projects, and some are on the drawing board, which are aimed at utilising PDPD further. The projects apply PDPD in various domains, and therefore also additional computer supports are to be developed. Some of these projects will specify the economical effect of the use of PDPD.

References

Ahola, J.L. 1998. Transient Studies of Exhaust Catalysis. Licentiate thesis. University of Oulu, Finland. 61 p. In Finnish.

Alha, M.K. and Pohjola, V.J. 1995. Eliciting Expert Knowledge; A Study on Connectionist Approach for Building a Cognitive Model of Design. Computers and chemical engineering, 19, Suppl., pp. S809–S814.

Andersson, M. 1994. An object-oriented modeling and simulation of hybrid systems. Doctoral dissertation. Department of automatic control, Lund institute of technology, Seweden.

Anttila, J.R. and Pasanen, A.P. 1999. Integration of Mathematica into an information system supporting chemical process development. Paper presented at International Mathematica Symposium, IMS'99. Linz, Austria, 1999. http://south.rotol.ramk.fi/~keranen/IMS99/paper19/juttu.pdf

ASPEN PLUS ver. 9.2. User Guide. 1995. Aspen Technology, Inc. Cambridge, Massachusetts, USA.

Asymmetrix ToolBook 3.0 1994. User's manual Austin, TX, USA. 214 p.

Bañares-Alcántara, R. and Labadidi, H.M.S. 1995. Decision support systems for process engineering – II. KBDS: An experimental prototype. Computers and chemical engineering, 19, 3, pp. 279–301.

Bañares-Alcántara, R., Josh King, M.P. and Ballinger, G.H. 1995. Égide: A Design Support System for Conceptual Chemical Process Design. AI System Support for Conceptual Design. (Ed. by Sharpe, J.E.E.) Springer-Verlag, Lancaster, UK.

Barton, P.I. 1992. The modeling and simulation of combined discrete/continuous processes. Ph.D. thesis, Department of Chemical Engineering, Imperial College, London, U.K.

Barton, P.I. and Pantelides, C.C. 1994. The modeling of combined discrete/continuous processes. AIChE Journal 40, pp. 966–979.

Becker, A. 1997. Enhancement of computer aided conceptual process design. Licentiate thesis. University of Oulu, Department of process engineering, Finland. 85 p. In Finnish.

Benson, R. 1995. Computer integrated management: an industrial perspective on the future. Computers and chemical engineering, 19, Suppl., pp. S543–551.

Bogusch, R. and Marquardt, W. 1995. A formal representation of process model equations. ESCAPE-5, Computers and chemical engineering, 19, Suppl., pp. S211–216.

Bogusch, R. and Marquardt, W. 1997. A formal representation of process model equations. Computers and chemical engineering, 21, 10, pp. 1105–1115.

Cellier, F.E. and Elmqvist, H. 1993. Automated formula manipulation supports object-oriented continuous-system modeling. IEEE Control Syst., April., pp. 28–38.

Christopher, J., Chonghun, H. And Stephanopoulos, G. 1995. Intelligent Systems in Process Engineering. Part I: Paradigms from Product and Process Design. Vol 21, Modeling language: Declarative and imperative descriptions of chemical reactions and processing systems. Academic Press, San Diego, California, USA. Pp. 73–91.

Costello, D.J., Fraga, E.S., Skilling, N., Ballinger, G.H., Bañares-Alcántara, R., Krabbe, J., Laing, D.M., Ponton, J.W. and Spenceley, M.W. 1996. Épée: A support environment for process engineering software. Computers and chemical engineering, 20, 12, pp. 1399–1412.

Cuena, J. 1993. Knowledge oriented software design. Madrid, Spain. North-Holland.

Douglas, J.M. 1985. A Hierarchical decision procedure for process synthesis. AIChE Journal 31, 3, pp. 353–362.

Douglas, J.M. 1995. A Hierarchical decision procedure for process synthesis. Intelligent Systems in Process Engineering. Part I: Paradigms from Product and Process Design. Vol 21, Modeling language: Declarative and imperative descriptions of chemical reactions and processing systems. Academic Press, San Diego, California, USA. Pp. 93–146.

Elmqvist, H.A. 1978. structured model language for large continuous systems. Ph.D. thesis. Department of Automatic Control, Lund institute of technology, Sweden.

Elmqvist, H., Cellier, F.E. and Otter, M. 1993. Object-oriented modeling of hybrid systems. ESS'93 European Simulation Symp. Delft, The Netherlands. Pp. 25–28.

Embley, D.W., Kurtz, B.D. and Woodfield, S.N. 1992. Object-Oriented System Analysis: A model-Driven Approach. Prentice-Hall, New Jersey.

Frezza, S.T., Levitan, S.P. and Chrysantis, P.K. 1996. Linking requirements and design data for automated functional evaluation. Computers in industry, 30, pp. 13–25.

Gavrila, I.S. and Iedema, P. 1996. Phenomena-driven process design, a knowledge-based approach. Computers and chemical engineering, 20, Suppl., pp. S103–108.

Gigh van, J.P. 1991. Systems design modeling and metamodeling. Plenum Press, New York, USA.

Goldberg, A. and Robson, D. 1983. Smalltalk-80: The language and its implementation. Addison–Wesley. Reading, Massachusettes, USA. 714 p.

Han, C. 1994. Human-aided computer-based design paradigm: The automation of conceptual process design. PhD. Thesis, Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA.

Han, S., Lee, T., Yoon, T. and Naka, Y. 1999. Information Sharing between Process and Engineering Design Activity in CAD Environment. Computers and chemical engineering, 23, Suppl., pp. S573–S576.

Herder, P.M. and Weijnen, M.P.C. 1998. Quality Criteria for Process Design in the Design Process – Industrial Case Studies and an Expert Panel, Computers and chemical engineering, 22, Suppl., pp. S513–S520.

Hirschheim, R., Klein, H.K. and Lyytinen, K. 1996. Exploring intellectual structures of information system development: a social action theoretic analysis. Accounting, management and information technologies, 6, 1/2, pp. 1–64.

Huuskonen, P. J. 1997. Model-based explanation of plant knowldege. Espoo: Technical Research Centre of Finland, VTT Publications 308. 173 p. + app. 59 p.

ISO 9001: 1994. 1994. Quality systems. Model for quality assurance in design, development, production, installation and servicing. ISO (the International Organisation for Standardisation), Geneve.

ISO 10303:1994. 1994. Quality systems. Model for quality assurance in design, development, production, installation and servicing. ISO (the International Organisation for Standardisation), Geneve.

Kelly, S., Lyytinen, K. and Rossi, M. 1996. MetaEdit+: A fully configurable multi-user and multi-tool CASE and CAME environment. In: Proceedigns of the 8th conference on advanced information systems engineering, CAISE’96. Crete, Creece, 1996. Pp. 1–21.

Koivisto, R. 1996. Safety-conscious process design. Doctoral dissertation. Espoo: Technical Research Centre of Finland, VTT Publications 264. 79 p. + app. 67 p.

Lund, P.C. 1992. An object-oriented environment for process modeling and simulation. Doctoral dissertation. Laboratory of chemical engineering, Norwegian institute of technology, Trondheim, Norway.

Lyytinen, K., Kaipala, J., Kelly, S., Koskinen, M., Liu, H., Luoma, J., Marttiin, P., Pohjonen, R., Kukkonen, H. Oinas, Rossi, M., Somppi, M., Tahvanainen, V.-P. and Tolvanen, J.-P. 1997. CAMSO: Computer Aided Modelling Support for Organisations, Technical Report TR-18, Department of Computer Science and Information Systems, University of Jyväskylä, Finland.

Maiorella, B., Wilke, C.R. and Blanch, H.W. 1984. Alcohol production and rexovery. In: Advances in biochemical engineering. (Ed. by Fiechter, A.) Vol. 20. Springer-Verlag.

Marquardt, W. 1992a. Rechnergestützte erstellung verfahrenstechnicher prozessmodelle. Chem.-Ingn.-Tech. 64, pp. 25–40. English translation in Int. Chem. Engng, 16S(1994), pp. S28–46.

Marquardt, W. 1992b. An object-oriented representation of structured process models. Computers and chemical engineering 16, Suppl., pp. S329–336.

Marquardt, W. 1996. Trends in computer-aided process modeling. Computers and chemical engineering, 20, 6/7, pp. 591–609.

Marquardt, W., Gerstlaur, A. and Gilles, E.D. 1993. Modeling and representation of complex objects: a chemical engineering perspective. Proc. 6th Int. Conf. IEA/AIE 1993, Edinburgh, Scotland. Pp. 219–228.

Mattson, S.E. and Andersson, M. 1994. OMOLA – an object-oriented modeling language. In: Recent advances in computer aided control engineering. (Eds. Jamshidi, M. and Herget, C.J.) Elsevier, Amsterdam, Netherlands. Pp. 291–310.

Maunula, T., Ahola, J., Salmi, T., Haario, H., Härkönen, M. Luoma, M. and Pohjola, V.J. 1997. Investigation of CO oxidation and NO reduction on three-way monolith catalysts with transient response techniques. Applied catalysis B: Environmental 12, pp. 287–308.

MetaCase Consulting. 1996. MetaEdit+ (version 2.5): User's Guide, Micro-Works, Jyväskylä, Finland.

MetaCase Consulting. 1999. MetaEdit+ (version 3.0): User’s Guide, MicroWorks, Jyväskylä, Finland.

Meyssami, B. and Åsbjørnsen, O.A. 1989. Process modeling from first principles – method and automation. In: Conference proceedings of Computer simulation 1989, Austin, TX, USA. Pp. 292–299.

Microsoft corp. 1994. Building applications, Microsoft Access 2.0 1994.

Modest user’s guide (by Haario, H.). 1994. ProfMath Oy. Helsinki, Finland.

Nilsson, B. 1993. Object-oriented modeling of chemical processes. Doctoral dissertation. Department of automatic control, Lund institute of technology, Seweden.

Oh, M. and Pantalides, C.C. 1994. A modeling and simulation language for combined lumped and distributed parameter systems. Proceedings of 5th international conference of process systems engineering PSE’94, Kyongju, Korea Vol. 1. 1995. Pp. 37–44.

Oinas-Kukkonen H. 1996. Debate Browser – An Argumentation Tool for MetaEdit+ Environment. Proceedings of the Seventh European Workshop on Next Generation of CASE Tools (NGCT '96), Crete, Greece, May 1996. Pp. 77–86.

Olin, M. 1994. Diffusion in crushed rock and in bentonite clay. Dissertation. Espoo: Technical Research Centre of Finland, VTT Publications 175. 98 p.

Olin, M., Pasanen, A., Ojaniemi, U., Mroueh, U.-M., Walavaara, M., Larjava, K., Vasara, P., Lobbas, P. and Lillandt, K. 2000. Implementation of IPPC-directive. Development of a new generation methodology and a case study in pulp industry. Espoo: Technical Research Centre of Finland, VTT Publications 429. 81 p.

Orvis, J.W. 1996. Excel for Scientists and engineers. SYBEX Inc., Alameda California, USA. 547 p.

Pantelides, C.C. and Barton, P.I. 1992. Equation-orienteddynamic simulation: Current status and future prospectives. Computers and chemical engineering, 17, Suppl., pp. S263–285.

Pasanen A. 1995. Systematisation of process design. Department of process engineering, University of Oulu. 101 p. In Finnish.

Pasanen, A. 1996. Presenting process design knowledge with a conceptual process design methodology. University of Oulu, Finland, Report 193. P. 33. ISBN 951-42-4534-2.

Pasanen, A.P. 1998. Tool supported methodology for process synthesis. AIChE Annual Seminar 1998, Nov. 15–20, 1998, Miami Beach, USA. www.aiche.org/docs/meetapp/programming/techprogram/abstracts/2861.htm

Pasanen, A.P. and Tolvanen J.-P. 1999. Computer tool supported process design methodology. Computers and chemical engineering, Suppl., pp. S723–S726.

Pasanen, A.P., Ahola, J.L., Tolvanen, J.-P. and Pohjola, V.J. 1998. Phenomenon Driven Process Design methodology: Tool support. CD-ROM Paper prints of CHISA'98, Prague, Czech Republic.

Pasanen, A.P., Pohjola, V.J., Turunen, I. and Luoma, M. 1999a. Phenomenon driven Process Design methodology: Industrial case. Computers and chemical engineering, Suppl., pp. S43–S4.

Pasanen, A., Ojaniemi, U., Olin, M. and Liimatainen, M. 1999b. Methodology development for integrated pollution prevention. Paper presented in AIChE Annual Seminar 1999 (17003 – Pollution Prevention in the Forest Products Industry), Dallas, TX, USA. 8 p.

Pasanen, A., Olin, M. and Lassila, P. 2000. Methodology for Integrated Pollution Prevention. In: Proc. of 4th International Conference on Environmental Impacts of the Pulp and Paper industry, 12.–15.6.2000, Helsinki, Finland. Pp. 64–68.

Perkins, J.D. and Sargent, W.H. 1982. SPEEDUP: A computer program for steady-state and dynamic simulation and design of chemical processes. AIChE Symp. Ser. 78(214), pp. 1–11.

Perkins, J.D., Sargent, W.H. and Vásques-Román, R. 1994. Computer generation of process models. In: Proceedings of 5th international conference of process systems engineering, PSE'94. Kyongju, Korea, Vol. 1. Pp. 37–44.

Perkins, J.D, Sargent, W.H, Vázqes-Román, R. and Cho, J.H. 1996. Computer generation of process models. Computers and chemical engineering 20, 6/7, pp. 635–639.

Peskin, A.P. and Hardin, R. 1996. An object-oriented approach to general purpose fluid dynamics software. Computers and chemical engineering, 20, 8, pp. 1043–1058.

Piela P.C. 1989. Ascend: An object-oriented computer environment for modelling and analysis. Ph.D. thesis, Carnegie–Mellon-University, Pittsburgh, PA, USA.

Piela, P.C., Epperly, T.G., Westerberg, K.M. and Westerberg, A.W. 1991. ASCEND: an object-oriented environment for modeling and analysis: the modelling language. Computers and chemical engineering, 15, pp. 53–72.

Piela, P.C., McKelvey, R.D. and Westerberg, A.W. 1992. An introduction to ASCEND: its language and interactive environment. Journal of manufacturing information science, 9, pp. 91–121.

Pohjola, V.J. 1995. Integration of Reactor Design to Process synthesis. Proceedings of the Course 'Novel Reactors for the 21st Century' of Finnish Graduate School of Chemical Engineers, July 2–5, 1995, Kuusamo, Finland.

Pohjola V.J. 1997. Systematization of Process Design. University of Oulu, Department of Process Engineering. Report 204. ISBN 951-42-4697-7.

Pohjola, V.J., Alha, M.K. and Ainassaari, J. 1993a. Methodology of Process Design. ESCAPE-3, Computers and chemical engineering, 18, Suppl., pp. S307–S311.

Pohjola, V. J., Koivisto, R. and Alha, M. K. 1993b. Conceptual association of safety with process design and operation, FOCAPO '93 Conference, Mount Crested Butte, Colorado, July 18–23, 1993. Published in: Foundations of Computer-Aided Process Operations, 1994, David W.T. Rippin, John C. Hale and James F. Davis (eds.), Austin TX, Cache. Pp. 457–463.

Pohjola, V.J., Alha, M.K., Pongrácz, E. & Keiski, R. (editors) 1997. Creativity and Chemical Process R&D. Proceedings of the GSCE Seminar. University of Oulu, Hailuoto, Oct. 17–18, 1996.

Pohjola, V.J. and Tanskanen, J. 1998. Phenomenon Driven Process Design Methodology: Formal representation. CD-ROM paper prints of CHISA'98, Prague, Czech Republic.

Pohjola, V.J., Ahola, J.L and Pulkkinen, U. 1999. Computers and chemical enginering, Suppl. 23, pp. S731–S734.

Pongrácz, E. and Pohjola, V.J. 1997. The conceptual model of waste management. ENTREE 1997, Vantaa. In: Proceedings of ENTREE 1997. Pp. 65–77.

Preisig, H.A. 1991. On computer-aided modeling for design. AIChE Annual meeting, Los Angeles, USA.

Preisig, H.A. 1992. An object-oriented approach to computer-aided modeling. Proceedings CHEMECA-1992, Australia.

Preisig, H.A. 1994a. Computer-aided modeling: species topology. Proceedings of international IFAC symposium and Advanced control of chemical processes ADHEM'94, Kyoto, Japan, 1994. Pp. 143–148.

Preisig, H.A. 1994b. Modeller – an object-oriented computer-aided modelling tool. Paper presented in Foundations on computer aided design FOCAPD'94, Snowmass, CO, USA, 1994.

Rantakylä, T.-K. 1995. Liquid phase hydrogenation. Licentiate thesis (in Finnish), Department of Process engineering, University of Oulu. 77 p.

Rossi, M. and Tolvanen, J.-P. 1995. Using reusable frameworks in development of a method support environment. In: Proceedings ofthe WITS 1995 Amsterdam, The Netherlands. (Eds. M. Jarke and S. Ram.) Pp. 240–249.

Roussel, P.A., Saad, K.N. & Erickson, T.J. 1991. Third Generation R&D. Harvard Business School Press, Boston.

Rousu, P.P. and Rousu, P. 1997. High Grade Nonwood Based Paper Economically and Environmentally Friendly. Pulping Conference, San Francisco, 1997. Atlanta. TAPPI. Book 2. Pp. 487–492.

Rousu, P.P, Anttila, J.R. and Rousu, E.J. 1999. Method for formic acid recovery (In Finnish). Chempolis Oy, Finland. Appl. FI973474. 23.2.1999.

SKI Report 96:36 1996. SKI SITE-94. Deep repository performance assesment project. Volume I. Swedish Nuclear Power Inspectorate. ISSN 1104-1374

Smolander, K. 1992. OPRR – A model for modelling systems development methods. In: Next generation CASE tools. (Eds. Lyytinen, K. and Tahvanainen, V.-P.) IOS Press, Amsterdam, Netherlands. Pp. 224–239.

Stephanopoulos G., Johnston J, Kriticos T., Lakshmanan R., Mavrovouniotis M.L. and Siletti, C. 1987. Design-Kit: An object-oriented environment for process engineering. Computers and chemical Engineering, 11, pp. 655–674.

Stephanopoulos, G. Henning, G. and Leone, H. 1990a. MODEL.LA. A modeling language for process engineering. I. The formal framework. Computers and chemical engineering, 14, 8, pp. 813–846.

Stephanopoulos, G. Henning, G. and Leone, H. 1990b. MODEL.LA. A modeling language for process engineering. II. Multifaced modeling of processing systems. Computers and chemical engineering, 14, 8, pp. 847–869.

Sørlie, C.F. 1990. A computer environment for process modeling. Doctoral dissertation, Laboratory of Chemical Engineering, Norwegian institute of technology, Trondheim, Norway.

Tanskanen, J. 1999. Phenomenon driven process design: Focus on multicomponent reactive and ordinary distillation. Doctoral dissertation thesis. Department of Process Engineering, University of Oulu. 60 p.

Tanskanen, J., Pohjola, V.J. and Lien, K.M. 1995. Phenomenon driven process design methodology: Focus on reactive distillation. ESCAPE-5, Computers and chemical engineering, 19, Suppl., pp. S77–S82.

Telnes, K. 1992. Computer-aided modeling of dynamic processes based on elementary physics. Doctoral dissertation. Division of technology, Trondheim, Norway.

Tolvanen J.-P. 1999. Incremental method engineering with modelling tools. (Dissertation). Jyväskylä studies in computer science, economics and statistics 47., University of Jyväskylä, Finland. 301 p.

Tolvanen, J.-P., Marttiin, P. and Smolander, K. 1993. An integrated model for information systems modeling. In: Proceedings of the 26th annual Hawaii international conference on systems scienc. (Eds. Nunamaker, J. and Sprague, R.) IEEE Computer society Press, Los Alamitos, USA. Pp. 470–479.

Turunen, I., 1997. Intensification of the Anthraquinone Process for Production of Hydrogen Peroxide. In: Proceedings of 2nd international conference on process intensification in practice. (Ed. by Semel, J.) Antwerpen, Belgium, October 21–23, 1997, BHR Group Conference Publication No 28.

Vieno, T. and Nordman, H. 1997. FEPs and scenarios. Auditing of TVO-92 and TILA-96 against International FEP database. POSIVA 97-1.

Westerberg, A.W., Abbot, K. and Allan, B. 1994. Plans for ASCEND IV: our next generation equation-based modeling environment. Paper presented at AspenWorld'94, Boston, MA.

Winograd, T. 1986. Understanding conputers and ognition. New Jersey, Ablex Publishing corp. 211 p.

Wold S, Esbensen K, Geladi P. 1987. Principal component analysis. Chemometrics and Intelligent Laboratory Systems, Vol 2, pp. 37–52.

Woods , E.A. 1993. The hybrid phenomena theory. Doctoral dissertation, Division of cybernetics, Norwegian institute of technology. Trondheim, Norway.

Wozny, M.J. 1990. Towards a theory and methodology of design. In Foundations of computer-aided process design. (Eds. Siirola, J.J, Grossmann, I.E. and Stephanopoulos, G.) Elsevier, Colorado, USA. Pp. 1–17.

Wolfram, S. 1988. Mathematica. A system for doing mathematics by computer. New York, Addison-Wesley Publishing Company. 749 p.

Zeng, Y., Jang, S. and Weng, C. 2000. Consider an internet-based simulation system. Chem. Engng. Prog., 96, 7, pp. 53–60.

Åsbjrønsen, O.A., Meyssami, B. and Sørlie, C. 1989. Structuring the knowledge for process modeling from first principles. Paper presented in IAKE'89, University of Maryland, College Park.

Appendix 1

Tools and editors of MetaEdit+:

Graph tool;	Functionality to set-up specifications for using flow diagrams, e.g. process flow diagrams or activity network in PDPD.
Object tool;	Functionality to set-up object types, e.g. generic project and generic chemical process.
Property tool;	Functionality to name generic properties (e.g. P, S, S, P attributes of 'GRT') and specific properties of object types (e.g. rate, and extent of 'physico-chemical phenomenon'). Also data types and ancestors are specified with the property tool.
Relationship tool;	Functionality to set-up relationship types between objects, e.g. 'object interaction' in PDPD categorised as 'info' | 'material' | 'energy' flows.
Role tool;	Functionality to specify the components that lie at the end of a relationship.
Dialog editor;	Functionality to set up the layout of the dialog, which is used for editing the properties of an instance of G, O, R and R (e.g. dialog for P, S, S and P expressions of GRT object).
Symbol editor;	Functionality to set up graphical symbols for O, R and R.
MetaEdit query language editor;	Functionality to specify predefined documentation reports, checks and code generation according to object repository, see Section 6.3.3 the queries compiled for PDPD.

ME+ provides the following editors and predefined procedures for a methodology user to use GOPRR-formatted methodologies and to manage the object repository:

Diagram editor;	Functionality to set-up and edit O, R and R in graphs as diagrams, e.g. to edit the network of a process system (through the topology attribute in PDPD).
Table editor;	Functionality for tabular or form-based views of the elements in graphs stored in the repository (not used in PDPD implementation).
Matrix editor;	Functionality to handle graphs as matrices, containing two axes and related cells (not used in PDPD implementation).
Property dialog;	Functionality to set and edit values for object properties.
Decomposition;	Functionality to decompose one or several objects into a new diagram (same as topological decomposition in PDPD but this utility has not been used in PDPD implementation)
Explosion;	Functionality to explode each O, R and R in a graph into other graphs (same as unit structural decomposition in PDPD but this utility has not been used in PDPD implementation).

Disaggregation via property; Functionality to set a value of a property as a collection of graphs. In PDPD, the topology attribute and unit structure attribute of an object are linked as topology graphs and unit structural graphs. Also design versions for an object can be generated in this way.

Report generation; Functionality to compile user's own queries and specification according to the data in the object repository.

Report execution; Functionality to run the predefined queries and user's own queries.

Report output window; for reporting the resulting output of queries in ASCII format.

Automated WWW, Java, C++, GIF, etc. -transcription of models; Functionality to facilitate the information exchange of the data in the object repository.

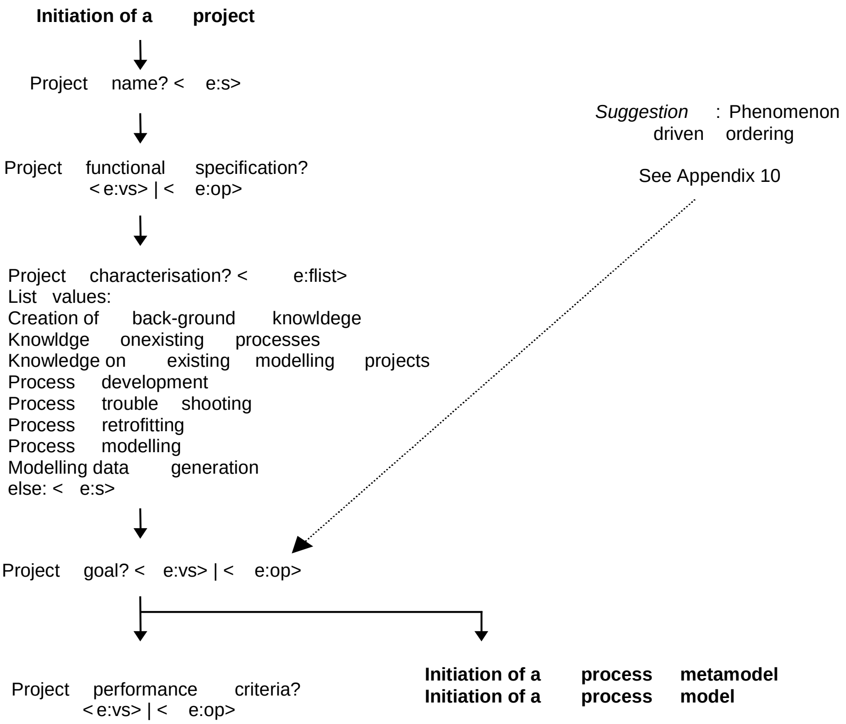
Information management tools; e.g. Type and Graph browsers for facilitating data management with compact and simpler browsers

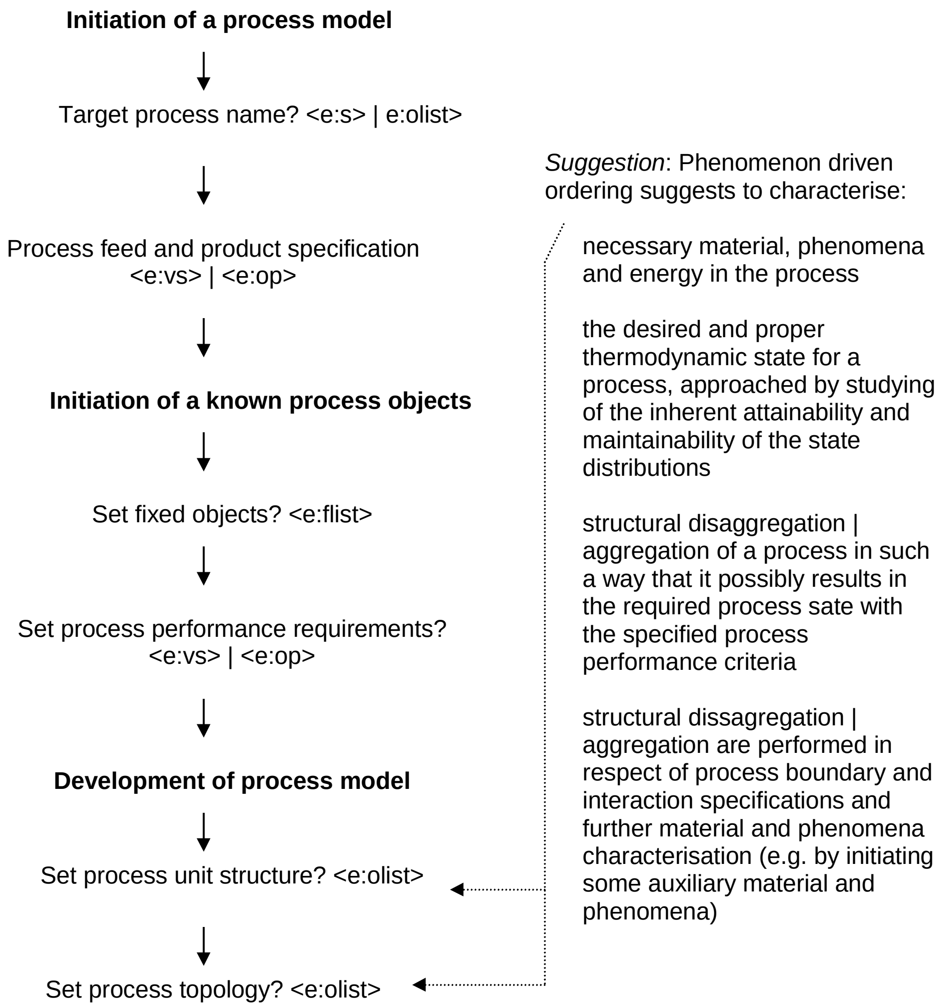
Appendix 2

A scheme of controls and data processes of PDPD tool.

Notation:

Question?	PDPD request
⟶	Next request or next object
┈▶	PDPD's suggestion
< e:s>	Expression: string
< e:vs>	Expression: set of verbal statements (including verbal satement target value)
< e:op>	Expression: object property (including object property target value)
< e:flist>	Expression: list of fixed values
< e:olist>	Expression: list of objects





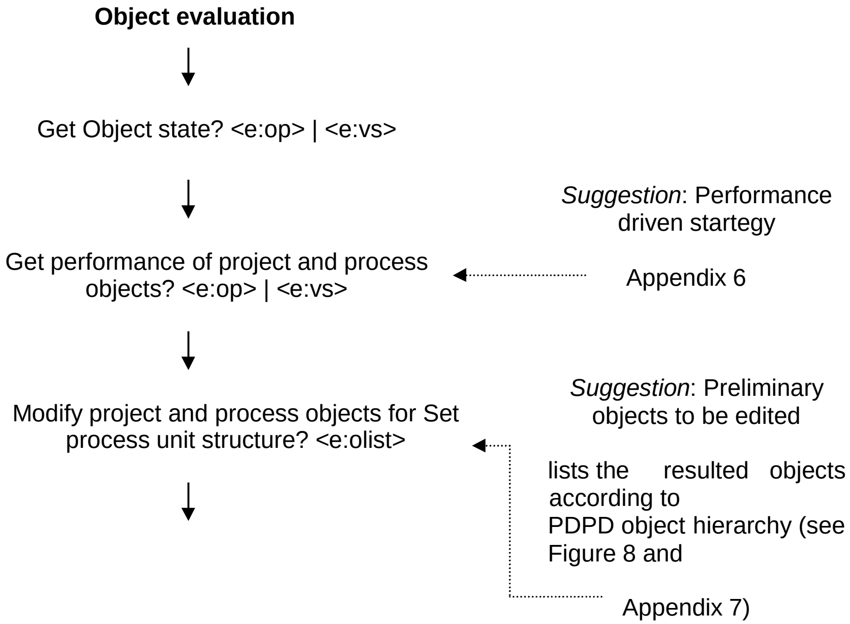
Suggestion: Phenomenon driven ordering suggests to characterise:

necessary material, phenomena and energy in the process

the desired and proper thermodynamic state for a process, approached by studying of the inherent attainability and maintainability of the state distributions

structural disaggregation | aggregation of a process in such a way that it possibly results in the required process sate with the specified process performance criteria

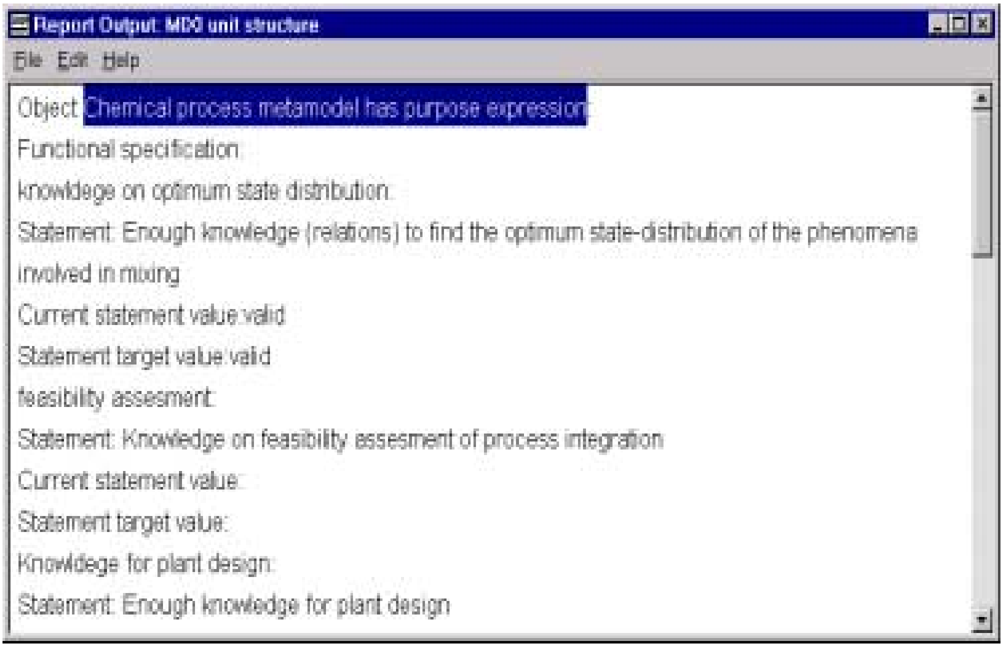
structural dissagregation | aggregation are performed in respect of process boundary and interaction specifications and further material and phenomena characterisation (e.g. by initiating some auxiliary material and phenomena)


Object evaluation
Get Object state? <e:op> | <e:vs>
Suggestion: Performance driven startegy
Get performance of project and process objects? <e:op> | <e:vs>
Appendix 6
Suggestion: Preliminary objects to be edited
Modify project and process objects for Set process unit structure? <e:olist>
lists the resulted objects according to
PDPD object hierarchy (see Figure 8 and
Appendix 7)

Appendix 3

Report on object purpose.

```
Report 'Object purpose'
Foreach .()
  {'Object '; id; ' has purpose expression: '; newline;
  do :Object purpose
    { newline; id; ':';
    do :Object property
      { newline; 'Property: '; id; ':   Property value: ' ; :Value; '  Unit: ':Unit;'   Property
        target value: ' ; :Target value; newline;
       };
    do :Verbal statement
      { newline; id; ':'; newline; 'Statement: '; :Statement; newline; 'Current statement
       value:' ; :Current Statement value; newline; 'Statement target value:' ;
       :Statement target value; newline;
       };
    }; newline;
  };
endreport
```



Object: Chemical process metamodel has purpose expression

Functional specification:

knowledge on optimum state distribution:

Statement: Enough knowledge (relations) to find the optimum state-distribution of the phenomena involved in mixing

Current statement value:valid

Statement target value:valid

feasibility assesment:

Statement: Knowledge on feasibility assesment of process integration

Current statement value:

Statement target value:

Knowldege for plant design:

Statement: Enough knowledge for plant design

Figure Appendix 3. ***An example report output.***

Appendix 4

Report on project properties.

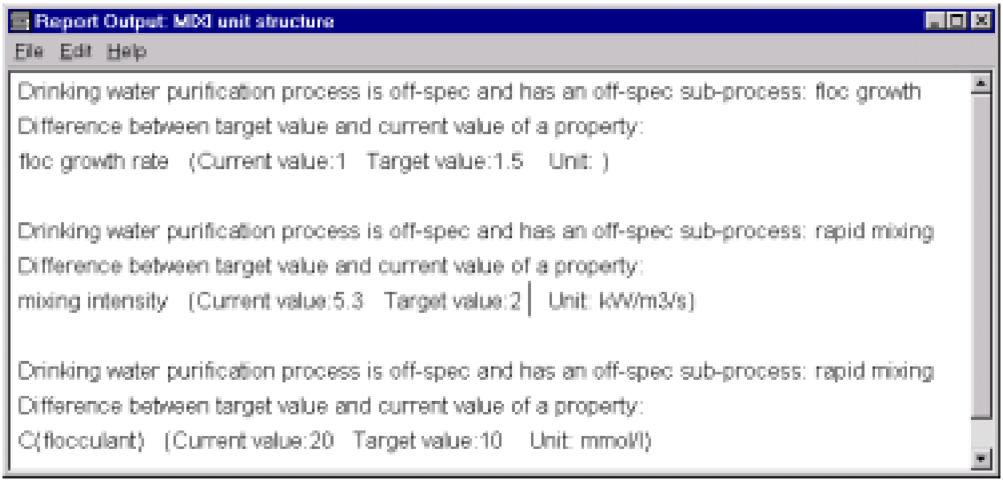
```
Report 'Project search'
'Report: which are the object properties for this project not meeting the target
value!'
newline;
foreach .()
  {dowhile :Object property
    {if :Property value; <> :Property target value; then
      newline; 'Object '; id;1; ' has difference between target value and current value
      of property: '; newline; id;'   (Current value:' ; :Property value; '   Target value:' ;
       :Property target value; ')';
    endif
    }
  dowhile :Verbal statement
    {if : Statement value <> :Statement target value; then
      newline; 'Object '; id;1; ' has difference between target value and current value
      of property: '; newline; id;'   (Current value:' ; :Statement value; '   Target value:'
       ; :Statement target value; ')';
    endif
  }
endreport
```

Appendix 5

Suggested sub-processes to be checked for an off- spec process in according to off-spec sub-process state variables

```
Report 'Suggestion on primarily checked sub-processes of an off-spec process'
foreach .Chemical process model
  {do :Object topology
    {foreach .Chemical process model
      {do :Object state
        {if :Value; <> :Target value; then
          newline;
          id;3; ' is off-spec and has an off-spec sub-process: '; id;1; newline;
         'Difference between target value and current value of a property: '; newline;
          id;'   (Current value:' ; :Value; '   Target value:' ; :Target value; :Unit; ')';
        endif
        newline;
        };
      };
    };
  };
endreport
```

In addition to topological dimensions, this piece of code can be extended to take acount of the unit structure of each sub-process: such as process interior objects (material, phenomena and energy), and interactions (material, energy and information).



```
Drinking water purification process is off-spec and has an off-spec sub-process: floc growth
Difference between target value and current value of a property:
floc growth rate   (Current value:1   Target value:1.5    Unit: )

Drinking water purification process is off-spec and has an off-spec sub-process: rapid mixing
Difference between target value and current value of a property:
mixing intensity   (Current value:5.3   Target value:2   Unit: kW/m3/s)

Drinking water purification process is off-spec and has an off-spec sub-process: rapid mixing
Difference between target value and current value of a property:
C(flocculant)   (Current value:20   Target value:10    Unit: mmol/l)
```

Figure Appendix 5. An example report output of the query

Appendix 6

Objects and code structure of the query: “possibly affected set of externals for a relation”.

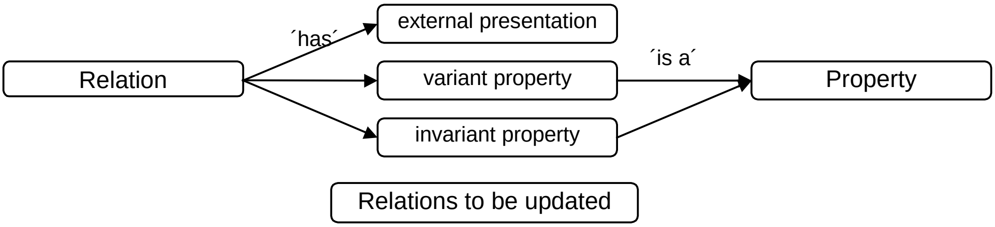


Figure Appendix 6a. Involved objects when searching of possibly affected external representations (e.g. mathematical tools) for a selected relation or set of relations.

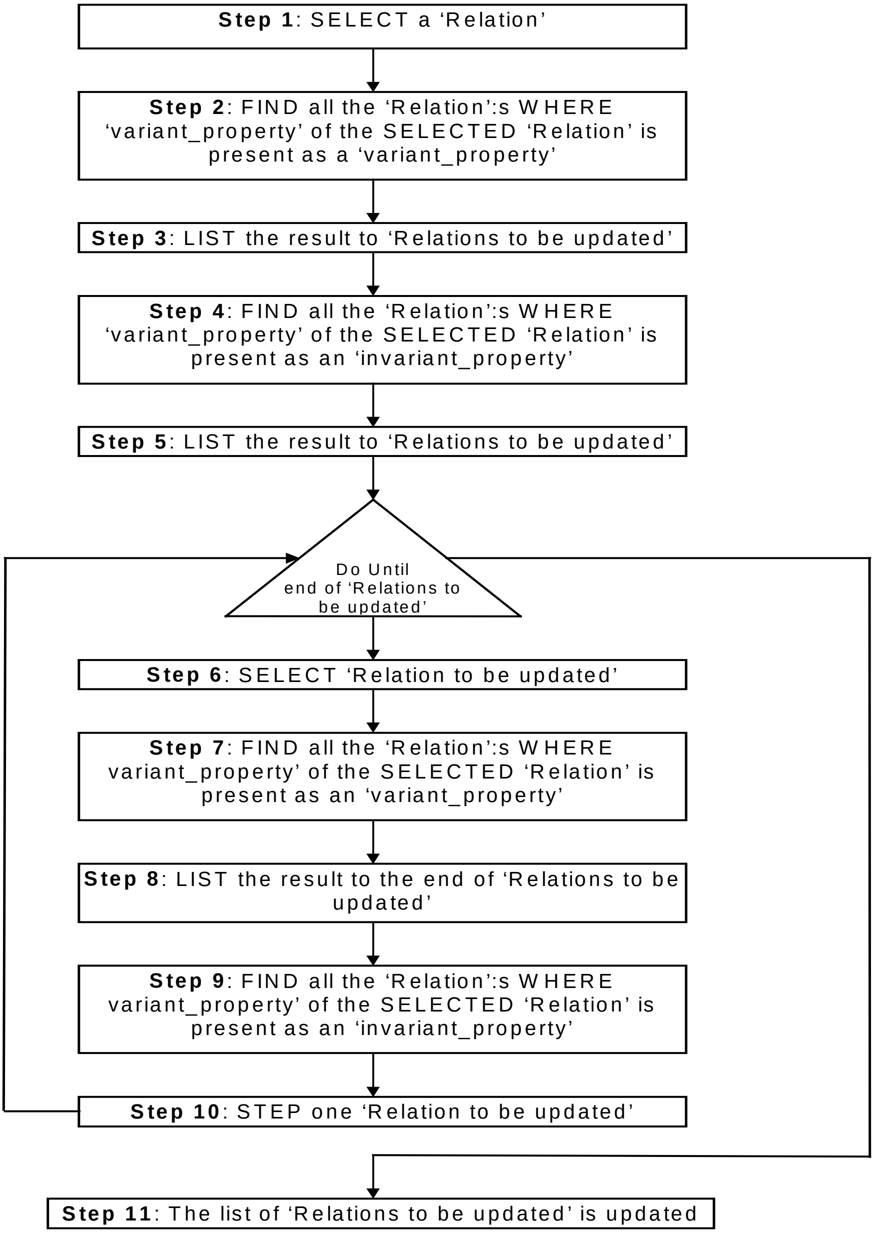


Figure Appendix 6b. Search algorithm for external representations possibly affected by an edited relation or set of relations

Appendix 7

Find all possibly affected relations for a selected relation, in MS Access 2, coded with and AccessBasic and SQL.

```
Sub Field1_Click ()
Dim MyDB As Database, Myset As Recordset, Myset2 As Recordset, Myset3 As
Recordset
Dim x As Integer
Set MyDB = DBEngine.Workspaces(0).Databases(0)
DoCmd RunSQL "DELETE *.* FROM tulos;"
DoCmd RunSQL "CREATE TABLE [apu] ([property_ID] TEXT);"
DoCmd RunSQL "CREATE TABLE [apu2] ([relation_ID] TEXT);"
field3.Requery

' Firstly, all the variant_properties of the selected relation are searched from the
' variant_link table. The result is put into “apu”-table.

DoCmd RunSQL "INSERT INTO apu SELECT variant_link.property_ID FROM
variant_link WHERE relation_ID=" & forms!form1!field1.column(0)

' Secondly, all the relations from variant_link table are searched into “apu2”-table, in
' which the properties listed into “apu”-table are present.

Set Myset2 = MyDB.OpenRecordset("apu", DB_OPEN_DYNASET)
Do Until Myset2.EOF 'begin loop
   DoCmd RunSQL "INSERT INTO apu2 SELECT variant_link.relation_ID
FROM variant_link WHERE
   variant_link.property_ID=" & Myset2!property_ID
   Myset2.MoveNext
Loop

' Thirdly, all the relations present in “apu2”-table are put into “tulos”-table.

Set Myset3 = MyDB.OpenRecordset("apu2", DB_OPEN_DYNASET)
Do Until Myset3.EOF 'begin loop
   DoCmd RunSQL "INSERT INTO tulos SELECT relation.* FROM relation
WHERE relation.relation_ID="
   & Myset3!relation_ID
   Myset3.MoveNext
Loop
Myset3.Close
DoCmd RunSQL "DELETE * FROM apu2;"
```

```
' The fourth step is to search form invariant_link–table all the relations in which
' “apu”-table properties are present. The result is put into “apu2” –table.

Myset2.MoveFirst
Do Until Myset2.EOF 'begin loop
   DoCmd RunSQL "INSERT INTO apu2 SELECT invariant_link.relation_ID
FROM invariant_link WHERE
   invariant_link.property_ID=" & Myset2!property_ID
   Myset2.MoveNext
Loop

'The fifth step is to search all the relations present in “apu2” –table and put them
into
' “tulos” –table.

Set Myset3 = MyDB.OpenRecordset("apu2", DB_OPEN_DYNASET)
Do Until Myset3.EOF 'begin loop
   DoCmd RunSQL "INSERT INTO tulos SELECT relation.* FROM relation
WHERE relation.relation_ID="
   & Myset3!relation_ID
   Myset3.MoveNext
Loop
DoCmd RunSQL "DELETE * FROM apu;" 'tuhotaan apun ja apu2:n sisältö
DoCmd RunSQL "DELETE * FROM apu2;"
field3.Requery
Myset2.Close
Myset3.Close

' Now the first round is completed and the rest is looped with End Of File (EOF)
' function analogously.

Set Myset = MyDB.OpenRecordset("tulos", DB_OPEN_DYNASET)

' The loop is started from the first row of “tulos” –table.

Do Until Myset.EOF
   DoCmd RunSQL "INSERT INTO apu SELECT variant_link.property_ID
FROM variant_link WHERE
   relation_ID=" & Myset!relation_ID

   ' All the relations in variant_link –table where “apu”-table properties are present
   ' are put into “apu2” –table.

   Set Myset2 = MyDB.OpenRecordset("apu", DB_OPEN_DYNASET)
   Do Until Myset2.EOF 'begin loop
```

```
    DoCmd RunSQL "INSERT INTO apu2 SELECT variant_link.relation_ID
FROM
    variant_link WHERE
    variant_link.property_ID=" & Myset2!property_ID
    Myset2.MoveNext
  Loop

  ' All the relations present in “apu2” –table are put into “tulos” –table.

  Set Myset3 = MyDB.OpenRecordset("apu2", DB_OPEN_DYNASET)
  Do Until Myset3.EOF 'begin loop
    DoCmd RunSQL "INSERT INTO tulos SELECT relation.* FROM relation
    WHERE
    relation.relation_ID=" & Myset3!relation_ID
    Myset3.MoveNext
  Loop
  Myset2.Close
  Myset3.Close
  DoCmd RunSQL "DELETE * FROM apu2;"

  ' All the relations in invariant_link –table where properties of “apu2” –table are
  ' present, are put into “apu2” –table.

  Set Myset2 = MyDB.OpenRecordset("apu", DB_OPEN_DYNASET)
  Do Until Myset2.EOF 'begin loop
    DoCmd RunSQL "INSERT INTO apu2 SELECT invariant_link.relation_ID
    FROM invariant_link
    WHERE invariant_link.property_ID=" & Myset2!property_ID
    Myset2.MoveNext
  Loop

  'All th relations in “tulos” –table where having 10. etsitään tulos-tauluun kaikki
relaatiot joissa esiintyy apu2-taulun relation_ID:t.

  Set Myset3 = MyDB.OpenRecordset("apu2", DB_OPEN_DYNASET)
  Do Until Myset3.EOF 'begin loop
    DoCmd RunSQL "INSERT INTO tulos SELECT relation.* FROM relation
    WHERE
    relation.relation_ID=" & Myset3!relation_ID
    Myset3.MoveNext
  Loop
  DoCmd RunSQL "DELETE * FROM apu;" 'tuhotaan apun ja apu2:n sisältö
  DoCmd RunSQL "DELETE * FROM apu2;"
  Myset.MoveNext
Loop
Myset2.Close
```

```
Myset3.Close
field3.Requery
DoCmd RunSQL "DROP TABLE [apu];"
DoCmd RunSQL "DROP TABLE [apu2];"

' Finally, the result is updated into screen

var.Requery
invar.Requery
End Sub
```

This piece of code finds an infinite hierarchy of mathematical models or external presentations.

Appendix 8

Find all possibly affected relations for a selected relation, with ME+ query language.

```
Report 'Effects of externals'
'Report: which are the effected externals of the relations of this graph!'
newline;
foreach .Mathematical model
  {newline;
  'Relation: '"'; id; '"' is edited in external ';
  do :External tool
    {id;
    newline;
    };
  dowhile :Object property values defined
    {foreach .Mathematical model
      {dowhile :Object property values defined
        {if id;=id;2; then
          Relation '"'; id;1; '"' effects to relation '"'; id;3; '""'; newline;
        endif;
        };
      };
  foreach .Mathematical model
    {dowhile :Object property values applied
      {if id;=id;2; then
        'Relation '"' id;1; '"' effects to relation '"' id;3; newline;
      endif;
      };
    };
  };
};
endreport
```

It should be noticed that the search-algorithms coded with ME query-language should be carried out with a loop-structure, which enables automatic search for i to i+1 level topologies. The reason for missing accounting from 1 to n topologies is that in ME+ query language there is no variable or temp-file, or end-of-file-, or read-to-table-functions. This particular piece of code searches only 2 levels of hierarchy but can be “copy - pasted” to account for more levels. In other words, to account, e.g., for a 10-level hierarchy requires 10 times more code lines when using ME+ query language.

Appendix 9

Aspen Plus Input language generation with ME+ query language. This piece of code utilises the ‘search for terminal topology (3 levels down)’.

```
Report 'aspen'
';******** Problem definition ********';newline;
foreach .Chemical process model
  {do explosions
    {if id;type;='Process unit structure' then
    foreach .inp
      {'TITLE '; '´'; :Title; '´'; newline; 'DESCRIPTION '; '´'; :Description; '´'; newline;
      ';~~~~~~~~~ Specify Units of Measurement ~~~~~~~~~'; newline;
      'IN-UNITS '; :IN-UNITS; newline;
      'OUT-UNITS '; :OUT-UNITS; newline;
      };
    endif;
    if id;type;='Process topology' then
    foreach .inp
      {'TITLE '; '´'; :Title; '´'; newline; 'DESCRIPTION '; '´'; :Description; '´'; newline;
      ';~~~~~~~~~ Specify Units of Measurement ~~~~~~~~~'; newline;
      'IN-UNITS '; :IN-UNITS; newline; 'OUT-UNITS '; :OUT-UNITS; newline;
      };
      endif;
    };
  };
  foreach .Chemical process model
    {do explosions
      {if id;type;='Process unit structure' then ';~~~~~~~~~ Specify the Components
        ~~~~~~~~~'; newline; 'COMPONENTS '; newline;
        foreach .Material model;
          { ' '; :Name; ' '; :Name; ' '; :Name; '/'; newline;};
      endif;
      if id;type;='Process topology' then
        ';~~~~~~~~~ Specify the Components ~~~~~~~~~'; newline;
        'COMPONENTS '; newline;
        foreach .Chemical process model
          {do explosions
            {if id;type;='Process unit structure' then
            foreach .Material model;
              {' '; :Name; ' '; :Name; ' '; :Name; '/'; newline;
              };
            endif;
          };
    };
```

```
    endif;
  };
};
foreach .Chemical process model
  {do explosions
    {if id;type;='Process unit structure' then
      ';~~~~~~~~~ Specify Flowsheet Connectivity ~~~~~~~~~'; newline;
      foreach .inp
        { 'FLOWSHEET '; :For process; newline; 'BLOCK '; :For process; '   '; 'IN= ';
        };
      foreach .Boundary
        {do ~To>Object interaction;
          { id; '   ';
          };
          '    OUT=   ';
        do ~From>Object interaction;
          {id; '   ';
           };newline;
        };
    endif;
    if id;type;='Process topology' then
      foreach .inp
      {'FLOWSHEET '; :For process; newline;
      };
    ';~~~~~~~~~ Specify Flowsheet Connectivity ~~~~~~~~~'; newline;
    foreach .Chemical process model
      {'BLOCK '; id; '   '; 'IN= ';
        do ~To>Object interaction;
          {id;
          };
        '    OUT=   ';
        do ~From>Object interaction;
          {id;};newline;
        };
    endif;
    };
  };
';~~~~~~~~~ Specify Feed Streams  ~~~~~~~~~'; newline;
foreach .Chemical process model
  {do explosions
    {if id;type;='Process unit structure' then
      foreach .Boundary
        {do ~To>Object interaction;
          {'STREAM '; id; ' TEMP=? [?] PRES=? [?]  MOLE-FLOW=? [?]  MOLE-FRAC=? [?]'; newline;
          };
```

```
        };
      endif;
      if id; type;='Process topology' then
        foreach .Chemical process model
          {do explosions
            {if id;type;='Process unit structure' then
              foreach .Boundary
                {do ~To>Object interaction;
                  {'STREAM '; id; ' TEMP=? [?] PRES=? [?]  MOLE-FLOW=? [?]
                  MOLE-FRAC=? [?]'; newline;};
                };
            endif;
            };
          };
      endif;
    };
  };
';~~~~~~~~ Specify Block Data  ~~~~~~~~'; newline;
foreach .Chemical process model
  {do explosions
    {if id;type;='Process unit structure' then
      foreach .inp
        {'BLOCK '; :For process; ' (BLOCK-TYPE?)';};
    endif;
    if id; type;='Process topology' then
      foreach .Chemical process model
        {'BLOCK '; :Name; ' (BLOCK-TYPE?)'; newline;};
    endif;
    };
  };
';~~~~~~~~ Specify Reactions  ~~~~~~~~'; newline; ';';newline; ';';newline;
;';newline; ';';newline;
endreport
```

It should be noticed that the search-algorithms coded with ME query-language should be carried out with a loop-structure, which enables automatic search for i to i+1 level topologies. The reason for missing accounting from 1 to n topologies is that in ME+ query language there is no variable or temp-file, or end-of-file-, or read-to-table-functions. This particular search can be easily extended (with “copy-paste” technique) to account for more than 3 levels down but the code lines will grow respectively.

Appendix 10

Suggestions for project management

```
Report 'Suggestions for project management'
foreach .Project
  {'For project: '; id; ' the suggested knowldge and data to be generated (goal
  setting) are:'; newline;
  if :Project characterization='' then 'Select project characterization type!';
  endif;
  if :Project characterization='creation of background data' then
    '1) How to use the gained knwledge of the study (process      metamodel)';
    newline;
    '2) Data on process requirements of the existing processes'; newline;
    '3) Data on materials and phenomena applied in the existing process'; newline;
    '4) Data on how successfully the phenomena has been made to advance as
    desired in the existing process'; newline;
    '5) Data on purpose, structure and state with respect to the performance of the
    existing process'; newline;
  endif;
  if :Project characterization='creation of knowledge on existing processes' then
    '1) How to use the gained knwledge of the study (process      metamodel)';
    newline;
    '2) Data on process requirements of the existing processes'; newline;
    '3) Data on materials and phenomena applied in the existing process'; newline;
    '4) Data on how successfully the phenomena has been made to advance as
    desired in the existing process'; newline;
    '5) Data on purpose, structure and state with respect to the performance of the
    existing process'; newline;
  endif;
  if :Project characterization='creation of knowledge on existing R&D projects' then
    '1) How to use the gained knowlegde of the study (process metamodel)';
    newline;
    '2) Data on modelling requirements (purpose) of existing models'; newline;
    '3) Data on model format and structural elements'; newline;
    '4) Data on model performance: credibility, robustness, computational criteria,
    ...'; newline;
  endif;
  if :Project characterization='Process development' then
    '1) Knowldge on process metamodel'; newline;
    '2) Knowldge on process reqirements and evaluation criteria'; newline;
    '3) Identified prcess to be developed: degrees of freedom and fixed process
    structures'; newline;
    '4) Knowledge on necessary material and phenomena involved in the process';
    newline;
```

```
    '5) Knowledge on desired and feasible state distribution of material, energy and
    phenomena in the process (thermodynamic feasibility)'; newline;
    '6) Knowldge how to create and maintain the desired state distribution in the
    process (technical feasibility)'; newline;
  endif;
  if :Project characterization='Process trouble shooting' then
    '1) Knowldge on process metamodel'; newline;
    '2) Knowldge on process reqirements and evaluation criteria'; newline;
    '3) Identified prcess to be developed: degrees of freedom and fixed process
    structures'; newline;
    '4) Identified phenomena being not proprely under control'; newline;
    '5) Knowledge on desired and feasible state distribution of phenomena not under
    control (thermodynamic feasibility)'; newline;
    '6) Knowldge how to create and maintain the desired state distribution in the
    process (technical feasibility)'; newline;
  endif;
  if :Project characterization='Process retrofit' then
    '1) Knowledge on process metamodel'; newline;
    '2) Knowledge on process performance rewuirements (improved) and evaluation
    criteria'; newline;
    '3) Identified process to be developed: degrees of freedom and fixed process
    structures'; newline;
    '4) Identified set(s) of phenomena and material to be "re-controlled"'; newline;
    '5) Knowldge on desired desired and feasible state distribution of phenomena
    not under control (thermodynamic feasibility)'; newline;
    '6) Knowldge how to create and maintain the desired state distribution in the
    process (technical feasibility)'; newline;
  endif;
  if :Project characterization='Process modelling' then
    '1) knowledge on process metamodel'; newline;
 endif;
 if :Project characterization='Modelling data generation' then
    '1) knowledge on data-metamodel: reqirements of data?, format?, generation
    method?, ...'; newline;
  endif;
};
endreport
```

Appendix 11

A ME+ query compiling the Mathematica program proposed for material balance calculations (partly in finnish).

```
Report 'mathematica'
'(*nimiP= *)                                  (*prosessin nimi*)'; newline;
'(*utot= Length[nimiU] *)                     (*prosessiyksiköiden lukumäärä*)';
newline; newline;
'(*prosessiyksiköiden nimet*)'; newline;
'nimiY={';
foreach .Process
   {id; ','; }; '};'; newline;
'nimiY={';
foreach .Process
   {id; ','; }; '} = Table[i, {i, 1, Length[nimiY]}]; utot = Length[nimiY];'; newline;
newline;
'(*ftot= Length[nimiV] *)                     (*virtojen lukumäärä*)'; newline;
newline;
'(*Virtojen nimet*)'; newline;
'nimiV={';
foreach >Object interaction
   {id; ',';}; '};'; newline; newline;
'nimiV={';
foreach >Object interaction
   {id; ',';}; '} = Table[i, {i, 1, Length[nimiV]}];'; newline; newline;
'(*Reaktioiden nimet*)'; newline;
'nimiR={';
foreach .Phenomenon model
   {id; ','; }; '};'; newline;
'nimiR={';
foreach .Phenomenon model
    {id; ','; }; '} = Table[i, {i, 1, Length[nimiR]}];'; newline; newline;
'(*rtot= *)                                   (*reaktioiden lukumäärä*)'; newline;
'(*ctot= *)                                   (*komponenttien lukumäärä*)';
newline;
newline;
'(*Komponenttien nimet*)'; newline;
'nimiK={';
foreach .Material model
   {id; ','; }; '};'; newline;
'nimiK={';
foreach .Material model
   {id; ','; }; '} = Table[i, {i, 1, Length[nimiK]}];'; newline; newline;
'(*Komponentit virroissa*)'; newline;
```

```
'cf={';
foreach >Object interaction
  { '{';
  dowhile :Interaction components (for material interaction)
    {
    foreach .Material model
      {if id;=id;1; then '1'; ','; else '0'; ','; endif;
      }; '}'; '+'; '{';
    }; '},'; newline;
  }; '};'; newline;
'cf=Transpose[cf];'; newline; newline;
'(*Prosessiyksiköt, joista virrat lähtevät*)'; newline;
'mistaV={';
foreach .Process
  {'{';
  dowhile ~From>Object interaction;
    {
    foreach >Object interaction
      {if id;=id;1; then '1'; ','; else '0'; ',';endif;
      }; '}'; '+'; '{';
    }; '},'; newline;
  }; '};'; newline;
'mistaV = pos[Transpose[mistaV]];'; newline; newline;
'(*Prosessiyksiköt, joihin virrat päättyvät*)'; newline;
'minneV={';
foreach .Process
  {'{';
  dowhile ~To>Object interaction;
    {
    foreach >Object interaction
    {if id;=id;1; then '1'; ','; else '0'; ','; endif;
    }; '}'; '+'; '{';
  }; '},'; newline;
  }; '};'; newline;
'minneV = pos[Transpose[minneV]];'; newline; newline;
'(*Reaktioihin osallistuvat komponentit*)'; newline; 'cr={';
foreach .Phenomenon model
  {'{';
  dowhile :Components
  {foreach .Material model
    {if id;=id;1; then '1'; ','; else '0'; ','; endif;
    }; '}'; '+'; '{';
    }; '},'; newline;
  }; '};'; newline;
'cr=Transpose[cr];'; newline; newline;
'(*Komponenttien stoikiometriset kertoimet reaktioissa*)'; newline; 'a={';
```

```
foreach .Phenomenon model
   { '{';
   do ~To>Stoichiometry;
      {
      do ~To>Stoichiometry~From.Material model;
         {
         foreach .Material model
            {if id;=id;1; then id;2; ','; else '0'; ','; endif;
            }; '}'; '+'; '{';
         };
      }; '},'; newline;
   }; '};'; newline; newline;
'(*Prosessiyksiköissä tapahtuvat reaktiot*)'; newline;
'ru={';
foreach .Phenomenon model
   {'{';
   do ~From>association
      {do ~From>association~To.Process;
         {foreach .Process
            {if id;=id;1 then '1,'; else '0,'; endif;
            }; '}'; '+'; '{';
         };
      }; '},'; newline;
   }; '};'; newline; newline;
'(*Virtojen tilarelaatiot*)'; newline;
'virtojenTilarelaatiot = {';
foreach >Object interaction
   {do :Relation
      { '(*'; :comment; '*)'; newline;
      :relation; ','; newline;
      };
   };
  '};'; newline; newline;
'(*Virtojen assigmentit*)'; newline;
foreach >Object interaction
   {do :Object purpose
      {do :Purpose expression
         {do :Object variable
            {if :fixed (assignment) = 'yes' then
               :Property; '='; :Property value; ';';
            endif; newline;
            };
         };
      };
   }; newline; newline;
'(*Prosessien tilarelaatiot*)'; newline;
```

```
'prosessienTilarelaatiot = {';
foreach .Process
   {do :Relation
      { '(*'; :comment; '*)'; newline;
      :relation; ',';  newline;
      };
   };
'};'; newline; newline;
'(*Prosessin interiorin assignmentit*)'; newline;
foreach .Process
   {do :Object purpose
      {do :Purpose expression
         {do :Object variable
            {if :fixed (assignment) = 'yes' then
               :Property; '='; :Property value; ';';
            endif; newline;
            };
         };
      };
   };
endreport
```

Published by

Vuorimiehentie 5, P.O.Box 2000, FIN–02044 VTT, Finland
Phone internat. +358 9 4561
Fax +358 9 456 4374

Series title, number and report code of publication

VTT Publications 438
VTT–PUBS–438

Author(s)

Pasanen, Antti

Title

Phenomenon-Driven Process Design methodology
Computer implementation and test usage

Abstract

This work focuses on how to develop a generic computer tool support for a design methodology proposed for chemical engineering purposes. The applied methodology is called Phenomenon Driven Process Design methodology (PDPD), and the formal language of the methodology is called PSSP language (an acronym for Purpose, Structure, State and Performance attributes). The focus of this work is divided into two theses

1. "Multi-characteristic and creative chemical process research and development work can be supported in a systematic manner – based on PDPD – and implemented as a computer system."
2. "The use of this computer system is expected to result in comprehensive process models and modelling work with efficient documentation and data management. These are prospective advantages for any chemical engineering R&D organisation."

The developed tool is tested and evaluated by employing it in a few academic and industrial pilot projects, but the methodology has not been taken into practice by industrial companies. The introduced test cases illustrate the formalisation of research data and knowledge of both the project and process models. The test cases also illustrate a phenomenon-driven way to argument process models and project management. The analysis and argumentation of both the developed tool and the test cases are written transparently for discussion. In addition, the usage and characteristics of the meta-level tool for capturing the methodology is analysed in an open-formatted way. The use of the developed prototype application is expected:

a) to save project meeting time and to improve meeting performance,
b) to enhance data and knowledge exchange among project staff,
c) to improve project and process model data retrieval,
d) to ease the utilisation of various process modelling software and
e) to improve the comprehension of the linkages between numerical results, conceptual process models and the set project goals.

The research work behind this thesis has been carried out during 1994 to 2000. PDPD and PSSP–based research work for building computer tool support are still ongoing.

Keywords

research and development, process design, systematization, conseptual design, models, modelling, simulation, computer tools, PSSP, chemical engineering

Activity unit

VTT Chemical Technology, Processes and Environment, Biologinkuja 7, P.O.Box 1401, FIN–02044 VTT, Finland

ISBN

951–38–5854–5 (soft back ed.)
951–38–5855–3 (URL:http://www.inf.vtt.fi/pdf/)

Project number

KETT94154

Date	Language	Pages	Price
August 2001	English	140 p. + app. 26 p.	D

Series title and ISSN

VTT Publications
1235–0621 (soft back ed.)
1455–0849 (URL: http://www.inf.vtt.fi/pdf/)

Sold by

VTT Information Service
P.O.Box 2000, FIN-02044 VTT, Finland
Phone internat. +358 9 456 4404
Fax +358 9 456 4374